Abstract

XHTML-Print is a member of the family of XHTML languages defined by the Modularization of XHTML [XHTMLMOD] [p.61]. It is designed to be appropriate for printing from mobile devices to low-cost printers that might not have a full-page buffer and that generally print from top-to-bottom and left-to-right with the paper in a portrait orientation. XHTML-Print is also targeted at printing in environments where it is not feasible or desirable to install a printer-specific driver and where some variability in the formatting of the output is acceptable.
Status of This Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

All sections of this document are normative unless noted as informative.

This document is a Proposed Edited Recommendation of XHTML Print. If approved, it will supersede the previous version. The only substantive changes in this version are the addition of an implementation of the markup language using XML Schema.

Publication as a Proposed Edited Recommendation does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to cite this document as other than work in progress.

W3C Advisory Committee Members are invited to send formal review comments on this Proposed Edited Recommendation to the W3C Team until 4 June 2009. Members of the W3C Advisory Committee will find the appropriate review form for this document by consulting their list of current WBS questionnaires.

This document has been produced by the W3C XHTML2 Working Group as part of the W3C HTML Activity.

This document is governed by the 24 January 2002 CPP as amended by the W3C Patent Policy Transition Procedure. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Please report errors in this document to www-html-editor@w3.org (archive).

Table of Contents

- 1. Introduction [p.5]
  - 1.1. XHTML for Printing [p.5]
  - 1.2. Terminology [p.5]
  - 1.3. Design Rationale [p.5]
- 2. Conformance [p.9]
  - 2.1. Document Conformance [p.9]
  - 2.2. Client Conformance [p.9]
  - 2.3. Printer Conformance [p.9]
- 3. The XHTML-Print Document Type [p.11]
  - 3.1. Attributes and Attribute Collections [p.12]
  - 3.2. Structure Module [p.13]
- 3 -
1. Introduction

All sections of this document are normative unless noted as informative.

1.1. XHTML for Printing

This section is informative.

This document specifies a simple XHTML based data stream suitable for printing as well as display. It is based on XHTML Basic [XHTMLBASIC][p.63]. Its targeted usage is for printing in environments where it is not feasible or desirable to install a printer-specific driver and where some variability in the formatting of the output is acceptable. Throughout this document this data stream is called "XHTML-Print."

XHTML-Print is designed to be appropriate for low-cost printers that might not have a full-page buffer and that generally print from top-to-bottom and left-to-right with the paper in a portrait orientation. For other printers (i.e., those that print in another direction or orientation) a full-page buffer could be needed.

XHTML-Print is not appropriate when strict layout consistency and repeatability across printers are needed. The design objective of XHTML-Print is to provide a relatively simple, broadly supportable page description format where content preservation and reproduction are the goal, i.e. "Content is King." Traditional printer page description formats such as PostScript or PCL are more suitable when strict layout control is needed. XHTML-Print does not utilize bi-directional communications with the printer either for capabilities or status inquiries.

This document creates a set of conformance criteria for XHTML-Print. It provides a strong basis for consistent printing results without a detailed understanding of each individual printer’s characteristics.

The document type definition for XHTML-Print is implemented based on the XHTML modules defined in Modularization of XHTML [XHTMLMOD][p.61].

1.2. Terminology

The keywords "MUST", "SHALL", "MUST NOT", "SHALL NOT", "REQUIRED", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" when used in this document are to be interpreted as described in RFC 2119 [RFC2119][p.61]. However, for readability, these words might not appear in all uppercase letters in this specification.

1.3. Design Rationale

This section explains why certain HTML features are not part of XHTML-Print and any special circumstances concerning a module and printing.
1.3.1. Script and Events

Scripts, as programs that are executed in conjunction with a document, are not relevant to the printed page. However, documents can provide information as an alternative to a script. Therefore, the script module is part of XHTML-Print. Scripts MUST NOT be executed and their results MUST NOT be printed. If a noscript element is present, it contains alternate content that MUST be printed in place of the content of the script element.

Events are not applicable to static, printed versions of a document. Therefore, the Intrinsic Events module is not part of XHTML-Print.

1.3.2. Presentation

Many simple printers cannot print a wider variety of fonts than generic serif, sans serif and monospace. It is RECOMMENDED that style sheets be used to create a presentation that is appropriate for a particular category of printer. How printers are categorized, what those categories are, how a printer identifies itself as a member of a category, and how style sheets are selectively applied based on category, is outside the scope of this document.

The Presentation module ([XHTMLMOD][p.61]), section 5.4.1) is supported to provide simple control over basic font variants and rules.

1.3.3. Forms

Basic XHTML forms ([XHTMLMOD][p.61]), section 5.5.1) are supported. Content developers SHOULD keep in mind that users might not be able to input many characters from some devices (e.g. from a mobile phone). Furthermore, developers are cautioned that a printer prints a static version of a form, and the visual appearance of a form depends heavily on the implementation.

1.3.4. Tables

Basic XHTML tables ([XHTMLMOD][p.61]), section 5.6.1) are supported, but tables can be difficult to format on very low resource devices. Furthermore, content developers are cautioned that in the Basic Tables Module, nesting of tables is prohibited.

1.3.5. Frames

Frames are not supported. Frames depend on a screen interface and therefore are not applicable to printers.

1.3.6. Attributes

XHTML-Print is a member of the family of XHTML languages defined by Modularization of XHTML ([XHTMLMOD][p.61]). Therefore, the elements and attributes in the modules that make up XHTML-Print are all valid constructs of the language. However, not all the attributes are applicable to a rendering of an XHTML-Print document in printed media, especially those that
are integral to a dynamic display of the document in a browser and the submission of a form. Furthermore, special attention is given to simple printers and some attributes are deemed too complex for such a printer to render. These attributes are treated as discretionary in that a conforming printer is not \textit{REQUIRED} to support them, but if a printer wishes to provide that support, there are requirements stated for consistency in the implementation of extensions.

1.3.7 Character Model

The W3C architectural specification \textit{Character Model for the World Wide Web 1.0} [CHARMOD \[p.61\]] gives the \textit{RECOMMENDED} representation of characters in XHTML-Print. Authors of XHTML-Print producing applications should be aware that low cost printers might be limited in both processing power and memory and therefore, that normalized utf-8 encoded documents could print more quickly than documents in other forms and encodings.
2. Conformance

2.1. Document Conformance

A conforming XHTML-Print document is a document that requires only the facilities described as mandatory in this specification. Such a document SHALL meet all of the following criteria:

1. The document SHALL conform to the constraints expressed in the DTD found in Appendix B [p.35] or the XML Schema found in Appendix B [p.43] and conform to the constraints expressed in Design Rationale [p.5].
2. The root element of the document MUST be html.
3. The name of the default namespace on the root element SHALL be the XHTML namespace name, http://www.w3.org/1999/xhtml.
4. There SHALL be a DOCTYPE declaration in the document prior to the root element. If present, the public identifier included in the DOCTYPE declaration SHALL reference the DTD found in Appendix B [p.35] of this specification, using its Formal Public Identifier. The system identifier MAY be modified appropriately.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML-Print 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-print10.dtd">
```

5. The DTD subset MUST NOT be used to override any parameter entities in the DTD.

The MIME type used to refer to a conforming XHTML-Print document SHALL be "application/xhtml+xml" with an OPTIONAL "profile" parameter of 'http://www.w3.org/Markup/Profile/Print'. An OPTIONAL "charset" parameter MAY be provided with the MIME type. Invalid values MUST be ignored and the result be as if the value were "utf-8". Usage of the OPTIONAL "charset" parameter is as described in Section 3.2 of RFC3023 - XML Media Types [RFC3023 [p.62]]. Usage of the OPTIONAL "profile" parameter is as described in Section 8 of RFC3236 - The 'application/xhtml+xml' Media Type [RFC3236 [p.62]].

2.2. Client Conformance

1. Clients SHALL produce a well-formed XHTML-Print document as defined in XHTML 1.0 [XHTML1 [p.61]] and in Document Conformance [p.9].
2. Beyond number 1 above, clients are not REQUIRED to use more of the XHTML-Print elements or style sheet attributes than necessary to get the desired output.

2.3 Printer Conformance

2.3.1 Formatting/Rendering Rules

A printer user agent MUST conform to the XHTML Family User Agent Conformance section of the Modularization of XHTML specification ([XHTMLMOD [p.61]], section 3.5) with the following exceptions and additions:
2.3.2 XHTML Requirements

1. A conforming printer SHALL support all XHTML Modules listed in The XHTML-Print Document Type[p.11].

2. A conforming printer SHALL print a static version of a form using default and selected values as specified in the form.

3. A conforming printer SHALL identify this datastream by the exact string: "XHTML-Print" (without the quotation marks) in all service discovery records and protocols, device identification records and protocols, and in other cases where a list of supported datastreams is to be presented by the printer. Where such datastreams are identified by a MIME media type, the identifier "application/xhtml+xml" SHALL be used in combination with a "profile" parameter of "http://www.w3.org/Markup/Profile/Print"; e.g., application/xhtml+xml; profile="http://www.w3.org/Markup/Profile/Print"
3. The XHTML-Print Document Type

The XHTML-Print document type is defined as a set of XHTML modules. All XHTML modules are defined in the *Modularization of XHTML* specification ([XHTMLMOD][p.61]).

XHTML-Print consists of the following XHTML modules:

- **Structure Module**
  - body, head, html, title

- **Text Module**
  - abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var

- **Hypertext Module**
  - a

- **List Module**
  - dl, dt, dd, ol, ul, li

- **Text Extension Module - Presentation**
  - b, big, hr, i, small, sub, sup, tt

- **Basic Forms Module**
  - form, input, label, select, option, textarea

- **Basic Tables Module**
  - caption, table, td, th, tr

- **Image Module**
  - img

- **Object Module**
  - object, param

- **Metainformation Module**
  - meta

- **Scripting Module**
  - noscript, script

- **Style Sheet Module**
  - style

- **Style Attribute Module**
  - style attribute

- **Link Module**
  - link

- **Base Module**
  - base

(*) = *This module is a REQUIRED [XHTML Host Language] module.*
(/**) = *These modules are not a part of XHTML Basic but are REQUIRED for XHTML-Print.*

An XML 1.0 DTD is available in [Appendix B][p.35] An XML Schema implementation is available in [Appendix C][p.43]
3.1 Attributes and Attribute Collections

Some of the attributes defined in the Modularization of XHTML ([XHTMLMOD][p.61]) are not applicable to the printed page or are not relevant due to the exclusion of their module from XHTML-Print. Other attributes are not REQUIRED but if supported by a printer, support SHOULD be provided in the RECOMMENDED manner.

Each attribute in the following sections is annotated with one of the following keywords indicating support options for a conforming printer:

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUST</td>
<td>Support is mandatory; a conforming printer MUST implement this attribute. (However, the inability of a printer to implement part of this specification due to the limitations of a particular device does not imply non-conformance. E.g., the fact that a monochrome printer user agent cannot render colors does not preclude its conformance to this specification.)</td>
</tr>
<tr>
<td>SHOULD</td>
<td>Support for the attribute is RECOMMENDED, but not REQUIRED.</td>
</tr>
<tr>
<td>MAY</td>
<td>The attribute’s functionality is entirely OPTIONAL.</td>
</tr>
</tbody>
</table>
| N/A | The attribute does Not Apply to the printed page; a conforming printer MAY ignore this attribute for one of the following reasons, but MUST NOT treat it as an error:  
  - The attribute applies to a user interface which is not represented on a printed page. For example, the accesskey attribute is irrelevant.  
  - The attribute applies to form submission which is not performed by the printer, the method attribute of the form element for example.  
  - The attribute, such as title, describes data which is not represented on a printed page  
  - The attribute applies to objects other than JPEG images, such as Java applets.  
  - The attribute does not apply since links specified by the anchor element are not followed. |

The Modularization of XHTML ([XHTMLMOD][p.61]), section 5.1) contains a set of attribute collections for ease of presentation. This specification continues this practice with the same conditions, that is, that the collections below are informative and their contents normative.
**3.2 Structure Module**

### Collection Name

<table>
<thead>
<tr>
<th>Attributes in Collection</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core desc (NMTOKENS)[p.13])</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td>Core id (ID)[p.13]</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td>Core title (CDATA)[p.13]</td>
<td>[N/A] [p.12]</td>
</tr>
<tr>
<td>Style style (CDATA)[p.13]</td>
<td>MUST [p.12]</td>
</tr>
</tbody>
</table>

**Table Note:**

† See Modularization of XHTML (XHTMLMOD[p.61], section 4.3)

Note that the `title` attribute of the Core collection is not applicable to the printed page since there is no place to display such supplementary information.

A printer MAY support special processing based on the natural language of the document, such as the use of guillemets for quotation marks in French text. If a printer implements processing based on the natural language of the document, that processing SHALL be controlled by the `xml:lang` attribute.

A printer SHOULD support CSS style sheets, as noted in section 1.3.2 Presentation[p.6], within the limits of its capabilities.

---

**3.2 Structure Module**

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>Common[p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>head I18N [p.13]</td>
<td></td>
<td>See Collection</td>
</tr>
<tr>
<td>head</td>
<td>profile (URI)[p.14]</td>
<td>MAY [p.12]</td>
</tr>
<tr>
<td>html</td>
<td>version (CDATA)[p.14]</td>
<td>[N/A] [p.12]</td>
</tr>
<tr>
<td>title</td>
<td>I18N[p.13]</td>
<td>See Collection</td>
</tr>
</tbody>
</table>

**Table Note:**
†See Modularization of XHTML ( [XHTMLMOD]p.61] ), section 4.3

If a printer implements support for meta data then it **MUST** support the *profile* attribute of the *head* element.

The *version* attribute is not applicable for printing since it was deprecated in the *HTML 4.01 Specification* [HTML4]p.61] in favor of version information within the DTD.

A printer **MAY** ignore the content of the *title* element since it is not part of the document’s body.

### 3.3 Text Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th><strong>REQUIRED</strong> Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>abbr, acronym, address</td>
<td>Common [p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>blockquote</td>
<td>cite (URI) [p.14]</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>br</td>
<td>Core [p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p</td>
<td>Common [p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>pre</td>
<td>Common [p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>pre</td>
<td>xml:space=&quot;preserve&quot;</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td>q</td>
<td>Common [p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>q</td>
<td>cite (URI) [p.14]</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>samp, span, strong, var</td>
<td>Common [p.13]</td>
<td>See Collection</td>
</tr>
</tbody>
</table>

Table Note:

†See Modularization of XHTML ( [XHTMLMOD]p.61] ), section 4.3

### 3.4 Hypertext Module
### 3.5 List Module

<table>
<thead>
<tr>
<th>Element</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td>See Collection</td>
</tr>
<tr>
<td>a</td>
<td>accesskey</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>charset</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>href</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>hreflang</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>rel</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>rev</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>tabindex</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>a</td>
<td>type</td>
<td>N/A [p.12]</td>
</tr>
</tbody>
</table>

**Table Note:**

†See *Modularization of XHTML* ([XHTMLMOD] p.61], section 4.3)

### 3.6 Presentation Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>b, big, hr, i, small, sub, sup, tt</td>
<td>Common [p.13]</td>
<td>See Collection</td>
</tr>
</tbody>
</table>

### 3.7 Basic Forms Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>form</td>
<td></td>
<td>See Collection</td>
</tr>
<tr>
<td>form</td>
<td>action</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>form</td>
<td>method</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>form</td>
<td>enctype</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td>Elements</td>
<td>Attributes</td>
<td>REQUIRED Processing</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>input</td>
<td>accesskey</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>checked</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>maxlength</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>name</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>size</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>src</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>tabindex</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>input</td>
<td>type</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>label</td>
<td>accesskey</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>label</td>
<td>for</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>select</td>
<td>name</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>select</td>
<td>size</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>select</td>
<td>tabindex</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>option</td>
<td>selected</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>option</td>
<td>value</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>Elements</td>
<td>Attributes</td>
<td>REQUIRED Processing</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>textarea</td>
<td>Common[p.13],</td>
<td>See Collection</td>
</tr>
<tr>
<td>textarea</td>
<td>accesskey (Character[p.17]),</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>textarea</td>
<td>cols[p.17] (Number[p.17]),</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>textarea</td>
<td>name (CDATA[p.17]),</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>textarea</td>
<td>rows[p.17] (Number[p.17]),</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>textarea</td>
<td>tabindex (Number[p.17])</td>
<td>N/A[p.12]</td>
</tr>
</tbody>
</table>

Table Notes:

†See *Modularization of XHTML* ([XHTMLMOD p.61], section 4.3)

* The attribute MUST be present.

** The value is the default.

The `src` attribute of the `input` element is not supported since the `image` type is not part of basic forms.

The `hidden` type for the `input` element MUST be supported even though nothing is printed, so that a printer can correctly recognize and ignore the element.

### 3.8 Basic Tables Module
<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>caption</td>
<td>[Common][p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>table</td>
<td>[Common][p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>table</td>
<td>summary ([Text][p.18])</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>td, th</td>
<td>[Common][p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>td, th</td>
<td>abbr ([Text][p.18])</td>
<td>MAY[p.12]</td>
</tr>
<tr>
<td>td, th</td>
<td>align (&quot;left&quot;</td>
<td>&quot;center&quot;</td>
</tr>
<tr>
<td>td, th</td>
<td>axis ([CDATA][p.18])</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>td, th</td>
<td>colspan ([Number][p.18])</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>td, th</td>
<td>headers ([IDREFS][p.18])</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>td, th</td>
<td>rowspan ([Number][p.18])</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>td, th</td>
<td>scope (&quot;row&quot;</td>
<td>&quot;col&quot;),</td>
</tr>
<tr>
<td>td, th</td>
<td>valign (&quot;top&quot;</td>
<td>&quot;middle&quot;</td>
</tr>
<tr>
<td>tr</td>
<td>[Common][p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>tr</td>
<td>align (&quot;left&quot;</td>
<td>&quot;center&quot;</td>
</tr>
<tr>
<td>tr</td>
<td>valign (&quot;top&quot;</td>
<td>&quot;middle&quot;</td>
</tr>
</tbody>
</table>

Table Note:

†See Modularization of XHTML ([XHTMLMOD][p.61]), section 4.3

If a printer implements a feature to truncate the contents of a cell because of space constraints, it MUST support the abbr attribute and print the value of the abbr attribute (if present) instead of the cell’s content.

A printer MUST support the values left, right, and center for the align attribute of the td, th, and tr elements; other values are OPTIONAL. If the align attribute is missing or has an unsupported value, a printer MUST act as if the align attribute has the value left for the td and tr elements, and as if the align attribute has the value center for the th element.

A printer MUST support the values top, middle, and bottom for the valign attribute of the td, th, and tr elements; other values are OPTIONAL. If the valign attribute is missing or has unrecognized value, a printer SHOULD act as if the valign attribute has the value middle. Vertical alignment is undefined across page boundaries.
3.9 Image Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>img</td>
<td>Common[p.13]</td>
<td>See Collection</td>
</tr>
<tr>
<td>img</td>
<td>height [Length][p.19]</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>img</td>
<td>src[p.19] [URI][p.19]</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>img</td>
<td>width [Length][p.19]</td>
<td>MUST[p.12]</td>
</tr>
</tbody>
</table>

Table Notes:
†See Modularization of XHTML ([XHTMLMOD]p.61], section 4.3)

* The attribute MUST be present.

Printers MUST support the http [RFC2616][p.62] URI scheme [RFC3986][p.62]. Support for other schemes is OPTIONAL. Printers MUST support the JPEG image format as defined in Appendix A[p.33].

Conforming documents SHOULD specify the width and height of the image using the width and height attributes or equivalent styling instructions. (See 2.3.1 Formatting/Rendering Rules [p.9]).

3.10 Object Module
<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td></td>
<td>See Collections</td>
</tr>
<tr>
<td></td>
<td>archive</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>classid</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>codebase</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td></td>
<td>codetype</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>data</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td></td>
<td>declare</td>
<td>MAY [p.12]</td>
</tr>
<tr>
<td></td>
<td>height</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>standby</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>tabindex</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>MUST [p.12]</td>
</tr>
<tr>
<td>param</td>
<td>id</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>type</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>value</td>
<td>N/A [p.12]</td>
</tr>
<tr>
<td></td>
<td>valuetype</td>
<td>N/A [p.12]</td>
</tr>
</tbody>
</table>

Table Notes:

† See Modularization of XHTML ([XHTMLMOD]p.61], section 4.3)

* The attribute **MUST** be present.

** The value is the default.

Printers **MUST** support the http [RFC2616] [p.62] URI scheme [RFC3986] [p.62]. Support for other schemes is **OPTIONAL**.

A printer **MUST** support resources of type "image/jpeg." A printer **MAY** support other types of image formats and therefore other values of the **type** attribute. A printer **MUST** process the content of the **object** element when it does not recognize or support the object type referenced by the value of the **type** attribute.
Conforming documents SHOULD specify the width and height of the image using the width and height attributes or equivalent styling instructions. (See 2.3.1 Formatting/Rendering Rules [p.9]).

The param element’s purpose is to pass data to an application specified in the enclosing object element. The param element MAY be completely ignored.

### 3.11 Metainformation Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>meta</td>
<td>[I18N][p.13],</td>
<td>See Collection</td>
</tr>
<tr>
<td>meta</td>
<td>content [p.21] (CDATA[p.21],)</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>meta</td>
<td>http-equiv [NMTOKEN][p.21],</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>meta</td>
<td>name [NMTOKEN][p.21],</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>meta</td>
<td>scheme [CDATA][p.21]</td>
<td>N/A[p.12]</td>
</tr>
</tbody>
</table>

Table Notes:

†See Modularization of XHTML ( [XHTMLMOD][p.61] ), section 4.3

* The attribute MUST be present.

A printer MAY implement support for this element and provide implementation specific processing of the meta-information. However, guidelines and/or recommendations for processing a document’s meta-information are beyond the scope of this document.

### 3.12 Scripting Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>noscript</td>
<td>[Common][p.13],</td>
<td>See Collections</td>
</tr>
<tr>
<td>script</td>
<td>charset [Charset],</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>script</td>
<td>defer (&quot;defer&quot;),</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>script</td>
<td>src[URI],</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>script</td>
<td>type [ContentType],</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>script</td>
<td>scheme [CDATA</td>
<td>N/A[p.12]</td>
</tr>
</tbody>
</table>
Scripts, as programs that are executed in conjunction with a document, are not relevant to the printed page and MUST NOT be executed or printed. The noscript element contains alternate content that MUST be printed in place of the content of the script element when present.

### 3.13 Style Sheet Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>style</td>
<td>media (MediaDesc[p.22]),</td>
<td>SHOULD[p.12]</td>
</tr>
<tr>
<td>style</td>
<td>title (Text[p.22]),</td>
<td>N/A[p.12]</td>
</tr>
<tr>
<td>style</td>
<td>type[p.22],</td>
<td>MUST[p.12]</td>
</tr>
<tr>
<td>style</td>
<td>xml:space=&quot;preserve&quot;</td>
<td>SHOULD[p.12]</td>
</tr>
</tbody>
</table>

Table Notes:

†See Modularization of XHTML ([XHTMLMOD][p.61], section 4.3)

* The attribute MUST be present.

A printer MUST read and process the content of style elements where the media attribute has the value print or all. A printer MAY read and process the content of style elements where the media attribute has the value projection. A printer SHOULD ignore the content of style elements where the media attribute has any other value. The absence of the media attribute MUST be treated as if the media attribute had the value all.

A printer SHOULD read and process the content of style elements where the value of the type attribute is "text/css"; a printer MAY read and process the content of style elements where the value of the type attribute is other than "text/css"; all unsupported values for type MUST cause the content to be ignored. Style elements without a type attribute will be treated in an implementation-dependent manner.

### 3.14 Style Sheet Attribute Module

This module adds the style attribute to the Common[p.13] attribute collection (section 3.1).

### 3.15 Link Module
### 3.16 Base Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>link</td>
<td>Common[p.13],</td>
<td>See Collection</td>
</tr>
<tr>
<td>link</td>
<td>charset [Charset][p.23],</td>
<td>[MUST][p.12]</td>
</tr>
<tr>
<td>link</td>
<td>href [URI][p.23],</td>
<td>[MUST][p.12]</td>
</tr>
<tr>
<td>link</td>
<td>hreflang [LanguageCode][p.23],</td>
<td>[MAY][p.12]</td>
</tr>
<tr>
<td>link</td>
<td>media [MediaDesc][p.23],</td>
<td>[MUST][p.12]</td>
</tr>
<tr>
<td>link</td>
<td>rel (&quot;stylesheet&quot;),</td>
<td>[MUST][p.12]</td>
</tr>
<tr>
<td>link</td>
<td>rev [LinkTypes][p.23],</td>
<td>[N/A][p.12]</td>
</tr>
<tr>
<td>link</td>
<td>type</td>
<td>[MUST][p.12]</td>
</tr>
</tbody>
</table>

Table Note:

†See Modularization of XHTML ([XHTMLMOD][p.61]), section 4.3

Printers MUST support the http [RFC2616][p.62] URI scheme [RFC3986][p.62]. Support for other schemes is OPTIONAL.

If the printer implements processing based on the natural language of the document, then the hreflang attribute MUST be supported.

A printer SHOULD read and process the content of external style sheets where the media attribute has the value print or all. A printer MAY read and process the content of external style sheets where the media attribute has the value projection. A printer SHOULD ignore the content of external style sheets where the media attribute has any other value. The absence of the media attribute MUST be treated as if the media attribute had the value all.

A printer SHOULD support the value stylesheet for the rel attribute along with the value "text/css" for the type attribute; all other values are OPTIONAL.

#### 3.16 Base Module

<table>
<thead>
<tr>
<th>Elements</th>
<th>Attributes</th>
<th>REQUIRED Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>href[p.24] [URI][p.23]</td>
<td>[MUST][p.12]</td>
</tr>
</tbody>
</table>

Table Notes:

†See Modularization of XHTML ([XHTMLMOD][p.61]), section 4.3
* The attribute *MUST* be present.

Printers *MUST* support the http [[RFC2616][p.62]] URI scheme [[RFC3986][p.62]]. Support for other schemes is *OPTIONAL*.

### 3.17 Character Entities

XHTML-Print is in the family of XHTML document types, since it is created by combining XHTML modules. The character entities that are part of XHTML-Print are, therefore, defined in [[XHTML Character Entities][XHTMLMOD][p.61]], Section F.1).
4. How to Use XHTML-Print

XHTML-Print inherits all the structure, encoding and other basic infrastructure specified by XHTML 1.0. The following sections describe and clarify the application and usage restrictions of XHTML-Print.

4.1 Images

This document specifies only one mandatory image format: baseline JPEG as defined in JPEG File Interchange Format. See Appendix A for a description of JPEG decoder requirements. Printers are not REQUIRED to support:

- Embedded thumbnails
- Rotation
- Progressive rendering

within the JFIF (JPEG File Interchange Format) and EXIF (Exchangeable Image File Format) files.

4.1.1 Recommended Attributes on the img and object Elements

Because many printers create the page in a serial manner from top to bottom, it is important for the printer to know the size of images before retrieving the image data itself. This information is then used to create portions of the page layout.

Therefore, the document SHOULD include the height and width attributes within the img or the object element (or equivalent styling instructions). These attributes MAY be expressed as pixels or percentages within the img or the object element. Percentages are relative to the parent element and not the page width or printable area.

4.1.2 Image Data

[Informative]

In traditional Web-based applications of XHTML, image data is contained in a separate file on a Web server that the user agent retrieves.

However, there are circumstances where it is desirable to include the image data along with the rest of the print data. For example, some low cost, resource constrained clients might want to include images in their print output but cannot afford to include an HTTP server. Furthermore, circumstances could require that all the print data be encapsulated in a single file for transportability, avoiding firewall issues, etc. Therefore, conforming XHTML-Print printers MAY optionally support a format that contains both a document and its referenced image data as well as the REQUIRED traditional format that contains only the document.
The format recommended for including image data along with xhtml-print markup is defined by RFC3391 - The MIME Application/Vnd.pwg-multiplexed Content-type. [MIME/MPPX [p.62]].

Including image data as defined in RFC2397 - The data URL scheme [RFC2397 [p.62]] may be appropriate for printers capable of buffering large amounts of data, but will not achieve the intended results for most cost- and memory-constrained printer UA’s. Because this method normally encodes the binary image data using base64 encoding, a significant increase in the size of the data transmitted will be experienced. This should be avoided over low speed connections. Printers supporting included data can support base64 encoding using the img or object element.

<object height="20 mm" width="20 mm" type="image/jpeg"
   data="data:image/jpeg;base64,aGh67Fghsap0Hji7dfGSweTa . . .">
   Example Image </object>
or
<img height="20 mm" width="20 mm" alt="Example Image"
   src="data:image/jpeg;base64,aGh67Fghsap0Hji7dfGSweTa . . ." />

Mechanisms for determining whether or not a printer supports either of the above OPTIONAL document formats is outside the scope of this specification.

4.1.3 Side-by-Side Images

Low-cost printers today often have very little memory into which page data can be stored before being printed. As such, they may build and print the page in swaths on the fly from the top of the page to the bottom. To enable the use of XHTML-Print in these low cost printers, some restrictions on the order of images contained in the XHTML-Print data stream must be added.

1. If two or more images will be even partially side-by-side on the printed page (i.e., a line across the short axis of the page will intersect more than one image), they SHOULD be included by reference; for example <img src="http://example.com/example.jpg">. This allows the printer to get chunks of the image, as it needs it, as it prints down the page. Interleaved or included image data, as discussed in Section 4.1.2 [p.25], is discouraged.

2. An XHTML-Print conforming printer lacking sufficient buffer space to hold multiple side-by-side images MAY choose to reformat the layout of the page to preserve content. Printers SHALL attempt to preserve content when encountering side-by-side images that MAY be impossible to print as specified within the XHTML-Print. Discarding the second and subsequent of the side-by-side images SHOULD be avoided unless preservation of content is best achieved by doing so. Other than attempting to best preserve content, this specification does not mandate any specific behavior when encountering this situation. Clients providing images SHOULD order them from left-to-right top-to-bottom unless the print direction is known to be otherwise.
4.2 Style Sheets

Conforming XHTML-Print printers SHALL support both in-line and referenced style sheets within the style element or link element in the head element of a document. Conforming XHTML-Print printers SHALL also support the style attribute (i.e. in-line style) when used within other elements as defined by XHTML 1.1[XHTML1.1][p.61]. Normal cascading rules apply.

4.3 Forms Usage

This section is informative.

An HTML form is a dynamic entity when the document is displayed in a browser: data can be entered into text fields, buttons can be pushed, selections made, and options checked. None of this dynamic activity can be rendered on a printed page. However, a printed page can permanently record a particular state of the form. For example, users might wish to print forms that record products ordered or payments made.

The following discussion illustrates the activity involved when interacting with and printing forms. Please refer to Sequence Diagram 1[p.27]

Sequence Diagram 1. Forms Usage
Steps:

1. The User enters a URL into the Browser
2. The Browser fetches the form from the Server and displays it
3. The User enters data into the form
4. The User asks the Browser to print the form
5. The Browser composes a page with the form and the user data
6. The Browser sends the newly composed form to the printer
7. The User selects the Submit button on the form
8. The Browser sends the user data to the Server

Detailed discussion of Steps:

1. The user interacts with a browser on a mobile device to access a form presented by a server on the network (steps 1 and 2 of [Sequence Diagram 1][p. 27]). The following fragment of an XHTML-Print document shows what the server sends to the browser to present to the user. Please note, that the form is blank when first presented to the user.

   ```html
   <form action="http://example.com/prog/adduser" method="post">
     <label for="firstname">First name: </label>
     <input type="text" id="firstname" /><br />
     <label for="lastname">Last name: </label>
     <input type="text" id="lastname" /><br />
     <label for="email">email: </label>
     <input type="text" id="email" size="40" /><br />
     <input type="checkbox" name="member" value="IEEE" /> IEEE <br />
     <input type="checkbox" name="member" value="ACM" /> ACM <br />
     <input type="submit" value="Send" /> <input type="reset" />
   </form>
   ``

   Here is an example presentation of the above form as the user would see it:

   First name: 
   Last name: 
   email: 
   [ ] IEEE
   [ ] ACM

2. The user enters data (step 3 of [Sequence Diagram 1][p. 27]) into the text fields and checks the IEEE check box so that the form now looks like the following:

   First name: 
   Last name: 

3. The user then clicks on the browser's print button (step 4 of Sequence Diagram 1[p. 27]), to print the form as it currently appears.

4. The browser then creates a, possibly new, document (step 5 of Sequence Diagram 1[p. 27]) containing the original form and the users data. Note in the XHTML-Print document below, created by the browser, that the user’s data is included either by a value attribute or a checked attribute.

```html
<form action="http://example.com/prog/adduser" method="post">
  <label for="firstname">First name: </label>
  <input type="text" id="firstname" value="John"/>
  <br/>
  <label for="lastname">Last name: </label>
  <input type="text" id="lastname" value="Doe"/>
  <br/>
  <label for="email">email: </label>
  <input type="text" id="email" value="johnd@example.org"/>
  <br/>
  <input type="checkbox" name="member" checked="checked" value="IEEE" /> IEEE
  <br/>
  <input type="checkbox" name="member" value="ACM" /> ACM
  <br/>
  <input type="submit" value="Send" />
  <input type="reset" />
</form>
```

5. The browser sends (step 6 of Sequence Diagram 1[p. 27]) the document created in step 5 to the printer.

6. Sometime later the user clicks on the submit form button (step 7 of Sequence Diagram 1[p. 27]) and the browser submits the form (step 8 of Sequence Diagram 1[p. 27]) using the procedures given in the HTML 4.01 Specification ([HTML4][p. 61], Forms Submission).
5. Acknowledgements

This section is informative.

This specification was prepared by the W3C HTML Working Group. The participants at the time of publication were:

- Mark Birbeck
- Beth Epperson
- Melinda Grant
- Luca Mascaro
- Shinichi Matsui
- Shane McCarron
- Steven Pemberton
- Brad Petit
- Richard Schwerdtfeger

At publication of the second edition, the membership was:

- Roland Merrick, IBM (XHTML 2 Working Group Co-Chair)
- Steven Pemberton, CWI (XHTML 2 Working Group Co-Chair)
- Mark Birbeck, webBackplane (Invited Expert)
- Susan Borgrink, Progeny Systems
- Christina Bottomley, Society for Technical Communication (STC)
- Alessio Cartocci, International Webmasters Association / HTML Writers Guild (IWA-HWG)
- Alexander Graf, University of Innsbruck
- Markus Gylling, DAISY Consortium
- Tina Holmboe, Greytower Technologies (Invited Expert)
- John Kugelman, Progeny Systems
- Luca Mascaro, International Webmasters Association / HTML Writers Guild (IWA-HWG)
- Shane McCarron, Applied Testing and Technology, Inc. (Invited Expert)
- Michael Rawling, IVIS Group Limited
- Gregory Rosmaita, Invited Expert
- Sebastian Schnitzenbaumer, Dreamlab Technologies AG
- Richard Schwerdtfeger, IBM
- Elias Torres, IBM
- Masataka Yakura, Mitsue-Links Co., Ltd.
- Toshihiko Yamakami, ACCESS Co., Ltd.

This specification is based in large part on the specification of the same name, XHTML™-Print [XHTMLPRINT][p.63], from the Printer Working Group, a program of and through the IEEE Industry Standards and Technology Organization, Inc.; and which was in turn based in large part upon an earlier work with the same name by Fujisawa, Grant, Wright, and Zehler. The editors wish to express their gratitude to all who contributed to this and earlier versions.
A. JPEG Decoder Requirements

A.1 Introduction

A.1.1 Intent

This appendix describes REQUIRED behaviors for JPEG decoders in XHTML-Print devices. Many of the behaviors described in this document follow directly from language already present in the relevant JPEG standards, but are repeated here to emphasize their importance.

A.1.2 Objectives

The decoder behaviors described in this document are intended to minimize implementation complexity, while retaining maximum compatibility with existing JPEG files. In particular, these recommendations seek to ensure compatibility with both EXIF (Exchangeable Image File Format) and baseline JFIF (JPEG File Interchange Format); i.e., the subset of JFIF files that use only baseline JPEG processes. Support for JPEG streams using non-baseline processes, such as arithmetic coding or progressive coding, is not mandated for XHTML-Print compliance.

A.2 Printer Behaviors

This section describes behaviors of JPEG decoders for XHTML-Print conformant implementations.

A.2.1 JPEG Processes

A JPEG decoder for an XHTML-Print printer SHALL support all baseline JPEG processes as defined in [CCITT p.62], except for 2- and 4-component images. These processes include grayscale and 3-component images, 8-bit/component sample depth, Huffman entropy coding, 444, 422, 411, and 400 subsampling modes, and sequential (i.e. non-progressive) scan.

A.2.2 Handling of APPx Markers

Baseline decoders MAY ignore application-specific markers, such as the JFIF APP0 marker and the EXIF APP1/APP2 markers; rotation fields within these markers SHOULD be ignored. (Specifically, conforming printers SHOULD NOT decode the TIFF IFDs embedded in the EXIF APP1 and APP2 markers, as described in Section 2.6.4 of [JEIDA p.62].) This implies images will print in the orientation in which they are stored, unless style markup indicates otherwise. The image size SHOULD be rendered as specified in the JPEG SOF marker, if not overridden by style mark-up. A JPEG decoder for a conforming printer SHALL NOT fail as a consequence of encountering an unsupported APPx marker (i.e. all such markers SHALL be correctly parsed, even if they are ignored).
A.2.3 Color Management

This section describes a RECOMMENDED color management approach for XHTML-Print printers.

Grayscale Images

Sample values in a grayscale (single-component) JPEG image MAY be converted to the sRGB color space by setting

\[ R_{\text{out}} = G_{\text{out}} = B_{\text{out}} = \text{Gray}_{\text{in}} \]

or by other suitable algorithm specific to the XHTML-Print device.

Color Images

Sample values in 3-component JPEG images SHALL be interpreted as YCbCr samples, as would be obtained by applying the matrices described in ITU BT.601 \[\text{[BT601.5[p.62]]}\] to sRGB input data.
B. XHTML-Print DTD and Modules

This section contains the pieces of the XHTML-Print DTD that are unique to XHTML-Print. The remaining entities and modules are as specified in reference [XHTMLMOD][p.61].

The following SHOULD be used from Modularization of XHTML [XHTMLMOD][p.61]:

1. xhtml-attribs-1.mod
2. xhtml-base-1.mod
3. xhtml-basic-form-1.mod
4. xhtml-basic-table-1.mod
5. xhtml-blkphras-1.mod
6. xhtml-blkpres-1.mod
7. xhtml-blkstruct-1.mod
8. xhtml-charent-1.mod
9. xhtml-datatypes-1.mod
10. xhtml-framework-1.mod
11. xhtml-hypertext-1.mod
12. xhtml-image-1.mod
13. xhtml-inlphras-1.mod
14. xhtml-inlpres-1.mod
15. xhtml-inlstruct-1.mod
16. xhtml-inlstyle-1.mod
17. xhtml-lat1.ent
18. xhtml-link-1.mod
19. xhtml-list-1.mod
20. xhtml-meta-1.mod
21. xhtml-notations-1.mod
22. xhtml-object-1.mod
23. xhtml-param-1.mod
24. xhtml-pres-1.mod
25. xhtml-qname-1.mod
26. xhtml-script-1.mod
27. xhtml-special.ent
28. xhtml-struct-1.mod
29. xhtml-style-1.mod
30. xhtml-symbol.ent
31. xhtml-text-1.mod
B.1. XHTML-Print 1.0 DTD

Available for download at xhtml-print10.dtd.

<!-- ................................................................. -->
<!-- XHTML-Print 1.0 DTD ......................................... -->
<!-- file: xhtml-print10.dtd -->

<!-- XHTML-Print 1.0 DTD

This is XHTML-Print 1.0, a variant of XHTML Basic for printing.

(Massachusetts Institute of Technology, European Research
Consortium for Informatics and Mathematics, Keio University).
All Rights Reserved.

Permission to use, copy, modify and distribute the XHTML-Print DTD and
its accompanying documentation for any purpose and without fee is hereby
granted in perpetuity, provided that the above copyright notice and
this paragraph appear in all copies. The copyright holders make no
representation about the suitability of the DTD for any purpose.

It is provided "as is" without expressed or implied warranty.

Primary Author: Jun Fujisawa <fujisawa.jun@canon.co.jp>
Editors: Jim Bigelow <jim.bigelow@hp.com>,
Shane McCarron <shane@aptest.com>,
Masayasu Ishikawa <mimasa@w3.org>
Revision: $Id: xhtml-print10.dtd,v 1.5 2003/10/24 14:24:40 fujisawa Exp $

-->

<!-- This is the driver file for version 1.0 of the XHTML-Print DTD.

This DTD is identified by the PUBLIC and SYSTEM identifiers:

PUBLIC "-//W3C//DTD XHTML-Print 1.0//EN"
SYSTEM "http://www.w3.org/MarkUp/DTD/xhtml-print10.dtd"

-->

<!ENTITY % XHTML.version "-//W3C//DTD XHTML-Print 1.0//EN" >

<!-- Use this URI to identify the default namespace:

"http://www.w3.org/1999/xhtml"

-->

<!ENTITY % NS.prefixed "IGNORE" >
<!ENTITY % XHTML.prefix "" >

<!-- Reserved for use with the XLink namespace:

-->

<!ENTITY % XLINK.xmlns "" >
<!ENTITY % XLINK.xmlns.attrib "" >

<!-- reserved for future use with document profiles -->
<!ENTITY % XHTML.profile "" >
<!-- Bidirectional Text features
This feature-test entity is used to declare elements
and attributes used for bidirectional text support.
-->
<!ENTITY % XHTML.bidi "IGNORE" >

<!-- :---------------------------------------------------------- -->

<!ENTITY % xhtml-events.module "IGNORE" >
<!ENTITY % xhtml-bdo.module "%XHTML.bidi;" >

<!-- Style Attribute Module ................................. -->
<!ENTITY % xhtml-inlstyle.module "INCLUDE" >
<![%xhtml-inlstyle.module;{
<!ENTITY % xhtml-inlstyle.mod
PUBLIC "-//W3C//ENTITIES XHTML Inline Style 1.0//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-inlstyle-1.mod" >
%xhtml-inlstyle.mod;]>

<!-- Document Model Module .............................. -->
<!ENTITY % xhtml-model.mod
PUBLIC "-//W3C//ENTITIES XHTML-Print 1.0 Document Model 1.0//EN"
"xhtml-print10-model-1.mod" >

<!-- Modular Framework Module (required) .............. -->
<!ENTITY % xhtml-framework.mod
PUBLIC "-//W3C//ENTITIES XHTML Modular Framework 1.0//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-framework-1.mod" >
%xhtml-framework.mod;

<!-- Text Module (required) ............................. -->
<!ENTITY % xhtml-text.mod
PUBLIC "-//W3C//ELEMENTS XHTML Text 1.0//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-text-1.mod" >
%xhtml-text.mod;

<!-- Hypertext Module (required) ........................ -->
<!ENTITY % xhtml-hypertext.mod
PUBLIC "-//W3C//ELEMENTS XHTML Hypertext 1.0//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-hypertext-1.mod" >
%xhtml-hypertext.mod;

<!-- Lists Module (required) ............................. -->
<!ENTITY % xhtml-list.mod
PUBLIC "-//W3C//ELEMENTS XHTML Lists 1.0//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-list-1.mod" >
%xhtml-list.mod;

<!-- :---------------------------------------------------------- -->

<!-- Presentation Module ................................... -->
<!ENTITY % xhtml-pres.module "INCLUDE" >
<![%xhtml-pres.module;{
<!ENTITY % xhtml-pres.mod
PUBLIC "-//W3C//ELEMENTS XHTML Presentation 1.0//EN"
"http://www.w3.org/TR/xhtml-modularization/DTD/xhtml-pres-1.mod" >
%xhtml-pres.mod;]>

- 37 -
B.2. XHTML-Print 1.0 Document Model Module

Available for download at xhtml-print10-model-1.mod.
<!ENTITY % Inline.mix "%Inline.class; %Misc.class;"
>
<!ENTITY % InlNoAnchor.mix "%InlNoAnchor.class; %Misc.class;"
>
<!ENTITY % Inline.extra;"
>
<!ENTITY % FlowNoTable.mix

B.2. XHTML-Print 1.0 Document Model Module
"%Heading.class;
| %List.class;
| %BlkStruct.class;
%BlkPhras.class;
%BlkPres.class;
%Form.class;
%Block.extra;
| %Inline.class;
%Misc.class;"
>

<!ENTITY % Flow.mix
"%Heading.class;
| %List.class;
| %Block.class;
| %Inline.class;
%Misc.class;"
>

<!-- end of xhtml-print10-model-1.mod -->
C. XHTML-Print Schema and Modules

This section contains the pieces of the XHTML-Print XML Schema implementation that are unique to XHTML-Print. The remaining entities and modules are as specified in reference [XHTMLMOD][p.61].

The following SHOULD be used from Modularization of XHTML [XHTMLMOD][p.61]:

1. xhtml-attrs-1.xsd
2. xhtml-base-1.xsd
3. xhtml-basic-form-1.xsd
4. xhtml-basic-table-1.xsd
5. xhtml-blkphras-1.xsd
6. xhtml-blkpres-1.xsd
7. xhtml-blkstruct-1.xsd
8. xhtml-charent-1.xsd
9. xhtml-datatypes-1.xsd
10. xhtml-framework-1.xsd
11. xhtml-hypertext-1.xsd
12. xhtml-image-1.xsd
13. xhtml-inlphras-1.xsd
14. xhtml-inlpres-1.xsd
15. xhtml-inlstruct-1.xsd
16. xhtml-inlstyle-1.xsd
17. xhtml-lat1.ent
18. xhtml-link-1.xsd
19. xhtml-list-1.xsd
20. xhtml-meta-1.xsd
21. xhtml-notations-1.xsd
22. xhtml-object-1.xsd
23. xhtml-param-1.xsd
24. xhtml-pres-1.xsd
25. xhtml-script-1.xsd
26. xhtml-special.ent
27. xhtml-struct-1.xsd
28. xhtml-style-1.xsd
29. xhtml-symbol.ent
30. xhtml-text-1.xsd
C.1. XHTML-Print 1.0 XML Schema

Available for download at xhtml-print-1.xsd.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
 xmlns:xs=http://www.w3.org/2001/XMLSchema
 targetNamespace="http://www.w3.org/1999/xhtml"
 xmlns:xh11d="http://www.w3.org/1999/xhtml/datatypes/
 xmlns="http://www.w3.org/1999/xhtml"
 elementFormDefault="qualified" >
  <xs:annotation>
    <xs:documentation>
      This is the XML Schema driver for XHTML Print 1.0
      Please use this namespace for XHTML elements:
      "http://www.w3.org/1999/xhtml"
    </xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation source="xhtml-copyright-1.xsd"/>
  </xs:annotation>
  <xs:documentation>
    This is the Schema Driver file for XHTML Print 1.0
    Document Type
  </xs:documentation>
  This schema
  + imports external schemas (xml.xsd)
  + redefines (and include)s schema modules for XHTML1.1 Document Type.
  + includes Schema for Named content model for the
    XHTML Print 1.0 Document Type

  XHTML Print 1.0 Document Type includes the following Modules
  XHTML Core modules (Required for XHTML Family Conformance)
    + text
    + hypertext
    + lists
    + structure
  Other XHTML modules
    + Edit
    + Bdo
    + Presentational
    + Link
    + Meta
    + Base
    + Scripting
    + Style
    + Image
    + Applet
    + Object
    + Param (Applet/Object modules require Param Module)
    + Basic Tables
    + Basic Forms
</xs:documentation>
</xs:annotation>
```
C.2. XHTML-Print 1.0 XML Schema Content Model Module

Available for download at xhtml-print-model-1.xsd.
XHTML Document Model

This module describes the groupings of elements/attributes that make up common content models for XHTML elements.

XHTML has the following basic content models:
- xhtml.Inline.mix; character-level elements
- xhtml.Block.mix; block-like elements, e.g., paragraphs and lists
- xhtml.Flow.mix; any block or inline elements
- xhtml.Head_OPTS.mix; Head Elements
- xhtml.InlinePre.mix; Special class for pre content model
- xhtml.InlineNoAnchor.mix; Content model for Anchor

Any groups declared in this module may be used to create element content models, but the above are considered ‘global’ (insofar as that term applies here). XHTML has the following Attribute Groups:
- xhtml.Core.extra.attrib
- xhtml.I18n.extra.attrib
- xhtml.Common.extra

The above attribute Groups are considered Global.

</xs:documentation>
</xs:annotation>
<xs:attributeGroup
  name="xhtml.I18n.extra.attrib">
  <xs:annotation>
    <xs:documentation> Extended I18n attribute </xs:documentation>
  </xs:annotation>
</xs:attributeGroup>
<xs:attributeGroup
  name="xhtml.Common.extra">
  <xs:annotation>
    <xs:documentation> Extended Common Attributes </xs:documentation>
  </xs:annotation>
  <xs:attributeGroup
    ref="xhtml.style.attrib">
    <xs:annotation>
      <xs:documentation>
        "style" attribute from Inline Style Module
      </xs:documentation>
    </xs:annotation>
  </xs:attributeGroup>
</xs:attributeGroup>
<xs:attributeGroup
  name="xhtml.Core.extra.attrib">
  <xs:annotation>
    <xs:documentation> Extend Core Attributes </xs:documentation>
  </xs:annotation>
</xs:attributeGroup>
<xs:attributeGroup
  name="xhtml.Global.core.extra.attrib">
  <xs:annotation>
    <xs:documentation> Extended Global Core Attributes </xs:documentation>
  </xs:annotation>
</xs:attributeGroup>
<xs:attributeGroup
  name="xhtml.Global.I18n.extra.attrib">
  <xs:annotation>
    <xs:documentation> Extended Global I18n attributes </xs:documentation>
  </xs:annotation>
</xs:attributeGroup>
</xs:annotation>
</xs:attributeGroup>
<xs:attributeGroup
    name="xhtml.Global.Common.extra">
    <xs:annotation>
        <xs:documentation> Extended Global Common Attributes </xs:documentation>
    </xs:annotation>
</xs:attributeGroup>
<xs:group
    name="xhtml.Head.extra">
    <xs:sequence/>
</xs:group>
<xs:group
    name="xhtml.HeadOpts.mix">
    <xs:choice>
        <xs:element
            name="script"
            type="xhtml.script.type"/>
        <xs:element
            name="style"
            type="xhtml.style.type"/>
        <xs:element
            name="meta"
            type="xhtml.meta.type"/>
        <xs:element
            name="link"
            type="xhtml.link.type"/>
        <xs:element
            name="object"
            type="xhtml.object.type"/>
        <xs:group
            ref="xhtml.Head.extra"/>
    </xs:choice>
</xs:group>
<xs:group
    name="xhtml.head.content">
    <xs:sequence>
        <xs:group
            ref="xhtml.HeadOpts.mix"
            minOccurs="0"
            maxOccurs="unbounded"/>
        <xs:choice>
            <xs:sequence>
                <xs:element
                    name="title"
                    minOccurs="1"
                    maxOccurs="1"
                    type="xhtml.title.type"/>
                <xs:group
                    ref="xhtml.HeadOpts.mix"
                    minOccurs="0"
                    maxOccurs="unbounded"/>
                <xs:sequence
                    minOccurs="0">
                    <xs:element
                        name="base"
                        type="xhtml.base.type"/>
                </xs:sequence>
            </xs:sequence>
        </xs:choice>
    </xs:sequence>
</xs:group>
<xs:element name="base" type="xhtml.base.type" minOccurs="1" maxOccurs="1"/>
<xs:element name="title" minOccurs="1" maxOccurs="1" type="xhtml.title.type"/>
</xs:group>
</xs:sequence>
</xs:choice>
</xs:sequence>
</xs:group>
</!--
script and noscript are used to contain scripts
and alternative content
-->
<xs:group name="xhtml.Script.class">
<xs:choice>
<xs:element name="script" type="xhtml.script.type"/>
<xs:element name="noscript" type="xhtml.noscript.type"/>
</xs:choice>
</xs:group>
<xs:group name="xhtml.Misc.extra">
</xs:sequence/>
</xs:group>
</!--
These elements are neither block nor inline, and can
essentially be used anywhere in the document body.
-->
<xs:group name="xhtml.Misc.class">
<xs:choice>
<xs:group ref="xhtml.Script.class"/>
<xs:group ref="xhtml.Misc.extra"/>
</xs:group>
<!-- Inline Elements -->
<xs:group name="xhtml.InlStruct.class">
  <xs:choice>
    <xs:element name="br" type="xhtml.br.type"/>
    <xs:element name="span" type="xhtml.span.type"/>
  </xs:choice>
</xs:group>
<xs:group name="xhtml.InlPhras.class">
  <xs:choice>
    <xs:element name="em" type="xhtml.em.type"/>
    <xs:element name="strong" type="xhtml.strong.type"/>
    <xs:element name="dfn" type="xhtml.dfn.type"/>
    <xs:element name="code" type="xhtml.code.type"/>
    <xs:element name="samp" type="xhtml.samp.type"/>
    <xs:element name="kbd" type="xhtml.kbd.type"/>
    <xs:element name="var" type="xhtml.var.type"/>
    <xs:element name="cite" type="xhtml.cite.type"/>
    <xs:element name="abbr" type="xhtml.abbr.type"/>
    <xs:element name="acronym" type="xhtml.acronym.type"/>
    <xs:element name="q" type="xhtml.q.type"/>
  </xs:choice>
</xs:group>
<xs:group name="xhtml.InlPres.class">
  <xs:choice>
    <xs:element name="tt" type="xhtml.InlPres.type"/>
  </xs:choice>
</xs:group>
<xs:element name="i" type="xhtml.InlPres.type"/>
<xs:element name="b" type="xhtml.InlPres.type"/>
<xs:element name="big" type="xhtml.InlPres.type"/>
<xs:element name="small" type="xhtml.InlPres.type"/>
<xs:element name="sub" type="xhtml.InlPres.type"/>
<xs:element name="sup" type="xhtml.InlPres.type"/>
</xs:choice>
</xs:group>
<xs:group name="xhtml.I18n.class">
</xs:group>
<xs:group name="xhtml.Anchor.class">
<xs:sequence>
<xs:element name="a" type="xhtml.a.type"/>
</xs:sequence>
</xs:group>
<xs:group name="xhtml.InlSpecial.class">
<xs:choice>
<xs:element name="img" type="xhtml.img.type"/>
<xs:element name="object" type="xhtml.object.type"/>
</xs:choice>
</xs:group>
<xs:group name="xhtml.InlForm.class">
<xs:choice>
<xs:element name="input" type="xhtml.input.type"/>
<xs:element name="select" type="xhtml.select.type"/>
<xs:element name="textarea" type="xhtml.textarea.type"/>
<xs:element name="label" type="xhtml.label.type"/>
</xs:choice>
</xs:group>
Inline.class includes all inline elements, used as a component in mixes

-->
<xs:group
name="xhtml.Inline.class">
  <xs:choice>
    <xs:group
      ref="xhtml.InlStruct.class"/>
    <xs:group
      ref="xhtml.InlPhras.class"/>
    <xs:group
      ref="xhtml.InlPres.class"/>
    <xs:group
      ref="xhtml.Anchor.class"/>
    <xs:group
      ref="xhtml.InlSpecial.class"/>
    <xs:group
      ref="xhtml.InlForm.class"/>
    <xs:group
      ref="xhtml.Inline.extra"/>
  </xs:choice>
</xs:group>

!---
InlinePre.mix
Used as a component in pre model
-->
<xs:group
name="xhtml.InlinePre.mix">
  <xs:choice>
    <xs:group
      ref="xhtml.InlStruct.class"/>
    <xs:group
      ref="xhtml.InlPhras.class"/>
    <xs:element
      name="tt"
      type="xhtml.InlPres.type"/>
    <xs:element
      name="i"
      type="xhtml.InlPres.type"/>
    <xs:element
      name="b"
      type="xhtml.InlPres.type"/>
    <xs:group
      ref="xhtml.Anchor.class"/>
    <xs:group
      ref="xhtml.Misc.class"/>
    <xs:group
      ref="xhtml.Inline.extra"/>
  </xs:choice>
</xs:group>
InlNoAnchor.class includes all non-anchor inlines, used as a component in mixes
-->
<xs:group
  name="xhtml.InlNoAnchor.class">
  <xs:choice>
    <xs:group
      ref="xhtml.InlStruct.class"/>
    <xs:group
      ref="xhtml.InlPhras.class"/>
    <xs:group
      ref="xhtml.InlPres.class"/>
    <xs:group
      ref="xhtml.InlSpecial.class"/>
    <xs:group
      ref="xhtml.InlForm.class"/>
    <xs:group
      ref="xhtml.Inline.extra"/>
  </xs:choice>
</xs:group>
<!--
InlNoAnchor.mix includes all non-anchor inlines
-->
<xs:group
  name="xhtml.InlNoAnchor.mix">
  <xs:choice>
    <xs:group
      ref="xhtml.InlNoAnchor.class"/>
    <xs:group
      ref="xhtml.Misc.class"/>
  </xs:choice>
</xs:group>
<!--
Inline.mix includes all inline elements, including Misc.class
-->
<xs:group
  name="xhtml.Inline.mix">
  <xs:choice>
    <xs:group
      ref="xhtml.Inline.class"/>
    <xs:group
      ref="xhtml.Misc.class"/>
  </xs:choice>
</xs:group>
<!--
In the HTML 4 DTD, heading and list elements were included in the block group. The Heading.class and List.class groups must now be included explicitly on element declarations where desired.
-->
<xs:group
  name="xhtml.Heading.class">
  <xs:choice>
    <xs:element
      name="h1"
      type="xhtml.h1.type"/>
    <xs:element
      name="h2"
<xs:choice>
  <xs:element
    name="pre"
    type="xhtml.pre.type"/>
  <xs:element
    name="blockquote"
    type="xhtml.blockquote.type"/>
  <xs:element
    name="address"
    type="xhtml.address.type"/>
</xs:choice>
</xs:group>
<xs:group
  name="xhtml.BblkPres.class">
  <xs:sequence>
    <xs:element
      name="hr"
      type="xhtml.hr.type"/>
  </xs:sequence>
</xs:group>
<xs:group
  name="xhtml.BblkSpecial.class">
  <xs:choice>
    <xs:group
      ref="xhtml.Table.class"/>
    <xs:group
      ref="xhtml.Form.class"/>
  </xs:choice>
</xs:group>
<xs:group
  name="xhtml.Bblk.class">
  <xs:choice>
    <xs:group
      ref="xhtml.BblkStruct.class"/>
    <xs:group
      ref="xhtml.BblkPhras.class"/>
    <xs:group
      ref="xhtml.BblkPres.class"/>
    <xs:group
      ref="xhtml.BblkSpecial.class"/>
    <xs:group
      ref="xhtml.Bblk.extra"/>
  </xs:choice>
</xs:group>
<xs:group
  name="xhtml.Bblk.mix">
  <xs:choice>
    <xs:group
      ref="xhtml.BblkStruct.class"/>
    <xs:group
      ref="xhtml.BblkPhras.class"/>
  </xs:group>
</xs:group>
</xs:group>
</xs:choice>
<!--
Block.class includes all block elements, used as an component in mixes
-->
<xs:group
  name="xhtml.Bblk.class">
  <xs:choice>
    <xs:group
      ref="xhtml.BblkStruct.class"/>
    <xs:group
      ref="xhtml.BblkPhras.class"/>
    <xs:group
      ref="xhtml.BblkPres.class"/>
    <xs:group
      ref="xhtml.BblkSpecial.class"/>
    <xs:group
      ref="xhtml.Bblk.extra"/>
  </xs:choice>
</xs:group>
<!--
Block.mix includes all block elements plus %Misc.class;
-->
<xs:group
  name="xhtml.Bblk.mix">
  <xs:choice>
All Content Elements

Flow.mix includes all text content, block and inline
Note that the "any" element included here allows us
to add data from any other namespace, a necessity
for compound document creation.
Note however that it is not possible to add
to any head level element without further
modification. To add RDF metadata to the head
of a document, modify the structure module.

BlkNoForm.mix includes all non-form block elements,
plus Misc.class

BlkNoForm.mix includes all non-form block elements,
plus Misc.class
C.3. XHTML-Print 1.0 XML Schema Modules

Available for download at xhtml-print-modules-1.xsd.

This schema includes all modules for XHTML1.1 Document Type.
$Id: xhtml-print-modules-1.xsd,v 1.3 2008/10/19 01:20:57 ahby Exp$

XHTML Core modules (Required for XHTML Family Conformance)
+ text
+ hypertext
+ lists
+ structure

Other XHTML modules
+ Edit
+ Presentational
+ Link
+ Meta
+ Scripting
+ Style
+ Image
+ Object
+ Param (Applet/Object modules require Param Module)
+ Basic Tables
+ Target
+ Basic Forms

Schema Framework Component Modules:
+ notations
+ datatypes
+ common attributes
+ character entities

This schema includes all modules (and redefinitions) for XHTML1.1 Document Type.
XHTML1.1 Document Type includes the following Modules
The Text module includes declarations for all core text container elements and their attributes.

+ block phrasal
+ block structural
+ inline phrasal
+ inline structural

Elements defined here:
* address, blockquote, pre, h1, h2, h3, h4, h5, h6
* div, p
* abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var
* br, span

Hypertext module

Elements defined here:
* a

Lists module

Elements defined here:
* dt, dd, dl, ol, ul, li

Structural module

Elements defined here:
* title, head, body, html

Presentational module

Elements defined here:
* hr, b, big, i, small, sub, sup, tt
C.3. XHTML-Print 1.0 XML Schema Modules

```
<xs:attributeGroup name="xhtml.link.attlist">
  <xs:annotation>
    <xs:documentation>
      Changes to XHTML Link Attlist
    </xs:documentation>
    <xs:attributeGroup ref="xhtml.link.attlist">
      <xs:annotation>
        <xs:documentation>
          Original Link Attributes (declared in Link Module)
        </xs:documentation>
        <xs:attributeGroup>
        </xs:attributeGroup>
      </xs:annotation>
    </xs:attributeGroup>
  </xs:annotation>
</xs:attributeGroup>

<xs:include schemaLocation="xhtml-meta-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Meta module
      Elements defined here:
      * meta
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_metamodule"/>
  </xs:annotation>
</xs:include>

<xs:redefine schemaLocation="xhtml-base-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Base module
      Elements defined here:
      * base
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_basemodule"/>
  </xs:annotation>
</xs:redefine>

<xs:include schemaLocation="xhtml-script-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Scripting module
      Elements defined here:
      * script, noscript
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_scriptmodule"/>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="xhtml-style-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Style module
      Elements defined here:
      * style
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_stylemodule"/>
  </xs:annotation>
</xs:include>

<xs:include schemaLocation="xhtml-inlstyle-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Style attribute module
      Attribute defined here:
      * style
    </xs:documentation>
    <xs:documentation source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#s_styleattributemodule"/>
  </xs:annotation>
</xs:include>

<xs:redefine schemaLocation="xhtml-image-1.xsd">
  <xs:annotation>
    <xs:documentation>
      Image module
      Elements defined here:
      * img
    </xs:documentation>
    <xs:attributeGroup name="xhtml.img.attlist">
      <xs:attributeGroup ref="xhtml.img.attlist">
      </xs:attributeGroup>
    </xs:attributeGroup>
  </xs:annotation>
</xs:redefine>
```

-58-
C.3. XHTML-Print 1.0 XML Schema Modules

<xs:annotation>
  <xs:documentation>
    Original Image Attributes (in Image Module)
  </xs:documentation>
</xs:annotation>
<xs:attributeGroup>
  <xs:attribute name="objtype" type="xs:string">
    <xs:documentation>
      Object module
    </xs:documentation>
  </xs:attribute>
</xs:attributeGroup>
<xs:redefine>
  <xs:schemaLocation="xhtml-object-1.xsd">
    <xs:attributeGroup name="xhtml.object.attlist">
      <xs:attributeGroup ref="xhtml.object.attlist">
        <xs:annotation>
          <xs:documentation source="http://www.w3.org/TR/xhtml-modularization/abstract_modules.html#a_objectmodule"/>
        </xs:annotation>
        <xs:attribute ref="xhtml.object.attlist"/>
        <xs:attributeGroup ref="xhtml.object.attlist">
          <xs:annotation>
            Original Object Attlist
          </xs:annotation>
        </xs:attributeGroup>
      </xs:attributeGroup>
    </xs:attributeGroup>
  </xs:schemaLocation>
</xs:redefine>
<xs:include schemaLocation="xhtml-param-1.xsd">
  <xs:attributeGroup name="xhtml.param.attlist">
    <xs:attributeGroup ref="xhtml.param.attlist">
      <xs:annotation>
        <xs:documentation>
          Param module
        </xs:documentation>
      </xs:annotation>
    </xs:attributeGroup>
  </xs:attributeGroup>
</xs:include>
<xs:include schemaLocation="xhtml-basic-table-1.xsd">
  <xs:attributeGroup name="xhtml.table.attlist">
    <xs:attributeGroup ref="xhtml.table.attlist">
      <xs:annotation>
        <xs:documentation>
          Tables module
        </xs:documentation>
      </xs:annotation>
    </xs:attributeGroup>
  </xs:attributeGroup>
</xs:include>
<xs:redefine schemaLocation="xhtml-basic-form-1.xsd">
  <xs:attributeGroup name="xhtml.form.attlist">
    <xs:attributeGroup ref="xhtml.form.attlist">
      <xs:annotation>
        <xs:documentation>
          Changes to XHTML Form Attlist
        </xs:documentation>
      </xs:annotation>
    </xs:attributeGroup>
  </xs:attributeGroup>
</xs:redefine>
<xs:include schemaLocation="xhtml-basic-form-1.xsd">
  <xs:attributeGroup name="xhtml.input.attlist">
    <xs:attributeGroup ref="xhtml.input.attlist">
      <xs:annotation>
        <xs:documentation>
          Changes to XHTML Form Input Element
        </xs:documentation>
      </xs:annotation>
    </xs:attributeGroup>
  </xs:attributeGroup>
</xs:include>
<xs:include schemaLocation="xhtml-label-1.xsd">
  <xs:attributeGroup name="xhtml.label.attlist">
    <xs:attributeGroup ref="xhtml.label.attlist">
      <xs:annotation>
        <xs:documentation>
          Changes to XHTML Form Label Element
        </xs:documentation>
      </xs:annotation>
    </xs:attributeGroup>
  </xs:attributeGroup>
</xs:include>
Original Select Attributes (in Forms Module)

Original TextArea Attributes (in Forms Module)

</xsd:schema>
D. References

C.1. Normative References

[CHARMOD]

[HTML4]

[XHTML1]

[XHTML1.1]

[XHTMLMOD]
The latest version is available at: http://www.w3.org/TR/xhtml-modularization

[XML]

[RFC2119]

[JPEG]
C.2. Informative References

[CCITT]

[JEIDA]

[RFC3986]

[RFC2616]

[RFC3023]

[RFC3236]

[BT601.5]

[CSSPP]

[MIMEMPX]

[RFC2392]

[RFC2397]

[RFC2557]
**RFC2557 - MIME Encapsulation of Aggregate Documents, such as HTML (MHTML)** J. Palme, A. Hopmann, N. Shelness, The Internet Engineering Task Force, March 1999. It is

[XHTMLBASIC]


[XHTMLPRINT]