Modularization of XHTML™

W3C Working Draft 5 January 2000

This version:
http://www.w3.org/TR/2000/WD-xhtml-modularization-20000105
(Single HTML file [p.1] , Postscript version, PDF version, ZIP archive, or Gzip'd TAR archive)
Latest version:
http://www.w3.org/TR/xhtml-modularization
Previous version:
http://www.w3.org/TR/1999/WD-xhtml-modularization-19990910/
Diff-marked version:
xhtml-modularization-diff-20000105.html
Editors:
Murray Altheim, Sun Microsystems
Frank Boumphrey, HTML Writers Guild
Sam Dooley, IBM
Shane McCarron, Applied Testing and Technology
Ted Wugofski, Gateway

Copyright ©2000 W3C® (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

This working draft specifies an abstract modularization of XHTML and an implementation of the abstraction using XML Document Type Definitions (DTDs). This modularization provide a means for subsetting and extending XHTML, a feature needed for extending XHTML's reach onto emerging platforms.

Status of this document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. The latest status of this document series is maintained at the W3C.

This is the "Last Call Working Draft" of "Modularization of XHTML". The Last Call review period ends on 2359Z on 1 Feburary 2000. Please send review comments before the review period ends to www-html-editor@w3.org.

The Working Group anticipates asking the W3C Director to advance this document to Proposed Recommendation after the Working Group processes Last Call review comments and incorporates resolutions into the Guidelines.

This document has been produced as part of the W3C HTML Activity. The goals of the HTML Working Group *(members only)* are discussed in the HTML Working Group charter *(members only)*.

This is a W3C Working Draft for review by W3C Members and other interested parties. It is a draft document and may be updated, replaced or obsoleted by other documents at any time. It is inappropriate to use W3C Working Drafts as reference material or to cite them as other than "work in progress". This is work in progress and does not imply endorsement by, or the consensus of, either W3C or participants of the HTML WG Group.

A list of current W3C Recommendations and other technical documents can be found at http://www.w3.org/TR.

Quick Table of Contents

1. Introduction 7
2. Terms and Definitions 11
3. Conformance Definition 15
4. XHTML Abstract Modules 19
A. XHTML 1.1 Schema Module Implementations 43
B. XHTML 1.1 DTD Module Implementations 45
C. References 115
D. Design Goals 119

Full Table of Contents

1. Introduction 7
1.1. What is XHTML? 7
1.2. What is XHTML Modularization? 7
1.3. Why Modularize XHTML? 7
1.3.1. Abstract modules 8
1.3.2. DTD modules 8
1.3.3. Hybrid document types 8
1.3.4. Validation 9
2. Terms and Definitions 11
3. Conformance Definition 15

3.1. XHTML Family Document Type Conformance 15
3.2. XHTML Family Document Conformance 15
3.3. XHTML Family User Agent Conformance 15
3.4. Naming Rules 17
3.4.1. Rationale for Naming Rules 17
4. XHTML Abstract Modules 19
4.1. Common Characteristics of Modules 19
4.1.1. Syntactic Conventions 19
4.1.2. Content Types 20
4.1.3. Attribute Types 20
4.1.4. Attribute Collections 25
4.2. Basic Modules 26
4.2.1. Structure Module 26
4.2.2. Basic Text Module 26
4.2.3. Hypertext Module 28
4.2.4. List Module 28
4.3. Applet Module 29
4.4. Text Extension Modules 29
4.4.1. Presentation Module 29
4.4.2. Edit Module 30
4.4.3. BDO Module 30
4.5. Forms Modules 30
4.5.1. Basic Forms Module 30
4.5.2. Forms Module 31
4.6. Table Modules 33
4.6.1. Basic Tables Module 33
4.6.2. Tables Module 34
4.7. Image Module 35
4.8. Client-side Image Map Module 35
4.9. Server-side Image Map Module 36
4.10. Object Module 36
4.11. Frames Module 37
4.12. Iframe Module 37
4.13. Intrinsic Events 38
4.14. Metainformation Module 39
4.15. Scripting Module 39
4.16. Stylesheet Module 39
4.17. Link Module 40
4.18. Base Module 40
4.19. Legacy Module 40
A. XHTML 1.1 Schema Module Implementations 43
B. XHTML 1.1 DTD Module Implementations 45
B.1. XHTML Character Entities 45

B.1.1. XHTML Latin 1 Character Entities 45
B.1.2. XHTML Special Characters 49
B.1.3. XHTML Mathematical, Greek, and Symbolic Characters 51
B.2. XHTML Modular Framework 55
B.2.1. XHTML Datatypes 56
B.2.2. XHTML Common Attribute Definitions 58
B.2.3. XHTML Character Entities 59
B.3. XHTML Module Implementations 59
B.3.1. XHTML Basic Modules 60
B.3.2. Applet 66
B.3.3. Text Modules 67
B.3.4. Forms 70
B.3.5. Tables 74
B.3.6. Image 83
B.3.7. Client-side Image Map 84
B.3.8. Server-side Image Map 85
B.3.9. Object 87
B.3.10. Frames 88
B.3.11. Iframe 90
B.3.12. Intrinsic Events 91
B.3.13. Metainformation 93
B.3.14. Scripting 94
B.3.15. Stylesheet 95
B.3.16. Link 96
B.3.17. Base 97
B.3.18. Legacy 98
B.4. XHTML DTD Support Modules 102
B.4.1. Block Phrasal 102
B.4.2. Block Presentational 105
B.4.3. Block Structural 106
B.4.4. Inline Phrasal 107
B.4.5. Inline Presentational 110
B.4.6. Inline Structural 113
B.4.7. Param 114
C. References 115
C.1. Normative References 115
C.2. Informative References 116
D. Design Goals 119
D.1. Requirements 119
D.1.1. Granularity 119
D.1.2. Composibility 119
D.1.3. Ease of Use 120
D.1.4. Compatibility 120

D.1.5. Conformance 120

1. Introduction

This section is *normative*.

1.1. What is XHTML?

XHTML is the reformulation of HTML 4.0 as an application of XML. XHTML 1.0 [XHTML1 [p.115]] specifies three XML document types that correspond to the three HTML 4.0 DTDs: Strict, Transitional, and Frameset. XHTML 1.0 is the basis for a family of document types that subset and extend HTML.

1.2. What is XHTML Modularization?

XHTML Modularization is a decomposition of XHTML 1.0, and by reference HTML 4.0, into a collection of abstract modules that provide specific types of functionality. These abstract modules are implemented in this specification using the XML Document Type Definition language, but an implementation using XML Schemas is expected. The mechanism for defining the abstract modules defined in this document, and for implementing them using XML DTDs, is defined in the document "Building XHTML Modules" [BUILDING [p.115]].

These modules may be combined with each other and with other modules to create XHTML subset and extension document types that qualify as members of the XHTML family of document types.

1.3. Why Modularize XHTML?

The modularization of XHTML refers to the task of specifying well-defined sets of XHTML elements that can be combined and extended by document authors, document type architects, other XML standards specifications, and application and product designers to make it economically feasible for content developers to deliver content on a greater number and diversity of platforms.

Over the last couple of years, many specialized markets have begun looking to HTML as a content language. There is a great movement toward using HTML across increasingly diverse computing platforms. Currently there is activity to move HTML onto mobile devices (hand held computers, portable phones, etc.), television devices (digital televisions, TV-based web browsers, etc.), and appliances (fixed function devices). Each of these devices has different requirements and constraints.

Modularizing XHTML provides a means for product designers to specify which elements are supported by a device using standard building blocks and standard methods for specifying which building blocks are used. These modules serve as "points of conformance" for the content community. The content community can now target the installed base that supports a certain collection of modules, rather than worry about the installed base that supports this permutation of XHTML elements or that permutation of XHTML elements. The use of standards is critical for

modularized XHTML to be successful on a large scale. It is not economically feasible for content developers to tailor content to each and every permutation of XHTML elements. By specifying a standard, either software processes can autonomously tailor content to a device, or the device can automatically load the software required to process a module.

Modularization also allows for the extension of XHTML's layout and presentation capabilities, using the extensibility of XML, without breaking the XHTML standard. This development path provides a stable, useful, and implementable framework for content developers and publishers to manage the rapid pace of technological change on the Web.

1.3.1. Abstract modules

An XHTML document type is defined as a set of abstract modules. A abstract module defines one kind of data that is semantically different from all others. Abstract modules can be combined into document types without a deep understanding of the underlying schema that define the modules.

1.3.2. DTD modules

A DTD module consists of a set of element types, a set of attribute list declarations, and a set of content model declarations, where any of these three sets may be empty. An attribute list declaration in a DTD module may modify an element type outside the element types in the module, and a content model declaration may modify an element type outside the element type set.

An XML DTD is a means of describing the structure of a class of XML documents, collectively known as an XML document type. XML document types are currently represented as DTDs, as described in the XML 1.0 Recommendation [XML] [p.116] . Where possible, this document also allows for the potential use of other schema languages that are currently under consideration by the W3C XML Schema Working Group. (e.g. DCD, SOX, DDML, XSchema)

1.3.3. Hybrid document types

A hybrid document type is an XML DTD composed from a collection of XML DTDs or DTD Modules. The primary purpose of the modularization framework described in this document is to allow a DTD author to combine elements from multiple abstract modules into a hybrid document type, develop documents against that hybrid document type, and to validate that document against the associated hybrid document type definition.

One of the most valuable benefits of XML over SGML is that XML reduces the barrier to entry for standardization of element sets that allow communities to exchange data in an interoperable format. However, the relatively static nature of HTML as the content language for the Web has meant that any one of these communities have previously held out little hope that their XML document types would be able to see widespread adoption as part of Web standards. The modularization framework allows for the dynamic incorporation of these diverse document types within the XHTML family of document types, further reducing the barriers to the incorporation of these domain-specific vocabularies in XHTML documents.

1.3.4. Validation

The use of well-formed, but not valid, documents is an important benefit of XML. In the process of developing a document type, however, the additional leverage provided by a validating parser for error checking is important. The same statement applies to XHTML document types with elements from multiple abstract modules.

The general problem of fragment validation - validation of XML documents with different schemas from multiple XML Namespaces [XMLNAMES] [p.117] in different portions of the document - is beyond the scope of this framework. An essential feature of this framework, however, is a collection of conventions for creating, from a set of abstract modules, hybrid DTDs.

2. Terms and Definitions

This section is *informative*.

While some terms are defined in place, the following definitions are used throughout this document. Familiarity with the W3C XML 1.0 Recommendation [XML] [p.116] is highly recommended.

abstract module
: a unit of document type specification corresponding to a distinct type of content, corresponding to a markup construct reflecting this distinct type.

compound document
: A compound document is a document that uses more than one XML Namespace. Compound documents may be defined as documents that contain elements or attributes from multiple document types.

content model
: the declared markup structure allowed within instances of an element type. XML 1.0 differentiates two types: elements containing only element content (no character data) and mixed content (elements that may contain character data optionally interspersed with child elements). The latter are characterized by a content specification beginning with the "#PCDATA" string (denoting character data).

document model
: the effective structure and constraints of a given document type. The document model constitutes the abstract representation of the physical or semantic structures of a class of documents.

document type
: a class of documents sharing a common abstract structure. The ISO 8879 [SGML] [p.115] definition is as follows: "a class of documents having similar characteristics; for example, journal, article, technical manual, or memo. (4.102)"

document type definition (DTD)
: a formal, machine-readable expression of the XML structure and syntax rules to which a document instance of a specific document type must conform; the schema type used in XML 1.0 to validate conformance of a document instance to its declared document type. The same markup model may be expressed by a variety of DTDs.

driver
: a generally short file used to declare and instantiate the modules of a DTD. A good rule of thumb is that a DTD driver contains no markup declarations that comprise any part of the document model itself.

element
: an instance of an element type.

element type
: the definition of an element, that is, a container for a distinct semantic class of document content.

entity
: an entity is a logical or physical storage unit containing document content. Entities may be composed of parse-able XML markup or character data, or unparsed (ie., non-XML,

possibly non-textual) content. Entity content may be either defined entirely within the document entity ("internal entities") or external to the document entity ("external entities"). In parsed entities, the replacement text may include references to other entities.

entity reference
: a mnemonic or numeric string used as a reference to the content of a declared entity (eg., "&" for "&", "<" for "<", "©" for "©".)

generic identifier
: the name identifying the element type of an element. Also, element type name.

instantiate
: to replace an entity reference with an instance of its declared content.

markup declaration
: a syntactical construct within a DTD declaring an entity or defining a markup structure. Within XML DTDs, there are four specific types: entity declaration defines the binding between a mnemonic symbol and its replacement content. element declaration constrains which element types may occur as descendants within an element. See also content model. attribute definition list declaration defines the set of attributes for a given element type, and may also establish type constraints and default values. notation declaration defines the binding between a notation name and an external identifier referencing the format of an unparsed entity

markup model
: the markup vocabulary (ie., the gamut of element and attribute names, notations, etc.) and grammar (ie., the prescribed use of that vocabulary) as defined by a document type definition (ie., a schema) The markup model is the concrete representation in markup syntax of the document model, and may be defined with varying levels of strict conformity. The same document model may be expressed by a variety of markup models.

module
: an abstract unit within a document model expressed as a DTD fragment, used to consolidate markup declarations to increase the flexibility, modifiability, reuse and understanding of specific logical or semantic structures.

modularization
: an implementation of a modularization model; the process of composing or de-composing a DTD by dividing its markup declarations into units or groups to support specific goals. Modules may or may not exist as separate file entities (ie., the physical and logical structures of a DTD may mirror each other, but there is no such requirement).

modularization model
: the abstract design of the document type definition (DTD) in support of the modularization goals, such as reuse, extensibility, expressiveness, ease of documentation, code size, consistency and intuitiveness of use. It is important to note that a modularization model is only orthogonally related to the document model it describes, so that two very different modularization models may describe the same document type.

parameter
: entity an entity whose scope of use is within the document prolog (ie., the external subset/DTD or internal subset). Parameter entities are disallowed within the document instance.

parent document type
: A parent document type of a compound document is the document type of the root element.

tag

descriptive markup delimiting the start and end (including its generic identifier and any attributes) of an element.

3. Conformance Definition

This section is *normative*.

In order to ensure that XHTML-family documents are maximally portable among XHTML-family user agents, this specification rigidly defines conformance requirements for both of these and for XHTML-family document types. While the conformance definitions can be found in this section, they necessarily reference normative text within this document, within the base XHTML specification [XHTML1 [p.115]], and within other related specifications. It is only possible to fully comprehend the conformance requirements of XHTML through a complete reading of all normative references.

3.1. XHTML Family Document Type Conformance

It is possible to modify existing document types and define wholly new document types using both modules defined in this specification and other modules. Such a document type conforms to this specification when it meets the following criteria:

1. The document type must be defined using one of the implementation methods defined by the W3C (currently this is limited to XML DTDs, but XML Schema will be available soon).
2. The document type must have a unique identifier as defined in Naming Rules [p.17] .
3. The document type must include, at a minimum, the Structure, Hypertext, Basic Text, and List modules defined in this specification.
4. For each of the W3C-defined modules that are included, all of the elements, attributes, and any required minimal content models must be included (and optionally extended) in the document type's content model.
5. The document type may define additional elements and attributes. However, these must be in their own XML Namespace [XMLNAMES [p.??]].

3.2. XHTML Family Document Conformance

Documents that rely upon XHTML-family document types are considered XHTML conforming if they validate against their referenced document type.

3.3. XHTML Family User Agent Conformance

A conforming user agent must meet all of the following criteria (as defined in [XHTML1 [p.115]]):

1. In order to be consistent with the XML 1.0 Recommendation [XML] [p.??] , the user agent must parse and evaluate an XHTML document for well-formedness. If the user agent claims to be a validating user agent, it must also validate documents against their referenced DTDs according to [XML] [p.??] .
2. When the user agent claims to support facilities defined within this specification or required by this specification through normative reference, it must do so in ways consistent with the facilities' definition.

3. When a user agent processes an XHTML document as generic XML, it shall only recognize attributes of type `ID` (e.g. the `id` attribute on most XHTML elements) as fragment identifiers.
4. If a user agent encounters an element it does not recognize, it must render the element's content.
5. If a user agent encounters an attribute it does not recognize, it must ignore the entire attribute specification (i.e., the attribute and its value).
6. If a user agent encounters an attribute value it doesn't recognize, it must use the default attribute value.
7. If it encounters an entity reference (other than one of the predefined entities) for which the User Agent has processed no declaration (which could happen if the declaration is in the external subset which the User Agent hasn't read), the entity reference should be rendered as the characters (starting with the ampersand and ending with the semi-colon) that make up the entity reference.
8. When rendering content, User Agents that encounter characters or character entity references that are recognized but not renderable should display the document in such a way that it is obvious to the user that normal rendering has not taken place.
9. The following characters are defined in [XML] as whitespace characters:
   - Space ( )
   - Tab (	)
   - Carriage return (
)
   - Line feed (
)

   The XML processor normalizes different system's line end codes into one single line-feed character, that is passed up to the application. The XHTML user agent in addition, must treat the following characters as whitespace:

   - Form feed ()
   - Zero-width space ()

   In elements where the 'xml:space' attribute is set to 'preserve', the user agent must leave all whitespace characters intact (with the exception of leading and trailing whitespace characters, which should be removed). Otherwise, whitespace is handled according to the following rules:

   - All whitespace surrounding block elements should be removed.
   - Comments are removed entirely and do not affect whitespace handling. One whitespace character on either side of a comment is treated as two white space characters.
   - Leading and trailing whitespace inside a block element must be removed.
   - Line feed characters within a block element must be converted into a space (except when the 'xml:space' attribute is set to 'preserve').
   - A sequence of white space characters must be reduced to a single space character (except when the 'xml:space' attribute is set to 'preserve').
   - With regard to rendition, the User Agent should render the content in a manner appropriate to the language in which the content is written. In languages whose primary

script is Latinate, the ASCII space character is typically used to encode both grammatical word boundaries and typographic whitespace; in languages whose script is related to Nagari (e.g., Sanskrit, Thai, etc.), grammatical boundaries may be encoded using the ZW 'space' character, but will not typically be represented by typographic whitespace in rendered output; languages using Arabiform scripts may encode typographic whitespace using a space character, but may also use the ZW space character to delimit 'internal' grammatical boundaries (what look like words in Arabic to an English eye frequently encode several words, e.g. 'kitAbuhum' = 'kitAbu-hum' = 'book them' == their book); and languages in the Chinese script tradition typically neither encode such delimiters nor use typographic whitespace in this way.

Whitespace in attribute values is processed according to [XML] [p.??] .

3.4. Naming Rules

Names for XHTML-conforming document types must adhere to strict naming conventions so that it is possible for software and users to readily determine the relationship of document types to XHTML. The names for document types implemented as XML Document Type Definitions are defined through XML Formal Public Identifiers (FPIs). Within FPIs, fields are separated by double slash character sequences (`//`). The various fields MUST be composed as follows:

1. The leading field identifies the resources relationship to a formal standard. For privately defined resources, this field MUST be "`-`". For formal standards, this field MUST be the formal reference to the standard (e.g. `ISO/IEC 15445:1999`).
2. The second field MUST contain the name of the organization responsible for maintaining the named item. There is no formal registry for these organization names. Each organization SHOULD define a name that is unique. The name used by the W3C is, for example, `W3C`.
3. The third field MUST take the form `DTD XHTML-` followed by an organization-defined unique identifier (e.g. MyML 1.0). This identifier SHOULD be composed of a unique name and a version identifier that can be updated as the document type evolves.
4. The fourth field defines the language in which the item is developed (e.g. `EN`).

Using these rules, the name for an XHTML family conforming document type might be `-//MyCompany//DTD XHTML-MyML 1.0//EN`.

3.4.1. Rationale for Naming Rules

Naming Rules are critical for portability of user agents and XHTML-conforming tools. These rules need to be simple enough that they can be readily adhered to, and need to convey upon document type and module designers the power to readily associate their creations with XHTML (for marketing purposes, if nothing else). The above rules address these concerns. There were some other possibilities for naming conventions, and they were not used for the following reasons:

- Use the XHTML version in the identifier.

  In the case of new modules, there is no need to associate the module with a specific version of XHTML - the name does not need to identify version dependencies. In the case of new document types, the new type does not necessarily have any relationship to a specific version of XHTML. Instead, the new document type should itself have versioning that will help in its evolution. Document types will necessarily evolve out of step with XHTML from the W3C.

4. XHTML Abstract Modules

This section is *normative*.

This section specifies the contents of the XHTML abstract modules. These modules are abstract definitions of collections of elements, attributes, and their content models. These abstract modules can be mapped onto any appropriate specification mechanism. The XHTML 1.1 Specification [p.??] , for example, maps these modules onto DTDs as described in [XML] [p.116] .

Content developers and device designers should view this section as a guide to the definition of the functionality provided by the various XHTML-defined modules. When developing documents or defining a profile for a class of documents, content developers can determine which of these modules are essential for conveying their message. When designing clients, device designers should develop their device profiles by choosing from among the abstract modules defined here.

4.1. Common Characteristics of Modules

Many of the abstract modules in this section describe elements, attributes on those elements, and minimal content models for those elements or element sets. This section identifies some shorthand expressions that are used throughout the abstract module definitions. These expressions should in no way be considered normative or mandatory. They are an editorial convenience for this document. When used in the remainder of this section, it is the expansion of the term that is normative, not the term itself.

4.1.1. Syntactic Conventions

The abstract modules are not defined in a formal grammar. However, the definitions do adhere to the following syntactic conventions (as defined in Building XHTML Modules [BUILDING [p.115]]). These conventions are similar to those of XML DTDs, and should be familiar to XML DTD authors. Each discrete syntactic element can be combined with others to make more complex expressions that conform to the algebra defined here.

element name
: When an element is included in a content model, its explicit name will be listed.

Content set
: Some modules define lists of explicit element names called *content sets*. When a content set is included in a content model, its name will be listed.

`expr ?`
: Zero or one instances of expr are permitted.

`expr +`
: One or more instances or expr are required.

`expr *`
: Zero or more instances of expr are permitted.

`a , b`
: Expression `a` is required, followed by expression `b`.

`a | b`
: Either expression a or expression b is required.

`a - b`
: Expression a is permitted, omitting elements in expression b.

parentheses
: When an expression is contained within parentheses, evaluation of any subexpressions within the parentheses take place before evaluation of expressions outside of the parentheses (starting at the deepest level of nesting first).

extending pre-defined elements
: In some instances, a module adds attributes to an element. In these instances, the element name is followed by an ampersand (&).

Defining the type of attribute values
: When a module defines the type of an attribute value, it does so by listing the type in parentheses after the attribute name.

Defining the legal values of attributes
: When a module defines the legal values for an attribute, it does so by listing the explicit legal values (enclosed in quotation marks), separated by vertical bars `|`, inside of parentheses following the attribute name.

4.1.2. Content Types

The abstract module definitions in this document define minimal, atomic content models for each module. These minimal content models reference the elements in the module itself. They may also reference elements in other modules upon which the abstract module depends. Finally, the content model in many cases requires that text be permitted as content to one or more elements. In these cases, the symbol used for text is PCDATA. This is a term, defined in the XML 1.0 Recommendation, that refers to processed character data. A content type can also be defined as `EMPTY`, meaning the element has no content in its minimal content model.

4.1.3. Attribute Types

In some instances, the types of attribute values or the explicit set of permitted values for attributes are defined. The following attribute types (defined in the XML 1.0 Recommendation) are used in the definitions of the Abstract Modules:

Attribute Type	Definition
CDATA	Character data
ID	A document-unique identifier
IDREF	A reference to a document-unique identifier
IDREFS	A space-separated list of references to document-unique identifiers
NAME	A name with the same character constraints as ID above
NMTOKEN	A name composed of CDATA characters but no whitespace
NMTOKENS	Multiple names composed of CDATA characters separated by whitespace
PCDATA	Processed character data

In addition to these pre-defined data types, XHTML Modularization defines the following data types and their semantics (as appropriate):

Data types	Description
Character	A single character from [ISO10646] [p.115] .
Charset	A character encoding, as per [RFC2045] [p.115] .
Charsets	A space separated list of character encodings, as per [RFC2045] [p.115] .
Color	The attribute value type "Color" refers to color definitions as specified in [SRGB] [p.115] . A color value may either be a hexadecimal number (prefixed by a hash mark) or one of the following sixteen color names. The color names are case-insensitive. **Color names and sRGB values** Black = "#000000" Green = "#008000" Silver = "#C0C0C0" Lime = "#00FF00" Gray = "#808080" Olive = "#808000" White = "#FFFFFF" Yellow = "#FFFF00" Maroon = "#800000" Navy = "#000080" Red = "#FF0000" Blue = "#0000FF" Purple = "#800080" Teal = "#008080" Fuchsia = "#FF00FF" Aqua = "#00FFFF" Thus, the color values "#800080" and "Purple" both refer to the color purple.
ContentType	A media type, as per [RFC2045] [p.115] .
ContentTypes	A comma-separated list of media types, as per [RFC2045] [p.115] .

Datetime	Date and time information.
LanguageCode	A language code, as per [RFC1766] [p.115] .
Length	The value may be either a Pixels or a percentage of the available horizontal or vertical space. Thus, the value "50%" means half of the available space.

LinkTypes	Authors may use the following recognized link types, listed here with their conventional interpretations. A LinkTypes value refers to a space-separated list of link types. White space characters are not permitted within link types. These link types are case-insensitive, i.e., "Alternate" has the same meaning as "alternate". User agents, search engines, etc. may interpret these link types in a variety of ways. For example, user agents may provide access to linked documents through a navigation bar. **Alternate** Designates substitute versions for the document in which the link occurs. When used together with the `xml:lang` attribute, it implies a translated version of the document. When used together with the `media` attribute, it implies a version designed for a different medium (or media). **Stylesheet** Refers to an external style sheet. See the Style Module [p.39] for details. This is used together with the link type "Alternate" for user-selectable alternate style sheets. **Start** Refers to the first document in a collection of documents. This link type tells search engines which document is considered by the author to be the starting point of the collection. **Next** Refers to the next document in a linear sequence of documents. User agents may choose to preload the "next" document, to reduce the perceived load time. **Prev** Refers to the previous document in an ordered series of documents. Some user agents also support the synonym "Previous". **Contents** Refers to a document serving as a table of contents. Some user agents also support the synonym *ToC* (from "Table of Contents"). **Index** Refers to a document providing an index for the current document. **Glossary** Refers to a document providing a glossary of terms that pertain to the current document. **Copyright** Refers to a copyright statement for the current document. **Chapter** Refers to a document serving as a chapter in a collection of documents. **Section** Refers to a document serving as a section in a collection of documents. **Subsection** Refers to a document serving as a subsection in a collection of documents. **Appendix** Refers to a document serving as an appendix in a collection of documents. **Help** Refers to a document offering help (more information, links to other sources information, etc.) **Bookmark** Refers to a bookmark. A bookmark is a link to a key entry point within an extended document. The title attribute may be used, for example, to label the bookmark. Note that several bookmarks may be defined in each document.

MediaDesc	The MediaDesc attribute is a comma separated list of media descriptors. The following is a list of recognized media descriptors:
	
	screen
	Intended for non-paged computer screens.
	tty
	Intended for media using a fixed-pitch character grid, such as teletypes, terminals, or portable devices with limited display capabilities.
	tv
	Intended for television-type devices (low resolution, color, limited scrollability).
	projection
	Intended for projectors.
	handheld
	Intended for handheld devices (small screen, monochrome, bitmapped graphics, limited bandwidth).
	print
	Intended for paged, opaque material and for documents viewed on screen in print preview mode.
	braille
	Intended for braille tactile feedback devices.
	aural
	Intended for speech synthesizers.
	all
	Suitable for all devices.
	
	Future versions of XHTML may introduce new values and may allow parameterized values. To facilitate the introduction of these extensions, conforming user agents must be able to parse the media attribute value as follows:
	
	1. The value is a comma-separated list of entries. For example,
	
	`media="screen, 3d-glasses, print and resolution > 90dpi"`
	
	is mapped to:
	
	`"screen"`
	`"3d-glasses"`
	`"print and resolution > 90dpi"`
	
	2. Each entry is truncated just before the first character that isn't a US ASCII letter [a-zA-Z] (ISO 10646 hex 41-5a, 61-7a), digit [0-9] (hex 30-39), or hyphen (hex 2d). In the example, this gives:
	
	`"screen"`
	`"3d-glasses"`
	`"print"`
	
	3. A case-sensitive match is then made with the set of media types defined above. User agents may ignore entries that don't match. In the example we are left with `screen` and `print`.
	
	Note. *Style sheets may include media-dependent variations within them (e.g., the CSS* ***@media*** *construct). In such cases it may be appropriate to use "media =all".*

MultiLength	The value may be a Length or a relative length. A relative length has the form "i*", where "i" is an integer. When allotting space among elements competing for that space, user agents allot pixel and percentage lengths first, then divide up remaining available space among relative lengths. Each relative length receives a portion of the available space that is proportional to the integer preceding the "*". The value "*" is equivalent to "1*". Thus, if 60 pixels of space are available after the user agent allots pixel and percentage space, and the competing relative lengths are 1*, 2*, and 3*, the 1* will be alloted 10 pixels, the 2* will be alloted 20 pixels, and the 3* will be alloted 30 pixels.
Number	One or more digits
Pixels	The value is an integer that represents the number of pixels of the canvas (screen, paper). Thus, the value "50" means fifty pixels. For normative information about the definition of a pixel, please consult [CSS1] [p.??]
Script	Script data can be the content of the "script" element and the value of intrinsic event attributes. User agents must not evaluate script data as HTML markup but instead must pass it on as data to a script engine. The case-sensitivity of script data depends on the scripting language. Please note that script data that is element content may not contain character references, but script data that is the value of an attribute may contain them.
Text	Arbitrary textual data, likely meant to be human-readable.
URI	A Uniform Resource Identifier, as per [URI] [p.116] .
URI	A space separated list of Uniform Resource Identifiers, as per [URI] [p.116] .

4.1.4. Attribute Collections

The following basic attribute sets are used on many elements. In each case where they are used, their use is identified via their name rather than enumerating the list.

Collection Name	Attributes in Collection
Core	class (NMTOKEN [p.21]), id (ID [p.21]), title (CDATA [p.21])
I18N	dir ("rtl" | "ltr"), xml:lang (NMTOKEN [p.21])
Events	onclick (Script [p.25]), ondblclick (Script [p.25]), onmousedown (Script [p.25]), onmouseup (Script [p.25]), onmouseover (Script [p.25]), onmousemove (Script [p.25]), onmouseout (Script [p.25]), onkeypress (Script [p.25]), onkeydown (Script [p.25]), onkeyup (Script [p.25])
Style	style (CDATA [p.21])
Common	Core [p.25] + Events [p.25] + I18N [p.25] + Style [p.25]

Note that the Events collection is only defined when the Intrinsic Events abstract module is selected. Otherwise, the Events collection is empty.

Also note that the Style collection is only defined when the Stylesheet Module is selected. Otherwise, the Style collection is empty.

4.2. Basic Modules

The basic modules are modules that are required to be present in any XHTML Family Conforming Document Type [p.15] .

4.2.1. Structure Module

The Structure Module defines the major structural elements for XHTML. These elements effectively act as the basis for the content model of many XHTML family document types. The elements and attributes included in this module are:

Elements	Attributes	Minimal Content Model
body	Common [p.25]	(Heading | Block | List)*
head	I18N [p.25] , profile (URI [p.25])	title
html	I18N [p.25] , version (CDATA [p.21]), xmlns (URI [p.25])	head, body
title	I18N [p.25]	PCDATA

This module is the basic structural definition for XHTML content. The `html` element acts as the root element for all XHTML Family Document Types.

4.2.2. Basic Text Module

This module defines all of the basic text container elements, attributes, and their content model:

Element	Attributes	Minimal Content Model
abbr	Common [p.25]	(PCDATA | Inline)*
acronym	Common [p.25]	(PCDATA | Inline)*
address	Common [p.25]	(PCDATA | Inline)*
blockquote	Common [p.25] , cite (URI [p.25])	(PCDATA | Heading | Block)*
br	Core [p.25]	EMPTY
cite	Common [p.25]	(PCDATA | Inline)*
code	Common [p.25]	(PCDATA | Inline)*
dfn	Common [p.25]	(PCDATA | Inline)*
div	Common [p.25]	(Heading | Block | List)*
em	Common [p.25]	(PCDATA | Inline)*
h1	Common [p.25]	(PCDATA | Inline)*
h2	Common [p.25]	(PCDATA | Inline)*
h3	Common [p.25]	(PCDATA | Inline)*
h4	Common [p.25]	(PCDATA | Inline)*
h5	Common [p.25]	(PCDATA | Inline)*
h6	Common [p.25]	(PCDATA | Inline)*
kbd	Common [p.25]	(PCDATA | Inline)*
p	Common [p.25]	(PCDATA | Inline)*
pre	Common [p.25]	(PCDATA | Inline)*
q	Common [p.25] , cite (URI [p.25])	(PCDATA | Inline)*
samp	Common [p.25]	(PCDATA | Inline)*
span	Common [p.25]	(PCDATA | Inline)*
strong	Common [p.25]	(PCDATA | Inline)*
var	Common [p.25]	(PCDATA | Inline)*

The minimal content model for this module defines some content sets:

Heading
h1 | h2 | h3 | h4 | h5 | h6
Block
address | blockquote | div | p | pre
Inline
abbr | acronym | br | cite | code | dfn | em | kbd | q | samp | span | strong | var
Flow
Heading | Block | Inline

4.2.3. Hypertext Module

The Hypertext Module provides the element that is used to define hypertext links to other resources. This module supports the following element and attributes:

Element	Attributes	Minimal Content Model
a	Common [p.25] , accesskey (Character [p.21]), charset (Charset [p.21]), href (URI [p.25]), hreflang (LanguageCode [p.22]), rel (LinkTypes [p.23]), rev (LinkTypes [p.23]), tabindex (Number [p.25]), type (ContentType [p.21])	(PCDATA | Inline - a)*

This module adds the `a` element to the Inline content set of the Basic Text Module.

4.2.4. List Module

As its name suggests, the List Module provides list-oriented elements. Specifically, the List Module supports the following elements and attributes:

Elements	Attributes	Minimal Content Model
dl	Common [p.25]	(dt | dd)+
dt	Common [p.25]	(PCDATA | Inline)*
dd	Common [p.25]	(PCDATA | Inline)*
ol	Common [p.25]	li+
ul	Common [p.25]	li+
li	Common [p.25]	(PCDATA | Inline)*

This module also defines the content set List with the minimal content model (dl | ol | ul)+ and adds this set to the Flow content set of the Basic Text Module.

4.3. Applet Module

The Applet Module provides elements for referencing external applications. Specifically, the Applet Module supports the following elements and attributes:

Element	Attributes	Minimal Content Model
applet	Core, alt (Text [p.25]), archive (CDATA [p.21]), code (CDATA [p.21]), codebase (URI [p.25]), height (Length [p.22]), name (CDATA [p.21]), object (CDATA [p.21]), width (Length [p.22])	param?
param	id (ID [p.21]), name (CDATA [p.21]), type (ContentType [p.21]), value (CDATA [p.21]), valuetype ("data" | "ref" | "object")	EMPTY

When the Applet Module is used, it adds the `applet` element to the Inline content set of the Basic Text Module.

4.4. Text Extension Modules

This section defines a variety of additional textual markup modules.

4.4.1. Presentation Module

This module defines elements, attributes, and a minimal content model for simple presentation-related markup:

Element	Attributes	Minimal Content Model
b	Common [p.25]	(PCDATA | Inline)*
big	Common [p.25]	(PCDATA | Inline)*
hr	Common [p.25]	EMPTY
i	Common [p.25]	(PCDATA | Inline)*
small	Common [p.25]	(PCDATA | Inline)*
sub	Common [p.25]	(PCDATA | Inline)*
sup	Common [p.25]	(PCDATA | Inline)*
tt	Common [p.25]	(PCDATA | Inline)*

When this module is used, the `hr` element is added to the Block content set of the Basic Text Module. In additional, the `b, big, i, small, sub, sup,` and `tt` elements are added to the Inline content set of the Basic Text Module.

4.4.2. Edit Module

This module defines elements and attributes for use in editing-related markup:

Element	Attributes	Minimal Content Model
del	Common [p.25] , cite (URI [p.25]), datetime (Datetime [p.22])	(PCDATA | Inline)*
ins	Common [p.25] , cite (URI [p.25]), datetime (Datetime [p.22])	(PCDATA | Inline)*

When this module is used, the `del` and `ins` elements are added to the Inline content set of the Basic Text Module.

4.4.3. BDO Module

The BDO module defines an element that can be used to declare the bi-directional rules for the element's content.

Elements	Attributes	Minimal Content Model
bdo	Common [p.25]	(PCDATA | Inline)*

When this module is used, the `bdo` element are added to the Inline content set of the Basic Text Module.

4.5. Forms Modules

4.5.1. Basic Forms Module

The Basic Forms Module provides the forms features found in HTML 3.2. Specifically, the Basic Forms Module supports the following elements, attributes, and minimal content model:

Elements	Attributes	Minimal Content Model
form	Common [p.25] , action (URI [p.25]), method ("get" | "put"), enctype (ContentType [p.21])	Heading | Block - form
input	Common [p.25] , checked ("checked"), maxlength (Number [p.25]), name (CDATA [p.21]), size (Number [p.25]), src (URI [p.25]), type ("text", "password", "checkbox", "radio", "submit", "reset", "file", "hidden", "image"), value (CDATA [p.21])	EMPTY
select	Common [p.25] , multiple ("multiple"), name (CDATA [p.21]), size (Number [p.25])	option+
option	Common [p.25] , selected ("selected"), value (CDATA [p.21])	Inline*
textarea	Common [p.25] , columns (Number [p.25]), name (CDATA [p.21]), rows (Number [p.25])	PCDATA*

This module defines two content sets:

Form
    form
Formctrl
    input | select | textarea

When this module is used, it adds the Form content set to the Block content set and it adds the Formctrl content set to the Inline content set as these are defined in the Basic Text Module.

4.5.2. Forms Module

The Forms Module provides all of the forms features found in HTML 4.0. Specifically, the Forms Module supports:

Elements	Attributes	Minimal Content Model
form	Common [p.25] , accept (ContentTypes [p.21]), accept-charset (Charsets [p.21]), action (URI [p.25]), method ("get" | "put"), enctype (ContentType [p.21])	(Heading | Block - form | fieldset)+
input	Common [p.25] , accept (ContentTypes [p.21]), accesskey (Character [p.21]), alt (CDATA [p.21]), checked ("checked"), disabled ("disabled"), maxlength (Number [p.25]), name (CDATA [p.21]), readonly ("readonly"), size (Number [p.25]), src (URI [p.25]), tabindex (Number [p.25]), type ("text", "password", "checkbox", "radio", "submit", "reset", "file", "hidden", "image"), value (CDATA [p.21])	EMPTY
select	Common [p.25] , disabled ("disabled"), multiple ("multiple"), name (CDATA [p.21]), size (Number [p.25]), tabindex (Number [p.25])	(optgroup | option)+
option	Common [p.25] , disabled ("disabled"), label (Text [p.25]), selected ("selected"), value (CDATA [p.21])	PCDATA
textarea	Common [p.25] , accesskey (Character [p.21]), columns (Number [p.25]), disabled ("disabled"), name (CDATA [p.21]), readonly ("readonly"), rows (Number [p.25]), tabindex (Number [p.25])	PCDATA
button	Common [p.25] , accesskey (Character [p.21]), disabled ("disabled"), name (CDATA [p.21]), tabindex (Number [p.25]), type ("button" | "submit" | "reset"), value (CDATA [p.21])	(PCDATA | Heading | List | Block - Form | Inline - Formctrl)*
fieldset	Common [p.25]	(PCDATA | legend | Flow)*
label	Common [p.25] , accesskey (Character [p.21]), for (IDREF [p.21])	(PCDATA | Inline - label)*
legend	Common [p.25] , accesskey (Character [p.21])	(PCDATA | Inline)+
optgroup	Common [p.25] , disabled ("disabled"), label (Text [p.25])	option+

This module defines two content sets:

Form
    form | fieldset

Formctrl
input | select | textarea | label | button

When this module is used, it adds the Form content set to the Block content set and it adds the Formctrl content set to the Inline content set as these are defined in the Basic Text Module.

The Forms Module is a superset of the Basic Forms Module. These modules may not be used together in a single document type.

4.6. Table Modules

4.6.1. Basic Tables Module

The Basic Tables Module provides table-related elements, but only in a limited form. Specifically, the Basic Tables Module supports:

Elements	Attributes	Minimal Content Model
caption	Common [p.25]	(PCDATA | Inline)*
table	Common [p.25] , border (Pixels [p.25]), cellpadding (Length [p.22]). cellspacing(Length [p.22]), summary (Text [p.25]), width (Length [p.22]) (	caption?, tr+
td	Common [p.25] , abbr (Text [p.25]), align ("left" | "center" | "right"), axis (CDATA [p.21]), colspan (Number [p.25]), headers (IDREFS [p.21]), rowspan (Number [p.25]), scope ("row" | "col" | "rowgroup" | "colgroup"), valign ("top" | "middle" | "bottom")	(PCDATA | Flow)*
th	Common [p.25] , abbr (Text [p.25]), align ("left" | "center" | "right"), axis (CDATA [p.21]), colspan (Number [p.25]), headers (IDREFS [p.21]), rowspan (Number [p.25]), scope ("row" | "col" | "rowgroup" | "colgroup"), valign ("top" | "middle" | "bottom")	(PCDATA | Flow)*
tr	Common [p.25] , align ("left" | "center" | "right"), valign ("top" | "middle" | "bottom")	(th | td)+

When this module is used, it adds the `table` element to the Block content set as defined in the Basic Text Module.

4.6.2. Tables Module

As its name suggests, the Tables Module provides table-related elements that are better able to be accessed by non-visual user agents. Specifically, the Tables Module supports the following elements, attributes, and content model:

Elements	Attributes	Minimal Content Model
caption	Common [p.25]	(PCDATA | Inline)*
table	Common [p.25] , border (Pixels [p.25]), cellpadding (Length [p.22]), cellspacing (Length [p.22]), datapagesize (CDATA [p.21]), frame ("void" | "above" | below" | "hsides" | "lhs" | "rhs" | "vsides" | "box" | "border"), rules ("none" | "groups" | "rows" | "cols" | "all"), summary (Text [p.25]), width (Length [p.22])	caption?, (col* | colgroup*), ((thead?, tfoot?, tbody+) | (tr+))
td	Common [p.25] , abbr (Text [p.25]), align ("left" | "center" | "right" | "justify" | "char"), axis (CDATA [p.21]), char (Character [p.21]), charoff (Length [p.22]), colspan (Number [p.25]), headers (IDREFS [p.21]), rowspan (Number [p.25]), scope ("row", "col", "rowgroup", "colgroup"), valign ("top" | "middle" | "bottom" | "baseline")	(PCDATA | Inline)*
th	Common [p.25] , abbr (Text [p.25]), align ("left" | "center" | "right" | "justify" | "char"), axis (CDATA [p.21]), char (Character [p.21]), charoff (Length [p.22]), colspan (Number [p.25]), headers (IDREFS [p.21]), rowspan (Number [p.25]), scope ("row", "col", "rowgroup", "colgroup"), valign ("top" | "middle" | "bottom" | "baseline")	(PCDATA | Inline)*
tr	Common [p.25] , align ("left" | "center" | "right" | "justify", "char"), char (Character [p.21]), charoff (Length [p.22]), valign ("top" | "middle" | "bottom" | "baseline")	(td | th)+
col	Common [p.25] , align ("left" | "center" | "right" | "justify", "char"), char (Character [p.21]), charoff (Length [p.22]), span (Number [p.25]), valign ("top" | "middle" | "bottom" | "baseline"), width (MultiLength [p.25])	EMPTY
colgroup	Common [p.25] , align ("left" | "center" | "right" | "justify", "char"), char (Character [p.21]), charoff (Length [p.22]), span (Number [p.25]), valign ("top" | "middle" | "bottom" | "baseline"), width (MultiLength [p.25])	col*

Elements	Attributes	Minimal Content Model
tbody	Common [p.25] , align ("left" | "center" | "right" | "justify", "char"), char (Character [p.21]), charoff (Length [p.22]), valign ("top" | "middle" | "bottom" | "baseline")	tr+
thead	Common [p.25] , align ("left" | "center" | "right" | "justify", "char"), char (Character [p.21]), charoff (Length [p.22]), valign ("top" | "middle" | "bottom" | "baseline")	tr+
tfoot	Common [p.25] , align ("left" | "center" | "right" | "justify", "char"), char (Character [p.21]), charoff (Length [p.22]), valign ("top" | "middle" | "bottom" | "baseline")	tr+

When this module is used, it adds the `table` element to the Block content set of the Basic Text Module.

4.7. Image Module

The Image Module provides basic image embedding, and may be used in some implementations independently of client side image maps. The Image Module supports the following element and attributes:

Elements	Attributes	Minimal Content Model
img	Common [p.25] , alt (Text [p.25]), height (Length [p.22]), longdesc (URI [p.25]), src (URI [p.25]), width (Length [p.22])	EMPTY

When this module is used, it adds the `img` element to the Inline content set of the Basic Text Module.

4.8. Client-side Image Map Module

The Client-side Image Map Module provides elements for client side image maps. It requires that the Image Module (or another module that supports the `img` element) be included. The Client-side Image Map Module supports the following elements:

Elements	Attributes	Minimal Content Model
a&	coords (CDATA [p.21]), shape ("rect" | "circle" | "poly" | "default")	n/a
area	Common [p.25] , accesskey (Character [p.21]), alt (Text [p.25]), coords (CDATA [p.21]), href (URI [p.25]), nohref ("nohref"), shape ("rect" | "circle" | "poly" | "default"), tabindex (Number [p.25])	EMPTY
img&	usemap (IDREF [p.21])	n/a
map	Common [p.25]	((Heading | Block) | area)+
object&	usemap (IDREF [p.21])	Note: Only when the object module is included

When this module is used, the `table` element is added to the Block content set of the Basic Text Module.

4.9. Server-side Image Map Module

The Server-side Image Map Module provides support for image-selection and transmission of selection coordinates. It requires that the Image Module (or another module that supports the `img` element) be included. The Server-side Image Map Module supports the following attributes:

Elements	Attributes	Minimal Content Model
img&	ismap ("ismap")	n/a

4.10. Object Module

The Object Module provides elements for general-purpose object inclusion. Specifically, the Object Module supports:

Elements	Attributes	Minimal Content Model
object	Common, archive (URIs [p.25]), classid (URI [p.25]), codebase (URI [p.25]), codetype (ContentType [p.21]), data (URI [p.25]), declare ("declare"), height (Length [p.22]), standby (Text [p.25]), tabindex (Number [p.25]), type (ContentType [p.21]), width (Length [p.22])	(PCDATA | Flow | param)*
param	id (ID [p.21]), name (CDATA [p.21]), type (ContentType [p.21]), value (CDATA [p.21]), valuetype ("data" | "ref" | "object")	EMPTY

When this module is used, it adds the `object` element to the Inline content set of the Basic Text Module.

4.11. Frames Module

As its name suggests, the Frames Module provides frame-related elements. Specifically, the Frames Module supports:

Elements	Attributes	Minimal Content Model
frameset	Core [p.25] , cols (MultiLength [p.25]), rows (MultiLength [p.25])	(frame | noframes)+
frame	Core [p.25] , frameborder ("1" | "0"), longdesc (URI [p.25]), marginheight (Pixels [p.25]), marginwidth (Pixels [p.25]), noresize ("noresize"), scrolling ("yes" | "no" | "auto"), src (URI [p.25])	EMPTY
noframes	Common [p.25]	body
a&	target (CDATA [p.21])	n/a
area&	target (CDATA [p.21])	n/a

4.12. Iframe Module

The Iframe Module defines an element that can be used to define a base URL against which relative URIs in the document will be resolved. The element and attribute included in this module are:

Elements	Attributes	Minimal Content Model
iframe	Core [p.25] , frameborder ("1" | "0"), height (Pixels [p.25]), longdesc (URI [p.25]), marginheight (Pixels [p.25]), marginwidth (Pixels [p.25]), scrolling ("yes" | "no" | "auto"), src (URI [p.25]), width (Length [p.22])	Flow

When this module is used, the `iframe` element is added to the Block content set as defined by the Basic Text Module.

4.13. Intrinsic Events

Intrinsic events are attributes that are used in conjunction with elements that can have specific actions occur when certain events are performed by the user. The attributes indicated in the following table are added to the attribute set for their respective elements ONLY when the modules defining those elements are selected. Note also that selection of this module defines the attribute collection Events [p.25] as described above. Attributes defined by this module are:

Elements	Attributes	Notes
a&	onblur (Script [p.25]), onfocus (Script [p.25])	
area&	onblur (Script [p.25]), onfocus (Script [p.25])	When the Client-side Image Map module is also used
form&	onreset (Script [p.25]), onsubmit (Script [p.25])	When the Basic Forms or Forms module is used
body&	onload (Script [p.25]), onunload (Script [p.25])	
label&	onblur (Script [p.25]), onfocus (Script [p.25])	When the Forms module is used
input&	onblur (Script [p.25]), onchange (Script [p.25]), onfocus (Script [p.25]), onselect (Script [p.25])	When the Basic Forms or Forms module is used
select&	onblur (Script [p.25]), onchange (Script [p.25]), onfocus (Script [p.25])	When the Basic Forms or Forms module is used
textarea&	onblur (Script [p.25]), onchange (Script [p.25]), onfocus (Script [p.25]), onselect (Script [p.25])	When the Basic Forms or Forms module is used
button&	onblur (Script [p.25]), onfocus (Script [p.25])	When the Forms module is used

4.14. Metainformation Module

The Metainformation Module defines an element that describes information within the declarative portion of a document (in XHTML within the head element). This module includes the following element:

Elements	Attributes	Minimal Content Model
meta	I18N [p.25] , content (CDATA [p.21]), http-equiv (NMTOKEN [p.21]), name (NMTOKEN [p.21]), scheme (CDATA [p.21])	EMPTY

4.15. Scripting Module

The Scripting Module defines elements that are used to contain information pertaining to executable scripts or the lack of support for executable scripts. Elements and attributes included in this module are:

Elements	Attributes	Minimal Content Model
noscript	Common [p.25]	(Heading | List | Block)+
script	charset (Charset [p.21]), defer ("defer"), src (URI [p.25]), type (ContentType [p.21])	PCDATA

When this module is used, it adds the `script and noscript` elements are added to the Block content set of the Basic Text Module.

4.16. Stylesheet Module

The Stylesheet Module enables style sheet processing. Note also that selection of this module defines the attribute collection Style [p.25] as described above. The element and attributes defined by this module are:

Elements	Attributes	Minimal Content Model
style	I18N [p.25] , media (MediaDesc [p.24]), title (Text [p.25]), type (ContentType [p.21])	PCDATA

When this module is used, it adds the `style` element to the Block content set of the Basic Text Module.

4.17. Link Module

The Link Module defines an element that can be used to define links to external resources. These resources are often used to augment the user agent's ability to process the associated XHTML document. The element and attributes included in this module are:

Elements	Attributes	Minimal Content Model
link	Common [p.25] , charset (Charset [p.21]), href (URI [p.25]), hreflang (LanguageCode [p.22]), media (MediaDesc [p.24]), rel (LinkTypes [p.23]), rev (LinkTypes [p.23]), type (ContentType [p.21])	EMPTY

When this module is used, it adds the `link` element to the content model of the `head` element as defined in the Structure Module.

4.18. Base Module

The Base Module defines an element that can be used to define a base URL against which relative URIs in the document will be resolved. The element and attribute included in this module are:

Elements	Attributes	Minimal Content Model
base	href (URI [p.25])	EMPTY

When this module is used, it adds the `base` element to the content model of the `head` element of the Structure Module.

4.19. Legacy Module

The Legacy Module defines elements and attributes that were deprecated in previous versions of HTML and XHTML. While the use of these elements and attributes is no longer encouraged, they are provided in this module to ease their integration should markup language authors wish to support them.

Elements	Attributes	Minimal Content Model
font	Common [p.25]	Inline
s	Common [p.25]	Inline
strike	Common [p.25]	Inline
u	Common [p.25]	Inline

Elements	Attributes	Notes
body&	background (URI [p.25]), bgcolor (Color [p.21]), text (Color [p.21]), link (Color [p.21]), vlink (Color [p.21]), alink (Color [p.21])	Inline
br&	clear ("left" | "all" | "right" | "none")	Inline
strike	Common [p.25]	Inline
u	Common [p.25]	Inline

When this module is used, it adds the `base` element to the content model of the `head` element of the Structure Module.

A. XHTML 1.1 Schema Module Implementations

This appendix is *normative*.

This appendix will contain implementations of the modules defined in XHTML Abstract Modules [p.19] via XML Schema [XMLSCHEMA] [p.116] when the XML Schema becomes a W3C approved recommendation.

B. XHTML 1.1 DTD Module Implementations

This appendix is *normative*.

This appendix will contain implementations of the modules defined in XHTML Abstract Modules [p.19] via XML DTDs. These module implementations can be used by XHTML Family Document Types.

B.1. XHTML Character Entities

XHTML DTDs make available a standard collection of named character entities. Those entities are defined in this section.

B.1.1. XHTML Latin 1 Character Entities

```
<!-- ..................................................................... -->
<!-- XML-compatible ISO Latin 1 Character Entity Set for XHTML ............ -->
<!-- file: xhtml-lat1.ent

     Typical invocation:

       <!ENTITY % xhtml-lat1
           PUBLIC "-//W3C//ENTITIES Latin 1 for XHTML//EN"
                  "xhtml-lat1.ent" >
       %xhtml-lat1;

     Revision:  $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     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 -->
<!-- end of xhtml-lat1.ent -->
```

B.1.2. XHTML Special Characters

```
<!-- ..................................................................... -->
<!-- XML-compatible ISO Special Character Entity Set for XHTML ............ -->
<!-- file: xhtml-lat1.ent

     Typical invocation:

       <!ENTITY % xhtml-special
           PUBLIC "-//W3C//ENTITIES Special for XHTML//EN"
                  "xhtml-special.ent" >
       %xhtml-special;

     Revision:  $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     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.
-->

<!-- Relevant ISO entity set is given unless names are newly introduced.
     New names (i.e., not in ISO 8879 list) do not clash with any
     existing ISO 8879 entity names. ISO 10646 character numbers
     are given for each character, in hex. CDATA values are decimal
     conversions of the ISO 10646 values and refer to the document
     character set. Names are Unicode 2.0 names.
-->

<!-- C0 Controls and Basic Latin -->
<!ENTITY quot    "&#34;" ><!-- quotation mark = APL quote, U+0022 ISOnum -->
<!ENTITY amp     "&#38;" ><!-- ampersand, U+0026 ISOnum -->
<!ENTITY lt      "&#60;" ><!-- less-than sign, U+003C ISOnum -->
<!ENTITY gt      "&#62;" ><!-- greater-than sign, U+003E ISOnum -->
```

```
<!-- 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 -->

<!-- end of xhtml-special.ent -->
```

B.1.3. XHTML Mathematical, Greek, and Symbolic Characters

```
<!-- ..................................................................... -->
<!-- ISO Math, Greek and Symbolic Character Entity Set for XHTML .......... -->
<!-- file: xhtml-lat1.ent

     Typical invocation:

       <!ENTITY % xhtml-symbol
           PUBLIC "-//W3C//ENTITIES Symbols for XHTML//EN"
                  "xhtml-symbol.ent" >
       %xhtml-symbol;

     Revision:  $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     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.
-->

<!-- Relevant ISO entity set is given unless names are newly introduced.
     New names (i.e., not in ISO 8879 list) do not clash with any
     existing ISO 8879 entity names. ISO 10646 character numbers
     are given for each character, in hex. CDATA values are decimal
     conversions of the ISO 10646 values and refer to the document
     character set. Names are Unicode 2.0 names.
-->

<!-- 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 -->

<!-- end of xhtml-symbol.ent -->
```

B.2. XHTML Modular Framework

In order to take advantage of the XHTML DTD Modules, DTD authors need to define the content model for their DTD. XHTML provides a variety of tools to ease this effort. They are defined in a set of support modules, instantiated by a main Framework module:

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Modular Framework Module  ..................................... -->
<!-- file: xhtml11-framework-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Modular Framework 1.0//EN"
     SYSTEM "xhtml11-framework-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Modular Framework

     This required module instantiates the modules needed
     to support the XHTML modularization model, including:

        +  datatypes
        +  common attributes
        +  document model
        +  character entities

     The Intrinsic Events module is ignored by default but
     occurs in this module because it must be instantiated
     prior to Attributes but after Datatypes.
-->

<!ENTITY % xhtml-datatypes.module "INCLUDE" >
<![%xhtml-datatypes.module;[
<!ENTITY % xhtml-datatypes.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Datatypes 1.0//EN"
            "xhtml11-datatypes-1.mod" >
%xhtml-datatypes.mod;]]>

<!ENTITY % xhtml-events.module "IGNORE" >
<![%xhtml-events.module;[
<!ENTITY % xhtml-events.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Intrinsic Events 1.0//EN"
```

```
            "xhtml11-events-1.mod" >
%xhtml-events.mod;]]>

<!ENTITY % xhtml-attribs.module "INCLUDE" >
<![%xhtml-attribs.module;[
<!ENTITY % xhtml-attribs.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Common Attributes 1.0//EN"
            "xhtml11-attribs-1.mod" >
%xhtml-attribs.mod;]]>

<!ENTITY % xhtml-model.module "INCLUDE" >
<![%xhtml-model.module;[
<!-- A content model MUST be defined by the driver file -->
%xhtml-model.mod;]]>

<!ENTITY % xhtml-charent.module "INCLUDE" >
<![%xhtml-charent.module;[
<!ENTITY % xhtml-charent.mod
     PUBLIC "-//W3C//ENTITIES XHTML Character Entities 1.0//EN"
            "xhtml-charent-1.mod" >
%xhtml-charent.mod;]]>

<!-- end of xhtml11-framework-1.mod -->
```

Note that the module above references a content model module. This module is defined on a per-document type basis in addition to the document type driver file. The Modular framework also relies upon the following component modules:

B.2.1. XHTML Datatypes

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Datatypes Module  ..................................... -->
<!-- file: xhtml11-datatypes-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Datatypes 1.0//EN"
     SYSTEM "xhtml11-datatypes-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Datatypes

     defines containers for the following datatypes, many of
     these imported from other specifications and standards.
-->

<!-- Length defined for cellpadding/cellspacing -->

<!-- nn for pixels or nn% for percentage length -->
```

```
<!ENTITY % Length.datatype "CDATA" >

<!-- space-separated list of link types -->
<!ENTITY % LinkTypes.datatype "NMTOKENS" >

<!-- single or comma-separated list of media descriptors -->
<!ENTITY % MediaDesc.datatype "CDATA" >

<!-- pixel, percentage, or relative -->
<!ENTITY % MultiLength.datatype "CDATA" >

<!-- one or more digits (NUMBER) -->
<!ENTITY % Number.datatype "CDATA" >

<!-- integer representing length in pixels -->
<!ENTITY % Pixels.datatype "CDATA" >

<!-- script expression -->
<!ENTITY % Script.datatype "CDATA" >

<!-- textual content -->
<!ENTITY % Text.datatype "CDATA" >

<!-- Imported Datatypes ................................ -->

<!-- a single character from [ISO10646] -->
<!ENTITY % Character.datatype "CDATA" >

<!-- a character encoding, as per [RFC2045] -->
<!ENTITY % Charset.datatype "CDATA" >

<!-- a space separated list of character encodings, as per [RFC2045] -->
<!ENTITY % Charsets.datatype "CDATA" >

<!-- media type, as per [RFC2045] -->
<!ENTITY % ContentType.datatype "CDATA" >

<!-- comma-separated list of media types, as per [RFC2045] -->
<!ENTITY % ContentTypes.datatype "CDATA" >

<!-- date and time information. ISO date format -->
<!ENTITY % Datetime.datatype "CDATA" >

<!-- a language code, as per [RFC1766] -->
<!ENTITY % LanguageCode.datatype "NMTOKEN" >

<!-- a Uniform Resource Identifier, see [URI] -->
<!ENTITY % URI.datatype "CDATA" >

<!-- a list of Uniform Resource Identifiers, see [URI] -->
<!ENTITY % URIs.datatype "CDATA" >

<!-- end of xhtml11-datatypes-1.mod -->
```

B.2.2. XHTML Common Attribute Definitions

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Common Attributes Module  ................................ -->
<!-- file: xhtml11-attribs-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Common Attributes 1.0//EN"
     SYSTEM "xhtml11-attribs-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Common Attributes

     This module declares many of the common attributes for the Strict DTD.
-->

<!ENTITY % Id.attrib
     "id           ID                      #IMPLIED"
>

<!ENTITY % Class.attrib
     "class        CDATA                   #IMPLIED"
>

<!ENTITY % Title.attrib
     "title        %Text.datatype;         #IMPLIED"
>

<!ENTITY % Core.extra.attrib "" >

<!ENTITY % Core.attrib
     "%Id.attrib;
      %Class.attrib;
      %Title.attrib;
      %Core.extra.attrib;"
>

<![%XHTML.I18n;[
<!ENTITY % I18n.attrib
     "xml:lang     %LanguageCode.datatype;  #IMPLIED
      dir          ( ltr | rtl )            #IMPLIED"
>
]]>
<!ENTITY % I18n.attrib "" >

<!-- intrinsic event attributes declared previously
-->
<!ENTITY % Events.attrib "" >
```

```
<!ENTITY % Common.attrib
     "%Core.attrib;
      %I18n.attrib;
      %Events.attrib;"
>

<!-- end of xhtml11-attribs-1.mod -->
```

B.2.3. XHTML Character Entities

```
<!-- ..................................................................... -->
<!-- XHTML Character Entities Module  ..................................... -->
<!-- file: xhtml-charent-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ENTITIES XHTML Character Entities 1.0//EN"
     SYSTEM "xhtml-charent-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Character Entities for XHTML

     This module declares the set of character entities for XHTML,
     including the Latin 1, Symbol and Special characters sets.
-->

<!ENTITY % xhtml-lat1
    PUBLIC "-//W3C//ENTITIES Latin 1 for XHTML//EN"
           "http://www.w3.org/TR/xhtml11/DTD/xhtml-lat1.ent" >
<!ENTITY % xhtml-symbol
    PUBLIC "-//W3C//ENTITIES Symbols for XHTML//EN"
           "http://www.w3.org/TR/xhtml11/DTD/xhtml-symbol.ent" >
<!ENTITY % xhtml-special
    PUBLIC "-//W3C//ENTITIES Special for XHTML//EN"
           "http://www.w3.org/TR/xhtml11/DTD/xhtml-special.ent" >
%xhtml-lat1;
%xhtml-symbol;
%xhtml-special;

<!-- end of xhtml-charent-1.mod -->
```

B.3. XHTML Module Implementations

This section contains the formal definition of each of the XHTML Abstract Modules as a DTD module.

B.3.1. XHTML Basic Modules

B.3.1.1. Structure

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Structure Module  ..................................... -->
<!-- file: xhtml11-struct-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Document Structure 1.0//EN"
     SYSTEM "xhtml11-struct-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Document Structure

        title, head, body, html

     The Structure Module defines the major structural elements
     and their attributes.

     Note that the content model of the head element type is
     redeclared when the Base Module is included in the DTD.
-->

<!-- title: Document Title ........................... -->

<!-- The title element is not considered part of the flow of text.
     It should be displayed, for example as the page header or
     window title. Exactly one title is required per document.
-->

<!ENTITY % Title.element  "INCLUDE" >
<![%Title.element;[
<!ENTITY % Title.content  "( #PCDATA )" >
<!ELEMENT title  %Title.content; >
<!-- end of Title.element -->]]>

<!ENTITY % Title.attlist  "INCLUDE" >
<![%Title.attlist;[
<!ATTLIST title
      %I18n.attrib;
>
<!-- end of Title.attlist -->]]>

<!-- head: Document Head ............................. -->

<!ENTITY % Head.element  "INCLUDE" >
<![%Head.element;[
```

```
<!ENTITY % Head.content
    "( %Head-opts.mix;, title, %Head-opts.mix; )"
>
<!ELEMENT head  %Head.content; >
<!-- end of Head.element -->]]>

<!ENTITY % Head.attlist  "INCLUDE" >
<![%Head.attlist;[
<!-- reserved for future use with document profiles -->
<!ENTITY % Profile.attrib
     "profile      %URI.datatype;           '%XHTML.profile;'"
>

<!ATTLIST head
      %I18n.attrib;
      %Profile.attrib;
>
<!-- end of Head.attlist -->]]>

<!-- body: Document Body ............................. -->

<!ENTITY % Body.element  "INCLUDE" >
<![%Body.element;[
<!ENTITY % Body.content
     "( %Block.mix; )+"
>
<!ELEMENT body  %Body.content; >
<!-- end of Body.element -->]]>

<!ENTITY % Body.attlist  "INCLUDE" >
<![%Body.attlist;[
<!ATTLIST body
      %Common.attrib;
>
<!-- end of Body.attlist -->]]>

<!-- html: XHTML Document Element ..................... -->

<!ENTITY % Html.element  "INCLUDE" >
<![%Html.element;[
<!ENTITY % Html.content  "( head, body )" >
<!ELEMENT html  %Html.content; >
<!-- end of Html.element -->]]>

<!ENTITY % Html.attlist  "INCLUDE" >
<![%Html.attlist;[
<!-- version attribute value defined in driver
-->
<!ENTITY % XHTML.version.attrib
     "version      CDATA                    #FIXED '%XHTML.version;'"
>

<!-- namespace attribute value for this document type defined in driver
-->
<!ENTITY % Xmlns.attrib
     "xmlns        %URI.datatype;           #FIXED '%XHTML.ns;'"
>
```

```
<!-- XLink namespace attribute
-->
<!ENTITY % XLinkns.attrib
     "xmlns:xlink  %URI.datatype;           #FIXED '%XLINK.ns;'"
>

<!ATTLIST html
      %I18n.attrib;
      %XHTML.version.attrib;
      %Xmlns.attrib;
      %XLinkns.attrib;
>
<!-- end of Html.attlist -->]]>

<!-- end of xhtml11-struct-1.mod -->
```

B.3.1.2. Basic Text

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Basic Text Module  ..................................... -->
<!-- file: xhtml11-text-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Basic Text 1.0//EN"
     SYSTEM "xhtml11-text-1.mod"

     Revisions:
     ..................................................................... -->

<!-- Textual Content

     The Text module includes declarations for all basic
     text container elements and their attributes.
-->

<!ENTITY % xhtml-inlstruct.module "INCLUDE" >
<![%xhtml-inlstruct.module;[
<!ENTITY % xhtml-inlstruct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Structural 1.0//EN"
            "xhtml11-inlstruct-1.mod" >
%xhtml-inlstruct.mod;]]>

<!ENTITY % xhtml-inlphras.module "INCLUDE" >
<![%xhtml-inlphras.module;[
<!ENTITY % xhtml-inlphras.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Phrasal 1.0//EN"
            "xhtml11-inlphras-1.mod" >
%xhtml-inlphras.mod;]]>

<!ENTITY % xhtml-blkstruct.module "INCLUDE" >
<![%xhtml-blkstruct.module;[
```

```
<!ENTITY % xhtml-blkstruct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Block Structural 1.0//EN"
            "xhtml11-blkstruct-1.mod" >
%xhtml-blkstruct.mod;]]>

<!ENTITY % xhtml-blkphras.module "INCLUDE" >
<![%xhtml-blkphras.module;[
<!ENTITY % xhtml-blkphras.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Block Phrasal 1.0//EN"
            "xhtml11-blkphras-1.mod" >
%xhtml-blkphras.mod;]]>

<!-- end of xhtml11-text-1.mod -->
```

B.3.1.3. Hypertext

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Hypertext Module  ....................................... -->
<!-- file: xhtml11-hypertext-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Hypertext 1.0//EN"
     SYSTEM "xhtml11-hypertext-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Hypertext

        a

     This module declares the anchor ('a') element type, which
     defines the source of a hypertext link. The destination
     (or link 'target') is identified via its 'id' attribute
     rather than the 'name' attribute as was used in HTML.
-->

<!-- .............  Anchor Element  ............. -->

<!ENTITY % A.element  "INCLUDE" >
<![%A.element;[
<!ENTITY % A.content
     "( #PCDATA | %Inline-noa.mix; )*"
>
<!ELEMENT a  %A.content; >
<!-- end of A.element -->]]>

<!ENTITY % A.attlist  "INCLUDE" >
<![%A.attlist;[
<!ATTLIST a
      %Common.attrib;
```

```
      href         %URI.datatype;           #IMPLIED
      charset      %Charset.datatype;       #IMPLIED
      type         %ContentType.datatype;   #IMPLIED
      hreflang     %LanguageCode.datatype;  #IMPLIED
      rel          %LinkTypes.datatype;     #IMPLIED
      rev          %LinkTypes.datatype;     #IMPLIED
      accesskey    %Character.datatype;     #IMPLIED
      tabindex     %Number.datatype;        #IMPLIED
>
<!-- end of A.attlist -->]]>

<!-- end of xhtml11-hypertext-1.mod -->
```

B.3.1.4. Lists

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Lists Module  ............................................. -->
<!-- file: xhtml11-list-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Lists 1.0//EN"
     SYSTEM "xhtml11-list-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Lists

        dl, dt, dd, ol, ul, li

     This module declares the list-oriented element types
     and their attributes.
-->

<!-- dl: Definition List ............................... -->

<!ENTITY % Dl.element  "INCLUDE" >
<![%Dl.element;[
<!ENTITY % Dl.content  "( dt | dd )+" >
<!ELEMENT dl  %Dl.content; >
<!-- end of Dl.element -->]]>

<!ENTITY % Dl.attlist  "INCLUDE" >
<![%Dl.attlist;[
<!ATTLIST dl
      %Common.attrib;
>
<!-- end of Dl.attlist -->]]>

<!-- dt: Definition Term ............................... -->
```

```
<!ENTITY % Dt.element  "INCLUDE" >
<![%Dt.element;[
<!ENTITY % Dt.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT dt  %Dt.content; >
<!-- end of Dt.element -->]]>

<!ENTITY % Dt.attlist  "INCLUDE" >
<![%Dt.attlist;[
<!ATTLIST dt
      %Common.attrib;
>
<!-- end of Dt.attlist -->]]>

<!-- dd: Definition Description ..... -->

<!ENTITY % Dd.element  "INCLUDE" >
<![%Dd.element;[
<!ENTITY % Dd.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT dd  %Dd.content; >
<!-- end of Dd.element -->]]>

<!ENTITY % Dd.attlist  "INCLUDE" >
<![%Dd.attlist;[
<!ATTLIST dd
      %Common.attrib;
>
<!-- end of Dd.attlist -->]]>

<!-- ol: Ordered List (numbered styles) ..... -->

<!ENTITY % Ol.element  "INCLUDE" >
<![%Ol.element;[
<!ENTITY % Ol.content  "( li )+" >
<!ELEMENT ol  %Ol.content; >
<!-- end of Ol.element -->]]>

<!ENTITY % Ol.attlist  "INCLUDE" >
<![%Ol.attlist;[
<!ATTLIST ol
      %Common.attrib;
>
<!-- end of Ol.attlist -->]]>

<!-- ul: Unordered List (bullet styles) ..... -->

<!ENTITY % Ul.element  "INCLUDE" >
<![%Ul.element;[
<!ENTITY % Ul.content  "( li )+" >
<!ELEMENT ul  %Ul.content; >
<!-- end of Ul.element -->]]>

<!ENTITY % Ul.attlist  "INCLUDE" >
<![%Ul.attlist;[
```

```
<!ATTLIST ul
      %Common.attrib;
>
<!-- end of Ul.attlist -->]]>

<!-- li: List Item ..................................... -->

<!ENTITY % Li.element  "INCLUDE" >
<![%Li.element;[
<!ENTITY % Li.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT li  %Li.content; >
<!-- end of Li.element -->]]>

<!ENTITY % Li.attlist  "INCLUDE" >
<![%Li.attlist;[
<!ATTLIST li
      %Common.attrib;
>
<!-- end of Li.attlist -->]]>

<!-- end of xhtml11-list-1.mod -->
```

B.3.2. Applet

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Java Applet Module  ....................................... -->
<!-- file: xhtml11-applet-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Java Applets 1.0//EN"
     SYSTEM "xhtml11-applet-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Java Applets

        applet

     This module declares the applet element type and its attributes,
     used to provide support for Java applets. The 'alt' attribute
     is now required (as it is on images). One of either code or
     object attributes must be present. In the document, place param
     elements before the object elements that require their content.

     Note that use of this module requires instantiation of the
     Param Element Module prior to this module.
-->
```

```
<!-- applet: Java Applet ............................... -->

<!ENTITY % Applet.element  "INCLUDE" >
<![%Applet.element;[
<!ENTITY % Applet.content
     "( param | %Flow.mix; )*"
>
<!ELEMENT applet  %Applet.content; >
<!-- end of Applet.element -->]]>

<!ENTITY % Applet.attlist  "INCLUDE" >
<![%Applet.attlist;[
<!ATTLIST applet
      %Core.attrib;
      alt          %Text.datatype;          #REQUIRED
      name         CDATA                    #IMPLIED
      archive      CDATA                    #IMPLIED
      code         CDATA                    #IMPLIED
      codebase     %URI.datatype;           #IMPLIED
      object       CDATA                    #IMPLIED
      width        %Length.datatype;        #REQUIRED
      height       %Length.datatype;        #REQUIRED
>
<!-- end of Applet.attlist -->]]>

<!-- end of xhtml11-applet-1.mod -->
```

B.3.3. Text Modules

B.3.3.1. Presentation

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Presentation Module ....................................... -->
<!-- file: xhtml11-pres-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Presentation 1.0//EN"
     SYSTEM "xhtml11-pres-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Presentational Elements

     This module defines elements and their attributes for
     simple presentation-related markup.
-->

<!ENTITY % xhtml-inlpres.module "INCLUDE" >
```

```
<![%xhtml-inlpres.module;[
<!ENTITY % xhtml-inlpres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Presentation 1.0//EN"
            "xhtml11-inlpres-1.mod" >
%xhtml-inlpres.mod;]]>

<!ENTITY % xhtml-blkpres.module "INCLUDE" >
<![%xhtml-blkpres.module;[
<!ENTITY % xhtml-blkpres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Block Presentation 1.0//EN"
            "xhtml11-blkpres-1.mod" >
%xhtml-blkpres.mod;]]>

<!-- end of xhtml11-pres-1.mod -->
```

B.3.3.2. Edit

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Editing Elements Module  ..................................... -->
<!-- file: xhtml11-edit-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Editing Markup 1.0//EN"
     SYSTEM "xhtml11-edit-1.mod"

     Revisions:
     ....................................................................... -->

<!-- Editing Elements

        ins, del

     This module declares element types and attributes used to indicate
     inserted and deleted content while editing a document.
-->

<!ENTITY % Ins.element  "INCLUDE" >
<![%Ins.element;[
<!ENTITY % Ins.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT ins  %Ins.content; >
<!-- end of Ins.element -->]]>

<!ENTITY % Ins.attlist  "INCLUDE" >
<![%Ins.attlist;[
<!ATTLIST ins
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED
      datetime     %Datetime.datatype;      #IMPLIED
>
<!-- end of Ins.attlist -->]]>
```

```
<!ENTITY % Del.element  "INCLUDE" >
<![%Del.element;[
<!ENTITY % Del.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT del  %Del.content; >
<!-- end of Del.element -->]]>

<!ENTITY % Del.attlist  "INCLUDE" >
<![%Del.attlist;[
<!ATTLIST del
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED
      datetime     %Datetime.datatype;      #IMPLIED
>
<!-- end of Del.attlist -->]]>

<!-- end of xhtml11-edit-1.mod -->
```

B.3.3.3. Bi-Directional Override Elements

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 BDO Element Module ......................................... -->
<!-- file: xhtml11-bdo-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 BDO Element 1.0//EN"
     SYSTEM "xhtml11-bdo-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Bidirectional Override (bdo) Element

     This modules declares the element 'bdo', used to override the
     Unicode bidirectional algorithm for selected fragments of text.

     DEPENDENCIES:
     Relies on the conditional section keyword %XHTML.I18n; declared
     as "INCLUDE" and the parameter entity %I18n.attrib; declared
     in the common attributes module. Internationalization support
     includes both the bdo element and the i18n attributes.
-->

<!ENTITY % Bdo.element  "INCLUDE" >
<![%Bdo.element;[
<!ENTITY % Bdo.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT bdo  %Bdo.content; >
```

```
<!-- end of Bdo.element -->]]>

<!ENTITY % Bdo.attlist  "INCLUDE" >
<![%Bdo.attlist;[
<!ATTLIST bdo
      %Core.attrib;
      xml:lang     %LanguageCode.datatype;  #IMPLIED
      dir          ( ltr | rtl )            #REQUIRED
>
]]>

<!-- end of xhtml11-bdo-1.mod -->
```

B.3.4. Forms

B.3.4.1. Basic Forms

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Simplified Forms Module  ..................................... -->
<!-- file: xhtml11-formb-1.mod

     This is a subset of XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Basic Forms 1.0//EN"
     SYSTEM "xhtml11-formb-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Basic Forms

     This forms module is based on the HTML 3.2 forms model, with
     the WAI-requested addition of the label element. While this
     module essentially mimics the content model and attributes of
     HTML 3.2 forms, the element types declared herein also include
     all HTML 4.0 common attributes.

        form, label, input, select, option, textarea

     When Basic Forms are used in lieu of the default XHTML 1.1
     Forms module, the parameter entity %Inline.extra; must
     be redeclared in the model module to include:

        "| input | select | textarea | label"
-->

<!-- %Block-noform.mix; includes all non-form block elements,
     plus %Misc.class;
-->
<!ENTITY % Block-noform.mix
     "%Heading.class;
```

```
      | %List.class;
      | %Blkstruct.class;
      %Blkphras.class;
      %Blkpres.class;
      | table
      %Misc.class;"
>

<!-- form: Form Element ............................... -->

<!ENTITY % Form.element  "INCLUDE" >
<![%Form.element;[
<!ENTITY % Form.content
     "( %Block-noform.mix; )+"
>
<!ELEMENT form  %Form.content; >
<!-- end of Form.element -->]]>

<!ENTITY % Form.attlist  "INCLUDE" >
<![%Form.attlist;[
<!ATTLIST form
      %Common.attrib;
      action       %URI.datatype;           #REQUIRED
      method       ( get | post )           'get'
      enctype      %ContentType.datatype;   'application/x-www-form-urlencoded'
>
<!-- end of Form.attlist -->]]>

<!-- label: Form Field Label Text ..................... -->

<!ENTITY % Label.element  "INCLUDE" >
<![%Label.element;[
<!-- Each label must not contain more than ONE field
-->
<!ENTITY % Label.content
     "( #PCDATA
      | input | select | textarea | button
      | %Inlstruct.class;
      %Inlphras.class;
      %I18n.class;
      %Inlpres.class;
      %Inlspecial.class;
      %Misc.class; )*"
>
<!ELEMENT label  %Label.content; >
<!-- end of Label.element -->]]>

<!ENTITY % Label.attlist  "INCLUDE" >
<![%Label.attlist;[
<!ATTLIST label
      %Common.attrib;
      for          IDREF                    #IMPLIED
      accesskey    %Character.datatype;     #IMPLIED
>
<!-- end of Label.attlist -->]]>

<!-- input: Form Control .............................. -->
```

```
<!ENTITY % Input.element  "INCLUDE" >
<![%Input.element;[
<!ENTITY % Input.content  "EMPTY" >
<!ELEMENT input  %Input.content; >
<!-- end of Input.element -->]]>

<!ENTITY % Input.attlist  "INCLUDE" >
<![%Input.attlist;[
<!ENTITY % InputType.class
     "( text | password | checkbox | radio | submit
      | reset | file | hidden | image )"
>
<!-- attribute name required for all but submit & reset
-->
<!ATTLIST input
      %Common.attrib;
      type         %InputType.class;        'text'
      name         CDATA                    #IMPLIED
      value        CDATA                    #IMPLIED
      checked      ( checked )              #IMPLIED
      size         CDATA                    #IMPLIED
      maxlength    %Number.datatype;        #IMPLIED
      src          %URI.datatype;           #IMPLIED
>
<!-- end of Input.attlist -->]]>

<!-- select: Option Selector ......................... -->

<!ENTITY % Select.element  "INCLUDE" >
<![%Select.element;[
<!ENTITY % Select.content  "( option )+" >
<!ELEMENT select  %Select.content; >
<!-- end of Select.element -->]]>

<!ENTITY % Select.attlist  "INCLUDE" >
<![%Select.attlist;[
<!ATTLIST select
      %Common.attrib;
      name         CDATA                    #IMPLIED
      size         %Number.datatype;        #IMPLIED
      multiple     ( multiple )             #IMPLIED
>
<!-- end of Select.attlist -->]]>

<!-- option: Selectable Choice ....................... -->

<!ENTITY % Option.element  "INCLUDE" >
<![%Option.element;[
<!ENTITY % Option.content  "( #PCDATA )" >
<!ELEMENT option  %Option.content; >
<!-- end of Option.element -->]]>

<!ENTITY % Option.attlist  "INCLUDE" >
<![%Option.attlist;[
<!ATTLIST option
      %Common.attrib;
```

```
      selected     ( selected )              #IMPLIED
      value        CDATA                     #IMPLIED
>
<!-- end of Option.attlist -->]]>

<!-- textarea: Multi-Line Text Field ................... -->

<!ENTITY % Textarea.element  "INCLUDE" >
<![%Textarea.element;[
<!ENTITY % Textarea.content  "( #PCDATA )" >
<!ELEMENT textarea  %Textarea.content; >
<!-- end of Textarea.element -->]]>

<!ENTITY % Textarea.attlist  "INCLUDE" >
<![%Textarea.attlist;[
<!ATTLIST textarea
      %Common.attrib;
      name         CDATA                     #IMPLIED
      rows         %Number.datatype;         #REQUIRED
      cols         %Number.datatype;         #REQUIRED
>
<!-- end of Textarea.attlist -->]]>

<!-- end of xhtml11-formb-1.mod -->
```

B.3.4.2. Forms

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Extended Forms 1.0 Module  ................................ -->
<!-- file: xhtml11-formx-1.mod

     This is an extension of XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Extended Forms 1.0//EN"
     SYSTEM "xhtml11-formx-1.mod"

     Revisions:
     ..................................................................... -->

<!-- C.7.15. Extended Forms Version 1.0

     [Any description here.]
-->

<!-- TBD -->

<!-- end of xhtml11-formx-1.mod -->
```

B.3.5. Tables

B.3.5.1. Basic Tables

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Basic Table Module  ....................................... -->
<!-- file: xhtml11-tableb-1.mod

     This module subsets XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Basic Tables 1.0//EN"
     SYSTEM "xhtml11-tableb-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Basic Tables

        table, caption, tr, th, td

     This table module declares elements and attributes defining
     a table model based fundamentally on features found in the
     widely-deployed HTML 3.2 table model.  While this module
     mimics the content model and table attributes of HTML 3.2
     tables, the element types declared herein also includes all
     HTML 4.0 common and most of the HTML 4.0 table attributes.
-->

<!-- The border attribute sets the thickness of the frame
     around the table. The default units are screen pixels.
-->

<!-- horizontal alignment attributes for cell contents
-->
<!ENTITY % CellHAlign.attrib
     "align        ( left
                   | center
                   | right )               #IMPLIED"
>

<!-- vertical alignment attributes for cell contents
-->
<!ENTITY % CellVAlign.attrib
     "valign       ( top
                   | middle
                   | bottom )              #IMPLIED"
>

<!-- scope is simpler than axes attribute for common tables
-->
<!ENTITY % Scope.attrib
```

```
     "scope        ( row
                   | col
                   | rowgroup
                   | colgroup )             #IMPLIED"
>

<!-- table: Table Element ............................. -->

<!ENTITY % Table.element  "INCLUDE" >
<![%Table.element;[
<!ENTITY % Table.content
     "( caption?, tr+ )"
>
<!ELEMENT table  %Table.content; >
<!-- end of Table.element -->]]>

<!ENTITY % Table.attlist  "INCLUDE" >
<![%Table.attlist;[
<!ATTLIST table
      %Common.attrib;
      width        %Length.datatype;        #IMPLIED
      border       %Pixels.datatype;        #IMPLIED
      cellspacing  %Length.datatype;        #IMPLIED
      cellpadding  %Length.datatype;        #IMPLIED
>
<!-- end of Table.attlist -->]]>

<!-- caption: Table Caption ........................... -->

<!ENTITY % Caption.element  "INCLUDE" >
<![%Caption.element;[
<!ENTITY % Caption.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT caption  %Caption.content; >
<!-- end of Caption.element -->]]>

<!ENTITY % Caption.attlist  "INCLUDE" >
<![%Caption.attlist;[
<!ENTITY % CaptionAlign.attrib
     "align        ( top | bottom )         #IMPLIED"
>

<!ATTLIST caption
      %Common.attrib;
      %CaptionAlign.attrib;
>
<!-- end of Caption.attlist -->]]>

<!-- tr: Table Row .................................... -->

<!ENTITY % Tr.element  "INCLUDE" >
<![%Tr.element;[
<!ENTITY % Tr.content  "( th | td )+" >
<!ELEMENT tr  %Tr.content; >
<!-- end of Tr.element -->]]>
```

```
<!ENTITY % Tr.attlist  "INCLUDE" >
<![%Tr.attlist;[
<!ATTLIST tr
      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Tr.attlist -->]]>

<!-- th: Table Header Cell ............................ -->

<!-- th is for header cells, td for data,
     but for cells acting as both use td
-->

<!ENTITY % Th.element  "INCLUDE" >
<![%Th.element;[
<!ENTITY % Th.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT th  %Th.content; >
<!-- end of Th.element -->]]>

<!ENTITY % Th.attlist  "INCLUDE" >
<![%Th.attlist;[
<!ATTLIST th
      %Common.attrib;
      abbr         %Text.datatype;          #IMPLIED
      axis         CDATA                    #IMPLIED
      headers      IDREFS                   #IMPLIED
      %Scope.attrib;
      rowspan      %Number.datatype;        '1'
      colspan      %Number.datatype;        '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Th.attlist -->]]>

<!-- td: Table Data Cell .............................. -->

<!ENTITY % Td.element  "INCLUDE" >
<![%Td.element;[
<!ENTITY % Td.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT td  %Td.content; >
<!-- end of Td.element -->]]>

<!ENTITY % Td.attlist  "INCLUDE" >
<![%Td.attlist;[
<!ATTLIST td
      %Common.attrib;
      abbr         %Text.datatype;          #IMPLIED
      axis         CDATA                    #IMPLIED
      headers      IDREFS                   #IMPLIED
      %Scope.attrib;
      rowspan      %Number.datatype;        '1'
```

```
       colspan      %Number.datatype;        '1'
       %CellHAlign.attrib;
       %CellVAlign.attrib;
>
<!-- end of Td.attlist -->]]>

<!-- end of xhtml11-tableb-1.mod -->
```

B.3.5.2. Tables

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Table Module  ................................................ -->
<!-- file: xhtml11-table-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Tables 1.0//EN"
     SYSTEM "xhtml11-table-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Tables

        table, caption, thead, tfoot, tbody, colgroup, col, tr, th, td

     This module declares element types and attributes used to provide
     table markup similar to HTML 4.0, including features that enable
     better accessibility for non-visual user agents.
-->

<!-- The border attribute sets the thickness of the frame around the
     table. The default units are screen pixels.

     The value "border" is included for backwards compatibility with
     <table border> which yields frame="border" and border="implied".
     For <table border="1"> you get border="1" and frame="implied".
     In this case, it is appropriate to treat this as frame="border"
     for backwards compatibility with deployed browsers.
-->

<!-- The frame attribute specifies which parts of the frame around
     the table should be rendered. The values are not the same as
     CALS to avoid a name clash with the valign attribute.
-->
<!ENTITY % TFrame.attrib
     "frame        ( void
                   | above
                   | below
                   | hsides
                   | lhs
                   | rhs
```

```
                    | vsides
                    | box
                    | border )               #IMPLIED"
>

<!-- The rules attribute defines which rules to draw between cells:

     If rules is absent then assume:

       "none" if border is absent or border="0" otherwise "all"
-->
<!ENTITY % TRules.attrib
     "rules         ( none
                    | groups
                    | rows
                    | cols
                    | all )                 #IMPLIED"
>

<!-- horizontal alignment attributes for cell contents
-->
<!ENTITY % CellHAlign.attrib
     "align         ( left
                    | center
                    | right
                    | justify
                    | char )                #IMPLIED
      char          %Character.datatype;     #IMPLIED
      charoff       %Length.datatype;        #IMPLIED"
>

<!-- vertical alignment attributes for cell contents
-->
<!ENTITY % CellVAlign.attrib
     "valign        ( top
                    | middle
                    | bottom
                    | baseline )            #IMPLIED"
>

<!-- scope is simpler than axes attribute for common tables
-->
<!ENTITY % Scope.attrib
     "scope         ( row
                    | col
                    | rowgroup
                    | colgroup )            #IMPLIED"
>

<!-- table: Table Element ........................... -->

<!ENTITY % Table.element  "INCLUDE" >
<![%Table.element;[
<!ENTITY % Table.content
     "( caption?, ( col* | colgroup* ),
      (( thead?, tfoot?, tbody+ ) | ( tr+ )))"
>
```

```
<!ELEMENT table  %Table.content; >
<!-- end of Table.element -->]]>

<!ENTITY % Table.attlist  "INCLUDE" >
<![%Table.attlist;[
<!ATTLIST table
      %Common.attrib;
      summary      %Text.datatype;          #IMPLIED
      width        %Length.datatype;        #IMPLIED
      border       %Pixels.datatype;        #IMPLIED
      %TFrame.attrib;
      %TRules.attrib;
      cellspacing  %Length.datatype;        #IMPLIED
      cellpadding  %Length.datatype;        #IMPLIED
      datapagesize CDATA                    #IMPLIED
>
<!-- end of Table.attlist -->]]>

<!-- caption: Table Caption ............................ -->

<!ENTITY % Caption.element  "INCLUDE" >
<![%Caption.element;[
<!ENTITY % Caption.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT caption  %Caption.content; >
<!-- end of Caption.element -->]]>

<!ENTITY % Caption.attlist  "INCLUDE" >
<![%Caption.attlist;[
<!ATTLIST caption
      %Common.attrib;
>
<!-- end of Caption.attlist -->]]>

<!-- thead: Table Header ............................... -->

<!-- Use thead to duplicate headers when breaking table
     across page boundaries, or for static headers when
     tbody sections are rendered in scrolling panel.
-->

<!ENTITY % Thead.element  "INCLUDE" >
<![%Thead.element;[
<!-- end of Thead.element -->]]>
<!ENTITY % Thead.content  "( tr )+" >
<!ELEMENT thead  %Thead.content; >

<!ENTITY % Thead.attlist  "INCLUDE" >
<![%Thead.attlist;[
<!ATTLIST thead
      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Thead.attlist -->]]>
```

```
<!-- tfoot: Table Footer ............................. -->

<!-- Use tfoot to duplicate footers when breaking table
     across page boundaries, or for static footers when
     tbody sections are rendered in scrolling panel.
-->

<!ENTITY % Tfoot.element  "INCLUDE" >
<![%Tfoot.element;[
<!ENTITY % Tfoot.content  "( tr )+" >
<!ELEMENT tfoot  %Tfoot.content; >
<!-- end of Tfoot.element -->]]>

<!ENTITY % Tfoot.attlist  "INCLUDE" >
<![%Tfoot.attlist;[
<!ATTLIST tfoot
      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Tfoot.attlist -->]]>

<!-- tbody: Table Body ............................... -->

<!-- Use multiple tbody sections when rules are needed
     between groups of table rows.
-->

<!ENTITY % Tbody.element  "INCLUDE" >
<![%Tbody.element;[
<!ENTITY % Tbody.content  "( tr )+" >
<!ELEMENT tbody  %Tbody.content; >
<!-- end of Tbody.element -->]]>

<!ENTITY % Tbody.attlist  "INCLUDE" >
<![%Tbody.attlist;[
<!ATTLIST tbody
      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Tbody.attlist -->]]>

<!-- colgroup: Table Column Group ..................... -->

<!-- colgroup groups a set of col elements. It allows you
     to group several semantically-related columns together.
-->

<!ENTITY % Colgroup.element  "INCLUDE" >
<![%Colgroup.element;[
<!ENTITY % Colgroup.content  "( col )*" >
<!ELEMENT colgroup  %Colgroup.content; >
<!-- end of Colgroup.element -->]]>

<!ENTITY % Colgroup.attlist  "INCLUDE" >
<![%Colgroup.attlist;[
```

```
<!-- end of Colgroup.attlist -->]]>
<!ATTLIST colgroup
      %Common.attrib;
      span         %Number.datatype;        '1'
      width        %MultiLength.datatype;   #IMPLIED
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!-- col: Table Column ............................... -->

<!-- col elements define the alignment properties for
     cells in one or more columns.

     The width attribute specifies the width of the
     columns, e.g.

       width="64"        width in screen pixels
       width="0.5*"      relative width of 0.5

     The span attribute causes the attributes of one
     col element to apply to more than one column.
-->

<!ENTITY % Col.element  "INCLUDE" >
<![%Col.element;[
<!ENTITY % Col.content  "EMPTY" >
<!ELEMENT col  %Col.content; >
<!-- end of Col.element -->]]>

<!ENTITY % Col.attlist  "INCLUDE" >
<![%Col.attlist;[
<!ATTLIST col
      %Common.attrib;
      span         %Number.datatype;        '1'
      width        %MultiLength.datatype;   #IMPLIED
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Col.attlist -->]]>

<!-- tr: Table Row ................................... -->

<!ENTITY % Tr.element  "INCLUDE" >
<![%Tr.element;[
<!ENTITY % Tr.content  "( th | td )+" >
<!ELEMENT tr  %Tr.content; >
<!-- end of Tr.element -->]]>

<!ENTITY % Tr.attlist  "INCLUDE" >
<![%Tr.attlist;[
<!ATTLIST tr
      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Tr.attlist -->]]>
```

```
<!-- th: Table Header Cell ............................ -->

<!-- th is for header cells, td for data,
     but for cells acting as both use td
-->

<!ENTITY % Th.element  "INCLUDE" >
<![%Th.element;[
<!ENTITY % Th.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT th  %Th.content; >
<!-- end of Th.element -->]]>

<!ENTITY % Th.attlist  "INCLUDE" >
<![%Th.attlist;[
<!ATTLIST th
      %Common.attrib;
      abbr         %Text.datatype;          #IMPLIED
      axis         CDATA                    #IMPLIED
      headers      IDREFS                   #IMPLIED
      %Scope.attrib;
      rowspan      %Number.datatype;        '1'
      colspan      %Number.datatype;        '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Th.attlist -->]]>

<!-- td: Table Data Cell .............................. -->

<!ENTITY % Td.element  "INCLUDE" >
<![%Td.element;[
<!ENTITY % Td.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT td  %Td.content; >
<!-- end of Td.element -->]]>

<!ENTITY % Td.attlist  "INCLUDE" >
<![%Td.attlist;[
<!ATTLIST td
      %Common.attrib;
      abbr         %Text.datatype;          #IMPLIED
      axis         CDATA                    #IMPLIED
      headers      IDREFS                   #IMPLIED
      %Scope.attrib;
      rowspan      %Number.datatype;        '1'
      colspan      %Number.datatype;        '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>
<!-- end of Td.attlist -->]]>

<!-- end of xhtml11-table-1.mod -->
```

B.3.6. Image

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Images Module  ............................................. -->
<!-- file: xhtml11-image-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Rovision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Images 1.0//EN"
     SYSTEM "xhtml11-image-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Images

        img

     This module provides markup to support basic image embedding.
-->

<!-- To avoid problems with text-only UAs as well as to make
     image content understandable and navigable to users of
     non-visual UAs, you need to provide a description with
     the 'alt' attribute, and avoid server-side image maps.
-->

<!ENTITY % Img.element  "INCLUDE" >
<![%Img.element;[
<!ENTITY % Img.content  "EMPTY" >
<!ELEMENT img  %Img.content; >
<!-- end of Img.element -->]]>

<!ENTITY % Img.attlist  "INCLUDE" >
<![%Img.attlist;[
<!ATTLIST img
      %Common.attrib;
      src          %URI.datatype;           #REQUIRED
      alt          %Text.datatype;          #REQUIRED
      longdesc     %URI.datatype;           #IMPLIED
      height       %Length.datatype;        #IMPLIED
      width        %Length.datatype;        #IMPLIED
>
<!-- end of Img.attlist -->]]>

<!-- end of xhtml11-image-1.mod -->
```

B.3.7. Client-side Image Map

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Client-side Image Map Module  ............................. -->
<!-- file: xhtml11-csismap-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Client-side Image Maps 1.0//EN"
     SYSTEM "xhtml11-csismap-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Client-side Image Maps

        area, map

     This module declares elements and attributes to support client-side
     image maps. This requires that the Image Module (or a module
     declaring the img element type) be included in the DTD.

     These can be placed in the same document or grouped in a
     separate document, although the latter isn't widely supported
-->

<!ENTITY % Area.element  "INCLUDE" >
<![%Area.element;[
<!ENTITY % Area.content  "EMPTY" >
<!ELEMENT area  %Area.content; >
<!-- end of Area.element -->]]>

<!ENTITY % Shape.datatype "( rect | circle | poly | default )">
<!ENTITY % Coords.datatype "CDATA" >

<!ENTITY % Area.attlist  "INCLUDE" >
<![%Area.attlist;[
<!ATTLIST area
      %Common.attrib;
      href         %URI.datatype;           #IMPLIED
      shape        %Shape.datatype;         'rect'
      coords       %Coords.datatype;        #IMPLIED
      nohref       ( nohref )               #IMPLIED
      alt          %Text.datatype;          #REQUIRED
      tabindex     %Number.datatype;        #IMPLIED
      accesskey    %Character.datatype;     #IMPLIED
>
<!-- end of Area.attlist -->]]>

<!-- modify anchor attribute definition list
     to allow for client-side image maps
```

```
-->
<!ATTLIST a
      shape        %Shape.datatype;         'rect'
      coords       %Coords.datatype;        #IMPLIED
>

<!-- modify img attribute definition list
     to allow for client-side image maps
-->
<!ATTLIST img
      usemap       IDREF                    #IMPLIED
>
<!-- 'usemap' points to the 'id' attribute of a MAP element,
     which must be in the same document; support for external
     document maps was not widely supported in HTML and is
     eliminated in XHTML.

     It is considered an error for the element pointed to by
     a usemap IDREF to occur in anything but a map element.
-->

<!ENTITY % Map.element  "INCLUDE" >
<![%Map.element;[
<!ENTITY % Map.content
     "(( %Block.mix; ) | area )+"
>
<!ELEMENT map  %Map.content; >
<!-- end of Map.element -->]]>

<!ENTITY % Map.attlist  "INCLUDE" >
<![%Map.attlist;[
<!ATTLIST map
      id           ID                       #REQUIRED
      %Class.attrib;
      %Title.attrib;
      %Core.extra.attrib;
      %I18n.attrib;
      %Events.attrib;
>
<!-- end of Map.attlist -->]]>

<!-- end of xhtml11-csismap-1.mod -->
```

B.3.8. Server-side Image Map

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Client-side Image Map Module  ............................... -->
<!-- file: xhtml11-csismap-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Client-side Image Maps 1.0//EN"
```

```
     SYSTEM "xhtml11-csismap-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Client-side Image Maps

        area, map

     This module declares elements and attributes to support client-side
     image maps. This requires that the Image Module (or a module
     declaring the img element type) be included in the DTD.

     These can be placed in the same document or grouped in a
     separate document, although the latter isn't widely supported
-->

<!ENTITY % Area.element  "INCLUDE" >
<![%Area.element;[
<!ENTITY % Area.content  "EMPTY" >
<!ELEMENT area  %Area.content; >
<!-- end of Area.element -->]]>

<!ENTITY % Shape.datatype "( rect | circle | poly | default )">
<!ENTITY % Coords.datatype "CDATA" >

<!ENTITY % Area.attlist  "INCLUDE" >
<![%Area.attlist;[
<!ATTLIST area
      %Common.attrib;
      href         %URI.datatype;           #IMPLIED
      shape        %Shape.datatype;         'rect'
      coords       %Coords.datatype;        #IMPLIED
      nohref       ( nohref )               #IMPLIED
      alt          %Text.datatype;          #REQUIRED
      tabindex     %Number.datatype;        #IMPLIED
      accesskey    %Character.datatype;     #IMPLIED
>
<!-- end of Area.attlist -->]]>

<!-- modify anchor attribute definition list
     to allow for client-side image maps
-->
<!ATTLIST a
      shape        %Shape.datatype;         'rect'
      coords       %Coords.datatype;        #IMPLIED
>

<!-- modify img attribute definition list
     to allow for client-side image maps
-->
<!ATTLIST img
      usemap       IDREF                    #IMPLIED
>
<!-- 'usemap' points to the 'id' attribute of a MAP element,
     which must be in the same document; support for external
```

```
     document maps was not widely supported in HTML and is
     eliminated in XHTML.

     It is considered an error for the element pointed to by
     a usemap IDREF to occur in anything but a map element.
-->

<!ENTITY % Map.element  "INCLUDE" >
<![%Map.element;[
<!ENTITY % Map.content
     "(( %Block.mix; ) | area )+"
>
<!ELEMENT map  %Map.content; >
<!-- end of Map.element -->]]>

<!ENTITY % Map.attlist  "INCLUDE" >
<![%Map.attlist;[
<!ATTLIST map
      id           ID                        #REQUIRED
      %Class.attrib;
      %Title.attrib;
      %Core.extra.attrib;
      %I18n.attrib;
      %Events.attrib;
>
<!-- end of Map.attlist -->]]>

<!-- end of xhtml11-csismap-1.mod -->
```

B.3.9. Object

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 External Inclusion Module  ................................. -->
<!-- file: xhtml11-object-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Object Element 1.0//EN"
     SYSTEM "xhtml11-object-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Embedded Objects

        object

     This module declares the object element type and its attributes,
     used to embed external objects as part of XHTML pages. In the
     document, place param elements prior to the object elements
     that require their content.
```

```
     Note that use of this module requires instantiation of the
     Param Element Module prior to this module.
-->

<!-- a space-separated list of Uniform Resource Identifiers, see [URI] -->
<!ENTITY % URIs.datatype "CDATA" >

<!-- object: Generic Embedded Object ................... -->

<!ENTITY % Object.element  "INCLUDE" >
<![%Object.element;[
<!ENTITY % Object.content
     "( #PCDATA | %Flow.mix; | param )*"
>
<!ELEMENT object  %Object.content; >
<!-- end of Object.element -->]]>

<!ENTITY % Object.attlist  "INCLUDE" >
<![%Object.attlist;[
<!ATTLIST object
      %Common.attrib;
      declare      ( declare )              #IMPLIED
      classid      %URI.datatype;           #IMPLIED
      codebase     %URI.datatype;           #IMPLIED
      data         %URI.datatype;           #IMPLIED
      type         %ContentType.datatype;   #IMPLIED
      codetype     %ContentType.datatype;   #IMPLIED
      archive      %URIs.datatype;          #IMPLIED
      standby      %Text.datatype;          #IMPLIED
      height       %Length.datatype;        #IMPLIED
      width        %Length.datatype;        #IMPLIED
      usemap       IDREF                    #IMPLIED
      name         CDATA                    #IMPLIED
      tabindex     %Number.datatype;        #IMPLIED
>
<!-- end of Object.attlist -->]]>

<!-- end of xhtml11-object-1.mod -->
```

B.3.10. Frames

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Frames Module  ..................................... -->
<!-- file: xhtml11-frames-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Frames//EN"
     SYSTEM "xhtml11-frames-1.mod"

     Revisions:
```

```
      (none)
      ..................................................................... -->

<!-- Frames

        frame, frameset, noframes

     This module declares frame-related element types and attributes.
-->

<!-- comma-separated list of MultiLength -->
<!ENTITY % MultiLengths.datatype "CDATA" >

<!-- render in this frame -->
<!ENTITY % FrameTarget.datatype "CDATA" >

<!-- The content model for XHTML documents depends on whether
     the <head> is followed by a <frameset> or <body> element.
-->

<!ENTITY % Frameset.content  "(( frameset | frame )+, noframes? )" >
<!ELEMENT frameset  %Frameset.content; >
<!ATTLIST frameset
      %Core.attrib;
      rows         %MultiLengths.datatype;  #IMPLIED
      cols         %MultiLengths.datatype;  #IMPLIED
>

<!-- reserved frame names start with "_" otherwise starts with letter -->

<!ENTITY % Frame.content  "EMPTY" >
<!ELEMENT frame  %Frame.content; >
<!ATTLIST frame
      %Core.attrib;
      longdesc     %URI.datatype;           #IMPLIED
      name         CDATA                    #IMPLIED
      src          %URI.datatype;           #IMPLIED
      frameborder  ( 1 | 0 )                '1'
      marginwidth  %Pixels.datatype;        #IMPLIED
      marginheight %Pixels.datatype;        #IMPLIED
      noresize     ( noresize )             #IMPLIED
      scrolling    ( yes | no | auto )      'auto'
>

<!-- changes to other declarations ..................... -->

<!-- redefine content model for html element,
     substituting frameset for body -->
<!ENTITY % Html.content "( head, frameset )" >

<!-- alternate content container for non frame-based rendering -->

<!ENTITY % Noframes.content "( body )">
<!ELEMENT noframes  %Noframes.content; >
<!ATTLIST noframes
      %Common.attrib;
>
```

```
<!-- add 'target' attribute to 'a' element -->
<!ATTLIST a
      target       %FrameTarget.datatype;   #IMPLIED
>

<!-- add 'target' attribute to 'area' element -->
<!ATTLIST area
      target       %FrameTarget.datatype;   #IMPLIED
>

<!-- add 'target' attribute to 'link' element -->
<!ATTLIST link
      target       %FrameTarget.datatype;   #IMPLIED
>

<!-- add 'target' attribute to 'form' element -->
<!ATTLIST form
      target       %FrameTarget.datatype;   #IMPLIED
>

<!-- add 'target' attribute to 'base' element -->
<!ATTLIST base
      target       %FrameTarget.datatype;   #IMPLIED
>

<!-- end of xhtml11-frames-1.mod -->
```

B.3.11. Iframe

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 IFrame Module  ............................................... -->
<!-- file: xhtml11-iframe-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Frame Element//EN"
     SYSTEM "xhtml11-iframe-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Inline Frames

        iframe

     This module declares the iframe element type and its attributes,
     used to create an inline frame within a document.
-->

<!-- Inline Frames ..................................... -->
```

```
<!ENTITY % Iframe.content  "( %Flow.mix; )*" >
<!ELEMENT iframe  %Iframe.content; >
<!ATTLIST iframe
      %Core.attrib;
      longdesc     %URI.datatype;           #IMPLIED
      name         CDATA                    #IMPLIED
      src          %URI.datatype;           #IMPLIED
      frameborder  ( 1 | 0 )                '1'
      marginwidth  %Pixels.datatype;        #IMPLIED
      marginheight %Pixels.datatype;        #IMPLIED
      scrolling    ( yes | no | auto )      'auto'
      height       %Length.datatype;        #IMPLIED
      width        %Length.datatype;        #IMPLIED
>

<!-- end of xhtml11-iframe-1.mod -->
```

B.3.12. Intrinsic Events

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Intrinsic Events Module  ..................................... -->
<!-- file: xhtml11-events-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ENTITIES XHTML 1.1 Intrinsic Events 1.0//EN"
     SYSTEM "xhtml11-events-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Intrinsic Event Attributes

     These are the event attributes defined in HTML 4.0, Section
     18.2.3 "Intrinsic Events". This module must be instantiated
     prior to the Attributes Module but after the Datatype Module
     in the Modular Framework module.

    "Note: Authors of HTML documents are advised that changes are
     likely to occur in the realm of intrinsic events (e.g., how
     scripts are bound to events). Research in this realm is carried
     on by members of the W3C Document Object Model Working Group
     (see the W3C Web site at http://www.w3.org/ for more information)."
-->

<!ENTITY % Events.attrib
     "onclick      %Script.datatype;        #IMPLIED
      ondblclick   %Script.datatype;        #IMPLIED
      onmousedown  %Script.datatype;        #IMPLIED
      onmouseup    %Script.datatype;        #IMPLIED
```

```
      onmouseover  %Script.datatype;        #IMPLIED
      onmousemove  %Script.datatype;        #IMPLIED
      onmouseout   %Script.datatype;        #IMPLIED
      onkeypress   %Script.datatype;        #IMPLIED
      onkeydown    %Script.datatype;        #IMPLIED
      onkeyup      %Script.datatype;        #IMPLIED"
>

<!-- additional attributes on anchor element
-->
<!ATTLIST a
     onfocus      %Script.datatype;         #IMPLIED
     onblur       %Script.datatype;         #IMPLIED
>

<!-- additional attributes on form element
-->
<!ATTLIST form
      onsubmit     %Script.datatype;        #IMPLIED
      onreset      %Script.datatype;        #IMPLIED
>

<!-- additional attributes on label element
-->
<!ATTLIST label
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
>

<!-- additional attributes on input element
-->
<!ATTLIST input
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
      onselect     %Script.datatype;        #IMPLIED
      onchange     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on select element
-->
<!ATTLIST select
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
      onchange     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on textarea element
-->
<!ATTLIST textarea
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
      onselect     %Script.datatype;        #IMPLIED
      onchange     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on button element
-->
```

```
<!ATTLIST button
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
>

<!-- additional attributes on body element
-->
<!ATTLIST body
      onload       %Script.datatype;        #IMPLIED
      onunload     %Script.datatype;        #IMPLIED
>

<!-- additional attributes on area element
-->
<!ATTLIST area
      onfocus      %Script.datatype;        #IMPLIED
      onblur       %Script.datatype;        #IMPLIED
>

<!-- end of xhtml11-events-1.mod -->
```

B.3.13. Metainformation

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Document Metainformation Module  ........................... -->
<!-- file: xhtml11-meta-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Metainformation 1.0//EN"
     SYSTEM "xhtml11-meta-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Meta Information

        meta

     This module declares the meta element type and its attributes,
     used to provide declarative document metainformation.
-->

<!-- meta: Generic Metainformation ..................... -->

<!ENTITY % Meta.element  "INCLUDE" >
<![%Meta.element;[
<!ENTITY % Meta.content  "EMPTY" >
<!ELEMENT meta  %Meta.content; >
<!-- end of Meta.element -->]]>
```

```
<!ENTITY % Meta.attlist  "INCLUDE" >
<![%Meta.attlist;[
<!ATTLIST meta
      %I18n.attrib;
      http-equiv   NMTOKEN                  #IMPLIED
      name         NMTOKEN                  #IMPLIED
      content      CDATA                    #REQUIRED
      scheme       CDATA                    #IMPLIED
>
<!-- end of Meta.attlist -->]]>

<!-- end of xhtml11-meta-1.mod -->
```

B.3.14. Scripting

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Document Scripting Module  ................................ -->
<!-- file: xhtml11-script-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Scripting 1.0//EN"
     SYSTEM "xhtml11-script-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Scripting

        script, noscript

     This module declares element types and attributes used to provide
     support for executable scripts as well as an alternate content
     container where scripts are not supported.
-->

<!-- script: Scripting Statement ....................... -->

<!ENTITY % Script.element  "INCLUDE" >
<![%Script.element;[
<!ENTITY % Script.content  "( #PCDATA )" >
<!ELEMENT script  %Script.content; >
<!-- end of Script.element -->]]>

<!ENTITY % Script.attlist  "INCLUDE" >
<![%Script.attlist;[
<!ATTLIST script
      charset      %Charset.datatype;       #IMPLIED
      type         %ContentType.datatype;   #REQUIRED
      src          %URI.datatype;           #IMPLIED
      defer        ( defer )                #IMPLIED
```

```
      xml:space    ( preserve )             #FIXED 'preserve'
>
<!-- end of Script.attlist -->]]>

<!-- noscript: No-Script Alternate Content ..... -->

<!ENTITY % Noscript.element  "INCLUDE" >
<![%Noscript.element;[
<!ENTITY % Noscript.content
     "( %Block.mix; )+"
>
<!ELEMENT noscript  %Noscript.content; >
<!-- end of Noscript.element -->]]>

<!ENTITY % Noscript.attlist  "INCLUDE" >
<![%Noscript.attlist;[
<!ATTLIST noscript
      %Common.attrib;
>
<!-- end of Noscript.attlist -->]]>

<!-- end of xhtml11-script-1.mod -->
```

B.3.15. Stylesheet

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Document Stylesheet Module  ............................... -->
<!-- file: xhtml11-style-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//DTD XHTML 1.1 Stylesheets 1.0//EN"
     SYSTEM "xhtml11-style-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Stylesheets

        style

     This module declares the style element type and its attributes,
     used to embed stylesheet information in the document head element.
-->

<!-- style: Stylesheet Information ..................... -->

<!ENTITY % Style.element  "INCLUDE" >
<![%Style.element;[
<!ENTITY % Style.content  "( #PCDATA )" >
<!ELEMENT style  %Style.content; >
```

```
<!-- end of Style.element -->]]>

<!ENTITY % Style.attlist  "INCLUDE" >
<![%Style.attlist;[
<!ATTLIST style
      %Title.attrib;
      %I18n.attrib;
      type          %ContentType.datatype;    #REQUIRED
      media         %MediaDesc.datatype;      #IMPLIED
      xml:space     ( preserve )              #FIXED 'preserve'
>
<!-- end of Style.attlist -->]]>

<!-- end of xhtml11-style-1.mod -->
```

B.3.16. Link

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Link Element Module  ....................................... -->
<!-- file: xhtml11-link-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Link Element 1.0//EN"
     SYSTEM "xhtml11-link-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Link element

        link

     This module declares the link element type and its attributes,
     which could (in principle) be used to define document-level links
     to external resources such as:

     a) for document specific toolbars/menus, e.g. start, contents,
        previous, next, index, end, help
     b) to link to a separate style sheet (rel="stylesheet")
     c) to make a link to a script (rel="script")
     d) by stylesheets to control how collections of html nodes are
        rendered into printed documents
     e) to make a link to a printable version of this document
        e.g. a postscript or pdf version (rel="alternate" media="print")
-->

<!-- link: Media-Independent Link ..................... -->

<!ENTITY % Link.element  "INCLUDE" >
<![%Link.element;[
```

```
<!ENTITY % Link.content  "EMPTY" >
<!ELEMENT link  %Link.content; >
<!-- end of Link.element -->]]>

<!ENTITY % Link.attlist  "INCLUDE" >
<![%Link.attlist;[
<!ATTLIST link
      %Common.attrib;
      charset      %Charset.datatype;       #IMPLIED
      href         %URI.datatype;           #IMPLIED
      hreflang     %LanguageCode.datatype;  #IMPLIED
      type         %ContentType.datatype;   #IMPLIED
      rel          %LinkTypes.datatype;     #IMPLIED
      rev          %LinkTypes.datatype;     #IMPLIED
      media        %MediaDesc.datatype;     #IMPLIED
>
<!-- end of Link.attlist -->]]>

<!-- end of xhtml11-link-1.mod -->
```

B.3.17. Base

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Base Element Module  ....................................... -->
<!-- file: xhtml11-base-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Base Element 1.0//EN"
     SYSTEM "xhtml11-base-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Base element

        base

     This module declares the base element type and its attributes,
     used to define a base URI against which relative URIs in the
     document will be resolved.

     Note that this module also redeclares the content model for
     the head element to include the base element.
-->

<!-- base: Document Base URI ........................... -->

<!ENTITY % Base.element  "INCLUDE" >
<![%Base.element;[
<!ENTITY % Base.content  "EMPTY" >
```

```
<!ELEMENT base  %Base.content; >
<!-- end of Base.element -->]]>

<!ENTITY % Base.attlist  "INCLUDE" >
<![%Base.attlist;[
<!ATTLIST base
      href         %URI.datatype;           #REQUIRED
>
<!-- end of Base.attlist -->]]>

<!ENTITY % Head.content
    "( %Head-opts.mix;,
     ( ( title, %Head-opts.mix;, ( base, %Head-opts.mix; )? )
     | ( base, %Head-opts.mix;, ( title, %Head-opts.mix; ))))"
>

<!-- end of xhtml11-base-1.mod -->
```

B.3.18. Legacy

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Legacy Markup Module ..................................... -->
<!-- file: xhtml11-legacy-1.mod

     This is an extension of XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Legacy Markup 1.0//EN"
     SYSTEM "xhtml11-legacy-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- HTML Legacy Markup

        font, s, strike, u (plus additional datatypes and attributes)

     This optional module declares additional markup for simple
     presentation-related markup based on features found in the
     HTML 4.0 Transitional and Frameset DTDs. This relies on
     inclusion of the Legacy Redeclarations module.

     See the note included in the Legacy Redeclarations Module
     for information on how to construct a DTD using this module.

     This is to allow XHTML 1.1 documents to be transformed for
     display on HTML browsers where CSS support is inconsistent
     or unavailable.
-->

<!-- Additional Element Types ................................... -->
```

```
<!ENTITY % Font.element  "INCLUDE" >
<![%Font.element;[
<!ENTITY % Font.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT font  %Font.content; >
<!-- end of Font.element -->]]>

<!ENTITY % Font.attlist  "INCLUDE" >
<![%Font.attlist;[
<!ATTLIST font
      %Core.attrib;
      %I18n.attrib;
      size         CDATA                    #IMPLIED
      color        %Color.datatype;         #IMPLIED
      face         CDATA                    #IMPLIED
>
<!-- end of Font.attlist -->]]>

<!ENTITY % S.element  "INCLUDE" >
<![%S.element;[
<!ENTITY % S.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT s  %S.content; >
<!-- end of S.element -->]]>

<!ENTITY % S.attlist  "INCLUDE" >
<![%S.attlist;[
<!ATTLIST s
      %Common.attrib;
>
<!-- end of S.attlist -->]]>

<!ENTITY % Strike.element  "INCLUDE" >
<![%Strike.element;[
<!ENTITY % Strike.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT strike  %Strike.content; >
<!-- end of Strike.element -->]]>

<!ENTITY % Strike.attlist  "INCLUDE" >
<![%Strike.attlist;[
<!ATTLIST strike
      %Common.attrib;
>
<!-- end of Strike.attlist -->]]>

<!ENTITY % U.element  "INCLUDE" >
<![%U.element;[
<!ENTITY % U.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT u  %U.content; >
<!-- end of U.element -->]]>
```

```
<!ENTITY % U.attlist  "INCLUDE" >
<![%U.attlist;[
<!ATTLIST u
      %Common.attrib;
>
<!-- end of U.attlist -->]]>

<!-- Additional Attributes ..................................... -->

<!-- Alignment attribute for Transitional use in HTML browsers
     (this functionality is generally well-supported in CSS,
     except within some contexts)
-->
<!ENTITY % Align.attrib
     "align        ( left | center | right | justify ) #IMPLIED"
>

<!ATTLIST a
      name         CDATA                    #IMPLIED
>

<!ATTLIST body
      background   %URI.datatype;           #IMPLIED
      bgcolor      %Color.datatype;         #IMPLIED
      text         %Color.datatype;         #IMPLIED
      link         %Color.datatype;         #IMPLIED
      vlink        %Color.datatype;         #IMPLIED
      alink        %Color.datatype;         #IMPLIED
>

<!ATTLIST br
      clear        ( left | all | right | none ) 'none'
>

<!ATTLIST caption
      %Align.attrib;
>

<!ATTLIST div
      %Align.attrib;
>

<!ATTLIST h1
      %Align.attrib;
>

<!ATTLIST h2
      %Align.attrib;
>

<!ATTLIST h3
      %Align.attrib;
>

<!ATTLIST h4
      %Align.attrib;
>
```

```
<!ATTLIST h5
      %Align.attrib;
>

<!ATTLIST h6
      %Align.attrib;
>

<!ATTLIST hr
      %Align.attrib;
      noshade      ( noshade )               #IMPLIED
      size         %Pixels.datatype;         #IMPLIED
      width        %Length.datatype;         #IMPLIED
>

<!ATTLIST img
      align       ( top | middle | bottom | left | right ) #IMPLIED
      border      %Pixels.datatype;         #IMPLIED
      hspace      %Pixels.datatype;         #IMPLIED
      vspace      %Pixels.datatype;         #IMPLIED
>

<!ATTLIST input
      %Align.attrib;
>

<!ATTLIST legend
      %Align.attrib;
>

<!ATTLIST li
      type         CDATA                     #IMPLIED
      value        CDATA                     #IMPLIED
>

<!ATTLIST map
      name         CDATA                     #IMPLIED
>

<!ATTLIST ol
      type         CDATA                     #IMPLIED
      compact      ( compact )               #IMPLIED
      start        CDATA                     #IMPLIED
>

<!ATTLIST p
      %Align.attrib;
>

<!ATTLIST pre
      width        CDATA                     #IMPLIED
>

<!ATTLIST table
      %Align.attrib;
      bgcolor      %Color.datatype;          #IMPLIED
```

```
>

<!ATTLIST tr
      bgcolor     %Color.datatype;          #IMPLIED
>

<!ATTLIST th
      nowrap      ( nowrap )                #IMPLIED
      bgcolor     %Color.datatype;          #IMPLIED
      width       %Pixels.datatype;         #IMPLIED
      height      %Pixels.datatype;         #IMPLIED
>

<!ATTLIST td
      nowrap      ( nowrap )                #IMPLIED
      bgcolor     %Color.datatype;          #IMPLIED
      width       %Pixels.datatype;         #IMPLIED
      height      %Pixels.datatype;         #IMPLIED
>

<!ATTLIST ul
      type         CDATA                    #IMPLIED
      compact      ( compact )              #IMPLIED
>

<!-- end of xhtml11-legacy-1.mod -->
```

B.4. XHTML DTD Support Modules

The modules in this section are elements of the XHTML DTD implementation that, while hidden from casual users, are important to understand when creating derivative markup languages using the Modularization architecture.

B.4.1. Block Phrasal

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Block Phrasal Module  ..................................... -->
<!-- file: xhtml11-blkphras-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Block Phrasal 1.0//EN"
     SYSTEM "xhtml11-blkphras-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Block Phrasal
```

```
        address, blockquote, pre, h1, h2, h3, h4, h5, h6

     This module declares the elements and their attributes used to
     support block-level phrasal markup.
-->

<!ENTITY % Address.element  "INCLUDE" >
<![%Address.element;[
<!ENTITY % Address.content
     "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT address  %Address.content; >
<!-- end of Address.element -->]]>

<!ENTITY % Address.attlist  "INCLUDE" >
<![%Address.attlist;[
<!ATTLIST address
      %Common.attrib;
>
<!-- end of Address.attlist -->]]>

<!ENTITY % Blockquote.element  "INCLUDE" >
<![%Blockquote.element;[
<!ENTITY % Blockquote.content
     "( %Block.mix; )+"
>
<!ELEMENT blockquote  %Blockquote.content; >
<!-- end of Blockquote.element -->]]>

<!ENTITY % Blockquote.attlist  "INCLUDE" >
<![%Blockquote.attlist;[
<!ATTLIST blockquote
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED
>
<!-- end of Blockquote.attlist -->]]>

<!ENTITY % Pre.element  "INCLUDE" >
<![%Pre.element;[
<!ENTITY % Pre.content
     "( #PCDATA
      | %Inlstruct.class;
      %Inlphras.class;
      | tt | i | b
      %I18n.class;
      %Anchor.class;
      | script | map
      %Inline.extra; )*"
>
<!ELEMENT pre  %Pre.content; >
<!-- end of Pre.element -->]]>

<!ENTITY % Pre.attlist  "INCLUDE" >
<![%Pre.attlist;[
<!ATTLIST pre
      %Common.attrib;
      xml:space    ( preserve )             #FIXED 'preserve'
>
```

```
<!-- end of Pre.attlist -->]]>

<!-- ...................  Heading Elements  ................... -->

<!ENTITY % Heading.content  "( #PCDATA | %Inline.mix; )*" >

<!ENTITY % H1.element  "INCLUDE" >
<![%H1.element;[
<!ELEMENT h1  %Heading.content; >
<!-- end of H1.element -->]]>

<!ENTITY % H1.attlist  "INCLUDE" >
<![%H1.attlist;[
<!ATTLIST h1
      %Common.attrib;
>
<!-- end of H1.attlist -->]]>

<!ENTITY % H2.element  "INCLUDE" >
<![%H2.element;[
<!ELEMENT h2  %Heading.content; >
<!-- end of H2.element -->]]>

<!ENTITY % H2.attlist  "INCLUDE" >
<![%H2.attlist;[
<!ATTLIST h2
      %Common.attrib;
>
<!-- end of H2.attlist -->]]>

<!ENTITY % H3.element  "INCLUDE" >
<![%H3.element;[
<!ELEMENT h3  %Heading.content; >
<!-- end of H3.element -->]]>

<!ENTITY % H3.attlist  "INCLUDE" >
<![%H3.attlist;[
<!ATTLIST h3
      %Common.attrib;
>
<!-- end of H3.attlist -->]]>

<!ENTITY % H4.element  "INCLUDE" >
<![%H4.element;[
<!ELEMENT h4  %Heading.content; >
<!-- end of H4.element -->]]>

<!ENTITY % H4.attlist  "INCLUDE" >
<![%H4.attlist;[
<!ATTLIST h4
      %Common.attrib;
>
<!-- end of H4.attlist -->]]>

<!ENTITY % H5.element  "INCLUDE" >
<![%H5.element;[
<!ELEMENT h5  %Heading.content; >
```

```
<!-- end of H5.element -->]]>

<!ENTITY % H5.attlist  "INCLUDE" >
<![%H5.attlist;[
<!ATTLIST h5
      %Common.attrib;
>
<!-- end of H5.attlist -->]]>

<!ENTITY % H6.element  "INCLUDE" >
<![%H6.element;[
<!ELEMENT h6  %Heading.content; >
<!-- end of H6.element -->]]>

<!ENTITY % H6.attlist  "INCLUDE" >
<![%H6.attlist;[
<!ATTLIST h6
      %Common.attrib;
>
<!-- end of H6.attlist -->]]>

<!-- end of xhtml11-blkphras-1.mod -->
```

B.4.2. Block Presentational

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Block Presentation Module  ................................. -->
<!-- file: xhtml11-blkpres-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Block Presentation 1.0//EN"
     SYSTEM "xhtml11-blkpres-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Block Presentational Elements

        hr

     This module declares the elements and their attributes used to
     support block-level presentational markup.
-->

<!ENTITY % Hr.element  "INCLUDE" >
<![%Hr.element;[
<!ENTITY % Hr.content  "EMPTY" >
<!ELEMENT hr  %Hr.content; >
<!-- end of Hr.element -->]]>
```

```
<!ENTITY % Hr.attlist  "INCLUDE" >
<![%Hr.attlist;[
<!ATTLIST hr
      %Common.attrib;
>
<!-- end of Hr.attlist -->]]>

<!-- end of xhtml11-blkpres-1.mod -->
```

B.4.3. Block Structural

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Block Structural Module  ..................................... -->
<!-- file: xhtml11-blkstruct-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Block Structural 1.0//EN"
     SYSTEM "xhtml11-blkstruct-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Block Structural

        div, p

     This module declares the elements and their attributes used to
     support block-level structural markup.
-->

<!ENTITY % Div.element  "INCLUDE" >
<![%Div.element;[
<!ENTITY % Div.content
     "( #PCDATA | %Flow.mix; )*"
>
<!ELEMENT div  %Div.content; >
<!-- end of Div.element -->]]>

<!ENTITY % Div.attlist  "INCLUDE" >
<![%Div.attlist;[
<!-- end of Div.attlist -->]]>
<!ATTLIST div
      %Common.attrib;
>

<!ENTITY % P.element  "INCLUDE" >
<![%P.element;[
<!ENTITY % P.content
     "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT p  %P.content; >
```

```
<!-- end of P.element -->]]>

<!ENTITY % P.attlist  "INCLUDE" >
<![%P.attlist;[
<!ATTLIST p
      %Common.attrib;
>
<!-- end of P.attlist -->]]>

<!-- end of xhtml11-blkstruct-1.mod -->
```

B.4.4. Inline Phrasal

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Inline Phrasal Module  ..................................... -->
<!-- file: xhtml11-inlphras-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Phrasal 1.0//EN"
     SYSTEM "xhtml11-inlphras-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Inline Phrasal

        abbr, acronym, cite, code, dfn, em, kbd, q, samp, strong, var

     This module declares the elements and their attributes used to
     support inline-level phrasal markup.
-->

<!ENTITY % Abbr.element  "INCLUDE" >
<![%Abbr.element;[
<!ENTITY % Abbr.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT abbr  %Abbr.content; >
<!-- end of Abbr.element -->]]>

<!ENTITY % Abbr.attlist  "INCLUDE" >
<![%Abbr.attlist;[
<!ATTLIST abbr
      %Common.attrib;
>
<!-- end of Abbr.attlist -->]]>

<!ENTITY % Acronym.element  "INCLUDE" >
<![%Acronym.element;[
<!ENTITY % Acronym.content
```

```
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT acronym  %Acronym.content; >
<!-- end of Acronym.element -->]]>

<!ENTITY % Acronym.attlist  "INCLUDE" >
<![%Acronym.attlist;[
<!ATTLIST acronym
      %Common.attrib;
>
<!-- end of Acronym.attlist -->]]>

<!ENTITY % Cite.element  "INCLUDE" >
<![%Cite.element;[
<!ENTITY % Cite.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT cite  %Cite.content; >
<!-- end of Cite.element -->]]>

<!ENTITY % Cite.attlist  "INCLUDE" >
<![%Cite.attlist;[
<!ATTLIST cite
      %Common.attrib;
>
<!-- end of Cite.attlist -->]]>

<!ENTITY % Code.element  "INCLUDE" >
<![%Code.element;[
<!ENTITY % Code.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT code  %Code.content; >
<!-- end of Code.element -->]]>

<!ENTITY % Code.attlist  "INCLUDE" >
<![%Code.attlist;[
<!ATTLIST code
      %Common.attrib;
>
<!-- end of Code.attlist -->]]>

<!ENTITY % Dfn.element  "INCLUDE" >
<![%Dfn.element;[
<!ENTITY % Dfn.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT dfn  %Dfn.content; >
<!-- end of Dfn.element -->]]>

<!ENTITY % Dfn.attlist  "INCLUDE" >
<![%Dfn.attlist;[
<!ATTLIST dfn
      %Common.attrib;
>
<!-- end of Dfn.attlist -->]]>
```

```
<!ENTITY % Em.element  "INCLUDE" >
<![%Em.element;[
<!ENTITY % Em.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT em  %Em.content; >
<!-- end of Em.element -->]]>

<!ENTITY % Em.attlist  "INCLUDE" >
<![%Em.attlist;[
<!ATTLIST em
      %Common.attrib;
>
<!-- end of Em.attlist -->]]>

<!ENTITY % Kbd.element  "INCLUDE" >
<![%Kbd.element;[
<!ENTITY % Kbd.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT kbd  %Kbd.content; >
<!-- end of Kbd.element -->]]>

<!ENTITY % Kbd.attlist  "INCLUDE" >
<![%Kbd.attlist;[
<!ATTLIST kbd
      %Common.attrib;
>
<!-- end of Kbd.attlist -->]]>

<!ENTITY % Q.element  "INCLUDE" >
<![%Q.element;[
<!ENTITY % Q.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT q  %Q.content; >
<!-- end of Q.element -->]]>

<!ENTITY % Q.attlist  "INCLUDE" >
<![%Q.attlist;[
<!ATTLIST q
      %Common.attrib;
      cite         %URI.datatype;           #IMPLIED
>
<!-- end of Q.attlist -->]]>

<!ENTITY % Samp.element  "INCLUDE" >
<![%Samp.element;[
<!ENTITY % Samp.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT samp  %Samp.content; >
<!-- end of Samp.element -->]]>

<!ENTITY % Samp.attlist  "INCLUDE" >
<![%Samp.attlist;[
<!ATTLIST samp
```

```
      %Common.attrib;
>
<!-- end of Samp.attlist -->]]>

<!ENTITY % Strong.element  "INCLUDE" >
<![%Strong.element;[
<!ENTITY % Strong.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT strong  %Strong.content; >
<!-- end of Strong.element -->]]>

<!ENTITY % Strong.attlist  "INCLUDE" >
<![%Strong.attlist;[
<!ATTLIST strong
      %Common.attrib;
>
<!-- end of Strong.attlist -->]]>

<!ENTITY % Var.element  "INCLUDE" >
<![%Var.element;[
<!ENTITY % Var.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT var  %Var.content; >
<!-- end of Var.element -->]]>

<!ENTITY % Var.attlist  "INCLUDE" >
<![%Var.attlist;[
<!ATTLIST var
      %Common.attrib;
>
<!-- end of Var.attlist -->]]>

<!-- end of xhtml11-inlphras-1.mod -->
```

B.4.5. Inline Presentational

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Inline Presentation Module  ............................... -->
<!-- file: xhtml11-inlpres-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Presentation 1.0//EN"
     SYSTEM "xhtml11-inlpres-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Inline Presentational Elements
```

```
        b, big, i, small, sub, sup, tt

     This module declares the elements and their attributes used to
     support inline-level presentational markup.
-->

<!ENTITY % B.element  "INCLUDE" >
<![%B.element;[
<!ENTITY % B.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT b  %B.content; >
<!-- end of B.element -->]]>

<!ENTITY % B.attlist  "INCLUDE" >
<![%B.attlist;[
<!ATTLIST b
      %Common.attrib;
>
<!-- end of B.attlist -->]]>

<!ENTITY % Big.element  "INCLUDE" >
<![%Big.element;[
<!ENTITY % Big.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT big  %Big.content; >
<!-- end of Big.element -->]]>

<!ENTITY % Big.attlist  "INCLUDE" >
<![%Big.attlist;[
<!ATTLIST big
      %Common.attrib;
>
<!-- end of Big.attlist -->]]>

<!ENTITY % I.element  "INCLUDE" >
<![%I.element;[
<!ENTITY % I.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT i  %I.content; >
<!-- end of I.element -->]]>

<!ENTITY % I.attlist  "INCLUDE" >
<![%I.attlist;[
<!ATTLIST i
      %Common.attrib;
>
<!-- end of I.attlist -->]]>

<!ENTITY % Small.element  "INCLUDE" >
<![%Small.element;[
<!ENTITY % Small.content
     "( #PCDATA | %Inline.mix; )*"
>
```

```
<!ELEMENT small  %Small.content; >
<!-- end of Small.element -->]]>

<!ENTITY % Small.attlist  "INCLUDE" >
<![%Small.attlist;[
<!ATTLIST small
      %Common.attrib;
>
<!-- end of Small.attlist -->]]>

<!ENTITY % Sub.element  "INCLUDE" >
<![%Sub.element;[
<!ENTITY % Sub.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT sub  %Sub.content; >
<!-- end of Sub.element -->]]>

<!ENTITY % Sub.attlist  "INCLUDE" >
<![%Sub.attlist;[
<!ATTLIST sub
      %Common.attrib;
>
<!-- end of Sub.attlist -->]]>

<!ENTITY % Sup.element  "INCLUDE" >
<![%Sup.element;[
<!ENTITY % Sup.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT sup  %Sup.content; >
<!-- end of Sup.element -->]]>

<!ENTITY % Sup.attlist  "INCLUDE" >
<![%Sup.attlist;[
<!ATTLIST sup
      %Common.attrib;
>
<!-- end of Sup.attlist -->]]>

<!ENTITY % Tt.element  "INCLUDE" >
<![%Tt.element;[
<!ENTITY % Tt.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT tt    %Tt.content; >
<!-- end of Tt.element -->]]>

<!ENTITY % Tt.attlist  "INCLUDE" >
<![%Tt.attlist;[
<!ATTLIST tt
      %Common.attrib;
>
<!-- end of Tt.attlist -->]]>

<!-- end of xhtml11-inlpres-1.mod -->
```

B.4.6. Inline Structural

```
<!-- ....................................................................... -->
<!-- XHTML 1.1 Inline Phrasal Module  ..................................... -->
<!-- file: xhtml11-inlstruct-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Inline Structural 1.0//EN"
     SYSTEM "xhtml11-inlstruct-1.mod"

     Revisions:
     (none)
     ....................................................................... -->

<!-- Inline Structural

        br, span

     This module declares the elements and their attributes used to
     support inline-level structural markup.
-->

<!ENTITY % Br.element  "INCLUDE" >
<![%Br.element;[

<!ENTITY % Br.content  "EMPTY" >
<!ELEMENT br  %Br.content; >

<!-- end of Br.element -->]]>

<!ENTITY % Br.attlist  "INCLUDE" >
<![%Br.attlist;[
<!ATTLIST br
      %Core.attrib;
>
<!-- end of Br.attlist -->]]>

<!ENTITY % Span.element  "INCLUDE" >
<![%Span.element;[
<!ENTITY % Span.content
     "( #PCDATA | %Inline.mix; )*"
>
<!ELEMENT span  %Span.content; >
<!-- end of Span.element -->]]>

<!ENTITY % Span.attlist  "INCLUDE" >
<![%Span.attlist;[
<!ATTLIST span
      %Common.attrib;
```

```
>
<!-- end of Span.attlist -->]]>

<!-- end of xhtml11-inlstruct-1.mod -->
```

B.4.7. Param

```
<!-- ..................................................................... -->
<!-- XHTML 1.1 Param Element Module  ..............................  -->
<!-- file: xhtml11-param-1.mod

     This is XHTML 1.1, a modular variant of XHTML 1.0.
     Copyright 1998-2000 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: $Id: xhtml-modularization-20000105.ps,v 1.2 2000/01/05 21:00:59 shane Exp $ SMI

     This DTD module is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC "-//W3C//ELEMENTS XHTML 1.1 Param Element 1.0//EN"
     SYSTEM "xhtml11-param-1.mod"

     Revisions:
     (none)
     ..................................................................... -->

<!-- Parameters for Java Applets and Embedded Objects

        param

     This module provides declarations for the param element,
     used to provide named property values for the applet
     and object elements.
-->

<!-- param: Named Property Value ..................... -->

<!ENTITY % Param.element  "INCLUDE" >
<![%Param.element;[
<!ENTITY % Param.content  "EMPTY" >
<!ELEMENT param  %Param.content; >
<!-- end of Param.element -->]]>

<!ENTITY % Param.attlist  "INCLUDE" >
<![%Param.attlist;[
<!ATTLIST param
      %Id.attrib;
      name         CDATA                    #REQUIRED
      value        CDATA                    #IMPLIED
      valuetype    ( data | ref | object )  'data'
      type         %ContentType.datatype;   #IMPLIED
>
<!-- end of Param.attlist -->]]>

<!-- end of xhtml11-param-1.mod -->
```

C. References

This appendix is *normative*.

C.1. Normative References

[BUILDING]
Building XHTML Modules, Murray Altheim, Shane P. McCarron, 5 January 2000.
See: http://www.w3.org/TR/2000/WD-xhtml-building-20000105

[HTML4]
HTML 4.01 Specification: W3C Recommendation, Dave Raggett, Arnaud Le Hors, Ian Jacobs, 24 December 1999.
See: http://www.w3.org/TR/1999/REC-html401-19991224

[RFC1766]
"Tags for the Identification of Languages", H. Alvestrand, March 1995. RFC1766 is expected to be updated by http://www.ietf.org/internet-drafts/draft-alvestrand-lang-tags-v2-00.txt, currently a work in progress.

[RFC2045]
"Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", N. Freed and N. Borenstein, November 1996. Note that this RFC obsoletes RFC1521, RFC1522, and RFC1590.

[ISO10646]
"Information Technology -- Universal Multiple-Octet Coded Character Set (UCS) -- Part 1: Architecture and Basic Multilingual Plane", ISO/IEC 10646-1:1993. This reference refers to a set of codepoints that may evolve as new characters are assigned to them. This reference therefore includes future amendments as long as they do **not** change character assignments up to and including the first five amendments to ISO/IEC 10646-1:1993. Also, this reference assumes that the character sets defined by ISO 10646 and Unicode remain character-by-character equivalent. This reference also includes future publications of other parts of 10646 (i.e., other than Part 1) that define characters in planes 1-16.

[SGML]
Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML), ISO 8879:1986.
Please consult http://www.iso.ch/cate/d16387.html for information about the standard, or http://www.oasis-open.org/cover/general.html#overview about SGML.

[SRGB]
"A Standard Default color Space for the Internet", version 1.10, M. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, 5 November 1996. This document is http://www.w3.org/Graphics/Color/sRGB

[XHTML1]
XHTML 1.0: The Extensible HyperText Markup Language, Steven Pemberton, et. al., 24 August 1999.
See: http://www.w3.org/TR/xhtml1

[XHTML11]
XHTML 1.1: Module-based XHTML, Murray Altheim, Shane McCarron, 5 January 2000.
See: http://www.w3.org/TR/2000/WD-xhtml11-20000105
[XML]
Extensible Markup Language (XML) 1.0: W3C Recommendation, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, 10 February 1998.
See: http://www.w3.org/TR/REC-xml
[XMLSCHEMA]
XML Schema Part 1: Structures Henry S. Thompson, et. al., 17 December 1999
See: http://www.w3.org/TR/1999/WD-xhtmlschema-1-19991217

C.2. Informative References

[CATALOG]
Entity Management: OASIS Technical Resolution 9401:1997 (Amendment 2 to TR 9401) Paul Grosso, Chair, Entity Management Subcommittee, SGML Open, 10 September 1997.
See: http://www.oasis-open.org/html/a401.htm
[DEVDTD]
Developing SGML DTDs: From Text to Model to Markup, Eve Maler and Jeanne El Andaloussi.
Prentice Hall PTR, 1996, ISBN 0-13-309881-8.
[STRUCTXML]
Structuring XML Documents, David Megginson. Part of the Charles Goldfarb Series on Information Management.
Prentice Hall PTR, 1998, ISBN 0-13-642299-3.
[DOCBOOK]
DocBook DTD, Eve Maler and Terry Allen.
Originally created under the auspices of the Davenport Group, DocBook is now maintained by OASIS. The *Customizer's Guide for the DocBook DTD V2.4.1* is available from this site.
See: http://www.oasis-open.org/docbook/index.html
[DUBLIN]
The Dublin Core: A Simple Content Description Model for Electronic Resources, The Dublin Core Metadata Initiative.
See: http://purl.oclc.org/dc/
[SMIL]
Synchronized Multimedia Integration Language (SMIL) 1.0 Specification, Philipp Hoschka, 15 June 1998.
See: http://www.w3.org/TR/REC-smil
[TEI]
The Text Encoding Initiative (TEI)
See: http://www.uic.edu/orgs/tei/
[URI]
Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter, August 1998.
See: http://www.ietf.org/rfc/rfc2396.txt. This RFC updates RFC 1738 [URL] [p.117] and [RFC1808] [p.117] .

[URL]
IETF RFC 1738, Uniform Resource Locators (URL), T. Berners-Lee, L. Masinter, M. McCahill.
See: http://www.ietf.org/rfc/rfc1738.txt
[RFC-1808]
Relative Uniform Resource Locators, R. Fielding.
See: http://www.ietf.org/rfc/rfc1808.txt
[CSS2]
"Cascading Style Sheets, level 2 (CSS2) Specification", B. Bos, H. W. Lie, C. Lilley, I. Jacobs, 12 May 1998.
Available at: http://www.w3.org/TR/REC-CSS2
[DOM]
"Document Object Model (DOM) Level 1 Specification", Lauren Wood *et al.*, 1 October 1998.
Available at: http://www.w3.org/TR/REC-DOM-Level-1
[RFC2119]
"RFC2119: Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997.
Available at: http://www.ietf.org/rfc/rfc2119.txt
[RFC2376]
"RFC2376: XML Media Types", E. Whitehead, M. Murata, July 1998.
Available at: http://www.ietf.org/rfc/rfc2376.txt
[TIDY]
"HTML Tidy" is a tool for detecting and correcting a wide range of markup errors prevalent in HTML. It can also be used as a tool for converting existing HTML content to be well formed XML. Tidy is being made available on the same terms as other W3C sample code, i.e. free for any purpose, and entirely at your own risk.
It is available from: http://www.w3.org/Status.html#TIDY
[XMLNAMES]
"Namespaces in XML", T. Bray, D. Hollander, A. Layman, 14 January 1999.
XML namespaces provide a simple method for qualifying names used in XML documents by associating them with namespaces identified by URI.
Available at: http://www.w3.org/TR/REC-xml-names
[XMLSTYLE]
"Associating stylesheets with XML documents Version 1.0", J. Clark, 14 January 1999.
This document describes a means for a stylesheet to be associated with an XML document by including one or more processing instructions with a target of xml-stylesheet in the document's prolog.
Available at: http://www.w3.org/TR/PR-xml-stylesheet

D. Design Goals

This appendix is **informative**

There are six major design goals for the modularization framework for XHTML:

- [G1] Provide a means for the W3C and third parties to integrate XHTML into other XML languages.
- [G2] Provide a means for the W3C to extend XHTML with new or optional features.
- [G3] Provide a means for third parties to extend XHTML with domain-specific features.
- [G4] Provide a means for third parties to integrate other XML languages into XHTML.
- [G5] Improve the ability to create a close approximation to the HTML 4.0 DTDs.
- [G6] Improve ease-of-use for DTD developers.

D.1. Requirements

The design goals listed in the previous section lead to a large number of requirements for the modularization framework. These requirements, summarized in this section, can be further classified according to the major features of the framework to be described.

D.1.1. Granularity

Collectively the requirements in this section express the desire that the modules defined within the framework hit the right level of granularity:

- [R1.1] Abstract modules should promote and maintain content portability.
- [R1.2] Abstract modules should promote platform profile standardization.
- [R1.3] Abstract modules should be large enough to promote interoperability.
- [R1.4] Abstract modules should be small enough to avoid the need for subsets.
- [R1.5] Abstract modules should collect elements with similar or related semantics.
- [R1.6] Abstract modules should separate elements with dissimilar or unrelated semantics.
- [R1.7] Modules should be small enough to allow single element document type modules.

D.1.2. Composibility

The composibility requirements listed here are intended to ensure that the modularization framework be able to express the right set of target modules required by the communities that will be served by the framework:

- [R2.1] The module framework should allow construction of abstract modules for XHTML 1.0.
- [R2.2] The module framework should allow construction of abstract modules that closely approximate HTML 4.0.
- [R2.3] The module framework should allow construction of abstract modules for other W3C Recommendations.
- [R2.4] The module framework should allow construction of abstract modules for other XML

document types.
- [R2.5] The module framework should allow construction of abstract modules for a wide range of platform profiles.

D.1.3. Ease of Use

The modularization framework will only receive widespread adoption if it describes mechanisms that make it easy for our target audience to use the framework:

- [R3.1] The module framework should make it easy for document type designers to subset and extend XHTML abstract modules.
- [R3.2] The module framework should make it easy for document type designers to create abstract modules for other XML document types.
- [R3.3] The module framework should make it easy for document authors to validate elements from different abstract modules.

D.1.4. Compatibility

The intent of this document is that the modularization framework described here should work well with the XML and other standards being developed by the W3C Working Groups:

- [R4.1] The module framework should strictly conform to the XML 1.0 Recommendation.
- [R4.2] The module framework should be compatible with the XML linking specification.
- [R4.3] The module framework should be compatible with the XML stylesheet specification.
- [R4.4] The module framework should be able to adopt new W3C recommendations where appropriate.
- [R4.5] The module framework should not depend on W3C work in progress.
- [R4.6] The module framework should not depend on work done outside W3C.

D.1.5. Conformance

The effectiveness of the framework will also be measured by how easy it is to test the behavior of modules developed according to the framework, and to test the documents that employ those modules for validation:

- [R5.1] It should be possible to validate documents constructed using elements and attributes from abstract modules.
- [R5.2] It should be possible to explicitly describe the behavior of elements and attributes from abstract modules.
- [R5.3] It should be possible to verify the behavior of elements and attributes from abstract modules.
- [R5.4] It should be possible to verify a compound document type as an XHTML document type.
- [R5.5] Modules defined in accordance with the methods in this document shall not duplicate the names of elements or parameter entities defined in XHTML modules.