Open eBook™ Publication Structure 1.0

September 16, 1999

Open eBook™ Publication Structure 1.0

September 16, 1999

TABLE OF CONTENTS

1 Overview

1.1 Purpose and Scope

The purpose of the Open eBook Publication Structure is to provide a specification for representing the content of electronic books. Specifically:

- The specification is intended to give content providers (e.g., publishers, and others who have content to be displayed) and tool providers minimal and common guidelines which ensure fidelity, accuracy, accessibility, and 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 provide the purveyors of electronic-book content (publishers, agents, authors et al.) a format for use in providing content to multiple reading systems.

This specification is based on the premise that 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.

1.2 About This Version

This specification has been developed through a cooperative effort, bringing together publishers, reading system vendors, software developers, and experts in the relevant standards. This "Open eBook Authoring Group" is composed of:

Facilitator:
Victor McCrary, National Institute of Standards and Technology,
      U.S. Department of Commerce

Authors:
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

1.3   Definitions

Basic OEB Document

An OEB document which restricts itself to the constructs defined in the 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 become obsolete in future revisions.

Extended OEB Document

An OEB document which uses constructs beyond those in this specification, but uses the extension mechanism defined herein.

OEB Core Media Type

A MIME media type that all reading systems must support.

OEB Document

An XML document which conforms to this specification.

OEB Package

A file that describes an OEB publication. It identifies all other files in the publication and provides descriptive and access information about them.

OEB Publication

A collection of OEB documents 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 OEB publications, and directly or indirectly makes them available to readers. Great variety is possible in the architecture of reading systems. A reading sytem may be implemented entirely on one device, or it may be split among several computers. In particular, a reading device that is part of a larger reading system need not directly accept OEB 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.

1.4 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 markup meta-language (http://www.w3.org/TR/REC-xml);

2. the XML namespace specification (http://www.w3.org/TR/REC-xml-names);

3. the HTML 4.0 document content markup language (http://www.w3.org/TR/REC-html40), with consideration of the pending XHTML 1.0 specification (http://www.w3.org/TR/xhtml1/);

4. the CSS 1 stylesheet language (http://www.w3.org/TR/REC-CSS1), with a very few properties also from CSS 2 (http://www.w3.org/TR/REC-CSS2);

5. the Dublin Core metadata language (http://purl.org/dc/) and the USMARC relator code list (http://www.loc.gov/marc/relators/re9802r1.html);

6. the Unicode character set (http://www.unicode.org);

7. particular MIME media types (http://www.ietf.org/rfc/rfc1738.txt).

1.4.1 Relationship to XML

OEB is based on XML because of its generality and simplicity, and because this increases the likelihood that documents will survive longer. XML also provides well-defined rules for the syntax of documents, which decreases the cost to implementers and reduces incompatibility across systems. Further, XML enables extensibility because 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 formatting and other types of computer processing.

OEB reading systems must be XML processors as defined in XML 1.0. All OEB documents must be well-formed XML documents, although they need not be valid XML documents. However, this specification ensures that for any basic OEB document, there is a syntax form that:

- is a valid XML document,

- conforms fully to the OEB document DTD,

- is expected to conform to XHTML 1.0 when that specification is issued, and

- is effectively previewable in typical version 4 HTML browsers.

The last point above does not claim full HTML 4.0 conformance for one primary reason: HTML 4.0 supports the XML empty element syntax in practice (i.e., it works in browsers) but not in formal HTML 4.0 specifications prior to XHTML 1.0.

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 "&";

- All element names and attribute names must be consistent in case (all OEB 1.0 names are, as expected for XHTML 1.0, lower-case);

- All empty elements must use the XML empty element syntax (this specification also requires whitespace before the trailing slash, although such space is optional in XML; for example, "<br />").

Empty elements are those (such as the HTML *br* and *hr* elements) 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 (and is strictly conformant XML, as XML ignores whitespace within tags. For this reason, this specification requires this (conforming) variation of the XML form (for example, "*
*"); this is the most portable syntax and it contributes the most 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 OEB documents. This is because this specification imposes further constraints in order to improve interoperability. These constraints are the "OEB Common Requirements," defined below.

This specification defines two XML DTDs – the package DTD and the basic OEB document DTD. The package forms the "root" of a complete publication, and reading systems should use it to find and organize publication components. The basic OEB document DTD formally defines the HTML subset described in this specification.

1.4.2   Relationship to XML Namespaces

This version of the specification does not demand that reading systems process XML namespace prefixes according to the XML Namespaces Recommendation at http://www.w3.org/TR/REC-xml-names.

Namespace prefixes are a method for prefixing element and attribute names to distinguish identical names that are drawn from different definition sets. A prefix is associated with a unique URI by an XML namespace declaration. Alternatively, a namespace declaration 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 by a colon.

This specification forbids the use of an *oeb:* namespace prefix within OEB documents . The use of the *dc:* prefix, however, is required for Dublin Core metadata element attributes in the OEB package. An element

within the OEB document that contains a namespace prefix is treated as an extended element, with the colon acting as a normal XML name character. Reading systems must recognize the colon as a valid name character within an OEB document.

For upwards compatibility, the element *metadata* in an OEB package is required to have an attribute of *xmlns:dc="http://purl.org/dc/elements/1.0/"* and *xmlns:oebpackage="http://openebook.org/namespaces/oeb-package/1.0/"*. In addition, the Dublin Core elements will be declared in the OEB package DTD with an explicit prefix of *dc:*.

1.4.3   Relationship to HTML

This specification recognizes the importance of current software tools, legacy data, publication practices, and market conditions, and so is based on HTML. This approach allows content providers to exploit current HTML 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 publication structure does not include all HTML elements and attributes. The elements and attributes were selected from the HTML 4.0 specification and were chosen to be consistent with current directions in HTML and XHTML development and the emergence of XML. See Appendix A for a complete table of the element types in HTML 3.2, HTML 4.0, and OEB documents.

Any HTML construct deprecated in HTML 4.0 is either omitted from this specification or is deprecated; CSS-based equivalents are provided in such cases. Stylesheet constructs are also used for new functionality beyond that provided in HTML 4.0..

To achieve predictable results and to support upwards compatibility with future versions of this specification, it is strongly recommended that basic OEB documents be valid XML documents with respect to the OEB document DTD.

1.4.4   Relationship to CSS

This specification defines a style language based on CSS 1 and CSS 2, with a media type of "*text/x-oeb1-css"*. The OEB Authoring Group is aware that this definition of a media type goes against the recommendation of the CSS Working Group (see http://www.w3.org/TR/REC-CSS1), but has chosen to do so due to practical considerations.

The CSS-based stylesheet constructs in this specification have been included to define baseline rendering functionality. To minimize the burden on reading system developers and device manufacturers, not all CSS 1 or CSS 2 properties are included. A few additional properties and values have been added for supporting 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 stylesheet 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 OEB. 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 inline *style* attribute, the *style* element, and externally linked stylesheets. This specification does not require that any handling of XML namespaces be performed by the reading system in the processing of stylesheets.

Stylesheets can be associated with an OEB document in several ways:

1.   by *style* attributes on specific HTML elements;

2.   by *style* elements within the HTML header;

3.   by an external stylesheet identified on a *link* element in the HTML *head*; and/or

4.   by an external stylesheet identified via a processing instruction as defined in the W3C proposed recommendation "Associating stylesheets with XML documents" (http://www.w3.org/TR/xml-stylesheet), or its final form if and when it is adopted.

The relative priority of the first three cases is as defined for HTML 4.0 and CSS 2. Stylesheets linked via a processing instruction are treated as if they had been linked via HTML *link* elements preceding any actual HTML *link* elements. As defined in the Conformance section, if no stylesheet is defined or no applicable style is found for a given element, HTML rendering is the default as defined elsewhere in this specification and the HTML4.0 specification.

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

Only those CSS constructs defined as supported in Section 4 of this specification may be included in stylesheets of type "*text/x-oeb1-css*". Stylesheets of other MIME media types may be substituted for the *text/x-oeb1-css* stylesheets at the discretion of the reading system.

The HTML 4.0 specification groups externally linked stylesheets into sets by their titles (including a "persistent" set for which the title is the null string). This specification requires that at least one stylesheet in each such set must be of MIME media type "*text/x-oeb1-css*".

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

1.4.5   Relationship to Dublin Core

The Dublin Core is designed to minimize the burden of cataloging on authors and publishers, while providing enough data to be useful. This specification supports the entire current set of Dublin Core metadata elements, 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.

1.4.6   Relationship to Unicode

Publications may use the entire Unicode character set, in UTF-8 or UTF-16 encodings, as defined by Internet RFC 2279 (see http://info.internet.isi.edu/in-notes/rfc/files/rfc2279.txt). The use of Unicode

facilitates internationalization and multilingual documents. However, reading systems are not required to provide glyphs for all Unicode characters.

Reading systems are required to 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).

1.4.7   MIME Media Types

This specification defines a list of "core" MIME media types that all reading systems must support and publications may include. Publications may include resources of other media types, but for each such resource must include an alternative resource of a core MIME media type (using methods defined in this specification).

The core OEB MIME media types are:

Media Type	Reference	Description
image/jpeg	RFC 2046	Used for raster graphics
image/png	RFC 2083	Used for raster graphics
text/x-oeb1-document	this specification	Used for basic or extended OEB documents
text/x-oeb1-css	this specification	Used for OEB CSS-subset stylesheets

1.5 Conformance

This section defines conformance for OEB documents, publications, and reading systems.

1.5.1   Document and Publication Conformance

This specification defines two named levels of conformance for OEB documents–basic and extended, and one conformance level for OEB publications.

1.5.1.1   Common Requirements

A conformant OEB document (both basic and extended), or conformant OEB 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)   for empty elements it uses only the XML empty element syntax with whitespace before the trailing slash;

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

(vi)   any attribute with a value of NMTOKEN, ID, or IDREF must be a XML name.

1.5.1.2   Basic OEB Document

A document is a basic OEB document if and only if:

(i)   it meets the common requirements;

(ii)   its DOCTYPE declaration, if any, references the basic OEB document DTD;

(iii)   it uses only the element names, attribute names, and attribute values defined in this specification; all element and attribute names must be in lower case;

(iv)   it uses element names, attribute names, and attribute values in a manner broadly consistent with intentions of the relevant descriptions in this specification and those of HTML 4.0 and the Dublin Core, with this specification taking precedence in the event of conflicts;

(v)   if external stylesheets are used, then at least one stylesheet in each title set (as described in the HTML 4.0 specification), including any "persistent" set, must be of MIME media type "*text/x-oeb1-css*"; and

(vi)   all style parameters specified within the document itself belong to the OEB CSS subset.

1.5.1.3   Extended OEB Document

A document is an extended OEB document if and only if

(i)   it meets the OEB common requirements;

(ii)   it uses elements, attributes, or attribute values not defined in this specification, or its DOCTYPE declaration references a DTD other than the basic OEB document DTD;

(iii)   for any element not in this specification it must provide an applicable style rule using only the OEB CSS subset;

(vii)   if external stylesheets are used, then at least one stylesheet in each title set (as described in the HTML 4.0 specification), including any "persistent" set, must be of MIME media type "*text/x-oeb1-css*"; and

(viii)   all style parameters specified within the document itself belong to the OEB CSS subset.

1.5.1.4   Validity

OEB documents, basic or extended, may or may not be valid (as defined in XML 1.0) with respect to an associated DTD. However, all OEB documents must be well-formed XML 1.0 documents.

1.5.1.5   Publication Conformance

A collection of files is a conforming OEB publication if and only if

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

(ii)   the OEB package file includes one and only one manifest entry corresponding to each other file in the 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 core MIME media types, conforms as defined for those MIME media types;

(v)   the *dc-metadata* element contains at least one *dc:Identifier* and at least one *dc:Title* 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:Contributor* element's *role* attribute begin with "oth."; and

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

1.5.2   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 OEB document the reading system

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

(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 HTML 4.0, CSS 2, and Dublin Core (in case of any conflict, this specification takes precedence).

B) When presented with an extended OEB document the reading system

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

(ii)   recognizes element instances not from this specification and renders them according to any applicable stylesheet rules, as described in section 1.4.4;

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

(iv)   does not render objects of unsupported media types, in the absence of fallbacks. These fallbacks are clearly defined in sections 2.3.1, 3.6.2, 3.7.3, 3.30 and 3.36.

C) When presented with an OEB package the reading system

(i)   must process all elements and attributes as described in chapter 2 of this specification.

D) When providing navigation via the OEB spine, the reading system

(i)   must not render content that does not have the media type *text/x-oeb1-document*; and

E) When presented with one or more style sheets via the HTML *link* mechanism or the XML-stylesheet processing instruction, described in http://www.w3.org/TR/xml-stylesheet, the reading system

(i)   is able to process the document in accordance with the *text/x-oeb1-css* style sheets; and

(ii)   if stylesheets of a MIME media type other than *text/x-oeb1-css* are provided, may choose to substitute those stylesheets for the *text/x-oeb1-css* stylesheets. Reading systems (although not necessarily reading devices) which support other stylesheet media types must provide a mechanism for requesting that those stylesheets be ignored in favor of the *text/x-oeb1-css* stylesheets.

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 OEB documents.

1.5.3   Compatibility with Future Versions

It is the intent of the authors of 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 OEB 1.0 documents) that are more restrictive than XML, requiring further XML processing capability of OEB 1.0 conformant reading systems, and perhaps supporting other XML-related standards such as Xlink;

- 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; and

- Future versions of OEB will support a clearer separation of content and formatting, while being fully backwards-compatible with this specification.

1.6 Extensibility

Use of extended OEB documents is the recommended mechanism for adding information and structure beyond that provided by the HTML subset defined in this specification, for example, to associate further semantics with content. Arbitrary non-OEB elements may be added as long as such elements are provided with style definitions in accompanying stylesheets.

For example, the following document would be an extended OEB document excerpt:

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

if associated with a style sheet that contains the following excerpt:

```
chapter      {page-break-before: always}
milestone    { }
chapterhead {
      font-weight: bold;
      font-family: sans-serif;
      text-align: center;
      display: block;
      margin-top: 4ex
}
```

1.7 Accessibility

This specification incorporates features that ensure content can be made accessible and usable by persons with disabilities. Existing accessibility features developed by the World Wide Web Consortium (W3C) for HTML 4.0 for content accessibility are incorporated into the OEB specification.

OEB 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 OEB implementers to ensure that reading systems will be in conformance with accessibility requirements.

1.8 Future Directions

This specification is designed to take advantage of current practices while preparing for future developments. Although the nature of subsequent versions of this specification remain to be determined, it is the expectation of the authors of this specification that these releases will continue in the same directions, including: increasing functionality, accessibility, internationalization, extensibility, and more rigorous separation of content and presentation, while maintaining alignment with relevant standards. Further, maintaining backward compatibility with this version of this specification is a high priority. Future directions can be tracked at http://www.openebook.org.

2 The OEB Package

A publication conforming to this specification must include exactly one OEB package file, which specifies the OEB documents, images, and other objects that make up the OEB publication and how they relate to each other.

It is recommended that all package files use extension ".opf", in order to make them readily identifiable within the group of files making up a 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 OEB 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 OEB package file are:

- **PACKAGE IDENTITY** – a unique identifier for the OEB 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 OEB package must be a valid XML document conforming to the OEB package DTD. An informal outline of the package is as follows:

```
<?xml version="1.0"?>
<!DOCTYPE package
  PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.0 Package//EN"
  "http://openebook.org/dtds/oeb-1.0/oebpkg1.dtd">
<package>
        metadata
        manifest
        spine
        guide
</package>
```

The following sections describe the parts of the OEB package.

2.1   Package Identity

The *package* element is the outermost 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 *dc-metadata* element contains specific publication-level metadata as defined by the Dublin Core initiative (http://purl.org/dc/). The descriptions below are included for convenience, and the Dublin Core's own definitions take precedence (see http://www.ietf.org/rfc/rfc2413.txt).

The *x-metadata* element must contain one or more instances of a *meta* element, analogous to the HTML 4.0 *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 language. Individual OEB documents may include the *meta* element directly (as in HTML 4.0) for document-specific metadata. This specification uses the OEB package file alone as the basis for expressing publication-level Dublin Core metadata.

For example:

```
<metadata>
   <dc-metadata>
     …
   </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 *metadata* in an OEB package is required to have an attribute of *xmlns:dc="http://purl.org/dc/elements/1.0/"* 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. The *dc:Title* and at least one *dc:Identifier* must be included in the *dc-metadata* element. Dublin Core elements, like any other elements in the OEB 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*.

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 a *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/1.0/"
xmlns:oebpackage="http://openebook.org/namespaces/oeb-
package/1.0/">
                  <dc:Title>Alice in Wonderland</dc:Title>
                  <dc:Type>Novel</dc:Type>
                  <dc:Identifier id="xyz"
                   scheme="ISBN">123456789X</dc:Identifier>
                  <dc:Creator role="aut">Lewis Caroll</dc:Creator>
            </dc-metadata>
      </metadata>
      ...
  </package>
```

Attributes of elements within the *dc-metadata* element are not defined by the 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. Although an OEB package must include at least one instance of this element type, 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.

Publication 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.

The dc:Creator elements has 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>

Plain text describing the publication's content.

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

The publisher as defined in RFC2413.

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 semantic 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.

The *dc:Contributor* element has 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 USMARC relator code list (http://www.loc.gov/marc/relators/re9802r1.html). When roles are specified, the 3-character USMARC 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 USMARC 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, afterword, or other critical matter, but who is not the chief author.

Note: The author of a specific part of the book may be expressed using a *meta* element in that part.

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 formats which have no equivalent in USMARC 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, modification.

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

Dublin Core provides examples of resource types "such as home page, novel, poem, working paper, technical report, essay, dictionary." The Dublin Core is currently considering revisions to the usage of this field, and should be consulted for updated definitions.

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

An enumerated list of formats for this content is being developed by the Dublin Core.

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

A string or number used to uniquely identify the resource.

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

The *dc:Identifier* element has an optional attribute, *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 RFC 2413).

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

If specified, the content must be an RFC 1766-compliant specification of the primary language of the content. (See http://www.ietf.org/rfc/rfc1766.txt.) US English (*en-us*) is the default. 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. An enumerated list of relationship types is currently under development by the Dublin Core.

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

The place or time which the publication's content addresses.

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

A statement about rights, or a link 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 *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 stylesheet, or other component that is considered part of the publication.

Each *item* element contained within a *manifest* element must have an *id* attribute. In addition, each *item* element in the *manifest* must have an *href* attribute, which is a URI (if relative, the URI is interpreted as relative to the package file itself). Additionally, each *item* element must have its MIME media type specified via the *media-type* attribute.

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*, may not use fragment identifiers.

2.3.1   Fallback items

This specification defines a set of core MIME media types that all conforming reading systems must support. For a publication that uses only those media types, the *manifest* merely needs to list the publication's component files directly. However, content providers may construct publications that reference items of additional media types. In order that such publications can nevertheless be read by all

conforming reading systems, content providers must always provide alternative "fallback" items for each such item. For every item that is not in a core media type, at least one of its associated fallback items must be of a type drawn from the set of core media types.

This specification defines two different mechanisms for specifying core media type fallbacks. For inline "replaced" elements, this specification relies on the HTML-based *object* element, described in section 3. For non-inline destinations, whether referenced from a document or a package, the *fallback* attribute of an *item* is used.

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. Every item in a publication that is not of one of the core MIME media types must, directly or indirectly, specify a fallback path to an item of one of the core MIME media types.

Fallback paths must terminate; circular references are not permitted. Nevertheless, reading systems should not fail catastrophically if they encounter such a cycle.

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 OEB 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* may refer only to *item* elements of media type *text/x-oeb1-document*. Content of other media types may be referenced via OEB documents, which can (and 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 OEB 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 OEB 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 OEB 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. 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 system 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 OEB 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).

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 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   HTML/XML Markup Constructs

This section lists all element types included in the OEB document specification. They are drawn from the HTML 4.0 specification found at http://www.w3.org/TR/REC-html40. See also Appendix A, which includes an index of element types and their status in HTML 4.0 and this specification.

3.1   Notational Conventions

This section does not list HTML 4.0 element types that are not supported in OEB documents (the table in Appendix A, however, does show them explicitly and marks them as excluded). For elements that are supported, this section does, however, explicitly indicate attributes that are ruled out. Reading systems may accept the presence of additional attributes (except in valid OEB documents), but need not support their semantics.

The following typographic conventions are used to indicate the status of element types, attributes, and attribute values:

3.1.1   ~~Strikethrough~~

Attributes and attribute values whose names are ~~struckthrough~~ below are HTML 4.0 constructs that are explicitly not supported by this specification. Reading systems may , but need not provide default presentation for them. In some cases, the constructs are intended to be permanently obsolete; in other cases, future revisions of this specification may add support for these constructs.

3.1.2   <u>Underline</u>

Elements, attributes, and attribute values whose names are <u>underlined below</u> are additions beyond the HTML 4.0 specification. Specifically:

- The *lowsrc* (de facto) attribute on *img; and*

- The *just* (de facto) value for *align* on various elements (which is, nevertheless, deprecated).

3.1.3   *(meaning changed)*

Elements, attributes, and attribute values whose names are followed by *"(meaning changed)"* below have had their semantics altered from those of the HTML 4.0 specification.

3.1.4   (deprecated)

Elements, attributes, and attribute values whose names are followed by *"(deprecated)"* below are *deprecated* (that is, permitted but not recommended) in OEB documents. The accompanying text indicates whether they are also deprecated in HTML 4.0. Although reading systems must recognize and support them, future releases of this specification are likely to prohibit them. Most but not all are deprecated in HTML 4.0. All elements, attributes, and attribute values that are deprecated in HTML 4.0 are either deprecated or prohibited by this specification.

3.1.5   <x> </x> versus

Elements that cannot contain subelements or text content are shown with the required empty-element syntax, while others are shown as start- and end-tag pairs. Note the XML documents may not omit start- or end-tags. The supported HTML elements for which this restriction is relevant include: *html, body, head, dt, li, p, caption, td*, and *th*.

3.2   Generally Applicable Attributes

HTML 4.0 defines the following four attributes that may be applied to nearly all elements. These attributes are therefore not listed in the following element descriptions, except to note their absence from the few exceptional elements.

3.2.1   id

This attribute is used to give a unique identifier to each element. Values for *id* must be unique across all elements in a single document, and may consist of the full range of XML name characters. For compatibility with HTML, restricting the *id* values to the HTML name characters (A-Za-z0-9.-) is recommended.

3.2.2   style

This attribute is included to allow for inline style specification. See the discussion of supported CSS properties elsewhere in this specification.

3.2.3   class

This attribute is included to allow selector-based style specifications. Its value must be a space-separated list of class names. OEB class names are limited to valid HTML 4.0 names, although HTML 4.0 allows arbitrary CDATA.

3.2.4   title

This attribute may be used to provide an "advisory title/amplification" for the element. Reading systems are not required to do anything in particular with its value.

3.3   Rendering on Limited Reading Systems

A number of elements and attributes that are allowed below permit semantics that are not required of all OEB 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 HTML construct, 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 HTML constructs. Some reading systems may choose to express the intent of elements in presentation by closely following web-browser usages – 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.

3.4   <?xml ?>

An OEB document must begin with the XML processing instruction

```
<?xml version="1.0" ?>
```

It is recommended that a character encoding be specified, for example:

```
<?xml version="1.0" encoding="UTF-16" ?>
```

3.5   <!DOCTYPE>

A *<!DOCTYPE>* declaration in a basic OEB document must reference the OEB document DTD, by system and optionally the public identifier. For example:

```
<!DOCTYPE html PUBLIC
  "+//ISBN 0-9673008-1-9//DTD OEB 1.0 Document//EN"
  "http://openebook.org/dtds/oeb-1.0/oebdoc1.dtd">
```

Note that the *<!DOCTYPE>* declaration must not include a trailing slash, but may include "[ ]" preceding the closing ">". "DOCTYPE" must be in upper case. Also, implementations should normalize whitespace in PUBLIC identifiers in accordance with the rules defined in XML, though the use of non-normalized whitespace in PUBLIC identifiers in OEB documents is prohibited.

3.6   <a>

Marks either the origin or destination for a hypertext link. Note that many browsers by default underline anchors with *href* attributes, to visually indicate links. Similarly, reading systems will use some sort of highlighting, perhaps some form of underline, to show links. The CSS-property setting *text-decoration: none* may be used to suppress such highlighting.

3.6.1   name *(deprecated)*

Used to provide an address to which hypertext links may refer.

Deprecated in HTML 4.0 and in this specification, in favor of the *id* attribute. HTML 4.0 allows any string as the value of this attribute; this specification limits them to being XML names.

3.6.2   href

Provides a URI to be used as the destination when traversal from the *a* element is requested. If this URI references a destination that is not an OEB core media type, a fallback must be provided in the manifest to an OEB core media type. If a fallback is not provided in the manifest, content at the end of this link must not be rendered by the reading system.

3.6.3   rel, rev

May be used to indicate the relationship between the document or portion linked, and the current document or portion.

3.6.4   ~~charset, type, hreflang, accesskey, shape, coords, tabindex, onfocus, onblur~~

Omitted from this specification.

3.7   *<area />*

3.7.1   shape

The possible values are:

rect	The default. Coordinates are left, top, right, bottom values.
circle	Coordinates are center x, center y, and radius.
poly	Coordinates are x, y pairs for successive points.
default	The default type, *rect*, can also be set explicitly.

3.7.2   coords

Comma-separated list of lengths, as integer coordinates or percentages. The values are interpreted according to the specific *shape* value.

3.7.3   href

Provides a URI to be used as the destination when traversal from this area is requested (for example by clicking within a hotspot). If this URI references a destination that is not an OEB core media type, a fallback must be provided in the manifest to an OEB core media type. If a fallback is not provided in the manifest, the destination resource must not be rendered by the reading system. A non-null value for the *alt* attribute of this element will be considered acceptable fallback.

3.7.4   nohref

If set to *nohref="nohref"*, indicates that the area isn't "hot," effectively "punching a hole" in a larger hot region.

3.7.5   alt

Should be used to provide an alternative textual representation for the hot-spot, accessible to print-disabled users.

3.7.6   ~~tabindex~~, ~~accesskey~~, ~~onfocus~~, ~~onblur~~

Omitted from this specification.

*3.8   *

The *b* tag is commonly used in HTML documents. Use of CSS *font-weight:bold* or the HTML *strong* tag is preferable.

3.9   <base />

Used in resolving links from OEB documents.

3.9.1   href

Provides a URI to be used as the basis for relative URIs elsewhere in the document.

3.9.2   ~~id, style, class, title~~

HTML 4.0 and this specification exclude these general attributes from this element.HTML 4.0 and this specification exclude these general attributes from this element.

3.10 <big> </big>

3.11 <blockquote> </blockquote>

3.11.1   cite

This optional attribute provides a URI for the source of the quoted material.

3.12 <body> </body>

This element defines a container for OEB content. It is assumed, in formatting, that its default rendering is consistent with the CSS property *page-break-before:always*, but may be overridden by an appropriate stylesheet declaration.

3.12.1 bgcolor *(deprecated)*

Deprecated in HTML 4.0 and in this specification, in favor of CSS *background-color* equivalent. See section 4.1, the CSS *color* and *background-color* properties, for discussion of acceptable values.

3.12.2 text *(deprecated)*

Deprecated in HTML 4.0 and in this specification, in favor of CSS *color* equivalent. See section 4.1, the CSS *color* and *background-color* properties, for discussion of acceptable values.

3.12.3 ~~link, vlink, alink, background~~

This specification does not require reading systems to support color control of link anchors or background bitmaps. These attributes are deprecated in HTML 4.0 and omitted from this specification.

3.12.4 onload, onunload

Omitted from this specification.

3.13

3.13.1 clear *(deprecated)*

> The possible values are:
> none (default)
> left
> right
> all

Deprecated in HTML 4.0, in favor of CSS *clear* property.

3.14 <caption> </caption>

When present, the *caption* element's text should describe the nature of the table.

3.14.1 align

Omitted from this specification.

3.15 <center> </center> (deprecated)

Equivalent to *<div align="center">*. Deprecated in HTML 4.0 and in this specification.

3.16 <cite> </cite>

Citations of other sources.

3.17 <code> </code>

Computer program fragments.

3.18 <dd> </dd>

Definition description.

3.19 <dfn> </dfn>

Inline definitions, such as the first instance of a definition.

3.20 <div> </div>

3.20.1 align *(deprecated)*

See section 3.25.1, *h1,* for additional discussion.

3.21 <dl> </dl>

Definition list.

3.21.1 ~~compact~~

See section 3.58.2, *ul*, for additional discussion.

3.22 <dt> </dt>

Definition term.

3.23

Emphasis, typically formatted like *i*.

3.24   (*deprecated*)

The *font* element is deprecated in HTML 4.0 and this specification, with its functionality replaced by CSS equivalents.

3.24.1 size *(deprecated)*

3.24.2 color *(deprecated)*

See section 3.12.1, the *bgcolor* and *text*.attributes of the *body* element, for additional discussion.

3.24.3 face *(deprecated)*

3.25 <h1> </h1> through <h6> </h6>

3.25.1 align *(deprecated)*

The possible values are:
        left
        center
        right

justify

Deprecated in favor of the equivalent CSS construct, *text-align*.

3.26 <head></head>

3.26.1   ~~profile~~

Omitted from this specification.

3.26.2   ~~id, style, class, title~~

HTML 4.0 excludes these general attributes from this element.

3.27 <hr />

3.27.1   align *(deprecated)*

The possible values are:
>   left
>   right
>   center

Deprecated in favor of the equivalent CSS construct, *text-align*.

3.27.2   ~~noshade~~

Deprecated in HTML 4.0, omitted from this specification.

3.27.3   size *(deprecated)*

Deprecated in HTML 4.0 and this specification, in favor of CSS *height* .

3.27.4   width *(deprecated)*

Deprecated in HTML 4.0 and this specification, in favor of CSS *width*. Value is specified in pixels or percentage.

3.28 <html> </html>

This element is required in OEB documents.

3.28.1   ~~id, style, class, title~~

HTML 4.0 and this specification exclude these general attributes from this element.

3.29 *<i> </i>*

The *i* tag is commonly used in HTML documents. Use of CSS *fontstyle:italic* or the HTML *em* tag is preferable.

3.30 *<img />*

A reference to an image to be displayed inline. This element may only be used to refer to images of OEB core media types (PNG and JFIF [JPEG]). Inline references to non-core media types require specification of alternate (fallback) rendering. The *object* element provides a general fallback mechanism by which all reading systems may locate an alternate of a core media type, and is the preferred mechanism for including non-core media types. However, a non-null value for the *alt* attribute is also considered an adequate fallback.

3.30.1 src

Images must be of type PNG (see http://www.ietf.org/rfc/rfc2083.txt) or JFIF (see ISO/IEC 10918-1:1994(E) and http://www.igd.fhg.de/icib/it/iso/is_10918-1/gen.html).

3.30.2 alt

Should be used to provide a non-graphical alternative to a graphic. This is crucial to accessibility of documents to print-disabled users, as well as for use on very low-resolution display devices. See http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505/#gl-provide-equivalents for further information on the use of this and related accessibility attributes.

3.30.3 longdesc

Value should be a URI pointing to a long description, typically more detailed than, but complementary to, *alt*. Use of this attribute is strongly recommended because it can greatly enhance accessibility for print-disabled users.

3.30.4 align *(deprecated)*

Deprecated in HTML 4.0 and this specification, in favor of CSS *float* and *vertical-align* properties.

The possible values are:
       top
       middle
       bottom
       left
       right

3.30.5 width *(deprecated)*

Deprecated in this specification, though not in HTML 4.0, in favor of CSS *width* property.

3.30.6 height *(deprecated)*

Deprecated in this specification, though not in HTML 4.0, in favor of CSS *height* property.

3.30.7 border *(deprecated)*

Deprecated in HTML 4.0 and this specification.

Pixel width for image border. The value 0 is particularly useful for *img* elements embedded in *a* elements, because typical HTML browsers apply a default border when rendering image-based hypertext links. This specification requires only that values of 0 and 1 be respected. Values greater than 1 may be mapped by the reading device to any smaller non-zero value.

3.30.8 hspace *(deprecated)*

Deprecated in HTML 4.0 and this specification, in favor of CSS margin mechanisms.

Reserves equal amount of whitespace to the left and right of an image.

3.30.9 vspace *(deprecated)*

Deprecated in HTML 4.0 and this specification, in favor of CSS margin mechanisms.

Reserves equal amount of whitespace above and below an image.

3.30.10 usemap

Points to a client-side map; the value of usemap refers to the map element's *name* or *id* attribute (see map 3.34).

3.30.11 ~~ismap~~

Omitted from this specification. In HTML 4.0, value is a reference to a server-side map for defining hotspots.

3.31 <kbd> </kbd>

The indication of text to be entered by a user, often used in software manuals.

3.32 <li>

3.32.1 ~~type~~

Deprecated in HTML 4.0 and omitted from this specification.

3.32.2 ~~value~~

Deprecated in HTML 4.0 and omitted from this specification.

3.33 <link />

In HTML the *link* element allows for specification of various relationships with other documents. Reading systems must recognize external stylesheet references specified via the *link* element. See section 1.4.4.

3.33.1   href

Provides the URI of a linked object.

3.33.2   rel, rev

Indicates the relationship to the linked document. Reading systems must support *rel="stylesheet"* and support *rel="alternate stylesheet"*. See the HTML 4.0 specification.

3.33.3   media

May be used to indicate the intended destination for style information, but may be ignored by reading systems.

3.33.4   type

This required attribute is used to specify the stylesheet language (e.g. "*text/x-oeb1-css*" or "text/css").

3.33.5   ~~charset, hreflang~~

Omitted from this specification.

3.34 <map> </map>

Contains *area* elements defining client-side hotspots for an image.

3.34.1   id

A required unique identifier for the map. See section 3.6, the *a* element, for restrictions on the characters permitted in names.

3.34.2   name *(deprecated)*

A unique identifier for the map. Deprecated in HTML 4.0 and in this specification, in favor of the *id* attribute; the *name* attribute is now optional. See section 3.6, the *a* element, for restrictions on the characters permitted in names.

3.35 <meta />

See the HTML 4.0 specification for further information.

For information regarding publication description (Dublin Core) metadata, see Chapter 2 of this specification.

3.35.1   name

3.35.2   content

3.35.3 scheme

3.35.4 http-equiv

Omitted from this specification.

3.35.5 id, style, class, title

HTML 4.0 and this specification exclude these general attributes from this element.

3.36 <object></object>

The *object* element is the preferred method for generic object inclusion. When adding objects whose *data* media type is not drawn from the core OEB media type list or which references an object implementation using the *classid* attribute, the *object* element must specify fallback information for the object, such as another *object*, an *img*, or descriptive text.

3.36.1 archive

See the HTML 4.0 specification for details.

3.36.2 classid

The URI of an implementation for the object. Conformant reading systems are not required to render objects that use external implementations, although they may do so.

3.36.3 codebase

The base path to be used for relative URIs in the *classid* attribute.

3.36.4 codetype

Specifies the MIME media type of the implementation referenced in *classid*.

Note: the type must match the MIME media type specified in the publication's manifest.

3.36.5 data

The URI of the data for this object.

3.36.6 type

The MIME media type of the information referenced by the *data* URI.

Note: the type must match the MIME media type specified in the publication's manifest.

3.36.7 align, height, width, border, hspace, vspace *(deprecated)*

Deprecated. See section 3.30, *img*, for additional discussion.

3.36.8 usemap

Refers to a client-side *map* element by its *name* attribute's value, in order to define hotspots.

3.36.9 ~~declare, standby, tabindex, name~~

Omitted from this specification.

3.37 <ol>

3.37.1 type *(deprecated)*
      1, a, A, i, I   Specifies a form of letters or numerals to use for the list items.

Deprecated in HTML 4.0 and this specification, in favor of CSS *list-style-type* .

3.37.2 ~~start~~

Deprecated in HTML 4.0 and omitted from this specification.

3.37.3 ~~compact~~

Omitted from this specification.

3.38 <p> </p>

3.38.1 align *(deprecated)*

See section 3.25, *h1*, for additional discussion.

3.39 <param />

The *param* 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*.

3.39.1 name

The name of a parameter used by the parent *object*.

3.39.2 value

The value of the parameter.

3.39.3 valuetype

The type of data stored in *value*:
      data   The default. The data stored in *value* is passed to the object as a string.
      ref   The data is a URI, and is passed to the object as specified, without being resolved.

object   An identifier that refers to an *object* declaration in the same document. The identifier must be the value of the <u>id</u> attribute set for the declared *object* s element.

3.39.4   type

Used when the *valuetype* is "ref". Specifies the MIME media type of the data a reading system should expect at the given URI.

~~3.39.5   style, class, title~~

HTML 4.0 and this specification exclude these general attributes from this element.

3.40 <pre> </pre>

Preformatted text to be rendered in a monospace font, such as program listings.

~~3.40.1   width~~

Deprecated in HTML 4.0 and omitted from this specification.

3.41 <q> </q>

Short inline quotation (see also section 3.11, *blockquote*).

3.41.1   cite

Provides a URI for the source of the quoted material. Reading systems are not required to make use of this attribute.

3.42 <s> </s> (deprecated)

Deprecated in HTML 4.0 and this specification, in favor of the CSS *text-decoration:line-through* property.

3.43 <samp> </samp>

Identifies the contents as program output, scripts, etc.

3.44 <script> </script>

Reading systems must not 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 1 *display* property to achieve this effect instead.

~~3.44.1   charset, type, src, defer, event, for~~

Omitted from this specification.

3.44.2 id, style, class, title

HTML 4.0 and this specification exclude these general attributes from this element.

3.45 <small> </small>

3.46

Particularly useful for associating CSS style information with inline content.

3.47 <strike> </strike> (deprecated)

Deprecated in HTML 4.0 and this specification, in favor of the CSS *text-decoration:line-through* property.

3.48

Strong emphasis, typically formatted like *b*.

3.49 <style> </style>

See section 1.4.4.

3.49.1 type

The value must be:
> Text/x-oeb1-css

OEB documents' embedded style information must always use only the CSS subset defined in this specification.

3.49.2 media

Omitted from this specification.

3.49.3 id, style, class

HTML 4.0 and this specification exclude these general attributes from this element.

3.50

3.51

3.52 <table> </table>

Reading systems may exercise great flexibility in laying out tables which would otherwise be too large to fit on the display.

3.52.1 align *(deprecated)*

The possible values are:

left	CSS equivalent is *float:left*
right	CSS equivalent is *float:right*
center	CSS equivalent is to enclose the table in a *div* element and set *text-align:center*

3.52.2 width

3.52.3 border

According to HTML 4.0, any number of pixels may be specified. This specification only requires that values of 0 and 1 be respected. Values greater than 1 may be mapped by the reading system to any smaller non-zero value.

3.52.4 cellspacing, cellpadding

3.52.5 bgcolor *(deprecated)*

See section 3.12.1, the *bgcolor* attribute of *body*, for additional discussion.

3.52.6 summary

A text summary of the table's purpose and/or structure, such as for speech output. This can be crucial in making documents accessible to print-disabled users.

3.52.7 ~~frame, rules~~

Omitted from this specification.

3.53 <td> </td> or <th> </th>

3.53.1 nowrap *(deprecated)*

Deprecated in HTML 4.0 and this specification. The non-deprecated alternative is to replace each run of whitespace with a single non-breaking space. Reading systems may choose to override this setting in cases where the default rendering of a particular *table* is bigger than the display surface.

3.53.2 rowspan, colspan

3.53.3 align

The possible values are:
    left
    center
    right
    justify

3.53.4 valign

The possible values are:
    top
    middle
    bottom

3.53.5 width *(deprecated),* height *(deprecated)*

Deprecated in HTML 4.0 and this specification. In pixels. Instead, use the CSS *width* and *height* to set value in pixels.

3.53.6 bgcolor *(deprecated)*

See section 3.12.1,the *bgcolor* attribute of *body,* for additional discussion.

3.53.7 abbr

An abbreviated form of the content, particularly to enhance accessibility for the print-disabled user, but also useful for other purposes such as rendering on small displays.

3.53.8 ~~axis, headers, scope~~

Omitted from the specification.

3.54 *<title> </title>*

See chapter 2 for information on the relationship between the HTML *title* and the Dublin Core metadata *Title*.

3.54.1 ~~id, style, class, title~~

HTML 4.0 and this specification exclude these general attributes from this element.

3.55 *<tr> </tr>*

3.55.1 ~~align~~

Omitted from this specification.

3.55.2   valign

The possible values are:
> top
> middle
> bottom

3.55.3   bgcolor *(deprecated)*

See section 3.12.1, the *bgcolor* attribute of *body*, for additional discussion.

3.56 *<tt> </tt>*

Monospace display inline.

3.57 *<u> </u>* (*deprecated*)

Deprecated in HTML 4.0 and this specification, in favor of CSS *text-decoration:underline*.

3.58 * *

3.58.1   ~~type~~

Deprecated in HTML 4.0 and omitted from this specification. Reading systems need support only one form of bullet, preferably *disc*.

3.58.2   ~~compact~~

Deprecated in HTML 4.0 and omitted from this specification.

3.59 *<var> </var>*

Identifies a mention of a variable or program argument.

4   CSS Constructs

This discussion refers to the CSS 1 and CSS 2 specifications, which can be found at
http://www.w3.org/TR/REC-CSS1/ and http://www.w3.org/TR/REC-CSS2/ respectively.

This specification requires support for only three kinds of CSS selectors. Specifically, those of the form

```
.classname   {}
 TAGNAME     {}
TAGNAME.classname {}
```

This specification does not require support for id-based selectors, or selectors that qualify element types based on the element types of their containing elements.

Multiple selectors may be combined with commas, as in

```
h1, h2, h3, h4, h5, h6 {text-indent: 0em}
```

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	m-width of current font
pt	Points
in	Inches
cm	Centimeters
mm	Millimeters
pc	Picas

Where percentage units are supported, informative text below explains their interpretation, which is as defined in the CSS specifications.

This specification does not list CSS properties omitted from this specification. It does identify, using the previous convention of ~~strike through~~, CSS 1 property *values* that are omitted from this specification. It underlines extensions beyond CSS 2. And it uses "*(meaning changed)*" to denote changes to the functionality of a property or keyword already defined in CSS.

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 not, however, required to support every distinction; the main example is that they are not required to render borders or to float anything but the specific elements *img* and *table*.

4.1   background-color: and color:

Some reading devices do not support colors, or even gray-scale. Use of colors other than white or black may not produce useful renderings on all reading devices.

Current browsers support a host of keyword color names. HTML 4.0 defines 16 named colors, as well as numeric values. OEB stylesheets 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	purple	
white	red	
aqua	silver	
blue	teal	
fuschia	yellow	
gray	#rrggbb	six-digit hexadecimal
green	#rgb	three-digit hexadecimal
lime	rgb(r, g, b)	integers in the range 0-255
maroon	rgb(r%, g%, b%)	floats in the range of
navy		0.0% to 100.0%
olive		

As with other names in this specification and in XML, these names are case-sensitive (unlike in HTML, which allows mixed case).

In addition, reading systems must recognize the keyword value:
        transparent

4.2   border:

This specification requires control of some aspects of borders, if only to provide non-deprecated alternatives to the *<table border="n">* and ** attributes. Of the twelve border properties defined by CSS 1, this specification includes only the *border* property. It can be used to set a single combination of style, width, and color values for all four borders of an element. This specification also does not require rendering support for color borders.

Reading systems must recognize all of the following style keywords, but may choose to map any style other than *none* to *solid*, consistent with the rules for the "CSS 1 core." Reading systems need not render borders except for *img, object,* and *table*.
        none   default, assumed if no style is specified
        solid
        dotted   may be mapped to *solid* by reading system
        dashed   may be mapped to *solid* by reading system
        double   may be mapped to *solid* by reading system
        groove   may be mapped to *solid* by reading system
        ridge   may be mapped to *solid* by reading system
        inset   may be mapped to *solid* by reading system
        outset   may be mapped to *solid* by reading system

Reading systems must recognize all of the following width values:
        thin
        medium   default, assumed if no style is specified
        thick
        [absolute width]

A reading system need not render three different widths for the three keyword values. For example, it may map non-zero absolute widths to 1 pixel.

Note: Reading systems must recognize the same set of color values for *border* as for *background-color:* and *color:* (see section 4.1 above), but need not actually distinguish them all in rendering.

4.3   clear:

Provide support equivalent to the deprecated *br clear* attribute. The possible values are:
        none (default)
        left
        right
        both

4.4   display: (values changed)

CSS 1 defines various values for this property, with CSS 2 adding more. Reading systems must respect the CSS 1 values *block, inline,* and *none*.

This specification also defines two new values: *oeb-page-head* and *oeb-page-foot*. Any content meant for presentation as a page-header or page-footer should be marked up and assigned these values via the *display* property.

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.

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 or footer as soon as nothing preceding the containing *div* is displayed:

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

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.

The possible values are:

block	Indicates the element is to be formatted as a block, with line-breaks at both ends.
inline	Indicates the element is to be formatted without forcing line-breaks at the ends.
~~list item~~	
oeb-page-head	Addition in support of page headers.
oeb-page-foot	Addition in support of page footers.
none	

Note: CSS enables a subtle mechanism for controlling whether or not a page-header should be displayed inline when rendered by existing browsers. The most straightforward construct:

```
<div style="display: oeb-page-head">
```

will render in an existing browser, for the browser will not recognize, and therefore ignore, the *display: oeb-page-head* setting. On the other hand,

```
<div style="display: none; display: oeb-page-head">
```

will not render in an existing browser. The browser will respond to *display: none*, ignore the *display: oeb-page-head*, and suppress formatting and rendering of the *div*.

4.5   float:

Although CSS 1 applies this to all elements, enabling any element to be extracted from inline flow, with normal text wrapping around. Reading systems need not actually float elements other than *img, object,* and *table*. The possible values are:

left
right

4.6 font-family:

Accepts a comma-separated list of family names, in order of preference. CSS defines five generic font family names, three of which reading systems must support :

serif
sans-serif
monospace

And two of which reading systems need not support:

~~cursive~~
~~fantasy~~

4.7 font-size:

Reading systems must accept the seven named font-sizes, as well as the forms for stepping one size (among the seven) larger or smaller than surrounding text. As discussed under *<font size="">*, there is no requirement that any given reading device actually render using seven distinct sizes.

Reading devices may provide means to change the mapping between the 7 named font-sizes and physical font sizes. Font-sizes specified with absolute units are presumed not to change size during normal operation (although they may do so in modes focused on universal access for the vision impaired). Content authors should avoid specifying absolute physical font sizes; doing so deprives the user of one of the significant benefits of an electronic book – user control over relative font size.

The possible values are:

xx-small
x-small
small
medium
large
x-large
xx-large
smaller
larger
absolute size

Percentage values are relative to the parent element's font size.

4.8 font-style:

Reading systems require support for only two out of three standard values:

normal
italic
~~oblique~~

Reading systems are free to treat italic and oblique synonymously; if a system sees either, it may apply either form (many font families provide only one of the two).

4.9   font-weight:

This specification requires support for only two values:

> normal
> bold

and omits the others (reading systems may accept these values, and are free to render them more simply, such as using only two degrees of boldness):

> ~~bolder~~
> ~~lighter~~
> ~~100, 200, 300, 400, 500, 600, 700, 800, 900~~

4.10 height:

Applies to block-level and replaced elements.

4.11 line-height:

This property sets the distance between the baselines of two adjacent lines. It accepts the values "normal", a number, or a length. Negative values are not allowed.

When "normal" is specified, CSS recommends that it function like a numeric value in the range of 1.0 to 1.2.

When a numerical value is specified, the line height is the font size of the current element multiplied by that value. This differs from a percentage value in the way it inherits: when a numerical value is specified, child elements will inherit the factor itself, not the resultant value (as is the case with percentage and other units).

Note also the following details from Section 4.4 of the CSS 1 specification:

> If a line of text contains sections with different 'line-height' values (because there are inline elements on the line), then each of those sections has its own half-leading above and below. The height of the line-box is from the top of the highest section to the bottom of the lowest one. Note that the top and bottom do not necessarily correspond to the tallest element, since elements can be positioned vertically with the 'vertical-align' property. To form a paragraph, each line-box is stacked immediately below the previous line.

> Note that any padding, border or margin above and below non-replaced inline elements does not influence the height of the line. In other words: if the 'line-height' is too small for the chosen padding or border, it will overlap with text on other lines.

> Replaced elements (e.g. images) on the line can make the line-box bigger, if the top of the replaced element (i.e., including all of its padding, border and margin) is above the tallest text section, or if the bottom is below the lowest.

4.12 list-style-type:

This property delivers the functionality of ordered and unordered lists, and their *type* attributes. CSS 2 introduces additional values beyond those listed below, none of which are currently supported by this specification.

The possible values are:

      decimal
      lower-roman
      upper-roman
      lower-alpha
      upper-alpha
      none

~~disc~~	omitted for simplicity
~~circle~~	omitted for simplicity
~~square~~	omitted for simplicity

Applies only to the predefined HTML 4.0 list types at this time, because the *display:list-item* value is not presently supported.

4.13 *margin-left:* and *margin-right:*

The value must be a length or a percentage. Percentage is relative to the width of the closest block-level ancestor. Reading systems are not required to support the value *~~auto~~*, which sets the value to that of the opposite margin.

4.14 *margin-top: and margin-bottom:*

The value must be a length or a percentage. Percentage is relative to the width of the closest block-level ancestor. Reading systems are not required to support the value *~~auto~~*, which sets the value that of the opposite margin.

Note that adjacent top and bottom margins coalesce. If a paragraph with *margin-bottom:* m *px* is followed by another with *margin-top:* n *px*, then the resulting whitespace is the greater of *m* and *n*, as opposed to their sum.

4.15 *text-align:*

Including this property eliminates the need for many uses of the *align* attribute. The possible values are:

      left
      right
      center
      justify

Applies to block-level elements.

4.16 *text-indent:*

The value must be a length or a percentage. Percentage is relative to the parent element's width.

Applies to block-level elements.

4.17 vertical-align: (meaning and applicability changed)

Although CSS applies this to any inline elements reading systems need only support it for images to achieve the functionality of **, **, and **.

The possible values are:
>   top
>   middle
>   bottom
>   ~~baseline~~
>   ~~sub~~
>   ~~super~~
>   ~~text-top~~
>   ~~text-bottom~~
>   ~~[percentage]~~

Applies to *img* and *object.*

4.18 width:

The value must be a length or a percentage. Percentage is relative to the width of the parent element. Reading systems are not required to support the value ~~*auto*~~.

Applies to block-level and replaced elements.

4.19 page-break-before:

Reading systems are free to treat *left* and *right* as *always*. The possible values are:
>   auto
>   always
>   left   Force one or two page breaks so that the following content begins on a left-hand page.
>   right   Force one or two page breaks so that the following content begins on a right-hand page.
>   ~~avoid~~
>   ~~inherit~~

Applies to block-level elements.

Note that this specification does not require support for the *page-break-after:* property.

4.20 page-break-inside:

Used to implement "keep-together" functionality. With *page-break-inside:avoid*, the entire element would be moved to a new page rather than broken across a page boundary. Explicit page breaks, as specified by *page-break-before:always*, take precedence over *page-break-inside:avoid*.

The possible values are:
>   auto
>   avoid
>   ~~inherit~~

Applies to block-level elements.

Note that the *page-break-inside* property is not supported in all current HTML browsers.

4.21 *text-decoration:*

This specification only requires support for the values:

> none   Particularly useful for suppressing special rendering of link anchors.
> underline
> ~~line-through~~

Reading systems need not support for:

> ~~overline~~
> ~~blink~~
> ~~inherit~~

4.22 <u>*oeb-column-number:*</u>

This specification allows use of this special property pending development of a CSS solution. It is suggested that this feature be accessed solely through the use of external stylesheets (not inline) so that a future standard method for accessing this functionality can be easily substituted. The possible values are:

> auto   Reading device decides on optimal number of columns in which to render the text, considering available width, font sizes, or any other metrics it considers relevant for readability.
> 1   Reading device should render in one column, regardless of the amount of horizontal space available to display the text.

Reading systems are free to support integer values other than 1, or may map them to 1.

Reading systems are not required to support column balancing.

Applies to block-level elements.

Appendix A: Element type table

The following table lists all the relevant HTML element type names alphabetically, along with their status in HTML 4.0 and OEB 1.0. Information about HTML 4.0 is based on the HTML 4.0 element table (http://www.w3.org/TR/REC-html40/index/elements.html). The status is marked as follows:

Empty: "E" when end-tag is forbidden because of declared content "Empty".

4.0:   Blank for the Strict DTD, "F" for the Frameset DTD; "L" for the Loose DTD, "NOT" if not in HTML 4.0 at all (3.2 only); "D" when deprecated (all "L" cases except *iframe*).

OEB: "+" when included, "-" when excluded, "D" when deprecated.

Element Type	Empty	4.0	OEB	Description
a			+	anchor
abbr			−	abbreviated form
acronym			−	acronym
address			−	information on author
applet		L, D	−	Java applet
area	E		+	client-side image map area
b			+	bold text style
base	E		+	document base URI
basefont	E	L, D	−	base font size
bdo			−	I18N BiDi over-ride
big			+	large text style
blockquote			+	long quotation
body			+	document body
br	E		+	forced line break
button			−	push button
caption			+	table caption
center		L, D	D	justification control
cite			+	citation
code			+	computer code fragment
col	E		−	table column
colgroup			−	table column group
dd			+	definition description
del			−	deleted text
dfn			+	instance definition
dir		L, D	−	directory list
div			+	generic language/style container
dl			+	definition list
dt			+	definition term
em			+	emphasis
fieldset			−	form control group
font		L, D	D	local change to font
form			−	interactive form

Element Type	Empty	4.0	OEB	Description
frame	E	F	−	subwindow
frameset		F	−	window subdivision
h1			+	heading
h2			+	heading
h3			+	heading
h4			+	heading
h5			+	heading
h6			+	heading
head			+	document head
hr	E		+	horizontal rule
html			+	document root element
i			+	italic text style
iframe		L	−	inline subwindow
img	E		+	Embedded image
input	E		−	form control
ins			−	inserted text
isindex	E	L, D	−	single line prompt
kbd			+	text to be entered by the user
label			−	form field label text
legend			−	fieldset legend
li			+	list item
link	E		+	a media-independent link
(listing)		NOT	−	(program listing, 3.2 only)
map			+	client-side image map
menu		L, D	−	menu list
meta	E		+	generic meta-information
noframes		F	−	alternate content container for non frame-based rendering
noscript			−	alternate content container for non script-based rendering
object			+	generic embedded object
ol			+	ordered list
optgroup			−	option group
option			−	selectable choice
p			+	paragraph
param	E		+	named property value
(plaintext)		NOT	−	(plaintext, 3.2 only)
pre			+	preformatted text
q			+	short inline quotation
s		L, D	D	strike-through text style
samp			+	program output, scripts, etc.
script			+	script statements
select			−	option selector
small			+	small text style
span			+	generic language/style container
strike		L, D	D	strike-through text
strong			+	strong emphasis
style			+	style info

Element Type	Empty	4.0	OEB	Description
sub			+	subscript
sup			+	superscript
table			+	
tbody			−	table body
td			+	table data cell
textarea			−	multi-line text field
tfoot			−	table footer
th			+	table header cell
thead			−	table header
title			+	document title
tr			+	table row
tt			+	teletype or monospaced text
u		L, D	D	underlined text style
ul			+	unordered list
var			+	instance of a variable or program argument
(xmp)		NOT	−	(example, 3.2 only)

Appendix B: The OEB Package DTD

```
<!--
    Document Type Definition for the Open eBook package version
    1.0.

    Version:  1.0
    Revision: 19990916-x

    Authors:   Steve DeRose <steven_derose@brown.edu>
               Gunter Hille <hille@abc.de>
               Ben Trafford <bent@exemplary.net>
               Garret Wilson <garret@globalmentor.com>

    Usage:
        <?xml version="1.0"?>
        <!DOCTYPE package
          PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.0 Package//EN"
          "http://openebook.org/dtds/oeb-1.0/oebpkg1.dtd">
        <package unique-identifier="foo">
          metadata
          manifest
          spine
          tours
          guide
        </package>

    References:
      Transitional XHTML 1.0 DTD at
      http://www.w3.org/TR/xhtml1/DTD/transitional.dtd
-->

<!-- ******** Character Mnemonic Entities ******** -->

<!-- OEB supports all XHTML mnemonics, but uses only one entity
file. -->
<!ENTITY % OEBEntities PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.0
  Entities//EN" "oeb1.ent">
%OEBEntities;

<!-- ******** Attribute Types ******** -->

<!-- LanguageCode: An RFC1766 language code. -->
<!ENTITY % LanguageCode "NMTOKEN">

<!-- URI: An RFC2396 Uniform Resource Identifier. -->
<!ENTITY % URI "CDATA">

<!-- ******** Common Attributes ******** -->

<!-- InternationalAttributes: Attributes for internationalization.
  xml:lang:     XML language code.
-->
<!ENTITY % InternationalAttributes
  "xml:lang %LanguageCode; #IMPLIED"
>
```

```
<!-- CoreAttributes: Most common attributes used by many elements.
   id:       ID unique to the entire document.
-->
<!ENTITY % CoreAttributes "id ID #IMPLIED">

<!-- CommonAttributes: Common attributes used by many elements.
   CoreAttributes:   Most common attributes.
   InternationalAttributes: Internationalization attributes.
-->
<!ENTITY % CommonAttributes
   "%CoreAttributes;
   %InternationalAttributes;"
>

<!-- DCNamespaceAttribute: Attribute that declare the Dublin Core
   namespace. Used mostly on each <dc:XXX> element to accomodate
   XMLparsers which require this unecessarily. -->
<!ENTITY % DCNamespaceAttribute
   "xmlns:dc   %URI;    #FIXED  'http://purl.org/dc/elements/1.0/'"
>

<!-- ******** OEB Package Elements ******** -->

<!-- A package must have metadata, a manifest, and a spine,
   and optionally may include a tours and/or a guide section. -->
<!ELEMENT package          (metadata, manifest, spine, tours?,
   guide?)>
<!ATTLIST package
   %CommonAttributes;
   unique-identifier    IDREF    #REQUIRED
>

<!-- The metadata section must have dc-metadata but may or may not
   include x-metadata. -->
<!ELEMENT  metadata       (dc-metadata, x-metadata?)>

<!-- These elements are optional in <dc-metadata>. -->
<!ENTITY % DCMetadataOptionalElements
   "dc:Contributor  | dc:Creator | dc:Subject | dc:Description
   | dc:Publisher |  dc:Date | dc:Type | dc:Format | dc:Source
   | dc:Language | dc:Relation | dc:Coverage | dc:Rights"
>

<!-- These are the optional and required elements of
   <dc-metadata>. -->
<!ENTITY % DCMetadataGeneralElements
   "%DCMetadataOptionalElements; | dc:Title |dc:Identifier"
>

<!-- A dc-metadata section must have a dc:Title and a
   dc:Identifier, and optionally other dc:XXX elements, all in any
   order. -->
<!ELEMENT dc-metadata ((%DCMetadataOptionalElements;)*,
   ((dc:Title, (%DCMetadataOptionalElements; | dc:Title)*,
     (dc:Identifier, (%DCMetadataGeneralElements;)*)) |
   (dc:Identifier, (%DCMetadataOptionalElements; | dc:Identifier)*,
     (dc:Title, (%DCMetadataGeneralElements;)*))))
>
```

```
<!ATTLIST dc-metadata
  %CommonAttributes;
  %DCNamespaceAttribute;
  xmlns:oebpackage     %URI; #FIXED
  "http://openebook.org/namespaces/oeb-package/1.0/"
>

<!-- A dc:Contributor element may optionally have a role and/or a
  file-asattribute. -->
<!ELEMENT dc:Contributor (#PCDATA)>
<!ATTLIST dc:Contributor
  %CommonAttributes;
  %DCNamespaceAttribute;
  role      NMTOKEN   #IMPLIED
  file-as   CDATA            #IMPLIED
>

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

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

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

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

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

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

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

```
  %DCNamespaceAttribute;
>

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

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

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

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

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

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

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

<!-- The <x-metadata> element must have at least one <meta>
  element. -->
<!ELEMENT x-metadata (meta+)>
<!ATTLIST x-metadata %CommonAttributes;>

<!ELEMENT meta EMPTY>
<!ATTLIST meta
  %CommonAttributes;
  name       NMTOKEN    #REQUIRED
  content    CDATA      #REQUIRED
  scheme     CDATA      #IMPLIED
>
```

```
<!-- The manifest must contain one or more items. -->
<!ELEMENT manifest (item+)>
<!ATTLIST manifest %CommonAttributes;>

<!-- The CommonAttributes entity is not used here because in this
case
  the "id" attribute is required. -->
<!ELEMENT item EMPTY>
<!ATTLIST item
  %InternationalAttributes;
  id            ID        #REQUIRED
  href          CDATA     #REQUIRED
  media-type    CDATA     #REQUIRED
  fallback      IDREF     #IMPLIED
>

<!-- The spine must contain one or more itemrefs. -->
<!ELEMENT spine           (itemref+)>
<!ATTLIST spine           %CommonAttributes;>

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

<!-- The tours element must have one or more tour elements. -->
<!ELEMENT tours           (tour+)>
<!ATTLIST tours           %CommonAttributes;>

<!-- Each tour of the set must contain at least one site. -->
<!ELEMENT tour (site+)>
<!ATTLIST tour
  %CommonAttributes;
  title   CDATA           #REQUIRED
>

<!-- Each site in a tour must have a title and an href. -->
<!ELEMENT site EMPTY>
<!ATTLIST site
  %CommonAttributes;
  title   CDATA   #REQUIRED
  href    CDATA   #REQUIRED
>

<!-- The guide element must have one or more reference
  elements. -->
<!ELEMENT guide           (reference+)>
<!ATTLIST guide           %CommonAttributes;>

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

Appendix C: The OEB Document DTD

```
<!--
    Document Type Definition for the Open eBook document version
    1.0.

    Version:  1.0
    Revision: 19990916-x

    Authors:  Gunter Hille <hille@abc.de>
              Ben Trafford <bent@exemplary.net>
              Garret Wilson <garret@globalmentor.com>

    Usage:
        <?xml version="1.0"?>
        <!DOCTYPE html PUBLIC
          "+//ISBN 0-9673008-1-9//DTD OEB 1.0 Document//EN"
          "http://openebook.org/dtds/oeb-1.0/oebdoc10.dtd">
        <html>
        ...
        </html>

    References:
      This DTD has been derived from XHTML 1.0 and HTML 4.0.
      It is a pure subset of neither.
      Transitional XHTML 1.0 DTD at
      http://www.w3.org/TR/xhtml1/DTD/transitional.dtd
-->

<!-- ******** Character Mnemonic Entities ******** -->

<!-- OEB supports all XHTML mnemonics, but uses only one entity
  file. -->
<!ENTITY % OEBEntities PUBLIC "+//ISBN 0-9673008-1-9//DTD OEB 1.0
  Entities//EN" "oeb1.ent">
%OEBEntities;

<!-- ******** Attribute Types ******** -->

<!-- Color: A color specification. -->
<!ENTITY % Color "CDATA">

<!-- Coords: Comma-separated coordinates for image maps. -->
<!ENTITY % Coords "CDATA">

<!-- LanguageCode: An RFC1766 language code. -->
<!ENTITY % LanguageCode "NMTOKEN">

<!-- Length: Number of pixels or percentage in one dimension. -->
<!ENTITY % Length "CDATA">

<!-- LinkTypes: List of types of document link types, used by
  "rel" and "rev". -->
<!ENTITY % LinkTypes "CDATA">

<!-- MediaType: An RFC2045 media type. -->
<!ENTITY % MediaType "CDATA">
```

```
<!-- MediaDest: Intended media destination. -->
<!ENTITY % MediaDest "CDATA">

<!-- A string of one or more digits. -->
<!ENTITY % Number "CDATA">

<!-- ObjectAlign: Non-text multidirectional alignment options. -->
<!ENTITY % ObjectAlign "(top|middle|bottom|left|right)">

<!-- ObjectHAlign: Horizontal non-text alignment options. -->
<!ENTITY % ObjectHAlign "(left|center|right)">

<!-- Shape: Shapes available for image maps. -->
<!ENTITY % Shape "(rect|circle|poly|default)">

<!-- StyleData: Style data (e.g. CSS) -->
<!ENTITY % StyleData "CDATA">

<!-- Text: Character data for such attributes as "title" and
  "alt". -->
<!ENTITY % Text "CDATA">

<!-- TextHAlign: Horizontal text alignment options. -->
<!ENTITY % TextHAlign "(left|center|right|justify)">

<!-- TextVAlign: Vertical text alignment options. -->
<!ENTITY % TextVAlign "(top|middle|bottom)">

<!-- URI: An RFC2396 Uniform Resource Identifier. -->
<!ENTITY % URI "CDATA">

<!-- A list of URIs separated by spaces. -->
<!ENTITY % URIList "CDATA">

<!-- ******** Common Attributes ******** -->

<!-- InternationalAttributes: Attributes for internationalization.
  xml:lang:     XML language code.
-->
<!ENTITY % InternationalAttributes
  "xml:lang %LanguageCode; #IMPLIED"
>

<!-- CoreAttributes: Most common attributes used by many elements.
  id:       ID unique to the entire document.
  class:    List of classes.
  style:    Style data.
  title:    Title or additional information.
-->
<!ENTITY % CoreAttributes
  "id      ID              #IMPLIED
  class   CDATA           #IMPLIED
  style   %StyleData;     #IMPLIED
  title   %Text;          #IMPLIED"
>

<!-- CommonAttributes: Common attributes used by many elements.
  CoreAttributes: Most common attributes.
```

```
    InternationalAttributes: Internationalization attributes.
-->
<!ENTITY % CommonAttributes
  "%CoreAttributes;
  %InternationalAttributes;"
>


<!-- ******** Common Elements ******** -->

<!-- HeadingElements: <h1>..<h6> -->
<!ENTITY % HeadingElements "h1|h2|h3|h4|h5|h6">

<!-- ListElements: Elements for lists. -->
<!ENTITY % ListElements "ul|ol|dl">

<!-- PhraseElements: Inline elements that contain a phrase of
  text. Note that while in current HTML implementations many
  PhraseElements are rendered identically to FontStyleElements
  counterparts (such as <em> and <i>, the former do not connotate
  rendering styles.
-->
<!ENTITY % PhraseElements
  "em | strong | dfn | code | q | sub | sup | samp | kbd | var |
  cite"
>


<!-- FontStyleElements: Inline font style elements.
  Note that many FontStyleElements have been deprecated in favor
  of their PhraseElements counterparts.
-->
<!ENTITY % FontStyleElements
  "tt | i | b | big | small | u | s | strike |font"
>


<!-- BlockElements: Elements at the block level. -->
<!ENTITY % BlockElements
  "%HeadingElements; | %ListElements; | p | pre | hr | blockquote
  | center | div | table"
>

<!-- InlineElements: Elements that are inline. -->
<!ENTITY % InlineElements
  "%PhraseElements; | %FontStyleElements; | a | br | span | img |
  object | map"
>

<!-- BlockOrInlineElements: Elements that can be either block or
  inline. -->
<!ENTITY % BlockOrInlineElements "script">

<!-- %FlowElements: Both block and inline elements, including
  those that can be both. -->
<!ENTITY % FlowElements "%BlockElements; | %InlineElements; |
  %BlockOrInlineElements;"
>

<!-- ******** OEB Document Elements ******** -->

<!ELEMENT br EMPTY>
```

```
<!ATTLIST br
  %CoreAttributes;
  clear (left|all|right|none) "none"
>

<!ELEMENT span (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST span %CommonAttributes;>

<!ELEMENT b (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST b %CommonAttributes;>

<!ELEMENT big (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST big %CommonAttributes;>

<!ELEMENT i (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST i %CommonAttributes;>

<!ELEMENT small (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST small %CommonAttributes;>

<!ELEMENT sub (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST sub %CommonAttributes;>

<!ELEMENT sup (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST sup %CommonAttributes;>

<!ELEMENT tt (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST tt %CommonAttributes;>

<!ELEMENT font (#PCDATA | %InlineElements; |
%BlockOrInlineElements;)*>
<!ATTLIST font
  %CommonAttributes;
  size    CDATA       #IMPLIED
  color   %Color;     #IMPLIED
  face    CDATA       #IMPLIED
>

<!ELEMENT s (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST s %CommonAttributes;>

<!ELEMENT strike (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST strike %CommonAttributes;>

<!ELEMENT u (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST u %CommonAttributes;>

<!ELEMENT cite (#PCDATA | %InlineElements; |
```

```
  %BlockOrInlineElements;)*>
<!ATTLIST cite %CommonAttributes;>

<!ELEMENT code (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST code %CommonAttributes;>

<!ELEMENT dfn (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST dfn %CommonAttributes;>

<!ELEMENT em (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST em %CommonAttributes;>

<!ELEMENT kbd (#PCDATA | %InlineElements; |
%BlockOrInlineElements;)*>
<!ATTLIST kbd %CommonAttributes;>

<!ELEMENT q (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST q
  %CommonAttributes;
  cite %URI; #IMPLIED
>

<!ELEMENT samp (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST samp %CommonAttributes;>

<!ELEMENT strong (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST strong %CommonAttributes;>

<!ELEMENT var (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST var %CommonAttributes;>

<!ELEMENT div (#PCDATA | %FlowElements;)*>
<!ATTLIST div
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT p (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST p
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT hr EMPTY >
<!ATTLIST hr
  %CommonAttributes;
  align    %ObjectHAlign;    #IMPLIED
  size     CDATA             #IMPLIED
  width    %Length;          #IMPLIED
>
```

```
<!ELEMENT center (#PCDATA | %FlowElements;)*>
<!ATTLIST center %CommonAttributes;>

<!ELEMENT blockquote (#PCDATA | %FlowElements;)*>
<!ATTLIST blockquote
  %CommonAttributes;
  cite %URI; #IMPLIED
>

<!ELEMENT pre
  (#PCDATA | %PhraseElements; | a | br | span | map | tt | i | b |
  u | s)*
>
<!ATTLIST pre
  %CommonAttributes;
  xml:space CDATA #FIXED "preserve"
>

<!-- Heading Elements -->

<!ELEMENT h1 (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST h1
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT h2 (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST h2
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT h3 (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST h3
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT h4 (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST h4
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT h5 (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST h5
  %CommonAttributes;
  align %TextHAlign; #IMPLIED
>

<!ELEMENT h6 (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST h6
  %CommonAttributes;
```

```
    align %TextHAlign; #IMPLIED
>

<!ELEMENT script (#PCDATA)>
<!ATTLIST script
  xml:space CDATA #FIXED "preserve"
>

<!ELEMENT style (#PCDATA)>
<!ATTLIST style
  %InternationalAttributes;
  type        %MediaType;   #FIXED  "text/x-oeb1-css"
  title       %Text;        #IMPLIED
  xml:space   CDATA         #FIXED  "preserve"
>

<!ELEMENT img  EMPTY >
<!ATTLIST img
  %CommonAttributes;
  align     %ObjectAlign;   #IMPLIED
  border    CDATA           #IMPLIED
  hspace    CDATA           #IMPLIED
  vspace    CDATA           #IMPLIED
  src       %URI;           #REQUIRED
  alt       %Text;          #REQUIRED
  longdesc  %URI;           #IMPLIED
  height    %Length;        #IMPLIED
  width     %Length;        #IMPLIED
  usemap    %URI;           #IMPLIED
>

<!ELEMENT a
  (#PCDATA | %FontStyleElements; | %PhraseElements; |
  %BlockOrInlineElements; | br | span | object | img | map)*
>
<!ATTLIST a
  %CommonAttributes;
  xml:link        CDATA          #FIXED   "simple"
  xlink:show      CDATA          #FIXED   "replace"
  xlink:actuate   CDATA          #FIXED   "user"
  name            NMTOKEN        #IMPLIED
  href            %URI;          #IMPLIED
  rel             %LinkTypes;    #IMPLIED
  rev             %LinkTypes;    #IMPLIED
>

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

<!ELEMENT link  EMPTY>
<!ATTLIST link
  %CommonAttributes;
  href    %URI;         #IMPLIED
  type    %MediaType;   #REQUIRED
  rel     %LinkTypes;   #IMPLIED
  rev     %LinkTypes;   #IMPLIED
  media   %MediaDest;   #IMPLIED
```

```
>

<!-- The CommonAttributes entity is not used here because in this
  case the "id" attribute is required. -->
<!ELEMENT map ((%BlockElements; | %BlockOrInlineElements;)+ |
  area+)>
<!ATTLIST map
  %InternationalAttributes;
  id       ID              #REQUIRED
  class    CDATA           #IMPLIED
  style    %StyleData;     #IMPLIED
  title    %Text;          #IMPLIED
  name     NMTOKEN         #IMPLIED
>

<!ELEMENT area  EMPTY>
<!ATTLIST area
  %CommonAttributes;
  href     %URI;      #IMPLIED
  shape    %Shape;    "rect"
  coords   %Coords;   #IMPLIED
  nohref   (nohref)   #IMPLIED
  alt      %Text;     #REQUIRED
>

<!ELEMENT object
  (#PCDATA | %BlockElements; | %InlineElements; |
  %BlockOrInlineElements; | param)*
>
<!ATTLIST object
  %CommonAttributes;
  classid   %URI;            #IMPLIED
  codebase  %URI;            #IMPLIED
  data      %URI;            #IMPLIED
  type      %MediaType;      #IMPLIED
  codetype  %MediaType;      #IMPLIED
  archive   %URIList;        #IMPLIED
  height    %Length;         #IMPLIED
  width     %Length;         #IMPLIED
  usemap    %URI;            #IMPLIED
  align     %ObjectAlign;    #IMPLIED
  border    CDATA            #IMPLIED
  hspace    CDATA            #IMPLIED
  vspace    CDATA            #IMPLIED
>

<!ELEMENT param EMPTY >
<!ATTLIST param
  id          ID                #IMPLIED
  name        CDATA             #REQUIRED
  value       CDATA             #IMPLIED
  valuetype   (data|ref|object) "data"
  type        %MediaType;       #IMPLIED
>

<!ELEMENT dl  (dt|dd)+ >
<!ATTLIST dl %CommonAttributes;>

<!ELEMENT dt (#PCDATA | %InlineElements; |
```

```
      %BlockOrInlineElements;)*>
<!ATTLIST dt %CommonAttributes;>

<!ELEMENT dd (#PCDATA | %FlowElements;)*>
<!ATTLIST dd %CommonAttributes;>

<!ELEMENT ol (li)+ >
<!ATTLIST ol
  %CommonAttributes;
  type (1|a|A|i|I) #IMPLIED
>

<!ELEMENT ul (li)+>
<!ATTLIST ul %CommonAttributes;>

<!ELEMENT li (#PCDATA | %FlowElements;)*>
<!ATTLIST li %CommonAttributes;>

<!ELEMENT table  (caption?, tr+) >
<!ATTLIST table
  %CommonAttributes;
  summary       %Text;          #IMPLIED
  width         %Length;        #IMPLIED
  border        CDATA           #IMPLIED
  cellspacing   %Length;        #IMPLIED
  cellpadding   %Length;        #IMPLIED
  align         %ObjectHAlign;  #IMPLIED
  bgcolor       %Color;         #IMPLIED
>

<!ELEMENT caption (#PCDATA | %InlineElements; |
  %BlockOrInlineElements;)*>
<!ATTLIST caption %CommonAttributes;>

<!ELEMENT tr  ( th | td )+ >
<!ATTLIST tr
  %CommonAttributes;
  valign    %TextVAlign;    #IMPLIED
  bgcolor   %Color;         #IMPLIED
>

<!ELEMENT th (#PCDATA | %FlowElements;)*>
<!ATTLIST th
  %CommonAttributes;
  abbr      %Text;          #IMPLIED
  rowspan   %Number;        "1"
  colspan   %Number;        "1"
  align     %TextHAlign;    #IMPLIED
  valign    %TextVAlign;    #IMPLIED
  nowrap    (nowrap)        #IMPLIED
  bgcolor   %Color;         #IMPLIED
  width     %Length;        #IMPLIED
  height    %Length;        #IMPLIED
>

<!ELEMENT td (#PCDATA | %FlowElements;)*>
<!ATTLIST td
  %CommonAttributes;
  abbr      %Text;          #IMPLIED
```

```
     rowspan    %Number;          "1"
     colspan    %Number;          "1"
     align      %TextHAlign;      #IMPLIED
     valign     %TextVAlign;      #IMPLIED
     nowrap     (nowrap)          #IMPLIED
     bgcolor    %Color;           #IMPLIED
     width      %Length;          #IMPLIED
     height     %Length;          #IMPLIED
>

<!ELEMENT title  (#PCDATA)>
<!ATTLIST title %InternationalAttributes;>

<!ELEMENT meta  EMPTY >
<!ATTLIST meta
  %InternationalAttributes;
  name       NMTOKEN    #IMPLIED
  content    CDATA      #REQUIRED
  scheme     CDATA      #IMPLIED
>

<!-- HeadElements: Elements that can appear many places within
  <head>.-->
<!ENTITY % HeadElements "script | style | meta | link | object">

<!-- <head> can have the common head elements (always optional),
  with one <title> and an optional <base> interspersed.
-->
<!ELEMENT head ((%HeadElements;)*,
 ((title, (%HeadElements;)*, (base, (%HeadElements;)*)?)
 | (base, (%HeadElements;)*, (title, (%HeadElements;)*))))
>
<!ATTLIST head %InternationalAttributes;>

<!ELEMENT body (#PCDATA | %FlowElements;)*>
<!ATTLIST body
  %CommonAttributes;
  bgcolor    %Color;   #IMPLIED
  text       %Color;   #IMPLIED
>

<!ELEMENT html (head?, body)>
<!ATTLIST html
  %InternationalAttributes;
  xmlns          %URI;   #FIXED
  "http://openebook.org/namespaces/oeb-document/1.0/"
  xmlns:xlink    %URI;   #FIXED  "http://www.w3.org/XML/XLink/"
>
```

Appendix D: Character Entities

```
<!--
    Character entities for the Open eBook publication structure
    version 1.0.

    Version:  1.0
    Revision: 19990916-x

    Authors:  Gunter Hille <hille@abc.de>
              Ben Trafford <bent@exemplary.net>
              Garret Wilson <garret@globalmentor.com>

    Usage:
        <!ENTITY % OEBEntities PUBLIC "+//ISBN 0-9673008-1-9//DTD
        OEB 1.0 Entities//EN" "oeb1.ent">
        %OEBEntities;

    References:
      This DTD fragment is merely a combination of the external
        entities included with XHTML 1.0.

      Transitional XHTML 1.0 DTD at
      http://www.w3.org/TR/xhtml1/DTD/transitional.dtd
      http://www.w3.org/TR/xhtml1/DTD/HTMLlat1x.ent
      http://www.w3.org/TR/xhtml1/DTD/HTMLspecialx.ent
      http://www.w3.org/TR/xhtml1/DTD/HTMLsymbolx.ent

        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.
-->

<!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 ordf   "&#170;" ><!-- feminine ordinal indicator,
                                    U+00AA 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 ordm   "&#186;" ><!-- masculine ordinal indicator,
                                    U+00BA 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 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 times  "&#215;" ><!-- multiplication sign,
                                            U+00D7 ISOnum -->
<!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 divide "&#247;" ><!-- division sign,
                                U+00F7 ISOnum -->
<!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 -->

<!-- C0 Controls and Basic Latin -->
<!ENTITY apos    "&#39;">

<!-- Latin Extended-A -->
<!ENTITY OElig   "&#338;" ><!-- latin capital ligature OE, U+0152
                                ISOlat2 -->
<!ENTITY oelig   "&#339;" ><!-- latin small ligature oe, U+0153
                                ISOlat2 -->

<!-- ligature is a misnomer, this is a separate character in some
  languages -->
<!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 -->

<!-- Spacing Modifier Letters -->
<!ENTITY circ    "&#710;" ><!-- modifier letter circumflex accent,
                                U+02C6 ISOpub -->
<!ENTITY tilde   "&#732;" ><!-- small tilde, U+02DC ISOdia -->

<!-- General Punctuation -->
<!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 permil   "&#8240;" ><!-- per mille sign, U+2030 ISOtech --
>

<!-- lsaquo is proposed but not yet ISO standardized -->
<!ENTITY lsaquo   "&#8249;" ><!-- single left-pointing angle
                                      quotation mark, U+2039 ISO
                                      proposed -->
<!-- rsaquo is proposed but not yet ISO standardized -->
<!ENTITY rsaquo   "&#8250;" ><!-- single right-pointing angle
                                      quotation mark, U+203A ISO
                                      proposed -->
<!ENTITY euro     "&#8364;" ><!-- euro sign, U+20AC NEW -->

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

<!-- 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 -->
<!-- there is no Sigmaf, and no U+03A2 character either -->
<!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 -->
<!ENTITY bull     "&#8226;" ><!-- bullet = black small circle,
                                U+2022 ISOpub  -->
<!-- bullet is NOT the same as bullet operator, U+2219 -->
<!ENTITY hellip   "&#8230;" ><!-- horizontal ellipsis = three dot
                                 leader, U+2026 ISOpub  -->
<!ENTITY prime    "&#8242;" ><!-- prime = minutes = feet, U+2032
                                 ISOtech -->
<!ENTITY Prime    "&#8243;" ><!-- double prime = seconds = inches,
                                 U+2033 ISOtech -->
<!ENTITY oline    "&#8254;" ><!-- overline = spacing overscore,
                                 U+203E NEW -->
<!ENTITY frasl    "&#8260;" ><!-- fraction slash, U+2044 NEW -->


<!-- Letterlike Symbols -->
<!ENTITY weierp   "&#8472;" ><!-- script capital P = power set
                                 = Weierstrass p, U+2118 ISOamso -->
<!ENTITY image    "&#8465;" ><!-- blackletter capital I =
                                    imaginary part, U+2111
                                    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 hebrew letter alef,
     U+05D0 although the same glyph could be used to depict both
     characters -->


<!-- Arrows -->
<!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. So ? lArr can be used for 'is implied by' as
     ISOtech suggests -->
<!ENTITY uArr       "&#8657;" ><!-- upwards double arrow, U+21D1
                                    ISOamsa -->
<!ENTITY rArr       "&#8658;" ><!-- rightwards double arrow,
                                   U+21D2 ISOtech -->
<!-- Unicode does not say this is the 'implies' character but does
     not have another character with this function so ? rArr can
     be used for 'implies' as ISOtech suggests -->
<!ENTITY dArr       "&#8659;" ><!-- downwards double arrow, U+21D3
                                    ISOamsa -->
<!ENTITY hArr       "&#8660;" ><!-- left right double arrow,
                                   U+21D4 ISOamsa -->


<!-- Mathematical Operators -->
<!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 -->
<!-- should there be a more memorable name than 'ni'? -->
<!ENTITY prod       "&#8719;" ><!-- n-ary product = product sign,
                                   U+220F ISOamsb -->
<!-- prod is NOT the same character as U+03A0 'greek capital
     letter pi' though the same glyph might be used for both -->
<!ENTITY sum        "&#8721;" ><!-- n-ary sumation, U+2211
                                    ISOamsb -->
<!-- sum is NOT the same character as U+03A3 'greek capital letter
     sigma'though the same glyph might be used for 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 the tilde,
     U+007E, although the same glyph might 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 -->
<!-- note that nsup, 'not a superset of, U+2283' is not covered by
     the Symbol font encoding and is not included. Should it be,
     for symmetry? It is in ISOamsn  -->
<!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 -->

<!-- Miscellaneous Technical -->
<!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' -->

       <!-- Geometric Shapes -->
       <!ENTITY loz     "&#9674;" ><!-- lozenge, U+25CA ISOpub -->

       <!-- Miscellaneous Symbols -->
       <!ENTITY spades  "&#9824;" ><!-- black spade suit, U+2660
                                    ISOpub -->
       <!-- black here seems to mean filled as opposed to hollow -->
       <!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 -->
```