Open eBook™ Publication Structure 1.2

RECOMMENDED SPECIFICATION August 27, 2002

Open eBook™ Open eBook Forum Publication Structure 1.2

RECOMMENDED SPECIFICATION

August 27, 2002

TABLE OF CONTENTS

1 Overview

1.1 Purpose and Scope

In order for electronic-book technology to achieve widespread success in the marketplace, Reading Systems must have convenient access to a large number and variety of titles. The Open eBook Publication Structure (OEBPS) is a specification for representing the content of electronic books. Specifically:

- The specification is intended to give content providers (e.g. publishers, authors, and others who have content to be displayed) and tool providers minimal and common guidelines which ensure fidelity, accuracy, accessibility, and adequate presentation of electronic content over various electronic book platforms.
- The specification seeks to reflect established content format standards.
- The goal of this specification is to define a standard means of content description for use by purveyors of electronic books (publishers, agents, authors et al.) allowing such content to be provided to multiple Reading Systems.

1.2 Definitions

BASIC OEBPS DOCUMENT

An OEBPS Document that restricts itself to the markup constructs defined in this specification.

CONTENT PROVIDER

A publisher, author, or other information provider who provides a publication to one or more Reading Systems in the form described in this specification.

DEPRECATED

A feature that is permitted, but not recommended, by this specification. Such features may be removed in future revisions.

EXTENDED OEBPS DOCUMENT

An OEBPS Document that uses markup constructs beyond those in this specification, but adheres to the extension mechanism defined herein.

OEBPS CORE MEDIA TYPE

A MIME media type that all Reading Systems must support.

OEBPS DOCUMENT

An XML document that conforms to this specification – generally containing textual content of an OEBPS Publication.

OEBPS PACKAGE

An XML file that describes an OEBPS Publication. It identifies all other files in the publication and provides descriptive information about them.

OEBPS PUBLICATION

A collection of OEBPS Documents, an OEBPS Package file, and other files, typically in a variety of media types, including structured text and graphics, that constitutes a cohesive unit for publication.

READER

A person who reads a publication.

READING DEVICE

The physical platform (hardware and software) on which publications are rendered.

READING SYSTEM

A combination of hardware and/or software that accepts OEBPS Publications and makes them available to readers. Great variety is possible in the architecture of Reading Systems. A Reading System may be implemented entirely on one device, or it may be split among several computers. In particular, a Reading Device that is a component of Reading System need not directly accept OEBPS Publications, but all Reading Systems must do so. Reading Systems may include additional processing functions beyond the scope of this specification, such as compression, indexing, encryption, rights management, and distribution.

1.3 Relationship to Other Specifications

This specification combines subsets and applications of other specifications. Together, these facilitate the construction, organization, presentation, and unambiguous interchange of electronic documents:

1. the XML 1.0 Extensible Markup Language specification (http://www.w3.org/TR/REC-xml);
2. the XML namespace specification (http://www.w3.org/TR/REC-xml-names);
3. the XHTML 1.1 Extensible HyperText Markup Language specification (http://www.w3.org/TR/xhtml11/);
4. the CSS2 Cascading Style Sheets language (http://www.w3.org/TR/REC-CSS2);
5. the Dublin Core metadata specification (http://dublincore.org/documents/1999/07/02/dces/) and the MARC relator code list (http://www.loc.gov/marc/relators/);
6. the Unicode character set (http://www.unicode.org); and
7. particular MIME media types (http://www.ietf.org/rfc/rfc2046.txt and http://www.iana.org/assignments/media-types/index.html).
8. the XML style sheet processing instruction (http://www.w3.org/TR/xml-stylesheet).

1.3.1 Relationship to XML

OEBPS is based on XML because of its generality and simplicity, and because XML documents are likely to adapt well to future technologies and uses. XML also provides well-defined rules for the syntax of documents, which decreases the cost to implementers and

reduces incompatibility across systems. Further, XML is extensible: it is not tied to any particular set of element types, it supports internationalization, and it encourages document markup that can represent a document's internal parts more directly, making them amenable to automated formatting and other types of computer processing.

- Reading Systems **must** be XML processors as defined in XML 1.0. All OEBPS Documents **must** be well-formed XML documents, although they need not be valid XML documents.

XML well-formedness requires characteristics beyond what HTML browsers typically require, such as:

- Elements must be bounded by both start- and end-tags;
- Elements must nest properly, with no overlaps;
- Attribute values must be quoted;
- Attribute assignments must use the non-minimized form (unlike some "border" usages);
- All "<" and "&" characters intended as content must be escaped as "<" and "&" or represented by their equivalent Unicode numerical character references;
- All element names and attribute names must be consistent in case (all OEBPS 1.2 names are, as in XHTML 1.1, lower-case); and
- All empty elements must use the XML empty element syntax (this specification also strongly encourages whitespace before the trailing slash, although such space is optional in XML; for example, "
").

Empty elements (such as the HTML **br** and **hr** elements) are those that permit no content. The XML and formal HTML syntaxes for these are incompatible, though the XML form, with whitespace before the trailing slash, is accepted by most HTML browsers. The addition of this whitespace remains strictly conformant XML, as XML ignores whitespace within tags. Hence, this specification strongly encourages, though does not require, this conforming variation of the XML form (for example, "
"). This is the most portable syntax and it contributes to document longevity, even though, strictly speaking, it is not valid in HTML.

Syntactic transformation from valid HTML to well-formed XML is trivial (though semantic transformations that also add brand-new structure and information value may not be). Transformation from invalid but moderately clean HTML is also usually an easy process and easily automated: several free tools already exist for this, such as "Tidy" (see http://www.w3.org/People/Raggett/tidy/). Transformation from extremely dirty HTML to XML, however, is of unpredictable complexity.

Not all well-formed XML 1.0 documents are conformant OEBPS Documents. This specification imposes further constraints in order to improve interoperability. These constraints are the "OEBPS Common Requirements," defined below.

This specification contains two XML DTDs – the OEBPS Package DTD (Appendix A) and the Basic OEBPS Document DTD (Appendix B). The OEBPS Package file (which, beyond well-formed, **must** be valid XML) provides the "framework" for a complete publication, and

Reading Systems **should** use it to find and organize publication components. The Basic OEBPS Document DTD formally defines the XHTML 1.1 subset described in this specification.

This specification ensures that for any Basic OEBPS Document, there is a syntax form that:

- is a valid XML document,
- conforms to the OEBPS Document DTD,
- conforms to the XHTML 1.1 specification, and
- is effectively previewable in typical HTML browsers.

1.3.2 Relationship to XML Namespaces

This version of the specification **does not require, but does allow**, Reading Systems to process XML namespaces according to the XML Namespaces Recommendation at http://www.w3.org/TR/REC-xml-names.

Namespace prefixes distinguish identical names that are drawn from different XML vocabularies. An XML namespace declaration in an XML document associates a namespace prefix with a unique URI. The prefix can then be employed on element or attribute names in the document. Alternatively, a namespace declaration in an XML document may identify a URI as the default namespace, applicable to elements lacking a namespace prefix. The XML namespace prefix is separated from the suffix element or attribute name by a colon.

OEBPS Documents **must not** contain declarations of default namespaces that reference namespaces other than the XHTML namespace (“`http://www.w3.org/1999/xhtml`”). Conversely, any declarations of prefixed namespaces within OEBPS Documents **must not** reference the XHTML namespace.

If a Reading System is not namespace-aware, any element within an OEBPS Document that contains a namespace prefix is treated as an Extended OEBPS Document element, with the colon acting as a normal XML Name character, afforded no special meaning.

The use of the dc: prefix, however, is required for Dublin Core metadata element attributes in the OEBPS Package file. For upwards compatibility, the element **`dc-metadata`** in an OEBPS Package file is required to have an attribute of *xmlns:dc="http://purl.org/dc/elements/1.1/"* and *xmlns:oebpackage="http://openebook.org/namespaces/oeb-package/1.0/"*. In addition, the Dublin Core elements are declared in the OEBPS Package DTD with an explicit prefix of dc:.

1.3.3 Relationship to XHTML

This specification recognizes the importance of current software tools, legacy data, publication practices, and market conditions, and has therefore based the Basic OEBPS Document vocabulary on XHTML 1.1. This approach allows content providers to exploit current XHTML content, tools, and expertise.

To minimize the implementation burden on Reading System implementers (who may be working with devices that have power and display constraints), the Basic OEBPS Document element set does not include all XHTML 1.1 elements and attributes. The elements and attributes were selected from the XHML 1.1 specification and were chosen to be consistent with current directions in XHTML.

Any construct deprecated in XHTML 1.1 is either deprecated or omitted from this specification; CSS-based equivalents are provided in most such cases. Style sheet constructs are also used for new presentational functionality beyond that provided in XHTML.

To achieve predictable results, for greater document interoperability, and to support upwards compatibility with future versions of this specification, it is strongly recommended that Basic OEBPS Documents be valid XML documents with respect to the Basic OEBPS Document DTD.

1.3.4 Relationship to CSS

This specification defines a style language based on CSS 2, with a media type of "text/x-oeb1-css". The Publication Structure Working Group is aware that this definition of a media type goes against the recommendation of the CSS Working Group, but has chosen to do so due to practical considerations.

The CSS-based style sheet constructs in this specification define required rendering functionality. To minimize the burden on Reading System developers and device manufacturers, not all CSS 2 properties are included. A few additional properties and values have been added to support page layout, headers, and footers.

In a number of cases, this specification does not require Reading Systems to provide the full range of rendering that a standard CSS style sheet might request. For example, some Reading Systems will use monochrome displays. It would neither be acceptable to limit all Reading Systems to monochrome, nor to declare color use a non-standardized extension beyond OEBPS. In such cases, the CSS settings are allowed, and keep their meanings; but a conforming Reading System **may** gracefully degrade to a simpler rendering.

This specification supports the ~~*style*~~ attribute (though deprecated), the **style** element, and externally linked style sheets. Reading Systems need not perform XML-namespace handling while processing style sheets.

Style sheets **may** be associated with an OEBPS Document in several ways:

1. by ~~*style*~~ attributes on specific XHTML elements (deprecated);
2. by **style** elements within the XHTML header;
3. by an external style sheet identified on a **link** elements in the XHTML **head**; and/or
4. by an external style sheet identified via the processing instruction **xml-stylesheet** (see Section 1.3.8).

The relative priority of the first three cases is as defined for XHTML 1.1 and CSS 2. Style sheets linked via a processing instruction are treated as if they had been linked via XHTML **link** elements preceding any actual XHTML **link** elements. As defined in the Conformance section, if no style sheet is defined or no applicable style is found for a given element, XHTML rendering is the default as defined elsewhere in this specification.

Styles attached via the first two methods listed above **must** use only those CSS constructs defined in Section 4 of this specification. External style sheets linked via the XHTML **link** element or by the processing instruction **xml-stylesheet**, however, **may** use this or any other style language, such as XSL (see http://www.w3.org/TR/xsl).

Style sheets of type "`text/x-oeb1-css`" **must** employ only those CSS constructs defined as supported in Section 4 of this specification. Style sheets of other MIME media types **may** be

substituted for the `text/x-oeb1-css` style sheets at the discretion of the Reading System.

The XHTML 1.1 specification groups externally linked style sheets into sets by their titles (including a "persistent" set for which the title is the null string). This specification requires that at least one style sheet in each such set **must** be of MIME media type "text/x-oeb1-css".

Reading Systems that implement only the OEBPS CSS subset **may** ignore any style sheets using other style languages. Reading Systems that support extended style sheet functionality **may** choose among any of the other external style sheets. It is strongly recommended that unique MIME media types be defined for any novel style sheet languages supported, and that style sheets in those languages be detected by examining the MIME media type.

1.3.5 Relationship to Dublin Core

The Dublin Core is designed to minimize the cataloging burden on authors and publishers, while providing enough metadata to be useful. This specification supports the set of Dublin Core 1.1 metadata elements (http://dublincore.org/documents/1999/07/02/dces/), supplemented with a small set of additional attributes addressing areas where more specific information may be useful. For example, the *`role`* attribute added to the **`dc:Contributor`** element allows for much more detailed specification of contributors to a publication, including their roles expressed via relator codes.

Content providers **must** include a minimum set of a metadata elements, defined in section 2.2, and **should** incorporate additional metadata to enable readers to discover publications of interest.

1.3.6 Relationship to Unicode

Publications **may** use the entire Unicode character set, in UTF-8 or UTF-16 encodings, as defined by Unicode (see http://www.unicode.org/). The use of Unicode facilitates internationalization and multilingual documents. However, Reading Systems **are not required to** provide glyphs for all Unicode characters.

Reading Systems **must** parse all UTF-8 and UTF-16 characters properly (as required by XML). Reading Systems **may** decline to display some characters, but **must** be capable of signaling in some fashion that undisplayable characters are present. They **must not** display Unicode characters merely as if they were 8-bit characters. For example, the biohazard symbol (0x2623) need not be supported by including the correct glyph, but **must not** be parsed or displayed as if its component bytes were the two characters "&#" (0x0026 0x0023).

1.3.7 MIME Media Types

This specification defines a list of OEBPS Core Media Types that all Reading Systems **must** support (as required by this specification) and publications may include. Publications **may** include resources of other media types, but for each such resource **must** include an alternative resource of an OEBPS Core Media Type (using methods defined in this specification).

The OEBPS Core Media Types are:

MIME Media Type	Reference	Description
`image/jpeg`	RFC 2046	Used for raster graphics
`image/png`	RFC 2083	Used for raster graphics
`text/x-oeb1-`	this	Used for Basic or Extended OEBPS

`document`	specification	Documents
`text/x-oeb1-css`	this specification	Used for OEBPS CSS-subset style sheets
`application/xml-dtd`	RFC 3023	Used for DTDs included with the publication
`application/xml-external-parsed-entity`	RFC 3023	Used for external parsed entity documents

1.3.8 XML Style Sheet Processing Instruction

This specification includes support for the XML style sheet processing instruction **`xml-stylesheet`**, defined in the W3C Recommendation "Associating Style Sheets with XML Documents" (http://www.w3.org/TR/xml-stylesheet). In this specification, the allowed pseudo-attributes for **`xml-stylesheet`** are those corresponding to the allowed attributes for XHTML **`link`** when used to identify an external style sheet. This processing instruction is placed in the prolog of the XML document. It can appear multiple times as **`link`** can.

1.4 Conformance

This section defines conformance for OEBPS Documents, Publications, and Reading Systems.

1.4.1 Document and Publication Conformance

This specification defines two named levels of conformance for OEBPS Documents—Basic and Extended, and one conformance level for OEBPS Publications. An OEBPS Document is conforming if and only if it is either a Basic OEBPS Document or an Extended OEBPS Document.

1.4.1.1 OEBPS Common Requirements

Each conformant OEBPS Document (whether Basic or Extended) and each conformant OEBPS Package File **must** meet these necessary conditions, referred to in this specification as the "Common Requirements:"

(i) it is a well-formed XML document (as defined in XML 1.0);

(ii) it begins with a correct XML declaration (e.g. *<?xml version='1.0'?>*);

(iii) it is encoded in UTF-8 or UTF-16;

(iv) it does not include an XML internal declaration subset; and

(v) any attribute with a type of NMTOKEN, ID, or IDREF must be an XML Name.

1.4.1.2 OEBPS Common Document Requirements

A conformant OEBPS Document (whether Basic or Extended) **must** meet these necessary conditions, referred to in this specification as the "common document requirements:"

(i) it meets the OEBPS Common Requirements;

(ii) it does not contain declarations of default namespaces referencing other than the XHTML namespace (“`http://www.w3.org/1999/xhtml`”);

(iii) any declarations of prefixed namespaces do not reference the XHTML namespace (“`http://www.w3.org/1999/xhtml`”);

(iv) if external style sheets are used, then at least one style sheet in each title set (as described in the XHTML 1.1 specification), including any “persistent” set, must be of MIME media type “`text/x-oeb1-css`”; and

(v) all style parameters specified within the document itself belong to the OEBPS CSS subset.

1.4.1.3 Basic OEBPS Document

A document is a Basic OEBPS Document if and only if:

(i) it meets the OEBPS Common Document Requirements;

(ii) its DOCTYPE declaration, if any, references the Basic OEBPS 1.2 Document DTD;

(iii) it uses only the element names, attribute names, and attribute values drawn from the Basic OEBPS Document Vocabulary with all element and attribute names in lower case; and

(iv) it uses element names, attribute names, and attribute values in a manner broadly consistent with the intentions of the relevant descriptions in this specification and those of XHTML 1.1, with this specification taking precedence in the event of conflicts.

1.4.1.4 Extended OEBPS Document

A document is an Extended OEBPS Document if and only if

(i) it meets the OEBPS Common Document Requirements;

(ii) it uses elements, attributes, or attribute values not drawn from the Basic OEBPS Document Vocabulary, or its DOCTYPE declaration references a DTD other than the Basic OEBPS 1.2 Document DTD; and

(iii) for any element not of the Basic OEBPS Document vocabulary it provides an applicable CSS style rule using only the OEBPS CSS subset.

1.4.1.5 Validity

OEBPS Documents, Basic or Extended, **may or may not** be valid (as defined in XML 1.0) with respect to an associated DTD. However, all OEBPS Documents **must** be well-formed XML 1.0 documents.

1.4.1.6 Publication Conformance

A collection of files is a conforming OEBPS Publication if and only if

(i) it includes a single OEBPS Package file that obeys the OEBPS Common Requirements listed above, and is a valid XML document conforming to the

OEBPS Package DTD;

(ii) the OEBPS Package file includes one and only one manifest entry corresponding to each other file in the OEBPS Publication;

(iii) the manifest entry for each file in the publication specifies a MIME media type for the file (see http://www.ietf.org/rfc/rfc2046.txt);

(iv) each file whose manifest entry identifies it as being in one of the OEBPS Core Media Types, conforms as defined for those MIME media types;

(v) the **dc-metadata** element contains at least one **dc:Identifier** element, at least one **dc:Title** element, and at least one **dc:Language** element;

(vi) the ***unique-identifier*** attribute of the **package** element is a correct XML IDREF to a **dc:Identifier** element;

(vii) any extended values specified for the dc:Creator and **dc:Contributor** elements' *role* attribute **must** be taken from the registered MARC Relator Code list or **must** begin with "oth."; and

(viii) any extended values specified for the **guide** element's *type* attribute begin with "other.".

1.4.2 Reading System Conformance

This specification defines only one level of conformance for a Reading System. A Reading System is conformant if and only if it processes documents as follows:

A) When presented with a Basic OEBPS Document the Reading System

(i) correctly processes XML as required in the XML 1.0 specification, including that specification's requirements for the handling of well-formedness errors;

(ii) recognizes all markup described as permitted in this specification and processes it consistently with the corresponding explanation(s) in this specification and in those of XHTML 1.1 and CSS 2 (in case of any conflict, this specification takes precedence); and

(iii) does not render objects of unsupported media types, in the absence of fallbacks. These fallbacks are clearly defined in section 2.3.1.

B) When presented with an Extended OEBPS Document, or a document whose MIME media type is 'text/x-oeb1-document' and which is not a Basic OEBPS Document, the Reading System

(i) performs as required in A.i, A.ii, and A.iii;

(ii) recognizes element instances not from this specification and renders them according to any applicable CSS style sheet rules, as described in section 1.3.4; and

(iii) continues processing, displaying the element inline, as if "display: inline" applied, for any element not dealt with by (i) and (ii).

C) When presented with an OEBPS Package file the Reading System

(i) processes all elements and attributes as described in section 2 of this specification.

D) When providing navigation via the OEBPS spine, the Reading System

(i) does not render content that does not have the media type `text/x-oeb1-document`.

E) When presented with one or more style sheets via the XHTML **`link`** mechanism or the **`xml-stylesheet`** processing instruction, described in Section 1.3.8, the Reading System:

(i) processes the document in accordance with the `text/x-oeb1-css` style sheets; and

(ii) if style sheets of a MIME media type other than `text/x-oeb1-css` are provided, **may** substitute those style sheets for the `text/x-oeb1-css` style sheets. Reading Systems (although not necessarily Reading Devices) which support other style sheet media types **must** provide a mechanism for requesting that those style sheets be ignored in favor of the `text/x-oeb1-css` style sheets.

F) When presented with an OEBPS 1.0 Document or Package file, the Reading System **must** process them as a conformant OEBPS 1.0 Reading System would.

Note: Reading Systems are **not required** to support XML entity and attribute declarations (beyond parsing past them as XML requires), because such constructs are not permitted in conforming OEBPS Documents.

1.4.3 Compatibility with Future Versions

It is the intent of the contributors to this specification that subsequent generations of this specification continue in the directions established by the 1.0 release. Specifically:

- Content format standards will be compatible with W3C (and IETF) standards;
- Future versions of this specification are expected to improve alignment with XML-based specifications, relaxing the constraints on OEBPS 1.2 Documents that are more restrictive than XML, requiring further XML processing capability of OEBPS-conformant Reading Systems, and perhaps supporting other XML-related standards such as XLink; and
- Any required functionality not present in relevant official standards shall be defined in a manner consistent with its eventual submission to an appropriate standards body as extensions to existing standards.

1.4.4 Compatibility of Version 1.2

Version 1.2 of the OEBPS Publication Structure is not meant to be a substantially “new” specification. However, version 1.2 does add functional enhancements over 1.0.1, largely supporting the goal of allowing enhanced control over content presentational fidelity. Specifically, the following are the most substantive additions:

- The Basic OEBPS Document element set has been expanded.
- All previously deprecated elements have been removed; it is now a true subset of XHTML 1.1.
- The OEBPS CSS subset has been augmented with numerous CSS2 properties and values.
- Most previously deprecated attributes have been removed.
- The OEBPS CSS subset has been augmented with a more extensive set of selectors.

It was a goal of version 1.2 that all documents conformant according to version 1.0.1 would remain conformant under 1.2. However, removal of elements deprecated in 1.0.1 (e.g. **`font`**) and the addition of namespace requirements (see Section 1.3.3) rendered full compatibility with version 1.0.1 impossible.

1.5 Extensibility

Use of Extended OEBPS Documents is the recommended mechanism for adding information and structure beyond that provided by the XHTML subset defined in this specification (e.g. to associate further semantics with content). Arbitrary non-OEBPS elements **may** be added as long as such elements are provided with style definitions in accompanying style sheets.

For example, the following document would be an Extended OEBPS Document excerpt:

```
<chapter>
<milestone n="257" />
<chapterhead>Chapter one</chapterhead>
<p>Now is the time… </p>
</chapter>
```

if associated with a style sheet containing the following excerpt:

```
chapter     {page-break-before: always; display: block}
milestone   {display: none}
chapterhead {
      font-weight: bold;
      font-family: sans-serif;
      text-align: center;
      display: block;
      margin-top: 4ex
}
```

1.6 Accessibility

This specification incorporates features that ensure content can be made accessible to, and usable by, persons with reading disabilities. Existing accessibility features developed by the World Wide Web Consortium (W3C) for XHTML 1.1 for content accessibility are incorporated into the OEBPS specification.

OEBPS Publications **should** be authored in accordance with the W3C Web Content Accessibility Guidelines 1.0 (http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/) to ensure that the broadest possible set of users will have access to books delivered in this format.

In addition, recommendations from the W3C HTML 4.0 Guidelines for Mobile Access (http://www.w3.org/TR/NOTE-html40-mobile/) and the W3C Web Accessibility Initiative's proposed User Agent Guidelines (http://www.w3.org/TR/WD-WAI-USERAGENT/) should be reviewed and applied by OEBPS implementers to ensure that Reading Systems will be in conformance with accessibility requirements.

1.7 Future Directions

This specification is designed to take advantage of current practices while preparing for future developments. Although details of subsequent versions of this specification remain to be determined, it is the expectation of the Publication Structure Working Group that continued evolutionary development will occur. The “themes” driving the creation of version 2.0 of the OEBPS Publication Structure are: standards compliance (e.g. full namespace support), metadata modularization, enhanced support for linking and navigation, and better support for international content. Other themes deemed important for future versions include: more rigorous separation of content and presentation, greater accessibility, Reading Device-specific presentation control and/or Reading Device profiles, application-specific markup (e.g. math, chemical), Publication container file format, multiple reading orders, and support for active content (e.g. multimedia, scripting), all while maintaining alignment with relevant standards. Additionally, maintaining backward compatibility to this version of this specification is a high priority. Future directions can be tracked at http://www.openebook.org.

Metadata support for OEBPS content is currently under development in other working groups within the OEBF; the Dublin Core constructs included in the OEBPS 1.2 Package File are only intended to provide a minimal level of metadata support while the work of those groups is being completed, as well as to maintain compatibility with 1.0.1.

2 The OEBPS Package

A publication conforming to this specification **must** include exactly one OEBPS Package file, which specifies the OEBPS Documents, images, and other objects that make up the OEBPS Publication and how they relate to each other.

The package file **should** be named using the extension “.opf”, in order to make it readily identifiable within the group of files making up the publication. Package files are of MIME media type “text/xml”. This specification does not define means for physically bundling files together to make one data transfer object (such as using zip or tar).

It is **not required** that the OEBPS Package DTD be physically included in every publication. If included, it **should** be referenced from the manifest (as described below for other files).

The major parts of the OEBPS Package file are:

PACKAGE IDENTITY

A unique identifier for the OEBPS Publication as a whole.

METADATA

Publication metadata (title, author, publisher, etc.).

MANIFEST

A list of files (documents, images, style sheets, etc.) that make up the publication. The manifest also includes fallback declarations for files of types not supported by this specification.

SPINE

An arrangement of documents providing a linear reading order.

TOURS

A set of alternate reading sequences through the publication, such as selective views for various reading purposes, reader expertise levels, etc.

GUIDE

A set of references to fundamental structural features of the publication, such as table of contents, foreword, bibliography, etc.

An OEBPS Package **must** be a valid XML document conforming to the OEBPS Package DTD (Appendix A). Appendix C includes the mnemonic character entities file associated with the OEBPS Pacakge DTD. An informal outline of the package is as follows:

```
<?xml version="1.0"?>
<!DOCTYPE package
  PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.2 Package//EN"
  "http://openebook.org/dtds/oeb-1.2/oebpkg12.dtd">
<package>
        metadata
        manifest
        spine
        guide
</package>
```

The following sections describe the parts of the OEBPS Package.

2.1 Package Identity

The **package** element is the root element in a package file; all other elements are nested within it.

The **package must** specify a value for its ***unique-identifier*** attribute. The ***unique-identifier*** attribute's value specifies which **dc:Identifier** element, described in section 2.2.10, provides the package's preferred, or primary, identifier. The package file's author is responsible for choosing a primary identifier that is unique to one and only one particular package (i.e., the set of files referenced from the package file's **manifest)**.

Notwithstanding the requirement for uniqueness, Reading Systems **must not** fail catastrophically should they encounter two distinct packages with the same purportedly unique primary identifier.

2.2 Publication Metadata

The required **metadata** element is used to provide information about the publication as a whole. It contains a Dublin Core metadata record within a **dc-metadata** element, and supplemental metadata in an **x-metadata** element.

The required **dc-metadata** element contains specific publication-level metadata as defined by the Dublin Core Metadata Initiative (http://dublincore.org/). The descriptions below are included for convenience, and the Dublin Core's own definitions take precedence (see http://dublincore.org/documents/1999/07/02/dces/).

The optional **x-metadata** element, if present, **must** contain one or more instances of a **meta** element, analogous to the XHTML 1.1 **meta** element, but applicable to the publication as a whole. The **x-metadata** element allows content providers to express arbitrary metadata beyond the data described by the Dublin Core specification. Individual OEBPS Documents may include the **meta** element directly (as in XHTML 1.1) for document-specific metadata. This specification uses the OEBPS Package file alone as the basis for expressing publication-level Dublin Core metadata.

For example:

```
<metadata>
   <dc-metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:oebpackage="http://openebook.org/namespaces/oeb-
package/1.0/">
     …
   </dc-metadata>
   <x-metadata>
     <meta name="price" content="USD 19.99" />
   </x-metadata>
</metadata>
```

The XML namespace mechanism (see http://www.w3.org/TR/REC-xml-names/) is used to identify the elements used for Dublin Core metadata without conflict. Note that there is no requirement on Reading Systems to process namespaces. This syntax is used to provide for upwards-compatibility.

The **dc-metadata** element can contain any number of instances of any Dublin Core elements. Dublin Core element names begin with the “dc:” prefix followed by a leading uppercase letter. Dublin Core metadata elements may occur in any order; in fact, multiple instances of the same element type (multiple **dc:Creator** elements, for example) can be interspersed with other metadata elements without change of meaning.

For upwards-compatibility, the element **dc-metadata** in an OEBPS Package **must** have an attribute of *xmlns:dc="http://purl.org/dc/elements/1.1/"* and *xmlns:oebpackage="http://openebook.org/namespaces/oeb-package/1.0/"*.

Each Dublin Core field is represented by an element whose content is the field’s value. At least one of each of **dc:Title**, **dc:Identifier** and **dc:Language** **must** be included in the **dc-metadata** element. Dublin Core elements, like any other elements in the OEBPS Package file, **may** have an *id* attribute specified. At least one **dc:Identifier**, that which is referenced from the **package** *unique-identifier* attribute, **must** have an ***id*** specified.

Because the Dublin Core metadata fields for Creator and Contributor do not distinguish roles of specific contributors (such as author, editor, and illustrator), this specification adds an optional *role* attribute for this purpose. See section 2.2.6 for a discussion of *role*.

To facilitate machine processing of **dc:Creator** and **dc:Contributor** fields, this specification adds the optional *file-as* attribute for those elements. This attribute is used to specify a normalized form of the contents. See section 2.2.6 for a discussion of *file-as*.

This specification also adds a *scheme* attribute to the **dc:Identifier** element to provide a structural mechanism to separate an identifier value from the system or authority that generated or defined that identifier value. See section 2.2.10 for a discussion of *scheme*.

This specification also adds an *event* attribute to the **dc:Date** element to enable content providers to distinguish various publication specific dates (for example, creation, publication, modification). See section 2.2.7 for a discussion of *event*.

For example:

```
<package unique-identifier="xyz">
      <metadata>
            <dc-metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:oebpackage="http://openebook.org/namespaces/oeb-package/1.0/">
                  <dc:Title>Alice in Wonderland</dc:Title>
                  <dc:Language>en</dc:Language>
                  <dc:Identifier id="xyz"
                   scheme="ISBN">123456789X</dc:Identifier>
                  <dc:Creator role="aut">Lewis Carroll</dc:Creator>
            </dc-metadata>
      </metadata>
      ...
</package>
```

There are no attributes for the elements within **dc-metadata** defined by Dublin Core – only the elements' contents are so defined.

The following subsections describe the individual Dublin Core metadata elements.

2.2.1 <dc:Title> </dc:Title>

The title of the publication. An OEBPS Package **must** include at least one instance of this element type, however multiple instances are permitted. Any Reading System that displays title metadata to the user **should** either use the first **dc:Title** only, or all **dc:Title** elements.

2.2.2 <dc:Creator> </dc:Creator>

A primary creator or author of the publication. Additional contributors whose contributions are secondary to those listed in **dc:Creator** elements **should** be named in **dc:Contributor** elements.

Publications with multiple co-authors **should** provide multiple **dc:Creator** elements, each containing one author. The order of **dc:Creator** elements is presumed to define the order in which the creators' names should be presented by the Reading System.

This specification recommends that the content of the **dc:Creator** elements hold the text for a single name as it would be presented to the user.

This specification adds to the **dc:Creator** element two optional attributes: *role* and *file-as*. The set of values for *role* are identical to those defined in section 2.2.6 for the **dc:Contributor** element. The *file-as* attribute **should** be used to specify a normalized form of the contents, suitable for machine processing. For example, one might find

```
<dc:Creator file-as="King, Martin Luther Jr." role="aut">
      Rev. Dr. Martin Luther King Jr.
</dc:Creator>
```

If a Reading System displays creator information, the Reading Systems **must** display the contents of all **dc:Creator** elements, in the order provided, with appropriate separating spacing and/or punctuation.

2.2.3 <dc:Subject> </dc:Subject>

Multiple instances of the **dc:Subject** element are supported, each including an arbitrary phrase or keyword. This specification makes no attempt to standardize subject naming schemes, such as the Library of Congress Subject Heading System.

2.2.4 <dc:Description> </dc:Description>

Description of the publication's content.

2.2.5 <dc:Publisher> </dc:Publisher>

The publisher as defined by the Dublin Core Metadata Element Set (http://dublincore.org/documents/1999/07/02/dces/).

2.2.6 <dc:Contributor> </dc:Contributor>

A party whose contribution to the publication is secondary to those named in **dc:Creator** elements.

Other than significance of contribution, the semantics of this element are identical to those of **dc:Creator**. Reading Systems are free to choose to display **dc:Creator** information without accompanying **dc:Contributor** information.

This specification adds to the **dc:Contributor** element two optional attributes: *role* and *file-as*. The *file-as* attribute is defined as for **dc:Creator**, and is documented in section 2.2.2.

The normative list of values used for the *role* attribute is defined by the MARC relator code list (http://www.loc.gov/marc/relators/). When roles are specified, the 3-character registered MARC values **must** be used when applicable. Although that list is extensive, other values **may** be added if a required role is not covered by those predefined values. Such values **must** begin with "oth.", and shall be considered subdivisions of the "other" relator code. Like other constructs in this specification, these values are case-sensitive and **must** be coded entirely in lower-case.

For convenience, some relator code values are listed here as examples. Consult the MARC code list cited above for the complete list.

Adapter [adp] Use for a person who 1) reworks a musical composition, usually for a different medium, or 2) rewrites novels or stories for motion pictures or other audiovisual medium.

Annotator [ann] Use for a person who writes manuscript annotations on a printed item.

Arranger [arr] Use for a person who transcribes a musical composition, usually for a different medium from that of the original; in an

arrangement the musical substance remains essentially unchanged.

Artist [art] Use for a person (e.g., a painter) who conceives, and perhaps also implements, an original graphic design or work of art, if specific codes (e.g., [egr], [etr]) are not desired. For book illustrators, prefer Illustrator [ill].

Associated name [asn] Use as a general relator for a name associated with or found in an item or collection, or which cannot be determined to be that of a Former owner [fmo] or other designated relator indicative of provenance.

Author [aut] Use for a person or corporate body chiefly responsible for the intellectual or artistic content of a work. This term may also be used when more than one person or body bears such responsibility.

Author in quotations or text extracts [aqt] Use for a person whose work is largely quoted or extracted in a works to which he or she did not contribute directly. Such quotations are found particularly in exhibition catalogs, collections of photographs, etc.

Author of afterword, colophon, etc. [aft] Use for a person or corporate body responsible for an afterword, postface, colophon, etc. but who is not the chief author of a work.

Author of introduction, etc. [aui] Use for a person or corporate body responsible for an introduction, preface, foreword, or other critical matter, but who is not the chief author.

Bibliographic antecedent [ant] Use for the author responsible for a work upon which the work represented by the catalog record is based. This may be appropriate for adaptations, sequels, continuations, indexes, etc.

Book producer [bkp] Use for the person or firm responsible for the production of books and other print media, if specific codes (e.g., [bkd], [egr], [tyd], [prt]) are not desired.

Collaborator [clb] Use for a person or corporate body that takes a limited part in the elaboration of a work of another author or that brings complements (e.g., appendices, notes) to the work of another author.

Commentator [cmm] Use for a person who provides interpretation, analysis, or a discussion of the subject matter on a recording, motion picture, or other audiovisual medium.

Compiler [com] Use for a person who produces a work or publication by selecting and putting together material from the works of various persons or bodies.

Designer [dsr] Use for a person or organization responsible for design if specific codes (e.g., [bkd], [tyd]) are not desired.

Editor [edt] Use for a person who prepares for publication a work not primarily his/her own, such as by elucidating text, adding

introductory or other critical matter, or technically directing an editorial staff.

Illustrator [ill] Use for the person who conceives, and perhaps also implements, a design or illustration, usually to accompany a written text.

Lyricist [lyr] Use for the writer of the text of a song.

Metadata contact [mdc] Use for the person or organization primarily responsible for compiling and maintaining the original description of a metadata set (e.g., geospatial metadata set).

Musician [mus] Use for the person who performs music or contributes to the musical content of a work when it is not possible or desirable to identify the function more precisely.

Narrator [nrt] Use for the speaker who relates the particulars of an act, occurrence, or course of events.

Other [oth] Use for relator codes from other lists which have no equivalent in the MARC list or for terms which have not been assigned a code.

Photographer [pht] Use for the person or organization responsible for taking photographs, whether they are used in their original form or as reproductions.

Printer [prt] Use for the person or organization who prints texts, whether from type or plates.

Redactor [red] Use for a person who writes or develops the framework for an item without being intellectually responsible for its content.

Reviewer [rev] Use for a person or corporate body responsible for the review of book, motion picture, performance, etc.

Sponsor [spn] Use for the person or agency that issued a contract, or under whose auspices a work has been written, printed, published, etc.

Thesis advisor [ths] Use for the person under whose supervision a degree candidate develops and presents a thesis, memoir, or text of a dissertation.

Transcriber [trc] Use for a person who prepares a handwritten or typewritten copy from original material, including from dictated or orally recorded material.

Translator [trl] Use for a person who renders a text from one language into another, or from an older form of a language into the modern form.

2.2.7 <dc:Date> </dc:Date>

Date of publication, in the format defined by “Date and Time Formats” at http://www.w3.org/TR/NOTE-datetime and by ISO 8601 on which it is based. In particular, dates without times are represented in the form YYYY[-MM[-DD]]: a mandatory 4-digit year, an optional 2-digit month, and if the month is given, an optional 2-digit day of month.

The **dc:Date** element has one optional attribute, *event*. The set of values for *event* are not defined by this specification; possible values may include: **creation**, **publication**, and

modification.

2.2.8 <dc:Type> </dc:Type>

Type includes terms describing general categories, functions, genres, or aggregation levels for content. Recommended best practice is to select a value from a controlled vocabulary.

2.2.9 <dc:Format> </dc:Format>

The media type or dimensions of the resource. Best practice is to use a value from a controlled vocabulary (e.g. MIME media types).

2.2.10 <dc:Identifier> </dc:Identifier>

A string or number used to uniquely identify the resource. An OEBPS Package **must** include at least one instance of this element type, however multiple instances are permitted.

At least one **dc:Identifier** **must** have an *id* specified, so it can be referenced from the package ***unique-identifier*** attribute described in Section 2.1.

The **dc:Identifier** element has an optional attribute defined by this specification: *scheme*. The *scheme* attribute names the system or authority that generated or assigned the text contained within the **dc:Identifier** element, for example “ISBN” or “DOI.” The values of the scheme attribute are case sensitive.

This specification does not standardize or endorse any particular publication identifier scheme. Specific use of URLs or ISBNs is not yet addressed by this specification. Identifier schemes are not currently defined by Dublin Core.

2.2.11 <dc:Source> </dc:Source>

Information regarding a prior resource from which the publication was derived; see the Dublin Core Metadata Element Set (http://dublincore.org/documents/1999/07/02/dces/).

2.2.12 <dc:Language> </dc:Language>

Identifies a language of the intellectual content of the Publication. An OEBPS Package **must** include at least one instance of this element type, however multiple instances are permitted. The content of this element **must** comply with RFC 3066 (see http://www.ietf.org/rfc/rfc3066.txt), or its successor on the IETF Standards Track. The Dublin Core permits other descriptions as well; this specification does not.

2.2.13 <dc:Relation> </dc:Relation>

An identifier of an auxiliary resource and its relationship to the publication.

2.2.14 <dc:Coverage> </dc:Coverage>

The extent or scope of the publication’s content. Recommended best practice is to select a value from a controlled vocabulary; see the Dublin Core Metadata Element Set (http://dublincore.org/documents/1999/07/02/dces/).

2.2.15 <dc:Rights> </dc:Rights>

A statement about rights, or a reference to one. In this specification, the copyright notice and any further rights description **should** appear directly.

This specification does not address the manner in which a Content Provider specifies to a secure distributor any licensing terms under which readership rights or copies of the content may be sold.

2.3 Manifest

The required **manifest** provides a list of all the files that are parts of the publication. The **manifest** element **must** contain one or more **item** elements. Each **item** describes a document, an image file, a style sheet, or other component that is considered part of the publication.

Each **item** element contained within a **manifest** element **must** have the attributes ***id***, ***href*** (a URI; if relative, the URI is interpreted as relative to the package file itself), and ***media-type*** (specifying the item's MIME media type).

The order of **item** elements in the **manifest** is not significant.

For example,

```
<manifest>
      <item id="intro" href="introduction.html"
            media-type="text/x-oeb1-document" />
      <item id="c1" href="chapter-1.html"
            media-type="text/x-oeb1-document" />
      <item id="c2" href="chapter-2.html"
            media-type="text/x-oeb1-document" />
      <item id="toc" href="contents.xml"
            media-type="text/x-oeb1-document" />
      <item id="oview" href="arch.png"
            media-type="image/png" />
</manifest>
```

The URIs in ***href*** attributes of **item** elements in the **manifest** **must not** use fragment identifiers.

2.3.1 Fallback items

This specification defines a set of OEBPS Core Media Types that all conforming Reading Systems **must** support (as required by this specification). For a publication that uses only those media types, the **manifest** merely lists the publication's component files directly. However, content providers **may** construct publications that reference items of additional media types. In order for such publications to be read by all conforming Reading Systems, content providers **must** provide alternative "fallback" items for each such item. For every item that is not an OEBPS Core Media Type, at least one of its associated fallback items **must** be of a type drawn from the set of OEBPS Core Media Types.

This specification defines three different mechanisms for specifying OEBPS Core Media Type fallbacks. First, for inline "replaced" resources referenced via the **object** element, this specification relies on that element's inherent replacement capabilities, described in section 3.3.6. Second, for non-inline destinations, whether referenced from a document or a package,

and for inline “replaced” resources referenced via the **img** element (described in section 3.3.4), the *fallback* attribute of the **item** is used. Third, for inline “replaced” resources referenced via the **img** element, the text value of the *alt* attribute provides a valid fallback.

An **item** identifies a fallback item using its *fallback* attribute, which **must** specify the ID of the **item** element that identifies the fallback. Items referenced from *fallback* attributes may each specify a *fallback* attribute in turn, forming a longer “fallback path.” For example,

```
<manifest>
      <item id="item1"
            href="FunDoc.txt"
            media-type="text/plain"
            fallback="fall1" />
      <item id="fall1" fallback="fall2"
            href="FunDoc.html"
            media-type="text/html" />
      <item id="fall2"
            href="FunDoc.oeb"
            media-type="text/x-oeb1-document" />
      <item ...>
</manifest>
```

If a *fallback* attribute points to an **item** that also has a *fallback* attribute, a Reading System **must** continue down the fallback path until it reaches a reference to an **item** of a media type it can display. A Reading System **may** continue further, and **may** display any item from the chain. In the absence of element-specific (i.e. **img** and **object**) fallback information, every item in a publication that is not of one of the OEBPS Core Media Types **must**, directly or indirectly, specify a fallback path to an item of one of the OEBPS Core Media Types.

Fallback paths **must** terminate; circular references are not permitted. Nevertheless, Reading Systems **should not** fail catastrophically if they encounter such a loop.

2.4 Spine

Following the **manifest**, there **must** be one **spine** element, which defines a primary linear reading order of the publication. It specifies an ordered list of one or more OEBPS Documents drawn from the manifest, using **itemref** elements contained within the **spine** element.

A publication **must** specify exactly one **spine**. Reading Systems **must** treat the file named in the first **itemref** element within the **spine** as the first file to be rendered in the reading of the book. The successive files named in its **itemref** elements are those that are to be rendered using “next-page”-type functionality that may be available in the Reading System.

The **spine** **must** refer only to **item** elements of media type `text/x-oeb1-document`. Content of other media types **may** be referenced via OEBPS Documents, which **should** provide text alternates and other information to enhance accessibility as appropriate.

The **spine** need not include references to every one of the manifest’s **item** elements that reference OEBPS Documents, because there are means other than the **spine** for accessing documents in the publication. For example, hypertext links **may** provide access to documents not in the **spine**, as **may** tours and guides (see below).

For example,

```
<manifest>
      <item id="toc"
            href="contents.html"
            media-type="text/x-oeb1-document" />
      <item id="c1"
            href="chap1.html"
            media-type="text/x-oeb1-document" />
      <item id="c2"
            href="chap2.html"
            media-type="text/x-oeb1-document" />
      <item id="c3"
            href="chap3.html"
            media-type="text/x-oeb1-document" />
      <item id="footnotes"
            href="footnotes.html"
            media-type="text/x-oeb1-document" />
      <item id="f1" href="fig1.jpg" media-type="image/jpeg" />
      <item id="f2" href="fig2.jpg" media-type="image/jpeg" />
      <item id="f3" href="fig3.jpg" media-type="image/jpeg" />
</manifest>

<spine>
      <itemref idref="toc" />
      <itemref idref="c1" />
      <itemref idref="c2" />
      <itemref idref="c3" />
</spine>
```

In the above example, suppose the document referenced by ID “c1” is being viewed by a reader. When the end of that document is reached, the next document in linear order would be that referenced by ID “c2”. Document “c1” might also have hypertext links to locations in another file such as the “footnotes”. Such a file **must** be listed in the **`manifest`**, but **need not** be named by any **`itemref`** of the **`spine`**. If a reader follows the hyperlink in “c1” to “footnotes”, and the end of that file is reached, then no successor in linear order is defined by this specification.

2.5 Tours

Much as a tour-guide might assemble points of interest into a set of sightseers’ tours, a content provider may assemble selected parts of a publication into a set of tours to enable convenient navigation.

An OEBPS Package **may, but need not**, contain one **`tours`** element, which in turn contains one or more **`tour`** elements. Each **`tour`** **must** have a *`title`* attribute, intended for presentation to the user. Reading Systems may use tours to provide various access sequences to parts of the publication, such as selective views for various reading purposes, reader expertise levels, etc. Because Reading Systems are not required to implement tour support, content providers **should** also provide other means of accessing content referenced from tours.

Each **`tour`** element contains one or more **`site`** elements, each of which **must** have an *`href`* attribute and a *`title`* attribute. The *`href`* attribute **must** refer to an OEBPS Document included in the manifest, and **may** include a fragment identifier as defined in section 4.1 of RFC 2396 (see http://www.ietf.org/rfc/rfc2396.txt). Each **`site`** element specifies a starting point from which the reader may explore freely. Reading Systems **may** use the bounds of the

referenced element to determine the scope of the site. If a fragment identifier is not used, the scope is considered to be the entire document. This specification does not require Reading Systems to mark or otherwise identify the entire scope of a referenced element. The order of **site** elements is presumed to be significant, and **should** be used by Reading Systems to aid navigation.

Example:

```
<tours>
      <tour id="tour1" title="Chicken Recipes">
            <site title="Chicken Fingers"
             href="appetizers.html#r3" />
            <site title="Chicken a la King"
             href="entrees.html#r5" />
      </tour>
      <tour id="tour2" title="Vegan Recipes">
            <site title="Hummus" href ="appetizer.html#r6" />
            <site title="Lentil Casserole" href="lentils.html" />
      </tour>
</tours>
```

2.6 Guide

Within the **package** there **may** be one **guide** element, containing one or more **reference** elements. The **guide** element identifies fundamental structural components of the publication, to enable Reading Systems to provide convenient access to them.

Example:

```
<guide>
      <reference type="toc" title="Table of Contents"
       href="toc.html" />
      <reference type="loi" title="List Of Illustrations"
       href="toc.html#figures" />
      <reference type="other.intro" title="Introduction"
       href="intro.html" />
</guide>
```

The structural components of the books are listed in **reference** elements contained within the **guide** element. These components may refer to the table of contents, list of illustrations, foreword, bibliography, and many other standard parts of the book. Reading Systems are **not required** to use the **guide** element in any way.

Each reference **must** have an *href* attribute referring to an OEBPS Document included in the manifest, and which **may** include a fragment identifier as defined in section 4.1 of RFC 2396 (see http://www.ietf.org/rfc/rfc2396.txt). Reading Systems **may** use the bounds of the referenced element to determine the scope of the reference. If a fragment identifier is not used, the scope is considered to be the entire document. This specification **does not require** Reading Systems to mark or otherwise identify the entire scope of a referenced element.

The required *type* attribute describes the publication component referenced by the *href* attribute. The values for the *type* attributes **must** be selected from the list defined below when applicable. Other types **may** be used when none of the predefined types are applicable; their names **must** begin with the string “other.”. The value for the *type* attribute is case-

sensitive.

The following list of *type* values is derived from the 13th edition of the *Chicago Manual of Style*:

cover	the book cover(s), jacket information, etc.
title-page	page with possibly title, author, publisher, and other metadata
toc	table of contents
index	back-of-book style index
glossary	glossary
acknowledgements	
bibliography	
colophon	
copyright-page	
dedication	
epigraph	
foreword	
loi	list of illustrations
lot	list of tables
notes	
preface	

3 Basic OEBPS Document Vocabulary

3.1 Introduction

OEBPS 1.0 provided document authors with a convenient “Basic” document vocabulary (a set of elements and attributes, the “tagset”) that all OEB Reading Systems **must** recognize. This vocabulary was selectively drawn from the HTML 4.01 tagset, essentially conforming to XHTML 1.0 Transitional. A Document Type Definition (DTD) of the Basic vocabulary (the “OEBPS 1.0 Document DTD”) was provided for optional validation purposes, to insure Basic OEBPS Documents conformed to the recommended content models and the allowed attribute values of the vocabulary.

This specification similarly continues support for a “Basic” document vocabulary which all OEBPS 1.2 Reading Systems **must** recognize.

The Basic OEBPS 1.2 Document vocabulary is a pure subset of XHTML 1.1 from which the elements and attributes selected for inclusion are listed in the table in Section 3.2.2. Appendix B includes the Basic OEBPS 1.2 Document DTD expressing the Basic OEBPS Document vocabulary (and is in strict conformance with the XHTML 1.1 DTD with modularization removed). Appendix C includes the mnemonic character entities file associated with the Basic OEBPS 1.2 Document DTD. Appendix D describes the differences between the Basic OEBPS 1.2 and 1.0.1 Document vocabularies.

All Basic OEBPS Documents that validate to the Basic OEBPS 1.2 Document DTD will also validate to the XHTML 1.1 DTD. It is strongly recommended that all Basic OEBPS Documents be valid XML documents with respect to the Basic OEBPS Document DTD.

Except where noted in this section and elsewhere, the semantics and expected rendering behavior of the Basic OEBPS 1.2 Document vocabulary are as defined in XHTML 1.1. XHTML 1.1 relies heavily upon HTML 4.01 for semantic definitions and expected User Agent rendering behavior (http://www.w3.org/TR/html401/).

3.2 Basic OEBPS Document Vocabulary Components

3.2.1 The Common Attributes

The Basic OEBPS Document vocabulary, following XHTML 1.1, defines five Common attributes that may be applied to nearly all the elements in the Basic OEBPS Document vocabulary. These *[Common]* attributes consist of *xml:lang* and the *[Core]* attributes *id*, ~~*style*~~, *class*, and *title*. These attributes are not individually listed in the element and attribute list in the following section 3.2.2, except to note their absence from the few exceptional elements.

These Common attributes **may** also be applied to non-Basic elements in Extended OEBPS Documents.

Because of their general importance, certain usage restrictions, and Reading System conformance issues, they are further described below. Except where further restricted, the data types for the attribute values conform with XHTML 1.1 (and the Basic OEBPS Document DTD in Appendix B.)

3.2.1.1 id

This attribute is used to give a unique identifier to an element. Its value **must** be of the XML data type ID with the token “Name” (the normative syntax of “Name” is precisely defined in section 2.3 of the XML 1.0 specification.)

Values for *id* **must** be unique across all elements in a single document. In addition, the value of *id* **should not** start with the string 'xml' (and all its case variants), since this is reserved in the XML specification for possible future standardization.

In this specification, the value of *id* **must** start with a “Letter” – it cannot start with an underscore (_) or colon (:) as otherwise allowed in XML 1.0. The character set defined by “Letter” is specified in Appendix B of the XML 1.0 specification.

For general HTML compatibility, document authors **should** further restrict the first character value of *id* to the Basic Latin letter characters (A-Za-z) and the remaining characters to (A-Za-z0-9.-_).

3.2.1.2 ~~style~~ *(deprecated)*

The core attribute ~~*style*~~, used to apply CSS styling directly to an element, is deprecated in this specification as it is in XHTML 1.1.

It is strongly recommended the ~~*style*~~ attribute not be used in OEBPS 1.2 Documents; instead use the **style** element or preferably an external style sheet to specify the styling of any element.

3.2.1.3 `class`

This attribute allows selector-based style specifications. Its value **must** be a space-separated list of class names.

3.2.1.4 `title`

This attribute **may** be used to provide an "advisory title/amplification" for the element. Reading Systems **may** ignore its value.

3.2.1.5 `xml:lang`

This attribute **may** be inserted in documents to specify the language used in the contents and attribute values of any element in an XML document. The attribute value of *`xml:lang`* **must** comply with RFC 3066 (see http://www.ietf.org/rfc/rfc3066.txt), or its successor on the IETF Standards Track.

3.2.2 Elements and Attributes of the Basic OEBPS Document Vocabulary

This section lists all the elements and associated attributes included in the Basic OEBPS 1.2 document vocabulary. They are drawn from the XHTML 1.1 vocabulary specified at http://www.w3.org/TR/xhtml11/.

Refer to the Basic OEBPS Document DTD (Appendix B), the XHTML 1.1 specification, and section 3.3 for attribute value and other restrictions.

Table Notes:

(i) The FIXED attribute of *`xmlns`* is currently declared for the root element <html>, with the value of **http://www.w3.org/1999/xhtml**. The FIXED attribute of *`xml:space`* has the value of **preserve**.

(ii) The "May Contain" category summarizes, for conformance with XHTML 1.1, the children elements and/or PCDATA ("parsed character data") the element can (and in a few cases must) contain. The XHTML 1.1 content model is reproduced in the Basic OEBPS Document DTD (Appendix B). As mentioned in Section 3.1, it is **strongly recommended** that any Basic OEBPS Document that is not valid XML with respect to the Basic OEBPS Document DTD, still follow the XHTML 1.1 content model.

Element	Short Description	Supported Attributes	Document Structure Level	May Contain (XHTML 1.1)
`a`	Anchor	*`[Common]`*, *`href`*, *`rel`*, *`rev`*	Inline	PCDATA; **`[Inline]`** (except **`a`**); **`[BlockOrInline]`**
`abbr`	Abbreviation	*`[Common]`*	Inline	PCDATA; **`[Inline]`**; **`[BlockOrInline]`**
`acronym`	Acronym	*`[Common]`*	Inline	PCDATA; **`[Inline]`**; **`[BlockOrInline]`**

address	Address	*[Common]*	Block	PCDATA; **[Inline]**; **[BlockOrInline]**
area	Client-Side Image Map Area	*[Common]*, ***alt***, *coords*, *href*, *nohref*, *shape*	Miscellaneous	[Empty]
b	Bold Text Style	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
base	Document Base URI	***href***	Head	[Empty]
big	Large Text Style	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
blockquote	Long Quotation	*[Common]*, *cite*	Block	**[Block]**; **[BlockOrInline]**
body	Document Body	*[Common]*	Top	**[Block]**; **[BlockOrInline]**
br	Forced Line Break	*[Core]*	Inline	[Empty]
caption	Table Caption	*[Common]*	Table	PCDATA; **[Inline]**; **[BlockOrInline]**
cite	Citation	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
code	Computer Code Fragment	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
col	Table Column	*[Common]*, *align*, *span*, *valign*, *width*	Table	[Empty]
colgroup	Table Column Group	*[Common]*, *align*, *span*, *valign*, *width*	Table	**col**
dd	Definition Description	*[Common]*	List	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
del	Deleted Text	*[Common]*, *cite*, *datetime*	Block Or Inline	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
dfn	Instance Definition	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**

div	Generic Block Level Container	*[Common]*	Block	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
dl	Definition List	*[Common]*	Block (List)	**dd**; **dt**
dt	Definition Term	*[Common]*	List	PCDATA; **[Inline]**; **[BlockOrInline]**
em	Emphasis	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
h1 to **h6**	Heading	*[Common]*	Block	PCDATA; **[Inline]**; **[BlockOrInline]**
head	Document Head	*xml:lang*	Top	**[Head]**; **object**; **script**
hr	Horizontal Rule	*[Common]*	Block	[Empty]
html	Document Root Element	*xmlns, xml:lang*	Top (Document Root)	**head**, **body**
i	Italic Text Style	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
img	Embedded Image	*[Common]*, ***alt***, *height*, *longdesc*, ***src***, *usemap*, *width*	Inline	[Empty]
ins	Inserted Text	*[Common]*, *cite*, *datetime*	Block Or Inline	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
kbd	Text Entered by the User	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
li	List Item	*[Common]*	List	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
link	Media-Independent Link	*[Common]*, *href*, *media*, *rel*, *rev*, *type*	Head	[Empty]
map	Client-Side Image Map	*[Common]* (***id*** is required)	Inline	**[Block]**; **[BlockOrInline]**; **area**
meta	Generic Metadata Information	***content***, *name*, *scheme*,	Head	[Empty]

		xml:lang		
noscript	Fallback Content For Non-Executable Script	*[Common]*	Block Or Inline	**[Block]**; **[BlockOrInline]**
object	Generic Embedded Object	*[Common]*, *archive*, *classid*, *codebase*, *codetype*, *data*, *height*, *type*, *usemap*, *width*	Inline	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**; **param**
ol	Ordered List	*[Common]*	Block (List)	**li**
p	Paragraph	*[Common]*	Block	PCDATA; **[Inline]**; **[BlockOrInline]**
param	Named Property Value	*id*, ***name***, *type*, *value*, *valuetype*	Miscellaneous	[Empty]
pre	Preformatted Text	*[Common]*, *xml:space*	Block	PCDATA; **script**; **[Inline]** except **big**, **img**, **object**, **small**, **sub**, **sup**
q	Inline Quotation	*[Common]*, *cite*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
samp	Program, Script, and Similar Output	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
script	Script Statements	***type***, *xml:space*	Block Or Inline	PCDATA
small	Small Text Style	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
span	Generic Inline Level Container	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
strong	Strong Emphasis	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
style	Style Information	*title*, ***type***, *xml:lang*, *xml:space*	Head	PCDATA
sub	Subscript	*[Common]*	Inline	PCDATA; **[Inline]**;

				[BlockOrInline]
sup	Superscript	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
table	Table	*[Common]*, *border*, *cellpadding*, *cellspacing*, *summary*, *width*	Block (Table)	**caption**; **col**; **colgroup**; **tbody**; **thead**; **tfoot**; **tr**
tbody	Table Body	*[Common]*, *align*, *valign*	Table	**tr**
td	Table Data Cell	*[Common]*, *abbr*, *align*, *colspan*, *rowspan*, *valign*	Table	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
tfoot	Table Footer	*[Common]*, *align*, *valign*	Table	**tr**
th	Table Header Cell	*[Common]*, *abbr*, *align*, *colspan*, *rowspan*, *valign*	Table	PCDATA; **[Block]**; **[Inline]**; **[BlockOrInline]**
thead	Table Header	*[Common]*, *align*, *valign*	Table	**tr**
title	Document Title	*xml:lang*	Head	PCDATA
tr	Table Row	*[Common]*, *align*, *valign*	Table	**td**; **th**
tt	Teletype or Monospaced Text	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**
ul	Unordered List	*[Common]*	Block (List)	**li**
var	Instance of a Variable or Program Argument	*[Common]*	Inline	PCDATA; **[Inline]**; **[BlockOrInline]**

3.3 Certain Element and Attribute Semantic Differences From, and Restrictions Beyond, XHTML 1.1

As noted in Section 3.1, the semantics and rendering behavior of the Basic OEBPS Document vocabulary (elements, attributes, and associated attribute values) strictly follows that of XHTML 1.1. However, there are several restrictions beyond that of XHTML 1.1, as noted

below. These restrictions have no effect on the XHTML 1.1 conformance of Basic 1.2 documents.

3.3.1 General Comments on URI References

A number of attributes reference resources using URI values (Uniform Resource Identifier, see RFC 2396, http://www.ietf.org/rfc/rfc2396.txt). Depending on the particular attribute, the URI referenced resource can either be an abstract entity or a physical object.

Except where noted or where not applicable, Reading Systems **may** use or render a URI referenced physical resource not listed in the Manifest (i.e., it is not a component of the Publication), but they are **not required** to do so.

3.3.2 body element

It is assumed, in formatting, that the default rendering for **body** is consistent with the CSS property **page-break-before** having been set to *right* (which behaves like *always* on one-page Reading Systems), but may be overridden by an appropriate style sheet declaration.

3.3.3 *cite* attribute

The optional attribute *cite* can be used in **blockquote**, **q**, **del** and **ins** to provide a URI citation for the element contents. Reading Systems are **not required** to process or use the referenced URI resource, whether or not the resource is listed in the Manifest.

3.3.4 img element

The inline element **img should only** be used to refer to images of OEBPS Core Media Types of PNG (http://www.ietf.org/rfc/rfc2083.txt) and JPG/JFIF (http://www.w3.org/Graphics/JPEG). The required URI attribute, *src*, is used to reference the image resource, which must be listed in the Manifest.

The **required** *alt* attribute **should** contain a brief and informative textual description of the image. This text **may** be used by Reading Systems as an alternative to, or in addition to, displaying the image. The text is also an acceptable fallback for an **img** with *src* referencing a non-OEBPS Core Media Type for which no viable fallback was found in the **manifest**. The *alt* textual description is useful for Reading Systems having limited resolution displays, or for non-visual presentation. Use of the **object** element is the preferred mechanism for including non-core media types in an OEBPS Document.

For greater accessibility, it is **strongly recommended** that OEBPS Document authors include a URI reference in the optional *longdesc* attribute referencing a resource (such as another OEBPS Document in the Publication) describing the image in finer detail. Reading System developers are also strongly urged to recognize and render in an appropriate fashion (and with accessibility in mind) the resource specified in *longdesc*. For further information on the use of this attribute and related accessibility attributes, see http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/#gl-provide-equivalents.

3.3.5 link element

The link element allows for the specification of various relationships with other documents. Reading Systems must recognize external style sheet references specified via the *href* attribute and the associated *rel* attribute (for the values *rel="stylesheet"* and *rel="alternate stylesheet"*.)

Reading Systems **may** ignore the *media* attribute, used to indicate the intended destination for style information.

3.3.6 object and param elements

The **object** element is the preferred method for generic object inclusion. When adding objects whose data media type is not drawn from the OEBPS Core Media Type list or which reference an object implementation using the *classid* attribute, the **object** element must specify fallback information for the object, such as another **object**, an **img** element, or descriptive text. Inline fallback information is provided as OEBPS content appearing immediately after the final **param** element that refers to the parent **object**. Descriptive text for the object, using inline content, an included OEBPS Document, or some other method, should be provided to allow access for people who are not able to access non-textual content.

The *classid* attribute for **object** gives the URI value of an implementation for the object – conformant Reading Systems are **not required** to render objects that use external implementations, although they **may** do so. The MIME media type values for the *codetype* and *type* attributes must match those specified in the Publication's Manifest.

The associated **param** empty element is used to specify initialization values for objects. The **param** element may only appear before the renderable content of an **object**. Reading Systems may examine only **param** elements that are direct children of the **object**.

Example:

```
<object classid="java:tictactoe" codetype="application/java">
      <param name="height" value="60" />
      <param name="width"  value="60" />
      <object type="image/png" data="tictactoe.png">
            <param name="height" value="60" />
            <param name="width"  value="60" />
            Tic-Tac-Toe, a <em>dull</em> game.
      </object>
</object>
```

3.3.7 script and noscript elements

Reading Systems **must not**, by default, render the textual content of the **script** element, but **may** choose to execute the script itself. To render the textual content of the **script** element, this specification recommends using the CSS **display** property to override the default *none* setting.

If **noscript** is included, whose purpose is to display some message should the Reading System not choose to execute the **script**, it must appear after the closing tag of the **script** it is associated with. Reading Systems must, by default, render the content contained in **noscript** if they cannot execute **script**, the default of which can be overridden by CSS **display**:*none*. Note that for XHTML 1.1 conformance the content model for **noscript** is Block.mix (Block level elements plus the "level-independent" elements); it cannot directly contain PCDATA or inline elements and is identical to the content model for **body** and **blockquote**, even if **noscript** itself appears inline.

The attribute *type*, which specifies the scripting language for **script**, is **required**.

One potential problem with **script**, whose content model is PCDATA, is that if the code contains the characters “<” and “&”, there is a potential conflict with XML. Thus, these

characters, if used, either must be escaped, or put into a CDATA section. Reading System developers who include certain **script** execution capability must be aware of this potential problem.

3.3.8 *type* attribute of the style element

The *type* attribute of the **style** element is **required** (per XHTML 1.1 requirements) and **must** be given the value of "text/x-oeb1-css". For browser rendering of an individual OEBPS Document as an XHTML 1.1 document, the value of the *type* attribute may need to be changed to "text/css" for the styling information in **style** to be recognized by the browser.

3.3.9 value of *align* attribute

The value of **char** for the *align* attribute is not included in the Basic OEB 1.2 Document Vocabulary. To achieve similar formatting, use the CSS **text-align** property with a <string> value.

3.4 Rendering of Documents on Reading Systems

A number of elements and attributes permit semantics that are not required of all OEBPS Reading Systems. For example, some devices may be monochrome, or provide mainly audio or tactile interfaces. In such cases this specification generally requires Reading Systems to accept all syntax (such as attribute values) permitted for the XHTML vocabulary, but does not require that they be honored. For example, a Reading System must parse and recognize the *border* attribute on **table** elements, but may choose to treat all values other than 0 the same as 1.

Note that this specification does not mandate specific rendering behavior for the Basic OEBPS Document vocabulary. Some Reading Systems may choose to express the intent of elements in presentation by closely following web-browser usage – a blank line before a paragraph, but no first-line text-indent, for example. Other Reading Systems may gear their presentation towards sustained novel-like readability: for example, no extra whitespace between paragraphs, but text-indent on the first line of each. Still other systems, such as speech generators, may present particular elements or entire documents in completely different ways.

4 OEBPS Style sheets

Like CSS style sheets, OEBPS style sheets are case-insensitive, except for parts that are not under the control of CSS. In particular, OEBPS Documents are XML documents and, as such, their element names and attribute values are case-sensitive. Therefore, element names and attribute values in OEBPS style sheets are case-sensitive. Currently, this applies only to element names, attribute names and attribute values used in selectors.

Where there are differences in the syntax specified by CSS1 and CSS2, OEBPS style sheets follow the CSS2 syntax. A list of these differences can be found in section D3 of the W3C Recommendation REC-CSS2-19980512, "Cascading Style Sheets, level 2 (CSS2) Specification" (http://www.w3.org/TR/REC-CSS2/grammar.html#tokenizer-diffs). OEBPS style sheets support the CSS construct of multiple declarations separated by semi-colons. Hence, the style sheet rules:

```
h1 { color: blue }
h1 { font-weight: bold }
h1 { font-size: 12pt }
```

are equivalent to:

```
h1 {  color: blue;
      font-weight: bold;
      font-size: 12pt }
```

Multiple rules with identical declaration blocks may be combined into one rule by separating the selectors with commas. Thus the rules:

```
h1 {text-indent: 0em}
h2 {text-indent: 0em}
h3 {text-indent: 0em}
```

may be combined into the equivalent form:

```
h1, h2, h3 {text-indent: 0em}
```

OEBPS style sheets support all CSS white space characters. Specifically, the characters “space” (Unicode code 32), “tab” (9), “line feed” (10), “carriage return” (13), and “form feed” (12) can occur in whitespace. Comments of the syntax defined in the CSS2 specification **may** be used in OEBPS-conforming CSS style sheets.

This specification supports the inline ~~*style*~~ attribute, the XHTML **style** element, and externally linked style sheets. This specification does **not require** that any handling of XML namespaces be performed by the Reading System in the processing of style sheets.

This specification assumes the use of selectors to be consistent with the definitions in the CSS 2 Specification (see http://www.w3.org/TR/REC-CSS2/selector.html for details). For example, the rules for determining which of multiple rules should be applied are determined by the rules of inheritance, cascading and selector specificity (see http://www.w3.org/TR/CSS2/cascade.html for details).

This specification does not require support for all CSS 2 selector forms; specifically, it does not require id-based selectors, or selectors that qualify element types using pseudo-classes. This specification does include pseudo elements, however, as described in this chapter.

If no style sheet is defined or no applicable style is found for a given Basic OEBPS 1.2 element, XHTML rendering is the default as defined elsewhere in this specification and the XHTML 1.1 specification.

This specification does **not require** that Reading Systems implement text-to-speech or other read-aloud technology. Those Reading Systems that do not implement such technology **may** ignore any CSS properties listed in this specification under the classification “Aural style sheets,” as well as the **speak-header** property listed under “Tables.”

All properties apply to elements as defined in CSS. That is, most properties can apply to all elements, while a few are limited based on the value of the **display** property (for example, **text-align** only applies when the display type is *block*, not *inline*). Reading Systems are, however, **not required** to support every distinction; for example they **may** choose to map a specific <length> value for border-width to one of the named values for that property.

4.1 Selectors

Selectors specify the patterns that must be matched in the target document for determining the elements to which the style declaration(s) in the accompanying declaration block apply. If all conditions in the pattern are true for a certain element, the selector matches the element and the declarations in the declaration block are applied. This specification assumes a use of selectors that is consistent with the CSS 2 Specification and, in some cases, adds additional constraints to OEBPS style sheet selectors.

The following table lists all CSS selectors that are allowed in OEBPS style sheets. Any selectors not listed in this table are not supported by OEBPS style sheets and **must** be treated as syntax errors by conforming Reading Systems, even if they would otherwise be legal CSS selectors. Errors in selectors **must** be treated as specified in CSS 2 section 4.1.7.:

Pattern	Meaning	CSS2 section
*	Matches any element.	5.3, Universal Selectors
E	Matches any E element (i.e., an element of type E).	5.4, Type Selectors
E F	Matches any F element that is a descendant of an E element.	5.5, Descendant Selectors
E > F	Matches any F element that is a child of an element E.	5.6, Child Selectors
E + F	Matches any F element immediately preceded by an element E.	5.7, Adjacent Sibling Selectors
E[foo]	Matches any E element with the “foo” attribute set (whatever the value).	5.8, Attribute Selectors
E[foo="warning"]	Matches any E element whose “foo” attribute value is exactly equal to “warning”.	5.8, Attribute Selectors
E[foo~="warning"]	Matches any E element whose “foo” attribute value is a list of space-separated values, one of which is exactly equal to “warning”.	5.8, Attribute Selectors
E.warning	Same as E[class~="warning"].	5.8, Attribute Selectors
E:first-line	Matches the first line of the block-level element E.	5.12.1, Pseudo Selectors
E:first-letter	Matches the first character of the block-level element E.	5.12.2, Pseudo Selectors
E:before	Generates content before the element E.	5.12.3, Pseudo Selectors
E:after	Generates content after the element E.	5.12.3, Pseudo Selectors
:link	Matches hyperlink source anchors.	5.11.2, Pseudo

Pattern	Meaning	CSS2 section
		Selectors

4.2 Value types

4.2.1 URI values

For those properties that take a URI value, the URI must point to a document of appropriate media type for the property in question. All such referenced documents must be contained within the package's manifest.

4.2.2 Integers and real numbers

Real numbers are denoted by <number>, integer values by <integer>. Either may have an optional sign value (one of "+" or "-"), though particular properties may restrict the ranges and sign of numeric values.

4.2.3 Length

All non-zero coordinate and size values must have specified units. All units defined by CSS 1 and 2 are supported:

`px`	Pixels
`ex`	x-height of current font
`em`	font-size of current font
`pt`	Points
`in`	Inches
`cm`	Centimeters
`mm`	Millimeters
`pc`	Picas

4.2.4 Percentages

Where percentage units are supported, they are used as defined for each property in the CSS specifications for which they are an allowed value.

4.2.5 Color

Current browsers support a host of keyword color names. XHTML 1.1 defines 16 named colors, as well as numeric values. OEBPS style sheets may use all CSS 1 forms. However,

Reading Systems are **not required** to distinguish all these colors for rendering (otherwise monochrome devices would necessarily be non-conforming, which is not the intent).

black

white

aqua

blue

fuchsia

gray

green

lime

maroon

navy

olive

purple

red

silver

teal

yellow

#rrggbb	six-digit hexadecimal
#rgb	three-digit hexadecimal
rgb(r, g, b)	integers in the range 0-255
rgb(r%, g%, b%)	floats in the range of 0.0% to 100.0%

4.2.6 Time

Units defined by CSS 2 are supported:

s	Seconds
ms	Milliseconds

4.2.7 Frequency

Units defined by CSS 2 are supported:

Hz	Hertz
kHz	Kilohertz

4.2.8 Strings

Strings must be quoted using either single or double quotes (Unicode codes 39 or 34, respectively). Nested strings must be escaped with a backslash (e.g. " a \"nested\" string") To embed a line break in a string, use the escape "\A". The hexadecimal "A" is the line feed character in Unicode, but represents the generic notion of "newline" in CSS.

4.3 Properties

Default values for all supported CSS properties are as listed in CSS2.

The following table lists all CSS properties and values supported by this specification. Where not all values given in the CSS2 specification are listed for a given property, those values not listed are not supported by this specification. The column "Alternate display" indicates acceptable fallback display for CSS values that a Reading System cannot display as intended.

Properties that are unique to this specification have been underlined.

CSS structure	Alternate display	CSS2 section
Media types		7
@media		7.2.1
aural		7.3
all		7.3
Page model		13.2
@page		13.2
:left		13.2.4
:right		13.2.4
:first		13.2.4
Box model		8
Margins		8.3
margin-top, margin-bottom, margin-left, margin-right		8.3
<length>		
<percentage>		
margin [2]		8.3
auto	0 [1]	
Padding		8.4
padding-top, padding-bottom, padding-left, padding-right		8.4
<length>		

CSS structure	Alternate display	CSS2 section
<percentage>		
padding [2]		8.4
Borders		8.5
border-top-width, border-bottom-width, border-left-width, border-right-width		8.5.1
thin		
medium		
thick		
<length>	*thin/medium/thick* [3]	
border-width [2]	*thin/medium/thick* [3]	8.5.1
border-top-color, border-bottom-color, border-left-color, border-right-color		8.5.2
<color>	[4]	
transparent		
border-color [2]		8.5.2
border-top-style, border-bottom-style, border-left-style, border-right-style		8.5.3
none		
hidden		
dotted	*solid*	
dashed	*solid*	
solid		
double	*solid*	
groove	*solid*	
ridge	*solid*	
inset	*solid*	
outset	*solid*	
border-style [2]		8.5.3
border-top, border-bottom, border-left, border-right [2]		8.5.4
border [2]		8.5.4
Visual display model		9
display [5]		9.2.5
none		

CSS structure	Alternate display	CSS2 section
inline		
block		
run-in		
table		
inline-table		
table-row-group		
table-header-group		
table-footer-group		
table-column-group		
table-row		
table-column		
table-cell		
table-caption		
inherit		
oeb-page-head [6]		
oeb-page-foot [6]		
float		9.5.1
left		
right		
none		
inherit		
Clear		9.5.2
none		
left		
right		
both		
inherit		
direction		9.10
ltr		
rtl		
inherit		
unicode-bidi		9.10
normal		

CSS structure	Alternate display	CSS2 section
embed		
bidi-override		
inherit		
oeb-column-number[13]		
auto		
<integer>	1	
Visual formatting model details		10
width		10.2
<length>		
<percentage>		
auto		
inherit		
min-width		10.4
<length>		
<percentage>		
inherit		
max-width		10.4
<length>		
<percentage>		
auto		
inherit		
Height		10.5
<length>		
<percentage>		
auto		
inherit		
min-height		10.7
<length>		
<percentage>		
inherit		
max-height		10.7
<length>		
<percentage>		

CSS structure	Alternate display	CSS2 section
none		
inherit		
line-height		10.8.1
normal		
<number>		
<length>		
<percentage>		
inherit		
vertical-align		10.8.1
baseline		
sub		
super		
top		
text-top	[7]	
middle		
bottom		
ext-bottom	[8]	
inherit		
Generated content, automatic numbering, and lists		12
content [9]		12.2
<string>		
inherit		
list-style-type		12.6.2
none		
disc		
circle		
square		
decimal		
decimal-leading-zero		
lower-roman		
upper-roman		
lower-greek	decimal	
upper-greek	decimal	

CSS structure	Alternate display	CSS2 section
lower-alpha		
lower-latin		
upper-alpha		
upper-latin		
hebrew	decimal	
armenian	decimal	
georgian	decimal	
cjk-ideographic	decimal	
hiragana	decimal	
katakana	decimal	
hiragana-iroha	decimal	
katakana-iroha	decimal	
inherit		
list-style-position		12.6.2
inside		
outside		
inherit		
list-style [2]		12.6.2
Paged media		13
page-break-before		13.3.1
auto		
always		
avoid		
left	[10]	
right	[10]	
inherit		
page-break-after		13.3.1
auto		
always		
avoid		
left	[10]	
right	[10]	
inherit		

CSS structure	Alternate display	CSS2 section
page-break-inside		13.3.1
auto		
avoid		
inherit		
orphans		13.3.3
<integer>		
inherit		
widows		13.3.3
<integer>		
inherit		
Colors and Backgrounds		14
color		14.1
<color>	[4]	
inherit		
background-color		14.2.1
<color>	[4]	
transparent		
inherit		
Fonts		15
font-family		15.2.2
<family-name>		
sans-serif		
serif		
monospace		
inherit		
font-style		15.2.3
normal		
italic	[11]	
oblique	[11]	
inherit		
font-variant		15.2.3
normal		
small-caps		

CSS structure	Alternate display	CSS2 section
font-weight		15.2.3
normal		
bold		
100–900	[3]	
inherit		
-size		15.2.4
xx-small		
x-small		
small		
medium		
large		
x-large		
xx-large		
smaller		
larger		
<length>	[3]	
<percentage>	[3]	
inherit		
font [2]		15.2.5
Text		16
text-indent		16.1
<length>		
<percentage>		
inherit		
text-align		16.2
left		
right		
center		
justify		
inherit		
text-decoration		16.3.1
none		
line-through		

CSS structure	Alternate display	CSS2 section
underline		
inherit		
white-space		16.6
normal		
pre		
nowrap		
inherit		
Tables		17
caption-side		17.4.1
top		
bottom		
left		
right		
inherit		
table-layout		17.5.2
fixed		
auto		
inherit		
speak-header		17.7.1
once		
always		
inherit		
Aural style sheets		19
volume		19.2
silent		
x-soft		
soft		
medium		
loud		
x-loud		
<percentage>	[3]	
0–100	[3]	

CSS structure	Alternate display	CSS2 section
inherit		
speak		19.3
normal		
none		
spell-out		
inherit		
pause-before		19.4
<time>		
<percentage>		
inherit		
pause-after		19.4
<time>		
<percentage>		
inherit		
pause [2]		19.4
cue-before		19.5
<uri>		
none		
inherit		
cue-after		19.5
<uri>		
none		
inherit		
cue [2]		19.5
speech-rate		19.8
x-slow		
slow		
medium		
fast		
x-fast		
faster		
slower		
<number> [12]		

CSS structure	Alternate display	CSS2 section
inherit		
voice-family		19.8
male		
female		
child		
inherit		
pitch		19.8
x-low		
low		
medium		
high		
x-high		
<frequency>		
inherit		
stress		19.8
0–100		
inherit		
richness		19.8
0–100		
inherit		
speak-punctuation		19.9
code		
none		
inherit		
speak-numeral		19.9
digits		
continuous		
inherit		

[1] Reading Systems **may** set the value of any margin property whose specified value is "*auto*" to 0.

[2] This is a shorthand property. The syntax for its value is as given in the CSS2 specification. Where this specification limits values or indicates alternate representations for properties abbreviated by this property, the same limits and alternate representations apply to this property.

[3] Reading Systems **may** map to one of the keyword values listed for this property.

[4] See section 4.2.5 on color units.

[5] CSS 2 provides a full description of the various table values and their correct renderings. Please refer to the CSS 2 Tables specification (http://www.w3.org/TR/REC-CSS2/tables.html) for a more detailed discussion of the various table values.

CSS 2 and XHTML provide similar but subtly different algorithms for rendering table data. These algorithms tend to generate the same results, but there are a few exceptions. In such cases, conforming Reading Systems must produce output consistent with the algorithm specified by CSS 2.

When using tables, authors should follow the Techniques for Web Accessibility Guidelines (http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/) for maintaining as much semantic information as possible. That document describes good practices for choosing how and when to use table tags, and when to use CSS properties. Specifically, see "Guideline 5: Create tables that transform gracefully" (http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/#gl-table-markup).

[6] The content of an element assigned *display: oeb-page-head* **should** be presented only as a header, and the content of an element assigned *display: oeb-page-foot* **should** be presented only as a footer. Neither **should** be simply presented as if it were inline or block. Reading Systems, however, are free to present headers and footers either in special areas as usual for paper publications, or to make them available in another way. For example, a device with a small screen might instead pop them up on demand. For purposes of page layout, these display values are similar to block boxes with an absolute position (i.e. a **position** value of "*fixed*" or "*absolute*"). That is, they are removed from the normal flow and a new block box is created with it's own flow. Margins, padding and other block characteristics are determined as if the element had *position: fixed* set.

An element assigned *display: oeb-page-head* or *display: oeb-page-foot* shall not be considered in effect while any preceding content remains presented. For example, when rendered to a screen with appropriate style settings, the **myhead** element below would become the page header as soon as nothing preceding the containing **div** is displayed:

```
<div>
      <myhead style="display: oeb-page-head">The OEB Publication
      Structure: Introduction</myhead>
      <h2>Introduction</h2>
      <p>…</p>
</div>
```

Such a header (or footer) remains in effect until another header (or footer) is in effect instead, or until no part of its parent element remains presented (such as when the **div** is no longer visible in the above example), whichever occurs first.

[7] Reading System **may** map to "*top*."

[8] Reading System **may** map to "*bottom*."

[9] **Must not** be used within a style sheet whose **@media** value is other than "*aural*."

[10] One-page Reading Systems must treat "*left*" and "*right*" as "*always*."

[11] Reading Systems need not distinguish "*italic*" and "*oblique*" from each other.

[12] Number specifies the speaking rate in words per minute.

[13] Specifies a number of columns in which to render content; **may** be applied to all block level elements. Reading Systems are free to support integer values other than 1, or **may** map them to 1. Reading Systems **may** support column balancing. A value of "*auto*" allows the Reading System to decide on the optimal number of columns in which to render content, considering available width, font sizes, or any other metrics it considers relevant for readability.

APPENDIX A: THE OEBPS PACKAGE DTD

```
<!--

Title:

     The Package Document Type Definition (DTD) for the Open
     eBook Publication Structure Version 1.2


Version:

     1.2


Revision:

     20020605-x


Previous Released Version:

     1.0.1 (Revision of 01-February-2001, "Document Type
            Definition for the Open eBook package version
            1.0.1")


Authors:

     Version 1.0; 1.0.1

          Steve DeRose <steven_derose@brown.edu>
          Gunter Hille <hille@abc.de>
          Ben Trafford <ben@legendary.org>
          Garret Wilson <garret@globalmentor.com>

     This Version 1.2 updated and edited by:

          Jon Noring <noring@olagrande.net>
          Benjamin Jung <benjamin.jung@deepx.com>


Usage:

     <?xml version="1.0"?>
     <!DOCTYPE package
               PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.2 Package//EN"
               "http://openebook.org/dtds/oeb-1.2/oebpkg12.dtd">
     <package unique-identifier="foo">
          metadata
          manifest
          spine
          tours
          guide
     </package>


Summary Description:

     This is the Package Document Type Definition (DTD) for
```

```
     the Open eBook Publication Structure version 1.2.

     Changes to this DTD from version 1.0.1 include:

          a. Upgrading the <dc-metadata> content model to
             fully conform with the OEBPS 1.2 specification
             requirements. Specifically, <dc:Language> is
             now required, while in OEBPS 1.0.1 it was
             optional.

          b. Updating the mnemonic character entity
             declaration to refer to version 1.2.

          c. Updating the xmlns:dc namespace to refer to
             version 1.1 of the Dublin Core Metadata
             Initiative.

          d. Editing and updating the various non-parsed
             comments.

          e. Revising the layout (e.g., white space
             alteration) to aid in readability.

-->


<!-- ................................................ -->

<!-- XHTML MNEMONIC CHARACTER ENTITIES ................ -->

<!ENTITY % OEBEntities
     PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.2 Entities//EN"
     "http://openebook.org/dtds/oeb-1.2/oeb12.ent">

%OEBEntities;

<!-- ................................................ -->

<!-- DATATYPE ENTITIES ................................ -->

<!-- Uniform Resource Identifier (URI), per [RFC2396] -->

<!ENTITY % URI "CDATA">

<!-- Language code, per [RFC3066] -->

<!ENTITY % LanguageCode "NMTOKEN">

<!-- ................................................ -->

<!-- NAMESPACE ENTITIES ............................... -->

<!ENTITY % dc.xmlns
     "'http://purl.org/dc/elements/1.1/'">

<!ENTITY % oebpk.xmlns
     "'http://openebook.org/namespaces/oeb-package/1.0/'">

<!-- ................................................ -->

<!-- ELEMENT ENTITIES ................................. -->

<!-- The entity 'DCMetadataOpt' includes the 12 optional
```

```
     <dc:Xxx> children elements of <dc-metadata>. It will
     be used in the <dc-metadata> content model. -->

<!ENTITY % DCMetadataOpt
     "dc:Contributor |
      dc:Coverage    |
      dc:Creator     |
      dc:Date        |
      dc:Description |
      dc:Format      |
      dc:Publisher   |
      dc:Relation    |
      dc:Rights      |
      dc:Source      |
      dc:Subject     |
      dc:Type       ">

<!-- ================================================= -->

<!-- ATTRIBUTE ENTITIES ............................... -->

<!ENTITY % CoreAttributes
     "id                ID              #IMPLIED">

<!ENTITY % InternationalAttributes
     "xml:lang          %LanguageCode;  #IMPLIED">

<!ENTITY % CommonAttributes
     "%CoreAttributes;
      %InternationalAttributes;">

<!-- 'DCNamespaceAttribute' is an attribute entity declaring
     the Dublin Core namespace. Used on each <dc:Xxx> element
     to accommodate XML parsers which unnecessarily require
     this. -->

<!ENTITY % DCNamespaceAttribute
     "xmlns:dc          %URI;           #FIXED %dc.xmlns;">

<!-- ================================================= -->

<!-- ELEMENTS AND ATTRIBUTES ........................... -->

<!-- <package> must have as children elements, in this order:
     <metadata>, <manifest>, and <spine>, and optionally may
     include <tours> and/or <guide>. The 'unique-identifier'
     attribute is required for <package> (see comment for
     <dc:Identifier>.) -->

<!ELEMENT package (metadata, manifest, spine, tours?, guide?)>
<!ATTLIST package
      %CommonAttributes;
      unique-identifier  IDREF           #REQUIRED
      xmlns              %URI;           #FIXED %oebpk.xmlns;>

<!-- <metadata> must contain one <dc-metadata>, and
     optionally contain one <x-metadata>. There are no
     attributes for <metadata>. -->

<!ELEMENT metadata (dc-metadata, x-metadata?)>

<!-- <dc-metadata> must contain at least one <dc:Title>,
     one <dc:Identifier>, and one <dc:Language>, and may
```

```
     contain one or more of each of the other twelve
     optional <dc:XXX> elements, all in any order. -->

<!ELEMENT dc-metadata
( (%DCMetadataOpt;)*,
  ( (dc:Title, (%DCMetadataOpt; | dc:Title)*,
      ( (dc:Identifier, (%DCMetadataOpt; | dc:Title | dc:Identifier)*,
         dc:Language) |
        (dc:Language, (%DCMetadataOpt; | dc:Title | dc:Language)*,
         dc:Identifier) ) ) |
    (dc:Identifier, (%DCMetadataOpt; | dc:Identifier)*,
      ( (dc:Title, (%DCMetadataOpt; | dc:Identifier | dc:Title)*,
         dc:Language) |
        (dc:Language, (%DCMetadataOpt; | dc:Identifier | dc:Language)*,
         dc:Title) ) ) |
    (dc:Language, (%DCMetadataOpt; | dc:Language)*,
      ( (dc:Identifier, (%DCMetadataOpt; | dc:Language | dc:Identifier)*,
         dc:Title) |
        (dc:Title, (%DCMetadataOpt; | dc:Language | dc:Title)*,
         dc:Identifier) ) ) ),
  (%DCMetadataOpt; | dc:Title | dc:Identifier | dc:Language)* )>
<!ATTLIST dc-metadata
      %CommonAttributes;
      %DCNamespaceAttribute;
      xmlns:oebpackage   %URI;           #FIXED %oebpk.xmlns;>

<!-- Required elements for <dc-metadata>. -->

<!ELEMENT dc:Title (#PCDATA)>
<!ATTLIST dc:Title
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!-- One <dc:Identifier> must specify an 'id' identical to
     the value of the required <package> 'unique-identifier'
     attribute. -->

<!ELEMENT dc:Identifier (#PCDATA)>
<!ATTLIST dc:Identifier
      %CommonAttributes;
      %DCNamespaceAttribute;
      scheme             NMTOKEN         #IMPLIED>

<!ELEMENT dc:Language (#PCDATA)>
<!ATTLIST dc:Language
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!-- Optional elements for <dc-metadata>. -->

<!ELEMENT dc:Contributor (#PCDATA)>
<!ATTLIST dc:Contributor
      %CommonAttributes;
      %DCNamespaceAttribute;
      file-as            CDATA           #IMPLIED
      role               NMTOKEN         #IMPLIED>

<!ELEMENT dc:Coverage (#PCDATA)>
<!ATTLIST dc:Coverage
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Creator (#PCDATA)>
```

```
<!ATTLIST dc:Creator
      %CommonAttributes;
      %DCNamespaceAttribute;
      file-as           CDATA          #IMPLIED
      role              NMTOKEN        #IMPLIED>

<!ELEMENT dc:Date (#PCDATA)>
<!ATTLIST dc:Date
      %CommonAttributes;
      %DCNamespaceAttribute;
      event             NMTOKEN        #IMPLIED>

<!ELEMENT dc:Description (#PCDATA)>
<!ATTLIST dc:Description
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Format (#PCDATA)>
<!ATTLIST dc:Format
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Publisher (#PCDATA)>
<!ATTLIST dc:Publisher
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Relation (#PCDATA)>
<!ATTLIST dc:Relation
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Rights (#PCDATA)>
<!ATTLIST dc:Rights
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Source (#PCDATA)>
<!ATTLIST dc:Source
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Subject (#PCDATA)>
<!ATTLIST dc:Subject
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!ELEMENT dc:Type (#PCDATA)>
<!ATTLIST dc:Type
      %CommonAttributes;
      %DCNamespaceAttribute;>

<!-- <x-metadata> must contain at least one <meta>. -->

<!ELEMENT x-metadata (meta+)>
<!ATTLIST x-metadata %CommonAttributes;>

<!-- Note that 'content' and 'name' are required attributes
     for <meta>. -->

<!ELEMENT meta EMPTY>
<!ATTLIST meta
      %CommonAttributes;
```

```
      content            CDATA           #REQUIRED
      name               NMTOKEN         #REQUIRED
      scheme             CDATA           #IMPLIED>

<!-- <manifest> must contain at least one <item>. -->

<!ELEMENT manifest (item+)>
<!ATTLIST manifest %CommonAttributes;>

<!-- Note that 'href', 'id' and 'media-type' are required
     attributes for <item>. -->

<!ELEMENT item EMPTY>
<!ATTLIST item
      %InternationalAttributes;
      fallback           IDREF           #IMPLIED
      href               %URI;           #REQUIRED
      id                 ID              #REQUIRED
      media-type         CDATA           #REQUIRED>

<!-- <spine> must contain at least one <itemref>. -->

<!ELEMENT spine (itemref+)>
<!ATTLIST spine %CommonAttributes;>

<!-- Note that 'idref' is a required attribute for
     <itemref>. -->

<!ELEMENT itemref EMPTY>
<!ATTLIST itemref
      %CommonAttributes;
      idref              IDREF           #REQUIRED>

<!-- <tours> must contain at least one <tour>. -->

<!ELEMENT tours (tour+)>
<!ATTLIST tours %CommonAttributes;>

<!-- <tour> must contain at least one <site>. Note that
     'title' is a required attribute for <tour>. -->

<!ELEMENT tour (site+)>
<!ATTLIST tour
      %CommonAttributes;
      title              CDATA           #REQUIRED>

<!-- Note that 'href' and 'title' are required attributes
     for <site>. -->

<!ELEMENT site EMPTY>
<!ATTLIST site
      %CommonAttributes;
      href               %URI;           #REQUIRED
      title              CDATA           #REQUIRED>

<!-- <guide> must contain at least one <reference>. -->

<!ELEMENT guide (reference+)>
<!ATTLIST guide %CommonAttributes;>

<!-- Note that 'href', 'title' and 'type' are required
     attributes for <reference>. -->
```

```
<!ELEMENT reference EMPTY>
<!ATTLIST reference
      %CommonAttributes;
      href              %URI;           #REQUIRED
      title             CDATA           #REQUIRED
      type              NMTOKEN         #REQUIRED>
```

APPENDIX B: THE BASIC OEBPS DOCUMENT DTD

```
<!--

Title:

     The Basic Document Type Definition (DTD) for the Open eBook
     Publication Structure Version 1.2


Version:

     1.2


Revision:

     20020327-x


Authors:

     Jon Noring <noring@olagrande.net>


Usage:

     <?xml version="1.0"?>
     <!DOCTYPE html PUBLIC
          "+//ISBN 0-9673008-1-9//DTD OEB 1.2 Document//EN"
          "http://openebook.org/dtds/oeb-1.2/oebdoc12.dtd">
     <html>
     ...
     </html>


Summary:

     This DTD is a pure subset of XHTML 1.1: any document validating
     to this DTD will also validate to the XHTML 1.1 DTD.

-->


<!-- ================================================= -->

<!-- GENERAL NOTATIONS ................................ -->

<!-- W3C XML 1.0 Recommendation -->

<!NOTATION w3c-xml
     PUBLIC "ISO 8879//NOTATION Extensible Markup Language (XML) 1.0//EN">

<!-- XML 1.0 CDATA -->

<!NOTATION cdata
     PUBLIC "-//W3C//NOTATION XML 1.0: CDATA//EN">

<!-- ================================================= -->
```

```
<!-- ENTITIES WITH DATATYPE NOTATIONS .................. -->

<!-- Content type, as per [RFC2045] -->

<!NOTATION contentType
    PUBLIC "-//W3C//NOTATION XHTML Datatype: ContentType//EN">
<!ENTITY % ContentType.datatype "CDATA">

<!-- Date and time information. ISO date format -->

<!NOTATION datetime
    PUBLIC "-//W3C//NOTATION XHTML Datatype: Datetime//EN">
<!ENTITY % Datetime.datatype "CDATA">

<!-- Language code, as per [RFC3066] -->

<!NOTATION languageCode
    PUBLIC "-//W3C//NOTATION XHTML Datatype: LanguageCode//EN">
<!ENTITY % LanguageCode.datatype "NMTOKEN">

<!-- Length defined for cellpadding/cellspacing -->
<!-- nn for pixels or nn% for percentage length -->

<!NOTATION length
    PUBLIC "-//W3C//NOTATION XHTML Datatype: Length//EN">
<!ENTITY % Length.datatype "CDATA">

<!-- Space-separated list of link types -->

<!NOTATION linkTypes
    PUBLIC "-//W3C//NOTATION XHTML Datatype: LinkTypes//EN">
<!ENTITY % LinkTypes.datatype "NMTOKENS">

<!-- Single or comma-separated list of media descriptors -->

<!NOTATION mediaDesc
    PUBLIC "-//W3C//NOTATION XHTML Datatype: MediaDesc//EN">
<!ENTITY % MediaDesc.datatype "CDATA">

<!-- Pixel, percentage, or relative -->

<!NOTATION multiLength
    PUBLIC "-//W3C//NOTATION XHTML Datatype: MultiLength//EN">
<!ENTITY % MultiLength.datatype "CDATA">

<!-- One or more digits (NUMBER) -->

<!NOTATION number
    PUBLIC "-//W3C//NOTATION XHTML Datatype: Number//EN">
<!ENTITY % Number.datatype "CDATA">

<!-- Integer representing length in pixels -->

<!NOTATION pixels
    PUBLIC "-//W3C//NOTATION XHTML Datatype: Pixels//EN">
<!ENTITY % Pixels.datatype "CDATA">

<!-- Textual content -->

<!NOTATION text
    PUBLIC "-//W3C//NOTATION XHTML Datatype: Text//EN">
<!ENTITY % Text.datatype "CDATA">
```

```
<!-- Uniform Resource Identifier -->

<!NOTATION uri
    PUBLIC "-//W3C//NOTATION XHTML Datatype: URI//EN">
<!ENTITY % URI.datatype "CDATA">

<!-- Space-separated list of Uniform Resource Identifiers -->

<!NOTATION uris
    PUBLIC "-//W3C//NOTATION XHTML Datatype: URIs//EN">
<!ENTITY % URIs.datatype "CDATA">

<!-- ................................................ -->

<!-- ELEMENT ENTITIES ................................ -->

<!ENTITY % Block.class
     "address | blockquote | div | dl |
      h1 | h2 | h3 | h4 | h5 | h6 |
      hr | ol | p | pre | table | ul">

<!ENTITY % Inline.class
     "a | abbr | acronym | b | big | br |
      cite | code | dfn | em | i | img | kbd |
      map | object | q | samp | small | span |
      strong | sub | sup | tt | var">

<!ENTITY % BlockOrInline.class
     "del | ins | noscript | script">

<!ENTITY % Block.mix
     "%Block.class; | %BlockOrInline.class;">

<!ENTITY % Inline.mix
     "%Inline.class; | %BlockOrInline.class;">

<!ENTITY % Flow.mix
     "%Block.class; | %Inline.class; | %BlockOrInline.class;">

<!ENTITY % HeadOpts.mix
     "(link | meta | object | script | style)*">

<!-- ................................................ -->

<!-- ATTRIBUTE ENTITIES .............................. -->

<!ENTITY % XHTML.xmlns "http://www.w3.org/1999/xhtml">

<!ENTITY % Core.attrib
     "class        NMTOKENS                #IMPLIED
      id           ID                      #IMPLIED
      style        CDATA                   #IMPLIED
      title        %Text.datatype;         #IMPLIED">

<!ENTITY % Common.attrib
     "%Core.attrib;
      xml:lang     %LanguageCode.datatype;  #IMPLIED">

<!ENTITY % CellHAlign
     "(left | center | right | justify)">

<!ENTITY % CellVAlign
     "(top | middle | bottom | baseline)">
```

```
<!-- ================================================ -->

<!-- XHTML MNEMONIC CHARACTER ENTITIES ................ -->

<!ENTITY % OEBEntities
     PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.2 Entities//EN"
     "http://openebook.org/dtds/oeb-1.2/oeb12.ent">

%OEBEntities;

<!-- ================================================ -->

<!-- ELEMENTS AND ATTRIBUTES ......................... -->

<!-- TOP LEVEL STRUCTURE ............................. -->

<!ELEMENT html (head, body)>
<!ATTLIST html
      xml:lang     %LanguageCode.datatype;  #IMPLIED
      xmlns        %URI.datatype;           #FIXED '%XHTML.xmlns;'>

<!ELEMENT head
      ( %HeadOpts.mix;,
        ( (title, %HeadOpts.mix;, (base,  %HeadOpts.mix;)?) |
          (base,  %HeadOpts.mix;, (title, %HeadOpts.mix;) ) ) )>
<!ATTLIST head
      xml:lang     %LanguageCode.datatype;  #IMPLIED>

<!ELEMENT body (%Block.mix;)+>
<!ATTLIST body %Common.attrib;>

<!-- HEAD LEVEL ...................................... -->

<!ELEMENT base EMPTY>
<!ATTLIST base
      href         %URI.datatype;           #REQUIRED>

<!ELEMENT link EMPTY>
<!ATTLIST link
      %Common.attrib;
      href         %URI.datatype;           #IMPLIED
      media        %MediaDesc.datatype;     #IMPLIED
      rel          %LinkTypes.datatype;     #IMPLIED
      rev          %LinkTypes.datatype;     #IMPLIED
      type         %ContentType.datatype;   #IMPLIED>

<!ELEMENT meta EMPTY>
<!ATTLIST meta
      content      CDATA                    #REQUIRED
      name         NMTOKEN                  #IMPLIED
      scheme       CDATA                    #IMPLIED
      xml:lang     %LanguageCode.datatype;  #IMPLIED>

<!ELEMENT style (#PCDATA)>
<!ATTLIST style
      title        %Text.datatype;          #IMPLIED
      type         %ContentType.datatype;   #REQUIRED
      xml:lang     %LanguageCode.datatype;  #IMPLIED
      xml:space    (preserve)               #FIXED 'preserve'>

<!ELEMENT title (#PCDATA)>
<!ATTLIST title
```

```
      xml:lang     %LanguageCode.datatype;  #IMPLIED>

<!-- BLOCK LEVEL ..................................... -->

<!ELEMENT address (#PCDATA | %Inline.mix;)*>
<!ATTLIST address %Common.attrib;>

<!ELEMENT blockquote (%Block.mix;)+>
<!ATTLIST blockquote
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED>

<!ELEMENT div (#PCDATA | %Flow.mix;)*>
<!ATTLIST div %Common.attrib;>

<!ELEMENT dl (dt | dd)+>
<!ATTLIST dl %Common.attrib;>

<!ELEMENT h1 (#PCDATA | %Inline.mix;)*>
<!ATTLIST h1 %Common.attrib;>

<!ELEMENT h2 (#PCDATA | %Inline.mix;)*>
<!ATTLIST h2 %Common.attrib;>

<!ELEMENT h3 (#PCDATA | %Inline.mix;)*>
<!ATTLIST h3 %Common.attrib;>

<!ELEMENT h4 (#PCDATA | %Inline.mix;)*>
<!ATTLIST h4 %Common.attrib;>

<!ELEMENT h5 (#PCDATA | %Inline.mix;)*>
<!ATTLIST h5 %Common.attrib;>

<!ELEMENT h6 (#PCDATA | %Inline.mix;)*>
<!ATTLIST h6 %Common.attrib;>

<!ELEMENT hr EMPTY>
<!ATTLIST hr %Common.attrib;>

<!ELEMENT ol (li)+>
<!ATTLIST ol %Common.attrib;>

<!ELEMENT p (#PCDATA | %Inline.mix;)*>
<!ATTLIST p %Common.attrib;>

<!ELEMENT pre
      (#PCDATA |
      a | abbr | acronym | b | br | cite |
      code | dfn | em | i | kbd | map | q |
      samp | span | strong | tt | var |
      script)*>
<!ATTLIST pre
      %Common.attrib;
      xml:space    (preserve)               #FIXED 'preserve'>

<!ELEMENT table
      ( caption?, (col* | colgroup*),
        ( (thead?, tfoot?, tbody+) | (tr+) ) )>
<!ATTLIST table
      %Common.attrib;
      border       %Pixels.datatype;        #IMPLIED
      cellpadding  %Length.datatype;        #IMPLIED
      cellspacing  %Length.datatype;        #IMPLIED
```

```
      summary      %Text.datatype;          #IMPLIED
      width        %Length.datatype;        #IMPLIED>

<!ELEMENT ul (li)+>
<!ATTLIST ul %Common.attrib;>

<!-- INLINE LEVEL ................................... -->

<!ELEMENT a (#PCDATA |
      abbr | acronym | b | big | br | cite |
      code | dfn | em | i | img | kbd | map |
      object | q | samp | small | span | strong |
      sub | sup | tt | var |
      %BlockOrInline.class;)*>
<!ATTLIST a
      %Common.attrib;
      href         %URI.datatype;           #IMPLIED
      rel          %LinkTypes.datatype;     #IMPLIED
      rev          %LinkTypes.datatype;     #IMPLIED>

<!ELEMENT abbr (#PCDATA | %Inline.mix;)*>
<!ATTLIST abbr %Common.attrib;>

<!ELEMENT acronym (#PCDATA | %Inline.mix;)*>
<!ATTLIST acronym %Common.attrib;>

<!ELEMENT b (#PCDATA | %Inline.mix;)*>
<!ATTLIST b %Common.attrib;>

<!ELEMENT big (#PCDATA | %Inline.mix;)*>
<!ATTLIST big %Common.attrib;>

<!ELEMENT br EMPTY>
<!ATTLIST br %Core.attrib;>

<!ELEMENT cite (#PCDATA | %Inline.mix;)*>
<!ATTLIST cite %Common.attrib;>

<!ELEMENT code (#PCDATA | %Inline.mix;)*>
<!ATTLIST code %Common.attrib;>

<!ELEMENT dfn (#PCDATA | %Inline.mix;)*>
<!ATTLIST dfn %Common.attrib;>

<!ELEMENT em (#PCDATA | %Inline.mix;)*>
<!ATTLIST em %Common.attrib;>

<!ELEMENT i (#PCDATA | %Inline.mix;)*>
<!ATTLIST i %Common.attrib;>

<!ELEMENT img EMPTY>
<!ATTLIST img
      %Common.attrib;
      alt          %Text.datatype;          #REQUIRED
      height       %Length.datatype;        #IMPLIED
      longdesc     %URI.datatype;           #IMPLIED
      src          %URI.datatype;           #REQUIRED
      usemap       IDREF                    #IMPLIED
      width        %Length.datatype;        #IMPLIED>

<!ELEMENT kbd (#PCDATA | %Inline.mix;)*>
<!ATTLIST kbd %Common.attrib;>
```

```
<!ELEMENT map (%Block.mix; | area)+>
<!ATTLIST map
      class        NMTOKENS                 #IMPLIED
      id           ID                       #REQUIRED
      style        CDATA                    #IMPLIED
      title        %Text.datatype;          #IMPLIED
      xml:lang     %LanguageCode.datatype;  #IMPLIED>

<!ELEMENT object (#PCDATA | %Flow.mix; | param)*>
<!ATTLIST object
      %Common.attrib;
      archive      %URIs.datatype;          #IMPLIED
      classid      %URI.datatype;           #IMPLIED
      codebase     %URI.datatype;           #IMPLIED
      codetype     %ContentType.datatype;   #IMPLIED
      data         %URI.datatype;           #IMPLIED
      height       %Length.datatype;        #IMPLIED
      type         %ContentType.datatype;   #IMPLIED
      usemap       IDREF                    #IMPLIED
      width        %Length.datatype;        #IMPLIED>

<!ELEMENT q (#PCDATA | %Inline.mix;)*>
<!ATTLIST q
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED>

<!ELEMENT samp (#PCDATA | %Inline.mix;)*>
<!ATTLIST samp %Common.attrib;>

<!ELEMENT small (#PCDATA | %Inline.mix;)*>
<!ATTLIST small %Common.attrib;>

<!ELEMENT span (#PCDATA | %Inline.mix;)*>
<!ATTLIST span %Common.attrib;>

<!ELEMENT strong (#PCDATA | %Inline.mix;)*>
<!ATTLIST strong %Common.attrib;>

<!ELEMENT sub (#PCDATA | %Inline.mix;)*>
<!ATTLIST sub %Common.attrib;>

<!ELEMENT sup (#PCDATA | %Inline.mix;)*>
<!ATTLIST sup %Common.attrib;>

<!ELEMENT tt (#PCDATA | %Inline.mix;)*>
<!ATTLIST tt %Common.attrib;>

<!ELEMENT var (#PCDATA | %Inline.mix;)*>
<!ATTLIST var %Common.attrib;>

<!-- BLOCK OR INLINE LEVEL ............................ -->

<!ELEMENT del (#PCDATA | %Flow.mix;)*>
<!ATTLIST del
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED
      datetime     %Datetime.datatype;      #IMPLIED>

<!ELEMENT ins (#PCDATA | %Flow.mix;)*>
<!ATTLIST ins
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED
      datetime     %Datetime.datatype;      #IMPLIED>
```

```
<!ELEMENT noscript (%Block.mix;)+>
<!ATTLIST noscript %Common.attrib;>

<!ELEMENT script (#PCDATA)>
<!ATTLIST script
      type         %ContentType.datatype;   #REQUIRED
      xml:space    (preserve)               #FIXED 'preserve'>

<!-- TABLE LEVEL ..................................... -->

<!ELEMENT caption (#PCDATA | %Inline.mix;)*>
<!ATTLIST caption %Common.attrib;>

<!ELEMENT col EMPTY>
<!ATTLIST col
      %Common.attrib;
      align        %CellHAlign;             #IMPLIED
      span         %Number.datatype;        '1'
      valign       %CellVAlign;             #IMPLIED
      width        %MultiLength.datatype;   #IMPLIED>

<!ELEMENT colgroup (col)*>
<!ATTLIST colgroup
      %Common.attrib;
      align        %CellHAlign;             #IMPLIED
      span         %Number.datatype;        '1'
      valign       %CellVAlign;             #IMPLIED
      width        %MultiLength.datatype;   #IMPLIED>

<!ELEMENT tbody (tr)+>
<!ATTLIST tbody
      %Common.attrib;
      align        %CellHAlign;             #IMPLIED
      valign       %CellVAlign;             #IMPLIED>

<!ELEMENT td (#PCDATA | %Flow.mix;)*>
<!ATTLIST td
      %Common.attrib;
      abbr         %Text.datatype;          #IMPLIED
      align        %CellHAlign;             #IMPLIED
      colspan      %Number.datatype;        '1'
      rowspan      %Number.datatype;        '1'
      valign       %CellVAlign;             #IMPLIED>

<!ELEMENT tfoot (tr)+>
<!ATTLIST tfoot
      %Common.attrib;
      align        %CellHAlign;             #IMPLIED
      valign       %CellVAlign;             #IMPLIED>

<!ELEMENT th (#PCDATA | %Flow.mix;)*>
<!ATTLIST th
      %Common.attrib;
      abbr         %Text.datatype;          #IMPLIED
      align        %CellHAlign;             #IMPLIED
      colspan      %Number.datatype;        '1'
      rowspan      %Number.datatype;        '1'
      valign       %CellVAlign;             #IMPLIED>

<!ELEMENT thead (tr)+>
<!ATTLIST thead
      %Common.attrib;
```

```
      align        %CellHAlign;             #IMPLIED
      valign       %CellVAlign;             #IMPLIED>

<!ELEMENT tr (th | td)+>
<!ATTLIST tr
      %Common.attrib;
      align        %CellHAlign;             #IMPLIED
      valign       %CellVAlign;             #IMPLIED>

<!-- LIST LEVEL ..................................... -->

<!ELEMENT dd (#PCDATA | %Flow.mix;)*>
<!ATTLIST dd %Common.attrib;>

<!ELEMENT dt (#PCDATA | %Inline.mix;)*>
<!ATTLIST dt %Common.attrib;>

<!ELEMENT li (#PCDATA | %Flow.mix;)*>
<!ATTLIST li %Common.attrib;>

<!-- MISCELLANEOUS ................................... -->

<!ELEMENT area EMPTY>
<!ATTLIST area
      %Common.attrib;
      alt          %Text.datatype;          #REQUIRED
      coords       CDATA                    #IMPLIED
      href         %URI.datatype;           #IMPLIED
      nohref       (nohref)                 #IMPLIED
      shape        (rect | circle |
                    poly | default)         'rect'>

<!ELEMENT param EMPTY>
<!ATTLIST param
      id           ID                       #IMPLIED
      name         CDATA                    #REQUIRED
      type         %ContentType.datatype;   #IMPLIED
      value        CDATA                    #IMPLIED
      valuetype    (data | ref | object)    'data'>
```

APPENDIX C: CHARACTER ENTITIES

```
<!--

Title:

     Mnemonic Character Entities For the Open eBook Publication
     Structure Version 1.2


Version:

     1.2


Revision:

     20020424-x


Previous Version:

     1.0.1 (Revision of 22-November-2000, "Character Entities for
            the Open eBook Publication Structure Version 1.0.1")


Authors:

     Version 1.0; 1.0.1

          Gunter Hille <hille@abc.de>
          Ben Trafford <ben@legendary.org>
          Garret Wilson <garret@globalmentor.com>


     This Version 1.2 updated and edited by:

          Jon Noring <noring@olagrande.net>


Usage:

     <!ENTITY % OEBEntities
               PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.2 Entities//EN"
               "http://openebook.org/dtds/oeb-1.2/oeb12.ent">

     %OEBEntities;


Summary:

     This DTD fragment exactly duplicates, with some reorganization,
     correction, and reformatting of the descriptive text, the 253
     character entity declarations in the XHTML 1.1 DTD. Refer to:

          http://www.w3.org/TR/xhtml1/DTD/xhtml-lat1.ent
          http://www.w3.org/TR/xhtml1/DTD/xhtml-symbol.ent
          http://www.w3.org/TR/xhtml1/DTD/xhtml-special.ent
```

Relation to OEBPS Version 1.0.1:

The 253 character entities declared herein include all 249 from Version 1.0.1 plus four of the five pre-defined XML 1.0 character entities of &, <, >, " (the fifth pre-defined XML character entity, ', is one of the 249 character entities already declared in Version 1.0.1.)

The five pre-defined XML 1.0 character entities are included for completeness and interoperability as recommended by W3C, and to follow XHTML practice. (Further information on the purpose and usage of these five pre-defined XML character entities, and the normative reference, is given in the Usage Note below.)

Relation to Unicode 3.2.0 and ISO/IEC 10646:

The mnemonic character entities declared herein substitute for numeric character references, the numeric values for the associated characters specified by Unicode (in turn, the Unicode Character Data Set conforms with the ISO/IEC 10646 character set which XML 1.0 specifies.) The current version of Unicode is 3.2.0. General information on Unicode, including information on the latest version, is found at

http://www.unicode.org/

In addition, Unicode has categorized the massive number of characters in its Character Database using two different systems: Character Blocks and Script Names. These two systems are used herein for general categorization of the 253 character entities. The text files listing the code points for these two systems are:

http://www.unicode.org/Public/UNIDATA/Blocks.txt
http://www.unicode.org/Public/UNIDATA/Scripts.txt

Tutorial Note to Document Authors: Character Entity Usage

To insert the desired special character into the content of an OEBPS Document or Package file (which are XML documents), prefix the associated mnemonic character entity with the '&' character and terminate with the ';' character.

Example: to insert the "em dash" character (which has the mnemonic 'mdash'), use — .

If preferred, the character can instead be inserted using the direct (Unicode) numerical character reference, the codes of which are given herein (see the above note on Unicode.) So, for the "em dash" character one can use, instead of —, either the decimal — or the hexadecimal equivalent — .

Importantly note that within the content (PCDATA) of all OEBPS documents and package files, the special XML characters '&' and '<', when intended to be used literally, MUST be represented with the mnemonic character entities of & and < (or the numerical character entity equivalents), respectively. In addition, it is considered good practice to use the > (or numerical equivalent) for the '>' symbol, although it is not necessary except in very unusual and rare circumstances. The two other special XML character entities, apostrophe (') and quote ("), are only necessary within element attribute values to literally represent

```
     these characters, and for similar non-content purposes.

     (The normative reference on the five XML pre-defined mnemonic
     character entities is given in Sections 2.4 and 4.6 of the XML
     1.0 Specification, Second Edition:

          http://www.w3.org/TR/2000/REC-xml-20001006

     )


     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       Portions (C) International Organization for
       Standardization 1986. Permission to copy in any
       form is granted for use with conforming SGML
       systems and applications as defined in ISO 8879,
       provided this notice is included in all copies.
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

-->

<!--

     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       XML 1.0 Pre-Defined Character Entities
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

       Drawn From Unicode 3.1.1 Character Sets:

             Block Name(s):  Basic Latin             (U+0000 to U+007F)
            Script Name(s):  (none)

-->


<!ENTITY quot       "&#34;" ><!-- quotation mark
                               APL quote
                               ==================== U+0022 ISOnum -->

<!ENTITY amp    "&#38;#38;" ><!-- ampersand
                               ==================== U+0026 ISOnum -->

<!ENTITY apos       "&#39;" ><!-- apostrophe mark
                               ==================== U+0027 ISOnum -->

<!ENTITY lt     "&#38;#60;" ><!-- less-than sign
                               ==================== U+003C ISOnum -->

<!ENTITY gt         "&#62;" ><!-- greater-than sign
                               ==================== U+003E ISOnum -->


<!--

     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       Extended Latin Script
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

       Drawn From Unicode 3.1.1 Character Sets:

             Block Name(s):  Latin-1 Supplement       (U+0080 to U+00FF)
                             Latin Extended-A         (U+0100 to U+017F)
                             Latin Extended-B         (U+0180 to U+024F)
```

```
            Script Name(s):  Latin

-->


<!ENTITY ordf      "&#170;" ><!-- feminine ordinal indicator
                                ==================== U+00AA ISOnum -->

<!ENTITY ordm      "&#186;" ><!-- masculine ordinal indicator
                                ==================== U+00BA ISOnum -->

<!ENTITY Agrave    "&#192;" ><!-- Latin capital letter A with grave
                                Latin capital letter A grave
                                ==================== U+00C0 ISOlat1 -->

<!ENTITY Aacute    "&#193;" ><!-- Latin capital letter A with acute
                                ==================== U+00C1 ISOlat1 -->

<!ENTITY Acirc     "&#194;" ><!-- Latin capital letter A with circumflex
                                ==================== U+00C2 ISOlat1 -->

<!ENTITY Atilde    "&#195;" ><!-- Latin capital letter A with tilde
                                ==================== U+00C3 ISOlat1 -->

<!ENTITY Auml      "&#196;" ><!-- Latin capital letter A with diaeresis
                                ==================== U+00C4 ISOlat1 -->

<!ENTITY Aring     "&#197;" ><!-- Latin capital letter A with ring above
                                Latin capital letter A ring
                                ==================== U+00C5 ISOlat1 -->

<!ENTITY AElig     "&#198;" ><!-- Latin capital letter AE
                                Latin capital ligature AE
                                ==================== U+00C6 ISOlat1 -->

<!ENTITY Ccedil    "&#199;" ><!-- Latin capital letter C with cedilla
                                ==================== U+00C7 ISOlat1 -->

<!ENTITY Egrave    "&#200;" ><!-- Latin capital letter E with grave
                                ==================== U+00C8 ISOlat1 -->

<!ENTITY Eacute    "&#201;" ><!-- Latin capital letter E with acute
                                ==================== U+00C9 ISOlat1 -->

<!ENTITY Ecirc     "&#202;" ><!-- Latin capital letter E with circumflex
                                ==================== U+00CA ISOlat1 -->

<!ENTITY Euml      "&#203;" ><!-- Latin capital letter E with diaeresis
                                ==================== U+00CB ISOlat1 -->

<!ENTITY Igrave    "&#204;" ><!-- Latin capital letter I with grave
                                ==================== U+00CC ISOlat1 -->

<!ENTITY Iacute    "&#205;" ><!-- Latin capital letter I with acute
                                ==================== U+00CD ISOlat1 -->

<!ENTITY Icirc     "&#206;" ><!-- Latin capital letter I with circumflex
                                ==================== U+00CE ISOlat1 -->

<!ENTITY Iuml      "&#207;" ><!-- Latin capital letter I with diaeresis
                                ==================== U+00CF ISOlat1 -->

<!ENTITY ETH       "&#208;" ><!-- Latin capital letter ETH
```

```
                                    ================== U+00D0 ISOlat1 -->

<!ENTITY Ntilde    "&#209;" ><!-- Latin capital letter N with tilde
                                    ================== U+00D1 ISOlat1 -->

<!ENTITY Ograve    "&#210;" ><!-- Latin capital letter O with grave
                                    ================== U+00D2 ISOlat1 -->

<!ENTITY Oacute    "&#211;" ><!-- Latin capital letter O with acute
                                    ================== U+00D3 ISOlat1 -->

<!ENTITY Ocirc     "&#212;" ><!-- Latin capital letter O with circumflex
                                    ================== U+00D4 ISOlat1 -->

<!ENTITY Otilde    "&#213;" ><!-- Latin capital letter O with tilde
                                    ================== U+00D5 ISOlat1 -->

<!ENTITY Ouml      "&#214;" ><!-- Latin capital letter O with diaeresis
                                    ================== U+00D6 ISOlat1 -->

<!ENTITY Oslash    "&#216;" ><!-- Latin capital letter O with stroke
                                    Latin capital letter O slash
                                    ================== U+00D8 ISOlat1 -->

<!ENTITY Ugrave    "&#217;" ><!-- Latin capital letter U with grave
                                    ================== U+00D9 ISOlat1 -->

<!ENTITY Uacute    "&#218;" ><!-- Latin capital letter U with acute
                                    ================== U+00DA ISOlat1 -->

<!ENTITY Ucirc     "&#219;" ><!-- Latin capital letter U with circumflex
                                    ================== U+00DB ISOlat1 -->

<!ENTITY Uuml      "&#220;" ><!-- Latin capital letter U with diaeresis
                                    ================== U+00DC ISOlat1 -->

<!ENTITY Yacute    "&#221;" ><!-- Latin capital letter Y with acute
                                    ================== U+00DD ISOlat1 -->

<!ENTITY THORN     "&#222;" ><!-- Latin capital letter THORN
                                    ================== U+00DE ISOlat1 -->

<!ENTITY szlig     "&#223;" ><!-- Latin small letter sharp s
                                    ess-zed
                                    ================== U+00DF ISOlat1 -->

<!ENTITY agrave    "&#224;" ><!-- Latin small letter a with grave
                                    Latin small letter a grave
                                    ================== U+00E0 ISOlat1 -->

<!ENTITY aacute    "&#225;" ><!-- Latin small letter a with acute
                                    ================== U+00E1 ISOlat1 -->

<!ENTITY acirc     "&#226;" ><!-- Latin small letter a with circumflex
                                    ================== U+00E2 ISOlat1 -->

<!ENTITY atilde    "&#227;" ><!-- Latin small letter a with tilde
                                    ================== U+00E3 ISOlat1 -->

<!ENTITY auml      "&#228;" ><!-- Latin small letter a with diaeresis
                                    ================== U+00E4 ISOlat1 -->

<!ENTITY aring     "&#229;" ><!-- Latin small letter a with ring above
```

```
                                    Latin small letter a ring
                                    ================== U+00E5 ISOlat1 -->

<!ENTITY aelig     "&#230;" ><!-- Latin small letter ae
                                    Latin small ligature ae
                                    ================== U+00E6 ISOlat1 -->

<!ENTITY ccedil    "&#231;" ><!-- Latin small letter c with cedilla
                                    ================== U+00E7 ISOlat1 -->

<!ENTITY egrave    "&#232;" ><!-- Latin small letter e with grave
                                    ================== U+00E8 ISOlat1 -->

<!ENTITY eacute    "&#233;" ><!-- Latin small letter e with acute
                                    ================== U+00E9 ISOlat1 -->

<!ENTITY ecirc     "&#234;" ><!-- Latin small letter e with circumflex
                                    ================== U+00EA ISOlat1 -->

<!ENTITY euml      "&#235;" ><!-- Latin small letter e with diaeresis
                                    ================== U+00EB ISOlat1 -->

<!ENTITY igrave    "&#236;" ><!-- Latin small letter i with grave
                                    ================== U+00EC ISOlat1 -->

<!ENTITY iacute    "&#237;" ><!-- Latin small letter i with acute
                                    ================== U+00ED ISOlat1 -->

<!ENTITY icirc     "&#238;" ><!-- Latin small letter i with circumflex
                                    ================== U+00EE ISOlat1 -->

<!ENTITY iuml      "&#239;" ><!-- Latin small letter i with diaeresis
                                    ================== U+00EF ISOlat1 -->

<!ENTITY eth       "&#240;" ><!-- Latin small letter eth
                                    ================== U+00F0 ISOlat1 -->

<!ENTITY ntilde    "&#241;" ><!-- Latin small letter n with tilde
                                    ================== U+00F1 ISOlat1 -->

<!ENTITY ograve    "&#242;" ><!-- Latin small letter o with grave
                                    ================== U+00F2 ISOlat1 -->

<!ENTITY oacute    "&#243;" ><!-- Latin small letter o with acute
                                    ================== U+00F3 ISOlat1 -->

<!ENTITY ocirc     "&#244;" ><!-- Latin small letter o with circumflex
                                    ================== U+00F4 ISOlat1 -->

<!ENTITY otilde    "&#245;" ><!-- Latin small letter o with tilde
                                    ================== U+00F5 ISOlat1 -->

<!ENTITY ouml      "&#246;" ><!-- Latin small letter o with diaeresis
                                    ================== U+00F6 ISOlat1 -->

<!ENTITY oslash    "&#248;" ><!-- Latin small letter o with stroke
                                    Latin small letter o slash
                                    ================== U+00F8 ISOlat1 -->

<!ENTITY ugrave    "&#249;" ><!-- Latin small letter u with grave
                                    ================== U+00F9 ISOlat1 -->

<!ENTITY uacute    "&#250;" ><!-- Latin small letter u with acute
```

```
                                    ================== U+00FA ISOlat1 -->

<!ENTITY ucirc     "&#251;" ><!-- Latin small letter u with circumflex
                                    ================== U+00FB ISOlat1 -->

<!ENTITY uuml      "&#252;" ><!-- Latin small letter u with diaeresis
                                    ================== U+00FC ISOlat1 -->

<!ENTITY yacute    "&#253;" ><!-- Latin small letter y with acute
                                    ================== U+00FD ISOlat1 -->

<!ENTITY thorn     "&#254;" ><!-- Latin small letter thorn with
                                    ================== U+00FE ISOlat1 -->

<!ENTITY yuml      "&#255;" ><!-- Latin small letter y with diaeresis
                                    ================== U+00FF ISOlat1 -->

<!ENTITY OElig     "&#338;" ><!-- Latin capital ligature OE
                                    ================== U+0152 ISOlat2 -->

<!ENTITY oelig     "&#339;" ><!-- Latin small ligature oe
                                    ================== U+0153 ISOlat2 -->

<!ENTITY Scaron    "&#352;" ><!-- Latin capital letter S with caron
                                    ================== U+0160 ISOlat2 -->

<!ENTITY scaron    "&#353;" ><!-- Latin small letter s with caron
                                    ================== U+0161 ISOlat2 -->

<!ENTITY Yuml      "&#376;" ><!-- Latin capital letter Y with diaeresis
                                    ================== U+0178 ISOlat2 -->

<!ENTITY fnof      "&#402;" ><!-- Latin small f with hook
                                    function
                                    florin
                                    ================== U+0192 ISOtech -->


<!--

     +-+-+-+-+-+-+-+
       Greek Script
     +-+-+-+-+-+-+-+

       Drawn From Unicode 3.1.1 Character Sets:

             Block Name(s):  Greek                    (U+0370 to U+03FF)
            Script Name(s):  Greek

-->


<!ENTITY Alpha     "&#913;" ><!-- Greek capital letter alpha
                                    ============================ U+0391 -->

<!ENTITY Beta      "&#914;" ><!-- Greek capital letter beta
                                    ============================ U+0392 -->

<!ENTITY Gamma     "&#915;" ><!-- Greek capital letter gamma
                                    ================== U+0393 ISOgrk3 -->

<!ENTITY Delta     "&#916;" ><!-- Greek capital letter delta
                                    ================== U+0394 ISOgrk3 -->
```

```
<!ENTITY Epsilon   "&#917;" ><!-- Greek capital letter epsilon
                                ============================ U+0395 -->

<!ENTITY Zeta      "&#918;" ><!-- Greek capital letter zeta
                                ============================ U+0396 -->

<!ENTITY Eta       "&#919;" ><!-- Greek capital letter eta
                                ============================ U+0397 -->

<!ENTITY Theta     "&#920;" ><!-- Greek capital letter theta
                                =================== U+0398 ISOgrk3 -->

<!ENTITY Iota      "&#921;" ><!-- Greek capital letter iota
                                ============================ U+0399 -->

<!ENTITY Kappa     "&#922;" ><!-- Greek capital letter kappa
                                ============================ U+039A -->

<!ENTITY Lambda    "&#923;" ><!-- Greek capital letter lambda
                                =================== U+039B ISOgrk3 -->

<!ENTITY Mu        "&#924;" ><!-- Greek capital letter mu
                                ============================ U+039C -->

<!ENTITY Nu        "&#925;" ><!-- Greek capital letter nu
                                ============================ U+039D -->

<!ENTITY Xi        "&#926;" ><!-- Greek capital letter xi
                                =================== U+039E ISOgrk3 -->

<!ENTITY Omicron   "&#927;" ><!-- Greek capital letter omicron
                                ============================ U+039F -->

<!ENTITY Pi        "&#928;" ><!-- Greek capital letter pi
                                =================== U+03A0 ISOgrk3 -->

<!ENTITY Rho       "&#929;" ><!-- Greek capital letter rho
                                ============================ U+03A1 -->

<!ENTITY Sigma     "&#931;" ><!-- Greek capital letter sigma
                                =================== U+03A3 ISOgrk3 -->

<!ENTITY Tau       "&#932;" ><!-- Greek capital letter tau
                                ============================ U+03A4 -->

<!ENTITY Upsilon   "&#933;" ><!-- Greek capital letter upsilon
                                =================== U+03A5 ISOgrk3 -->

<!ENTITY Phi       "&#934;" ><!-- Greek capital letter phi
                                =================== U+03A6 ISOgrk3 -->

<!ENTITY Chi       "&#935;" ><!-- Greek capital letter chi
                                ============================ U+03A7 -->

<!ENTITY Psi       "&#936;" ><!-- Greek capital letter psi
                                =================== U+03A8 ISOgrk3 -->

<!ENTITY Omega     "&#937;" ><!-- Greek capital letter omega
                                =================== U+03A9 ISOgrk3 -->

<!ENTITY alpha     "&#945;" ><!-- Greek small letter alpha
                                =================== U+03B1 ISOgrk3 -->
```

```
<!ENTITY beta      "&#946;" ><!-- Greek small letter beta
                                ==================== U+03B2 ISOgrk3 -->

<!ENTITY gamma     "&#947;" ><!-- Greek small letter gamma
                                ==================== U+03B3 ISOgrk3 -->

<!ENTITY delta     "&#948;" ><!-- Greek small letter delta
                                ==================== U+03B4 ISOgrk3 -->

<!ENTITY epsilon   "&#949;" ><!-- Greek small letter epsilon
                                ==================== U+03B5 ISOgrk3 -->

<!ENTITY zeta      "&#950;" ><!-- Greek small letter zeta
                                ==================== U+03B6 ISOgrk3 -->

<!ENTITY eta       "&#951;" ><!-- Greek small letter eta
                                ==================== U+03B7 ISOgrk3 -->

<!ENTITY theta     "&#952;" ><!-- Greek small letter theta
                                ==================== U+03B8 ISOgrk3 -->

<!ENTITY iota      "&#953;" ><!-- Greek small letter iota
                                ==================== U+03B9 ISOgrk3 -->

<!ENTITY kappa     "&#954;" ><!-- Greek small letter kappa
                                ==================== U+03BA ISOgrk3 -->

<!ENTITY lambda    "&#955;" ><!-- Greek small letter lambda
                                ==================== U+03BB ISOgrk3 -->

<!ENTITY mu        "&#956;" ><!-- Greek small letter mu
                                ==================== U+03BC ISOgrk3 -->

<!ENTITY nu        "&#957;" ><!-- Greek small letter nu
                                ==================== U+03BD ISOgrk3 -->

<!ENTITY xi        "&#958;" ><!-- Greek small letter xi
                                ==================== U+03BE ISOgrk3 -->

<!ENTITY omicron   "&#959;" ><!-- Greek small letter omicron
                                ======================= U+03BF NEW -->

<!ENTITY pi        "&#960;" ><!-- Greek small letter pi
                                ==================== U+03C0 ISOgrk3 -->

<!ENTITY rho       "&#961;" ><!-- Greek small letter rho
                                ==================== U+03C1 ISOgrk3 -->

<!ENTITY sigmaf    "&#962;" ><!-- Greek small letter final sigma
                                ==================== U+03C2 ISOgrk3 -->

<!ENTITY sigma     "&#963;" ><!-- Greek small letter sigma
                                ==================== U+03C3 ISOgrk3 -->

<!ENTITY tau       "&#964;" ><!-- Greek small letter tau
                                ==================== U+03C4 ISOgrk3 -->

<!ENTITY upsilon   "&#965;" ><!-- Greek small letter upsilon
                                ==================== U+03C5 ISOgrk3 -->

<!ENTITY phi       "&#966;" ><!-- Greek small letter phi
                                ==================== U+03C6 ISOgrk3 -->
```

```
<!ENTITY chi       "&#967;" ><!-- Greek small letter chi
                                ==================== U+03C7 ISOgrk3 -->

<!ENTITY psi       "&#968;" ><!-- Greek small letter psi
                                ==================== U+03C8 ISOgrk3 -->

<!ENTITY omega     "&#969;" ><!-- Greek small letter omega
                                ==================== U+03C9 ISOgrk3 -->

<!ENTITY thetasym  "&#977;" ><!-- Greek small letter theta symbol
                                ======================= U+03D1 NEW -->

<!ENTITY upsih     "&#978;" ><!-- Greek upsilon with hook symbol
                                ======================= U+03D2 NEW -->

<!ENTITY piv       "&#982;" ><!-- Greek pi symbol
                                ==================== U+03D6 ISOgrk3 -->


<!--

     +-+-+-+-+-+-+-+-+-+-+-+-+
       General Punctuation
     +-+-+-+-+-+-+-+-+-+-+-+-+

       Drawn From Unicode 3.1.1 Character Sets:

             Block Name(s):  General Punctuation      (U+2000 to U+206F)
            Script Name(s):  (none)

-->


<!ENTITY ensp      " " ><!-- en space
                                ==================== U+2002 ISOpub -->

<!ENTITY emsp      " " ><!-- em space
                                ==================== U+2003 ISOpub -->

<!ENTITY thinsp    " " ><!-- thin space
                                ==================== U+2009 ISOpub -->

<!ENTITY zwnj      "&#8204;" ><!-- zero width non-joiner
                                ============== U+200C NEW RFC 2070 -->

<!ENTITY zwj       "&#8205;" ><!-- zero width joiner
                                ============== U+200D NEW RFC 2070 -->

<!ENTITY lrm       "&#8206;" ><!-- left-to-right mark
                                ============== U+200E NEW RFC 2070 -->

<!ENTITY rlm       "&#8207;" ><!-- right-to-left mark
                                ============== U+200F NEW RFC 2070 -->

<!ENTITY ndash     "&#8211;" ><!-- en dash
                                ==================== U+2013 ISOpub -->

<!ENTITY mdash     "&#8212;" ><!-- em dash
                                ==================== U+2014 ISOpub -->

<!ENTITY lsquo     "&#8216;" ><!-- left single quotation mark
                                ==================== U+2018 ISOnum -->
```

```
<!ENTITY rsquo    "&#8217;" ><!-- right single quotation mark
                               ==================== U+2019 ISOnum -->

<!ENTITY sbquo    "&#8218;" ><!-- single low-9 quotation mark
                               ======================= U+201A NEW -->

<!ENTITY ldquo    "&#8220;" ><!-- left double quotation mark
                               ==================== U+201C ISOnum -->

<!ENTITY rdquo    "&#8221;" ><!-- right double quotation mark
                               ==================== U+201D ISOnum -->

<!ENTITY bdquo    "&#8222;" ><!-- double low-9 quotation mark
                               ======================= U+201E NEW -->

<!ENTITY dagger   "&#8224;" ><!-- dagger
                               ==================== U+2020 ISOpub -->

<!ENTITY Dagger   "&#8225;" ><!-- double dagger
                               ==================== U+2021 ISOpub -->

<!ENTITY bull     "&#8226;" ><!-- bullet
                               black small circle
                               ==================== U+2022 ISOpub -->
                           <!-- bullet is NOT the same as U+2219,
                               'bullet operator' -->

<!ENTITY hellip   "&#8230;" ><!-- horizontal ellipsis
                               three dot leader
                               ==================== U+2026 ISOpub -->

<!ENTITY permil   "&#8240;" ><!-- per mille sign
                               ==================== U+2030 ISOtech -->

<!ENTITY prime    "&#8242;" ><!-- prime
                               minutes
                               feet
                               ==================== U+2032 ISOtech -->

<!ENTITY Prime    "&#8243;" ><!-- double prime
                               seconds
                               inches
                               ==================== U+2033 ISOtech -->

<!ENTITY lsaquo   "&#8249;" ><!-- single left-pointing angle quotation
                                    mark
                               ============== U+2039 ISO proposed -->

<!ENTITY rsaquo   "&#8250;" ><!-- single right-pointing angle quotation
                               ============== U+203A ISO proposed -->

<!ENTITY oline    "&#8254;" ><!-- overline
                               spacing overscore
                               ======================= U+203E NEW -->

<!ENTITY frasl    "&#8260;" ><!-- fraction slash
                               ======================= U+2044 NEW -->


<!--

     +-+-+-+-+-+-+-+-+-+-+-+
```

```
       Spacing Modifiers
     +-+-+-+-+-+-+-+-+-+-+

       Drawn From Unicode 3.1.1 Character Sets:

             Block Name(s):  Spacing Modifier Letters (U+0280 to U+02FF)
           Script Name(s):  (none)

       Note: The Spacing Modifier Letters are an unusual class of
             characters. They are an assorted collection of small signs
             used to indicate modifications of the preceding or
             following character, and sometimes to be an independent
             character. They differ from diacritical marks in that they
             are treated as free-standing, independent characters, which
             form part of the word and do not break up the word. They
             have the "letter" property. Most of the characters are
             phonetic modifiers. For further information, refer to
             Section 7.8 of the Unicode 3.0 manual, an online version is
             at http://www.unicode.org/unicode/uni2book/ch07.pdf .

-->


<!ENTITY circ      "&#710;" ><!-- modifier letter circumflex accent
                               ==================== U+02C6 ISOpub -->

<!ENTITY tilde     "&#732;" ><!-- small tilde
                               ==================== U+02DC ISOdia -->


<!--

     +-+-+-+-+-+-+-+-+-+-+
       Various Symbols
     +-+-+-+-+-+-+-+-+-+-+

       Drawn From Unicode 3.1.1 Character Sets:

             Block Name(s):  Latin-1 Supplement       (U+0080 to U+00FF)
                             Currency Symbols         (U+20A0 to U+20CF)
                             Letterlike Symbols       (U+2100 to U+214F)
                             Arrows                   (U+2190 to U+21FF)
                             Mathematical Operators   (U+2200 to U+22FF)
                             Miscellaneous Technical  (U+2300 to U+23FF)
                             Geometric Shapes         (U+25A0 to U+25FF)
                             Miscellaneous Symbols    (U+2600 to U+26FF)
           Script Name(s):  (none, except Greek for "micro", U+00B5)

-->


<!ENTITY nbsp      " " ><!-- no-break space
                               non-breaking space
                               ==================== U+00A0 ISOnum -->

<!ENTITY iexcl     "&#161;" ><!-- inverted exclamation mark
                               ==================== U+00A1 ISOnum -->

<!ENTITY cent      "&#162;" ><!-- cent sign
                               ==================== U+00A2 ISOnum -->

<!ENTITY pound     "&#163;" ><!-- pound sign
                               ==================== U+00A3 ISOnum -->
```

```
<!ENTITY curren    "&#164;" ><!-- currency sign
                                ==================== U+00A4 ISOnum -->

<!ENTITY yen       "&#165;" ><!-- yen sign
                                yuan sign
                                ==================== U+00A5 ISOnum -->

<!ENTITY brvbar    "&#166;" ><!-- broken bar
                                broken vertical bar
                                ==================== U+00A6 ISOnum -->

<!ENTITY sect      "&#167;" ><!-- section sign
                                ==================== U+00A7 ISOnum -->

<!ENTITY uml       "&#168;" ><!-- diaeresis
                                spacing diaeresis
                                ==================== U+00A8 ISOdia -->

<!ENTITY copy      "&#169;" ><!-- copyright sign
                                ==================== U+00A9 ISOnum -->

<!ENTITY laquo     "&#171;" ><!-- left-pointing double angle quotation
                                     mark
                                left pointing guillemet
                                ==================== U+00AB ISOnum -->

<!ENTITY not       "&#172;" ><!-- not sign
                                ==================== U+00AC ISOnum -->

<!ENTITY shy       "&#173;" ><!-- soft hyphen
                                discretionary hyphen
                                ==================== U+00AD ISOnum -->

<!ENTITY reg       "&#174;" ><!-- registered sign
                                registered trade mark sign
                                ==================== U+00AE ISOnum -->

<!ENTITY macr      "&#175;" ><!-- macron
                                spacing macron
                                overline
                                APL overbar
                                ==================== U+00AF ISOdia -->

<!ENTITY deg       "&#176;" ><!-- degree sign
                                ==================== U+00B0 ISOnum -->

<!ENTITY plusmn    "&#177;" ><!-- plus-minus sign
                                plus-or-minus sign
                                ==================== U+00B1 ISOnum -->

<!ENTITY sup2      "&#178;" ><!-- superscript two
                                superscript digit two
                                squared
                                ==================== U+00B2 ISOnum -->

<!ENTITY sup3      "&#179;" ><!-- superscript three
                                superscript digit three
                                cubed
                                ==================== U+00B3 ISOnum -->

<!ENTITY acute     "&#180;" ><!-- acute accent
                                spacing acute
```

```
                                    ==================== U+00B4 ISOdia -->

<!ENTITY micro     "&#181;" ><!-- micro sign
                                    ==================== U+00B5 ISOnum -->

<!ENTITY para      "&#182;" ><!-- pilcrow sign
                                    paragraph sign
                                    ==================== U+00B6 ISOnum -->

<!ENTITY middot    "&#183;" ><!-- middle dot
                                    Georgian comma
                                    Greek middle dot
                                    ==================== U+00B7 ISOnum -->

<!ENTITY cedil     "&#184;" ><!-- cedilla
                                    spacing cedilla
                                    ==================== U+00B8 ISOdia -->

<!ENTITY sup1      "&#185;" ><!-- superscript one
                                    superscript digit one
                                    ==================== U+00B9 ISOnum -->

<!ENTITY raquo     "&#187;" ><!-- right-pointing double angle quotation
                                         mark
                                    right pointing guillemet
                                    ==================== U+00BB ISOnum -->

<!ENTITY frac14    "&#188;" ><!-- vulgar fraction one quarter
                                    fraction one quarter
                                    ==================== U+00BC ISOnum -->

<!ENTITY frac12    "&#189;" ><!-- vulgar fraction one half
                                    fraction one half
                                    ==================== U+00BD ISOnum -->

<!ENTITY frac34    "&#190;" ><!-- vulgar fraction three quarters
                                    fraction three quarters
                                    ==================== U+00BE ISOnum -->

<!ENTITY iquest    "&#191;" ><!-- inverted question mark
                                    turned question mark
                                    ==================== U+00BF ISOnum -->

<!ENTITY times     "&#215;" ><!-- multiplication sign
                                    ==================== U+00D7 ISOnum -->

<!ENTITY divide    "&#247;" ><!-- division sign
                                    ==================== U+00F7 ISOnum -->

<!ENTITY euro     "&#8364;" ><!-- euro sign
                                    ======================= U+20AC NEW -->

<!ENTITY image    "&#8465;" ><!-- blackletter capital I
                                    imaginary part
                                    ==================== U+2111 ISOamso -->

<!ENTITY weierp   "&#8472;" ><!-- script capital P
                                    power set
                                    Weierstrass p
                                    ==================== U+2118 ISOamso -->

<!ENTITY real     "&#8476;" ><!-- blackletter capital R
                                    real part symbol
```

```
                                    ================== U+211C ISOamso -->

<!ENTITY trade    "&#8482;" ><!-- trade mark sign
                                    ================== U+2122 ISOnum -->

<!ENTITY alefsym  "&#8501;" ><!-- alef symbol
                                    first transfinite cardinal
                                    ====================== U+2135 NEW -->
                               <!-- alef symbol is NOT the same as
                                    U+05D0, 'Hebrew letter alef',
                                    although the same glyph could be
                                    used to represent both -->

<!ENTITY larr     "&#8592;" ><!-- leftwards arrow
                                    ================== U+2190 ISOnum -->

<!ENTITY uarr     "&#8593;" ><!-- upwards arrow
                                    ================== U+2191 ISOnum -->

<!ENTITY rarr     "&#8594;" ><!-- rightwards arrow
                                    ================== U+2192 ISOnum -->

<!ENTITY darr     "&#8595;" ><!-- downwards arrow
                                    ================== U+2193 ISOnum -->

<!ENTITY harr     "&#8596;" ><!-- left right arrow
                                    ================== U+2194 ISOamsa -->

<!ENTITY crarr    "&#8629;" ><!-- downwards arrow with corner leftwards
                                    carriage return
                                    ====================== U+21B5 NEW -->

<!ENTITY lArr     "&#8656;" ><!-- leftwards double arrow
                                    ================== U+21D0 ISOtech -->
                               <!-- Unicode does not say that lArr is
                                    the same as the 'is implied by'
                                    arrow, but also does not have any
                                    other character for that function.
                                    As ISOtech suggests, lArr can be
                                    used for 'is implied by'. -->

<!ENTITY uArr     "&#8657;" ><!-- upwards double arrow
                                    ================== U+21D1 ISOamsa -->

<!ENTITY rArr     "&#8658;" ><!-- rightwards double arrow
                                    ================== U+21D2 ISOtech -->
                               <!-- Unicode does not say that rArr is
                                    the same as the 'implies' arrow,
                                    but also does not have any other
                                    character for that function. As
                                    ISOtech suggests, rArr can be used
                                    for 'implies'. -->

<!ENTITY dArr     "&#8659;" ><!-- downwards double arrow
                                    ================== U+21D3 ISOamsa -->

<!ENTITY hArr     "&#8660;" ><!-- left right double arrow
                                    ================== U+21D4 ISOamsa -->

<!ENTITY forall   "&#8704;" ><!-- for all
                                    ================== U+2200 ISOtech -->

<!ENTITY part     "&#8706;" ><!-- partial differential
```

```
                                    ================== U+2202 ISOtech -->

<!ENTITY exist    "&#8707;" ><!-- there exists
                                    ================== U+2203 ISOtech -->

<!ENTITY empty    "&#8709;" ><!-- empty set
                                    null set
                                    diameter
                                    ================== U+2205 ISOamso -->

<!ENTITY nabla    "&#8711;" ><!-- nabla
                                    backward difference
                                    ================== U+2207 ISOtech -->

<!ENTITY isin     "&#8712;" ><!-- element of
                                    ================== U+2208 ISOtech -->

<!ENTITY notin    "&#8713;" ><!-- not an element of
                                    ================== U+2209 ISOtech -->

<!ENTITY ni       "&#8715;" ><!-- contains as member
                                    ================== U+220B ISOtech -->

<!ENTITY prod     "&#8719;" ><!-- n-ary product
                                    product sign
                                    ================== U+220F ISOamsb -->
                               <!-- prod is NOT the same character as
                                    U+03A0, 'Greek capital letter pi',
                                    although the same glyph could be
                                    used to represent both -->

<!ENTITY sum      "&#8721;" ><!-- n-ary summation
                                    ================== U+2211 ISOamsb -->
                               <!-- sum is NOT the same character as
                                    U+03A3, 'Greek capital letter sigma',
                                    although the same glyph could be
                                    used to represent both -->

<!ENTITY minus    "&#8722;" ><!-- minus sign
                                    ================== U+2212 ISOtech -->

<!ENTITY lowast   "&#8727;" ><!-- asterisk operator
                                    ================== U+2217 ISOtech -->

<!ENTITY radic    "&#8730;" ><!-- square root
                                    radical sign
                                    ================== U+221A ISOtech -->

<!ENTITY prop     "&#8733;" ><!-- proportional to
                                    ================== U+221D ISOtech -->

<!ENTITY infin    "&#8734;" ><!-- infinity
                                    ================== U+221E ISOtech -->

<!ENTITY ang      "&#8736;" ><!-- angle
                                    ================== U+2220 ISOamso -->

<!ENTITY and      "&#8743;" ><!-- logical and
                                    wedge
                                    ================== U+2227 ISOtech -->

<!ENTITY or       "&#8744;" ><!-- logical or
                                    vee
```

```
                                    ==================== U+2228 ISOtech -->

<!ENTITY cap      "&#8745;" ><!-- intersection
                                    cap
                                    ==================== U+2229 ISOtech -->

<!ENTITY cup      "&#8746;" ><!-- union
                                    cup
                                    ==================== U+222A ISOtech -->

<!ENTITY int      "&#8747;" ><!-- integral
                                    ==================== U+222B ISOtech -->

<!ENTITY there4   "&#8756;" ><!-- therefore
                                    ==================== U+2234 ISOtech -->

<!ENTITY sim      "&#8764;" ><!-- tilde operator
                                    varies with
                                    similar to
                                    ==================== U+223C ISOtech -->
                               <!-- tilde operator is NOT the same
                                    character as U+007E, 'tilde',
                                    although the same glyph could be
                                    used to represent both -->

<!ENTITY cong     "&#8773;" ><!-- approximately equal to
                                    ==================== U+2245 ISOtech -->

<!ENTITY asymp    "&#8776;" ><!-- almost equal to
                                    asymptotic to
                                    ==================== U+2248 ISOamsr -->

<!ENTITY ne       "&#8800;" ><!-- not equal to
                                    ==================== U+2260 ISOtech -->

<!ENTITY equiv    "&#8801;" ><!-- identical to
                                    ==================== U+2261 ISOtech -->

<!ENTITY le       "&#8804;" ><!-- less-than or equal to
                                    ==================== U+2264 ISOtech -->

<!ENTITY ge       "&#8805;" ><!-- greater-than or equal to
                                    ==================== U+2265 ISOtech -->

<!ENTITY sub      "&#8834;" ><!-- subset of
                                    ==================== U+2282 ISOtech -->

<!ENTITY sup      "&#8835;" ><!-- superset of
                                    ==================== U+2283 ISOtech -->

<!ENTITY nsub     "&#8836;" ><!-- not a subset of
                                    ==================== U+2284 ISOamsn -->

<!ENTITY sube     "&#8838;" ><!-- subset of or equal to
                                    ==================== U+2286 ISOtech -->

<!ENTITY supe     "&#8839;" ><!-- superset of or equal to
                                    ==================== U+2287 ISOtech -->

<!ENTITY oplus    "&#8853;" ><!-- circled plus
                                    direct sum
                                    ==================== U+2295 ISOamsb -->
```

```
<!ENTITY otimes   "&#8855;" ><!-- circled times
                                vector product
                                ==================== U+2297 ISOamsb -->

<!ENTITY perp     "&#8869;" ><!-- up tack
                                orthogonal to
                                perpendicular
                                ==================== U+22A5 ISOtech -->

<!ENTITY sdot     "&#8901;" ><!-- dot operator
                                ==================== U+22C5 ISOamsb -->
                             <!-- dot operator is NOT the same
                                character as U+00B7, 'middle dot' -->

<!ENTITY lceil    "&#8968;" ><!-- left ceiling
                                APL upstile
                                ==================== U+2308 ISOamsc -->

<!ENTITY rceil    "&#8969;" ><!-- right ceiling
                                ==================== U+2309 ISOamsc -->

<!ENTITY lfloor   "&#8970;" ><!-- left floor
                                APL downstile
                                ==================== U+230A ISOamsc -->

<!ENTITY rfloor   "&#8971;" ><!-- right floor
                                ==================== U+230B ISOamsc -->

<!ENTITY lang     "&#9001;" ><!-- left-pointing angle bracket
                                bra
                                ==================== U+2329 ISOtech -->
                             <!-- lang is NOT the same character as
                                U+003C, 'less than', or U+2039,
                               'single left-pointing angle quotation
                                mark' -->

<!ENTITY rang     "&#9002;" ><!-- right-pointing angle bracket
                                ket
                                ==================== U+232A ISOtech -->
                             <!-- rang is NOT the same character as
                                U+003E, 'greater than', or U+203A,
                               'single right-pointing angle quotation
                                mark' -->

<!ENTITY loz      "&#9674;" ><!-- lozenge
                                ==================== U+25CA ISOpub -->

<!ENTITY spades   "&#9824;" ><!-- black spade suit
                                ==================== U+2660 ISOpub -->

<!ENTITY clubs    "&#9827;" ><!-- black club suit
                                shamrock
                                ==================== U+2663 ISOpub -->

<!ENTITY hearts   "&#9829;" ><!-- black heart suit
                                valentine
                                ==================== U+2665 ISOpub -->

<!ENTITY diams    "&#9830;" ><!-- black diamond suit
                                ==================== U+2666 ISOpub -->
```

APPENDIX D: Differences Between the Basic OEBPS 1.2 and 1.0.1 Document Vocabularies

The Basic OEBPS Document vocabulary in this specification (hereafter referred to in this section as “Basic 1.2”) is similar to that for version 1.0.1 (“Basic 1.0”). The most significant difference is the addition of new elements and associated attributes (most notably improved table support), and the removal of nearly all the deprecated elements and attributes in Basic 1.0. It is noted that Basic 1.0 Document authors who avoided using any of the deprecated elements and attributes will, in general, find it easier to upgrade their documents to conform to Basic 1.2.

Following are the specific differences between Basic 1.2 and Basic 1.0.

D.1 Elements in Basic 1.0 Removed in Basic 1.2

Element

~~**center**~~

~~**font**~~

~~**s**~~

~~**strike**~~

~~**u**~~

All of these elements were deprecated in Basic 1.0, and are removed in Basic 1.2 since they are not included in XHTML 1.1. In place of these elements, use CSS.

D.2 Attributes in Basic 1.0 Removed in Basic 1.2

Elements	Attributes
a	~~*name*~~
body	~~*bgcolor*~~, ~~*text*~~
br	~~*clear*~~
div	~~*align*~~
h1 to **h6**	~~*align*~~
hr	~~*align*~~, ~~*size*~~, ~~*width*~~
img	~~*align*~~, ~~*border*~~, ~~*hspace*~~, ~~*vspace*~~
li	~~*type*~~
map	~~*name*~~
object	~~*align*~~, ~~*border*~~, ~~*hspace*~~, ~~*vspace*~~
ol	~~*type*~~

p	~~*align*~~
table	~~*align*~~, ~~*bgcolor*~~
td	~~*bgcolor*~~, ~~*height*~~, ~~*nowrap*~~, ~~*width*~~
th	~~*bgcolor*~~, ~~*height*~~, ~~*nowrap*~~, ~~*width*~~
tr	~~*bgcolor*~~

All of these attributes were deprecated in Basic 1.0, and are removed in Basic 1.2 since they are not included in XHTML 1.1. In place of the stylistic-oriented attributes, use CSS. For the *name* attribute, removed for **a** and **map**, use *id* instead.

D.3 Deprecated Basic 1.0 Attributes Undeprecated in Basic 1.2

Elements	Attributes
img	*height*, *width*
object	*height*, *width*

D.4 Deprecated Core Attribute ~~**style**~~

Following XHTML 1.1, the Core/Common attribute ~~*style*~~ is deprecated in Basic 1.2. It may be removed in a future version of this specification. Thus, for future upgradeability of documents, it is **strongly recommended** the ~~*style*~~ attribute not be used in Basic 1.2 documents; instead, use either the **style** element or external style sheets.

D.5 New Elements (and Included Attributes) Added in Basic 1.2

Elements	Attributes
abbr	
acronym	
address	
col	*align*, *span*, *valign*, *width*
colgroup	*align*, *span*, *valign*, *width*
del	*cite*, *datetime*
ins	*cite*, *datetime*
noscript	
tbody	*align*, *valign*
tfoot	*align*, *valign*
thead	*align*, *valign*

All of these new elements include support for the Common attribute set described in Section 3.2.1 (for brevity the Common attributes are not included in the above table.)

D.6 New Basic 1.2 Attributes Added to Pre-Existing Elements

Elements	Attributes
script	***type***
tr	*align*

D.7 Miscellaneous Differences in DTD Content Models, Elements and Attributes

Following are various differences in strict DTD content models, attribute data types, etc., between Basic 1.2 and Basic 1.0. These differences arise by Basic 1.2 being a pure subset of XHTML 1.1 as detailed in Section 3.1. Note that some of the following items apply only to Basic OEBPS Documents that are valid XML with respect to the Basic OEBPS Document DTD; however, it is strongly recommended that all Basic OEBPS 1.2 documents completely conform with the Basic OEBPS Document DTD (and thus XHTML 1.1) as mentioned in Section 3.1.

(i) In Basic 1.2, the *type* attribute for **style** is REQUIRED. In Basic 1.0 it was FIXED.

(ii) In Basic 1.2, the content model for **body** is Block.mix (Block level elements plus the "level-independent" elements), while for Basic 1.0 **body** could also contain PCDATA and Inline elements.

(iii) In Basic 1.2, **blockquote** can contain only Block.mix. In Basic 1.0, this element could also contain PCDATA and Inline elements. In essence, **blockquote** is identical to **body** in content model and can be thought of as sort of a "document within a document".

(iv) In Basic 1.2, **head** is required, while in Basic 1.0 it was optional.

(v) In Basic 1.2, the *type* attribute in **link** is IMPLIED (optional), while in Basic 1.0 it was REQUIRED. When identifying an external style sheet, *type* should be used to identify the MIME media type of the style sheet, such as **text/x-oeb1-css**.

(vi) In Basic 1.2, the data type for the *class* attribute is NMTOKENS. In Basic 1.0 it was CDATA.

(vii) In Basic 1.2, the data type for the attribute *usemap* in **object** is IDREF. In Basic 1.0 it was CDATA.

(viii) Basic 1.2 supports the value of **baseline** for the *valign* attribute (used in several table-related elements), while Basic 1.0 does not support it.

(ix) Note that because several new Table elements are added, the Basic 1.2 content model for **`table`** is significantly more complex than that for Basic 1.0. Refer to the Basic OEBPS Document DTD (Appendix B) for the exact content model.

APPENDIX E: CONTRIBUTORS

This specification has been developed through a cooperative effort, bringing together publishers, Reading System vendors, software developers, and experts in the relevant standards.

Version 1.2 of this specification was prepared by the Open eBook Forum Publication Structure Working Group. Active members of the working group at the time of publication of revision 1.2 were:

Jeff Alexander, Franklin Electronic Publishers

Garth Conboy, Gemstar eBook Group Limited (Version 1.2 Co-Editor)

Eva Conkright, Random House

Brady Duga, Gemstar eBook Group Limited (Version 1.2 Co-Editor)

Jerry Dunietz, Microsoft Corporation (Working Group Vice Chair)

George Kerscher, Daisy Consortium

Gene Golovchinsky, FX Palo Alto Laboratory, Inc. (Development Sub-Group Co-Chair)

Donald Goyette, McGraw-Hill

Rebecca Guenther, Library of Congress

Markku Hakkinen, isSound Corporation

Benjamin Jung, deepX

Christopher R, Maden, crism consulting

Bob Mathews, Adobe

Michael Moodie, National Library Service for the Blind and Physically Handicapped, Library of Congress

Jon Noring, Windspun Inc. (Maintenance Sub-Group Chair)

David Pawson, Royal National Institute for the Blind

Chris Powell, University of Michigan Library

James Pritchett, Recording for the Blind and Dyslexic

Allen Renear, Brown University Scholarly Technology Group (Working Group Chair)

Dorothea Salo, OverDrive, Inc. (Working Group Scribe)

Sheela Sethuraman, CAST

Roger Sperberg, e-bks, LLC

Patrick Stickler, Nokia

Mikhail Vaysbukh, Data Conversion Laboratory

Garret Wilson, GlobalMentor, Inc. (Development Sub-Group Co-Chair)

Harrison Yee, Apex

Contributors to the Version 1.2 effort who were no longer active working group members at the time of publication include:

David Goldstein, Versaware Inc.

Barbara Hammond, Nokia

Ilan Sender, Microsoft Corporation

Lisa Rupe, Questia Media Inc.

Version 1.0.1 of this specification was prepared by the Open eBook Forum Publication Structure Working Group. Active members of the working group at the time of publication of revision 1.0.1 were:

Fred Beuthel, LizardTech

Garth Conboy, Gemstar eBook Technology (Working Group Vice Chair)

Marisa DeMeglio, isSound Corporation

Kimmo Djupsjöbacka, Nokia

Brady Duga, Gemstar eBook Technology

Jerry Dunietz, Microsoft Corporation

George Kerscher, Daisy Consortium

David Goldstein, Versaware Technologies, Inc.

Gene Golovchinsky, FX Palo Alto Laboratory, Inc.

Rebecca Guenther, Library of Congress

Markku Hakkinen, isSound Corporation

Tony Lavender, netLibrary, Inc.

Michael Moodie, National Library Service for the Blind and Physically Handicapped, Library of Congress

Jon Noring, Windspun Inc.

David Pawson, Royal National Institute for the Blind

James Pritchett, Recording for the Blind and Dyslexic

Allen Renear, Brown University Scholarly Technology Group (Working Group Chair)

Lisa Rupe, Questia Media Inc.

Dorothea Salo, Impressions Book and Journal Services (Working Group Scribe)

Ilan Sender, Microsoft Corporation

Roger Sperberg, e-bks, LLC

Mikhail Vaysbukh, Data Conversion Laboratories

Garret Wilson, GlobalMentor, Inc. (Maintenance Sub-Group Chair)

Sarah Wood, OverDrive, Inc.

Version 1.0 of the specification was developed by the “Open eBook Authoring Group” which was composed of the following people (affiliations given are those at the time of publishing version 1.0, September 16, 1999):

FACILITATOR:

Victor McCrary, National Institute of Standards and Technology,
U.S. Department of Commerce

CONTRIBUTORS:

Jeff Alger, Microsoft Corporation

Garth Conboy, SoftBook Press

Horace Dediu, BCL Computers

Steve DeRose, Brown University Scholarly Technology Group

Kimmo Djupsjöbacka, Nokia

Brady Duga, SoftBook Press

Jerry Dunietz, Microsoft Corporation

David Goldstein, Versaware Inc.

Gene Golovchinsky, FX Palo Alto Laboratory, Inc

Markku Hakkinen, The Productivity Works

Gunter Hille, Project Gutenberg-DE

Kate Hughes, Microsoft Corporation

George Kerscher, DAISY Consortium

Steve Kotrch, Simon & Schuster

Brian Lambert, Glassbook Inc.

Jon Noring, Exemplary Technologies

Aleksey Novicov, Softbook Press

David Ornstein, NuvoMedia, Inc.

Dhiren Patel, NuvoMedia, Inc.

Steve Potash, OverDrive Systems

Allen Renear, Brown University Scholarly Technology Group

Mike Riley, R.R. Donnelley & Sons Company

Ben Trafford, Exemplary Technologies

John Voiklis, Red Figure, Inc.

Garret Wilson, GlobalMentor, Inc.

OTHER CONTRIBUTING ORGANIZATIONS:

Adobe

EAST Co., Ltd

IBM

Librius

Vadem