Modularization of XHTML™

W3C Working Draft 06 April 1999

This version:
http://www.w3.org/TR/1999/xhtml-modularization-19990406
(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:
None.
Editors:
Murray Altheim, Sun Microsystems
Daniel Austin, CNET: The Computer Network
Frank Boumphrey, HTML Writers Guild
Sam Dooley, IBM
Shane McCarron, The Open Group
Ted Wugofski, Gateway

Abstract

This working draft specifies a modularization of XHTML 1.0. There are two aspects to the proposed modularization: modularization into semantic modules, and implementation of these semantic modules through a document type definition (DTD). Semantic modules provide a means for subsetting and extending XHTML, a feature desired for extending XHTML's reach onto emerging platforms. Modularization at the DTD level improves the ability to create new complete DTDs from XHTML and other DTD modules.

Status of this document

This document is a working draft of the W3C's HTML Working Group. This working draft may be updated, replaced or rendered obsolete by other W3C 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 document is work in progress and does not imply endorsement by the W3C membership.

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)*.

Please send detailed comments on this document to www-html-editor@w3.org. We cannot guarantee a personal response, but we will try when it is appropriate. Public discussion on HTML features takes place on the mailing list www-html@w3.org.

Quick Table of Contents

Full Table of Contents

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 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. This document describes how to create additional members of the XHTML family of document types.

1.2. Modularization Framework

This framework provides mechanisms for defining members of the XHTML family of document types, including the three standard XHTML 1.0 document types (Strict, Transitional, and Frameset), subset document types that include only some of the elements from one of the standard document types, and extension document types that incorporate elements from other XML document types.

The modularization framework defines a collection of semantic modules that form the basis for the XHTML family of document types. These semantic modules may be combined with each other and with semantic modules defined for other XML document types to create XHTML subset and extension document types that qualify as members of the XHTML family of document types.

The modularization framework also defines a collection of DTD modules that represent the underlying building blocks used to define the XHTML semantic modules. These DTD modules are created according to certain conventions, as specified by the framework, to allow their combination into semantic modules and complete, functional DTDs.

The conventions specified by the modularization framework may also be used to create new semantic and DTD modules for other XML document types. These new semantic modules can then be used with the XHTML semantic modules to create XHTML document types that incorporate elements from other XML document types.

The modularization framework provides instructions to document authors for associating an XHTML document type with a document instance, and for verifying that the document is a valid instance of the XHTML document type associated with the document.

1.3. Modularization of 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 (handheld 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.

The modularization of XHTML is accomplished on two major levels: at the semantic level, and at the document type level. Roughly speaking, the semantic level provides a conceptual approach to the modularization of XHTML, while the document type level provides DTD-level building blocks that allow document type designers to support the semantic modules.

1.3.1. Semantic modules

An XHTML document type is defined as a set of semantic modules. A semantic module defines, in a document type, one kind of data that is semantically different from all others. Semantic modules can be combined into document types without a deep understanding of the underlying schema that defines 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 schemas are currently represented as DTDs, as described in the XML 1.0 Recommendation [XML] [p.58] . 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. Compound document types

A compound 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 semantic modules into a compound document type, develop documents against that compound document type, and to validate that document against the associated compound document type.

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 semantic modules.

The general problem of fragment validation - validation of XML documents with different schemas from multiple XML Namespaces [XMLNS] [p.61] 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 semantic modules, compound DTDs.

1.3.5. Conformance

This section introduces three notions of conformance relating to the modularization of XHTML: Document type conformance, document conformance, and browser conformance.

1.3.5.1. Document type conformance

The goal of the modularization framework is to support the creation of new modules beyond those envisioned for XHTML. To support this activity, it is the intent of this document that semantic modules form the atomic building blocks for new XHTML document types. For a compound document type to be considered an XHTML document type, it must satisfy the following properties:

- It must incorporate at least one of the XHTML semantic modules.
- For each XHTML semantic module it uses, it must incorporate the module in its entirety.

These requirements are intended to be extremely permissive, while ensuring that document authors can rely on the behavior of a module at the semantic level.

1.3.5.2. Document conformance

An XHTML document that conforms to the modularization framework must meet the following requirements:

- It must specify a conforming XHTML document type.
- It must validate against its associated XHTML document type.

1.3.5.3. Browser conformance

A browser conforming to this document shall be a conforming browser as defined in [XHTML1] [p.??] . In addition, such a browser shall support the following functionality:

1. The browser shall use the value of the `xmlns` attribute of the `html` element to uniquely identify the default namespace of the document as associated with the document's XHTML document type.

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.58] is highly recommended.

document type
: a class of documents sharing a common abstract structure. The ISO 8879 [SGML] [p.59] definition is as follows: "a class of documents having similar characteristics; for example,

journal, article, technical manual, or memo. (4.102)"

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.

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.

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.

reference DTD
: a DTD whose markup model represents the foundation of a complete document type. A reference DTD provides the basis for the design of a "family" of related DTDs, such as subsets, extensions and variants.

subset DTD
: a DTD whose document model is the proper subset of a reference document type, whose conforming document instances are still valid according to the reference DTD. A subset may place tighter restrictions on the markup than the reference, remove elements or attributes, or both.

extension DTD
: a DTD whose document model extends a reference document type (usually by the addition of element types or attributes), but generally makes no profound changes to the reference document model other than required to add the extension's semantic components. An extension can also be considered a proper superset if the reference document type is a proper subset of the extension.

variant DTD
: a DTD whose document model alters (through subsetting, extension, and/or substitution) the basic data model of a reference document type. It is often difficult to transform without loss between instances conforming to a variant DTD and the reference DTD.

fragment DTD
: a portion of a DTD used as a component either for the creation of a compound or variant document type, or for validation of a document fragment. SGML nor XML current have standardized methods for such partial validation.

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).

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

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

element
: an instance of an element type.

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

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

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

entity
: an entity is a logical or physical storage unit containing document content. Entities may be composed of parseable 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 "Copyright 1999 Sun Microsystems, Inc.")

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

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.

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.

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.

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

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.
: A module is a collection of elements or attributes.
: A profile is metadata about an XML document type and possible related technologies, such as scripting languages and style-sheets. A browser can support one or many profiles. The purpose of profiles is to provide content developers with machine-readable information about features that can be expected from a particular browser. For example, a Television profile would include a DTD for Television sets and describe technologies that can be used (e.g. scripts and style-sheets).

3. XHTML Semantic Modules

This section is *normative*.

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

Throughout this section, some elements are marked as only being available when the transitional aspects of XHTML are supported. Transitional elements, attributes, and behaviors are defined in HTML 4. These transitional aspects continue to be defined by XHTML, but XHTML does not require their support in all conforming implementations.

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.

3.1. Applet Module

The Applet Module provides elements for including external applets. Specifically, the Applet Module supports:

- applet
- param

3.2. Block Modules

Block modules define "block level" elements and their attributes. Block level elements are those that cause a break in the rendered output when they are encountered in a document.

3.2.1. Block Phrasal Module

This module defines block level elements that have special "phrasal" rendering characteristics - they transform the enclosed content in a special, specific way to assist in its interpretation by the user. Elements included in the Block Phrasal Module are:

- address
- blockquote
- pre
- h1
- h2
- h3
- h4
- h5
- h6

3.2.2. Block Presentational Module

This module defines block level elements that are used strictly to improve the presentation of a document. Elements included in the Block Presentation Module are:

- center*
- hr

Items marked with an asterisk are only available when the Transitional aspects of XHTML are enabled.

3.2.3. Block Structural Module

This module defines block level elements to help control the structure of their enclosed content. Elements included are:

- div
- p

3.3. Inline modules

Inline modules defined elements and their attributes that, when used in a document, effect their contents but do not cause a break in the rendered output.

3.3.1. Inline Phrasal Module

This module defines inline level elements that have special "phrasal" rendering characteristics - they transform the enclosed content in a special, specific way to assist in its interpretation by the user. Elements included in the Inline Phrasal Module are:

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

3.3.2. Inline Presentational Module

This module defines inline level elements that are used strictly to improve the presentation of a document. Elements included in the Inline Presentation Module are:

- b
- basefont[*]
- big
- font[*]
- i
- s[*]
- small
- strike[*]
- sub
- sup
- tt
- u[*]

Items marked with an asterisk are only available when the Transitional aspects of XHTML are enabled.

3.3.3. Inline Structural Module

This module defines inline level elements to help control the structure of their enclosed content. Elements included are:

- bdo
- br
- del
- ins
- span

3.4. Linking Module

The Linking Module provides elements that are used for linking an HTML document to other documents or resources. Specifically, the Linking Module supports:

- a
- base
- link

3.5. List Module

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

- dir*
- dl
- dt
- dd
- ol
- ul
- li
- menu*

Items marked with an asterisk are only available when the Transitional aspects of XHTML are enabled.

3.6. HTML 3.2 Forms Module

The HTML 3.2 Forms Module provides the forms features found in HTML 3.2. Specifically, the HTML 3.2 Forms Module supports:

- form
- input
- select
- option
- textarea

3.7. HTML 4.0 Forms Module

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

- button
- fieldset
- form
- input
- label
- legend
- optgroup
- option
- select
- textarea

The HTML 4.0 Forms Module is a superset of the HTML 3.2 Forms Module.

3.8. HTML 3.2 Table Module

The HTML 3.2 Table Module provides table-related elements, but only in a limited form. Specifically, the HTML 3.2 Table Module supports:

- caption
- table
- td
- th
- tr

3.9. HTML 4.0 Table Module

As its name suggests, the HTML 4.0 Table Module provides table-related elements, with the full capabilities of HTML 4.0. Specifically, the HTML 4.0 Table Module supports:

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

This module is a proper superset of the HTML 3.2 Table Module.

3.10. 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:

- img

3.11. Image Map Module

The Image Map Module provides elements for images and client side image maps. Specifically, the Image Map Module supports:

- area
- map

The Image Map Module is typically used in conjunction with the image [p.17] and linking [p.15] modules

3.12. Object Module

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

- object
- param

3.13. Frames Module

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

- frameset
- frame
- iframe
- noframes

The elements in the Frame Module are included in the HTML 4.0 Frameset document type.

3.14. 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. Attributes included in this module are:

- onclick
- ondblclick
- onmousedown
- onmouseup
- onmouseover
- onmousemove
- onmouseout
- onkeypress
- onkeydown
- onkeyup

3.15. Metainformation Module

The Metainformation Module defines elements that are typically used within the declarative portion of a document (in XHTML within the head element). This module includes the following elements:

- meta
- title

3.16. 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 included in this module are:

- noscript
- script

3.17. Stylesheet Module

The Stylesheet Module defines style sheet handling. The module includes:

- style

3.18. Structure Module

The Structure Module defines the major structural elements for XHTML. These elements effectively act as the basis for the content model of HTML (although the content model itself is defined in other, DTD-specific modules). The elements included in this module are:

- body
- head
- html

4. XHTML DTD Modules

This section is *normative.*

This section specifies the XHTML DTD modules, the type declarations found in each module, and the file and parameter entity naming conventions used throughout the DTD.

4.1. Implementing Document Model Modules in the DTD

Partitioning of the document model occurs at the semantic module level. This partitioning is implemented in the markup model by two primary methods: parameterization, the use of parameter entities as reusable strings, and modularization, the creation of DTD fragments called *modules*.

4.1.1. Parameterization

This specification classifies parameter entities into six categories and names them consistently using the following suffixes:

.mod
: parameter entities use the suffix `.mod` when they are used to represent a DTD module (a collection of element classes). In this specification, each module is an atomic unit and may be represented as a separate file entity.

.module
: parameter entities use the suffix `.module` when they are used to control the inclusion of a DTD module by containing either of the conditional section keywords `INCLUDE` or `IGNORE`.

.content
: parameter entities use the suffix `.content` when they are used to represent the content model of an element type.

.class
: parameter entities use the suffix `.class` when they are used to represent elements of the same class.

.mix
: parameter entities use the suffix `.mix` when they are used to represent a collection of element types from different classes.

.attrib
: parameter entities use the suffix `.attrib` when they are used to represent a group of tokens representing one or more complete attribute specifications within an ATTLIST declaration.

For example, in HTML 4.0, the `%block;` parameter entity is defined to represent the heterogenous collection of element types that are block-level elements. In this specification, the corollary parameter entity is `%Block.mix;`.

4.1.2. Modularization

DTD modules are often used to encompass the markup declarations of a specific semantic component or "feature", from higher-level document features like tables and forms, to lower-level components such as specific elements or element groups. Modules can even contain modules, creating a hierarchical structure mirroring the document model. Note that modules are not always implemented as separate file entities, and modular DTDs can be easily normalized into single file versions for more efficient distribution over the Web.

The relationship between document model components and how they are implemented in markup as modules, entities and files (i.e., the *granularity* of the parameterization or modularization, how the markup model is structured and stored as separate entities, etc.) is not necessarily direct, as design style and implementation issues properly play a part. Higher-level modules are sometimes delivered as individual file entities to facilitate portability and reusability. To promote interoperability, the XHTML DTD design considers each module as atomic, with the notion that implementations should support the semantics of an entire module without further subdivision.

While the notion of "plug and play" with DTD modules is very attractive, in practice this is not quite so simple. Complex document models often resort to extensive parameterization of semantic modules to facilitate understanding, markup reuse, extensibility, and maintenance. The resultant modules may have have many interdependencies, and may require a fair amount of "rewiring" when adding or removing a DTD module. In light of this, a compromise must be made between markup flexibility, complexity of the DTD, and ease of maintainability.

The XHTML DTD attempts to ameliorate this by localizing many of the more "global" parameter entities to several modules that are declared early in the DTD. These are labelled common modules, and include declarations for common names, attributes, parameter and character entities.

XHTML elements are classified into the following categories:

structural element types
: element types that create the overall structure of an XHTML document.

block element types
: element types that should cause a line break.

inline element types
: element types that are displayed inline to an existing block.

phrasal element types
: element types that specify a domain-relevant connotation

presentational element types
: element types that indicate a desire on the part of the author for a specific rendering effect.

special case (or "feature") element types
: element types that provide XHTML with special features, such as linking, forms, etc.

Following is a module-by-module catalog of element types and parameter entities declared in the three XHTML DTDs, what constitutes in effect the XHTML *namespace*.

More detailed *DTD Modularization Interface* ('DMI') documentation is also available for each of the three XHTML 1.0 DTDS:

- XHTML 1.0 Strict DMI
- XHTML 1.0 Transitional DMI
- XHTML 1.0 Frameset DMI

4.2. Common Declarations

4.2.1. Names (XHTML1-names.mod)

The XHTML1-names.mod DTD module defines the following common names, many of these imported from other specifications and standards.

%ContentType;
: media type, as per [RFC2045]

%ContentTypes;
: a comma-separated list of media types, as per [RFC2045]

%Charset;
: a character encoding, as per [RFC2045]

%Charsets;
: a space-separated list of character encodings, as per [RFC2045]

%Datetime;
: date and time information, ISO date format

%Character;
: a single character from [ISO10646]

%LanguageCode;
: a language code, as per [RFC1766]

%LinkTypes;
: a space-separated list of link types

%MediaDesc;
: a single or comma-separated list of media descriptors

%Number;
: one or more digits

%URI;
: a Uniform Resource Identifier, see [URI]

%URIs;
: a space-separated list of Uniform Resource Identifiers, see [URI]

%Script;
: a script expression

%StyleSheet;
: style sheet data

%Text;
: simple text

%Length;
: the length defined in the HTML strict DTD for cellpadding and cellspacing

%MultiLength;
: pixel, percentage, or relative

%MultiLengths;
: a comma-separated list of MultiLength

%Pixels;
: integer representing length in pixels

%FrameTarget;
: render in this frame

%Color;
: a color using sRGB

4.2.2. Attributes (XHTML1-attribs.mod)

The XHTML1-attribs.mod DTD modules defines the following common attributes:

%Core.attrib;
: defines the attributes `id`, `class`, `style`, and `title`

%I18n.attrib;
: defines the internationalization (i18n) attributes `lang`, `xml:lang` and `dir`

%Events.attrib;
: in this module, `%Events.attrib;` is declared as an empty string, should the Events module not already have been instantiated.

%Common.attrib;
: combines `%Core.attrib;`, `%I18n.attrib;`, and `%Events.attrib;`

%Align.attrib;
: in this module, `%Align.attrib;` is declared as an empty string, as a default in the Strict DTD.

%XLink.attribs;
: a conditional section keyword controlling declaration of additional XLink attributes on the `a` element type

%Alink.attrib;
: additional XLink attributes on the `a` element type

4.2.3. Transitional Attributes (XHTML1-attribs-t.mod)

The XHTML1-attribs-t.mod DTD module defines the common attributes associated with the HTML 4.0 Transitional DTD:

%Core.attrib;
: defines the attributes `id`, `class`, `style`, and `title`

%I18n.attrib;
: defines the internationalization (i18n) attributes `lang`, `xml:lang` and `dir`

%Common.attrib;
: combines `%Core.attrib;`, `%I18n.attrib;`, and `%Events.attrib;`

%Align.attrib;
: horizontal text alignment

%IAlign.attrib;
: horizontal and vertical text alignment

%XLink.attribs;
: a conditional section keyword controlling declaration of additional XLink attributes on the `a` element type

%Alink.attrib;
: additional XLink attributes on the `a` element type

4.2.4. Strict Document Model (XHTML1-model.mod)

The XHTML1-model DTD module declares parameter entities describing the structure of the XHTML Strict document model. It provides an effective implementation of the document model occurring within `body` in one location, simplifying modification. It's also a good place to gain an understanding of the structures of the DTD.

NOTE: because the `#PCDATA` token must occur first in a mixed content model in XML, it is not included in any of the following parameter entities, and is declared explicitly on each element where it is to occur.

%Misc.class;
: these elements are neither block nor inline, and can essentially be used anywhere in the document body

%Inlstruct.class;
: the class of inline structural element types

%Inlpres.class;
: the class of inline presentational element types

%Inlphras.class;
: the class of inline phrasal element types

%Inlspecial.class;
: the class of special inline element types

%Formctrl.class;
: the class of form control element types

%Inline.class;
: includes all inline elements, used as a component in mixes

%Inline.mix;
: includes all inline elements, including `%Misc.class;`

%Inline-noa.class;
: includes all non-anchor inlines, used as a component in mixes

%Inline-noa.mix;
: includes all non-anchor inlines

%Heading.class;
: the class of heading element types `h1` to `h6`

%List.class;
: the class of list element types

%Blkstruct.class;
: the class of block structural element types

%Blkpres.class;
: the class of block presentational element types

%Blkphras.class;
: the class of block phrasal element types

%Blkform.class;
: the class of block-level form element types

%Blkspecial.class;
: the class of special block-level element types

%Block.class;
: includes all block elements, used as an component in mixes

%Block.mix;
: includes all block elements plus `%Misc.class;`

%Block-noform.class;
: includes all non-form block elements, used as a component in mixes

%Block-noform.mix;
: includes all non-form block elements, plus `%Misc.class;`

%Flow.mix;
: includes all text content, block and inline

4.2.5. Transitional Document Model (XHTML1-model-t.mod)

The XHTML1-model DTD module declares parameter entities describing the structure of the XHTML Transitional document model. It provides an effective implementation of the document model occurring within `body` in one location, simplifying modification. It's also a good place to gain an understanding of the structures of the DTD.

NOTE: because the `#PCDATA` token must occur first in a mixed content model in XML, it is not included in any of the following parameter entities, and is declared explicitly on each element where it is to occur.

%Misc.class;
: these elements are neither block nor inline, and can essentially be used anywhere in the document body

%Inlstruct.class;
: the class of inline structural element types

%Inlpres.class;
: the class of inline presentational element types

%Inlphras.class;
: the class of inline phrasal element types

%Inlspecial.class;
: the class of special inline element types

%Inlspecial.class;
: [description]

%Formctrl.class;
: the class of form control element types

%Inline.class;
: includes all inline elements, used as a component in mixes

%Inline.mix;
: includes all inline elements, including `%Misc.class;`

%Inline-noa.class;
: includes all non-anchor inlines, used as a component in mixes

%Inline-noa.mix;
: includes all non-anchor inlines

%Heading.class;
: the class of heading element types `h1` to `h6`

%List.class;
: the class of list element types

%Blkstruct.class;
: the class of block structural element types

%Blkpres.class;
: the class of block presentational element types

%Blkphras.class;
: the class of block phrasal element types

%Blkform.class;
: the class of block-level form element types

%Blkspecial.class;
: the class of special block-level element types

%Blkspecial.class;
: the class of special block-level element types

%Block.class;
: includes all block elements, used as an component in mixes

%Block.mix;
: includes all block elements plus `%Misc.class;`

%Block-noform.class;
: includes all non-form block elements, used as a component in mixes

%Block-noform.mix;
: includes all non-form block elements, plus `%Misc.class;`

%Flow.mix;
: includes all text content, block and inline

4.2.6. Intrinsic Event Attributes (XHTML1-events.mod)

The XHTML1-events.mod DTD module defines the intrinsic event attributes specified in HTML 4.0.

ATTLIST a
: additional event-related attributes for the `a` element type

%Events.attrib;
: defines the intrinsic events such as `onclick` and `onmouseout`

4.2.7. Character Entities (XHTML1-charent.mod)

This module acts as a wrapper for declaring the three character entity sets based on ISO Latin 1 and other special symbols used in HTML, such as `<`, ` `, `Ä`, etc.

%XHTML1.ents;
: a conditional section keyword which can be set to `IGNORE` during normalization of the DTD to avoid having the character entity declarations included in the resultant DTD (this should be done in internal subset of the dummy document used for normalization, rather than explicitly editing this module)

%XHTML1-lat1;
: the extended ISO Latin 1 set of character entities, the same as in HTML 4.0 DTD.

%XHTML1-symbol;
: symbol characters for XHTML, including mathematical and Greek characters

%XHTML1-special;
: special characters for XHTML, including typographic symbols. (this set now includes the new `€` symbol)

4.3. Document Structure Element Types

4.3.1. Normal Document Structure (XHTML1-struct.mod)

The XHTML1-struct.mod DTD module defines element types that support the general structure of an XHTML document, as apart from the content’s structure.

ELEMENT head
: the `head` element type and its attributes

%Head.content;
: the content model for the `head` element type

%Head-opts.mix;
: the optional, repeatable element types that can appear in the `head` of a document

ELEMENT body
: the `body` element type and its attributes

ATTLIST body
: additional Transitional attributes on `body` (in a `%XHTML.Transitional;` conditional section)

%Body.content;
: the content model for the `body` element type

ELEMENT html
: the `html` element type and its attributes, the *document element* for XHTML.

%Html.content;
: the content model for the `html` element type

%Version.attrib;
: the Formal Public Identifier (FPI) for this DTD. This parameter entity is a historical legacy of past HTML DTDs, and is preserved in case any application software uses it. It is also the only place within the DTD (excluding comments) that the DTD's FPI is declared. This declares the attribute specification containing the value as declared in the DTD driver.

%Profile.attrib;
: the declared profile identifier for this DTD.
FIXME This should be changed in the DTD to be named as a namespace, not profile identifier.

4.3.2. Frames (XHTML1-frames.mod)

The XHTML1-frames.mod DTD module defines element types that provide frame functionality.

ELEMENT frameset
: the `frameset` element type and its attributes

%Frameset.content;
: the content model for the `frameset` element type

ELEMENT frame
: the `frame` element type and its attributes

%Frame.content;
: the content model for the `frame` element type

ELEMENT iframe
: the `iframe` element type and its attributes

%Iframe.content;
: the content model for the `frame` element type

ELEMENT noframes
: the `noframe` element type and its attributes

%Noframes.content;
: the content model for the `noframes` element type

ATTLIST a
: additional frame-related attributes on the `a` element type

%Html.content;
: redeclares the content model of the `html` element type

4.3.3. Lists (XHTML1-list.mod)

The XHTML1-lists.mod DTD module defines elements that provide list functionality.

ELEMENT dl
: the `dl` (definition list) element type and its attributes

%Dl.content;
: the content model for the `dl` element type

ELEMENT dt
: the `dl` (definition term) element type and its attributes

%Dt.content;
: the content model for the `dt` element type

ELEMENT dd
: the `dl` (definition description) element type and its attributes

%Dd.content;
: the content model for the `dd` element type

ELEMENT ol
: the `dl` (ordered list) element type and its attributes

%Ol.content;
: the content model for the `ol` element type

%OlStyle;
: ordered lists (ol) numbering style

ELEMENT ul
: the `dl` (unordered list) element type and its attributes

%Ul.content;
: the content model for the `ul` element type

%UlStyle;
: unordered list (ul) bullet styles

ELEMENT li
: the `dl` (list item) element type and its attributes

%Li.content;
: the content model for the `li` element type

ELEMENT dir
: declares the Transitional element type `dir` (deprecated in HTML 4.0)

%Dir.content;
: the content model for the `dir` element type

ELEMENT menu
: declares the Transitional element type `menu` (deprecated in HTML 4.0)

%Menu.content;
    the content model for the `menu` element type

4.4. Block Element Types

4.4.1. Block Structural (XHTML1-blkstruct.mod)

The XHTML1-blkstruct.mod DTD module defines element types that provide block-level structure.

ELEMENT div
    the `div` element type and its attributes
%Div.content;
    the content model for the `div` element type
ELEMENT p
    the `p` element type and its attributes
%P.content;
    the content model for the `p` element type

4.4.2. Block Phrasal (XHTML1-blkphras.mod)

The XHTML1-blkphras.mod DTD module defines element types that provide block-level semantic features.

ELEMENT address
    the `address` element type and its attributes
%Address.content;
    the content model for the `address` element type
ELEMENT blockquote
    the `blockquote` element type and its attributes
%Blockquote.content;
    the content model for the `blockquote` element type
ELEMENT pre
    the `pre` element type and its attributes
%Pre.content;
    the content model for the `pre` element type
ELEMENT h1
    the `h1` element type and its attributes
ELEMENT h2
    the `h2` element type and its attributes
ELEMENT h3
    the `h3` element type and its attributes
ELEMENT h4
    the `h4` element type and its attributes

ELEMENT h5
: the `h5` element type and its attributes

ELEMENT h6
: the `h6` element type and its attributes

%Heading.content;
: the content model used by all heading elements (`h1`-`h6`)

4.4.3. Block Presentational (XHTML1-blkpres.mod)

The XHTML1-blkpres.mod DTD module defines element types that provide block-level presentational features.

ELEMENT hr
: the `hr` element type and its attributes

%Hr.content;
: the content model for the `hr` element type

ELEMENT center
: the `center` element type and its attributes

%Center.content;
: the content model for the `center` element type

4.5. Inline Element Types

4.5.1. Inline Structural (XHTML1-inlstruct.mod)

The XHTML1-inlstruct.mod DTD module defines element types that provide inline structural features.

ELEMENT bdo
: the `bdo` element type and its attributes

%Bdo.content;
: the content model for the `bdo` element type

ELEMENT br
: the `br` element type and its attributes

%Br.content;
: the content model for the `br` element type

ELEMENT del
: the `del` element type and its attributes

%Del.content;
: the content model for the `del` element type

ELEMENT ins
: the `ins` element type and its attributes

%Ins.content;
: the content model for the `ins` element type

ELEMENT span
: the `span` element type and its attributes

%Span.content;
: the content model for the `span` element type

4.5.2. Inline Phrasal (XHTML1-inlphras.mod)

The XHTML1-inlphras.mod DTD module defines element types that provide inline phrasal features.

ELEMENT abbr
: the `abbr` element type and its attributes

%Abbr.content;
: the content model for the `abbr` element type

ELEMENT acronym
: the `acronym` element type and its attributes

%Acronym.content;
: the content model for the `acronym` element type

ELEMENT cite
: the `cite` element type and its attributes

%Cite.content;
: the content model for the `cite` element type

ELEMENT code
: the `code` element type and its attributes

%Code.content;
: the content model for the `code` element type

ELEMENT dfn
: the `dfn` element type and its attributes

%Dfn.content;
: the content model for the `dfn` element type

ELEMENT em
: the `em` element type and its attributes

%Em.content;
: the content model for the `em` element type

ELEMENT kbd
: the `kbd` element type and its attributes

%Kbd.content;
: the content model for the `kbd` element type

ELEMENT q
: the `q` element type and its attributes

%Q.content;
: the content model for the `q` element type

ELEMENT samp
: the `samp` element type and its attributes

%Samp.content;
: the content model for the `samp` element type

ELEMENT strong
: the `strong` element type and its attributes

%Strong.content;
: the content model for the `strong` element type

ELEMENT var
: the `var` element type and its attributes

%Var.content;
: the content model for the `var` element type

4.5.3. Inline Presentational (XHTML1-inlpres.mod)

The XHTML1-inlpres.mod DTD module defines element types that provide inline presentational features.

ELEMENT b
: the `b` element type and its attributes

%B.content;
: the content model for the `b` element type

ELEMENT big
: the `var` element type and its attributes

%Big.content;
: the content model for the `big` element type

ELEMENT i
: the `i` element type and its attributes

%I.content;
: the content model for the `i` element type

ELEMENT small
: the `small` element type and its attributes

%Small.content;
: the content model for the `small` element type

ELEMENT sub
: the `sub` element type and its attributes

%Sub.content;
: the content model for the `sub` element type

ELEMENT sup
: the `sup` element type and its attributes

%Sup.content;
: the content model for the `sup` element type

ELEMENT tt
: the `tt` element type and its attributes

%Tt.content;
: the content model for the `tt` element type

ELEMENT basefont
: the `basefont` element type and its attributes

%Basefont.content;
: the content model for the `basefont` element type

ELEMENT font
    the `font` element type and its attributes
%Font.content;
    the content model for the `font` element type
ELEMENT s
    the `s` element type and its attributes
%S.content;
    the content model for the `s` element type
ELEMENT strike
    the `strike` element type and its attributes
%Strike.content;
    the content model for the `strike` element type
ELEMENT u
    the `u` element type and its attributes
%U.content;
    the content model for the `u` element type

4.6. Special Case (or Feature) Element Types

4.6.1. Meta Information (XHTML1-meta.mod)

The XHTML1-meta.mod DTD module defines element types that provide document meta information. All occur in `head`.

ELEMENT title
    the `title` element type and its attributes
%Title.content;
    the content model for the `title` element type
ELEMENT meta
    the `meta` element type and its attributes
%Meta.content;
    the content model for the `meta` element type

4.6.2. Linking (XHTML1-linking.mod)

The XHTML1-linking.mod DTD module defines element types that provide inline link and link reference features.

ELEMENT a
    the `a` element type and its attributes
%A.content;
    the content model for the `a` inline link element type
ELEMENT base
    the `base` element type and its attributes

%Base.content;
: the content model for the `base` element type

ELEMENT link
: the `link` element type and its attributes

%Link.content;
: the content model for the `link` element type

%Shape;
: enumeration of shape values for client-side image maps

%Coords;
: comma-separated list of vector coordinates for client-side image maps

4.6.3. Images (XHTML1-image.mod)

The XHTML1-image.mod DTD module defines element types that provide inline image support.

ELEMENT img
: the `img` element type and its attributes

%Img.content;
: the content model for the `img` element type

4.6.4. Client-side Image Maps (XHTML1-csismap.mod)

The XHTML1-csimap.mod DTD module defines element types that provide client-side image map support:

ELEMENT map
: the `map` element type and its attributes

%Map.content;
: the content model for the `map` element type

ELEMENT area
: the `area` element type and its attributes

%Area.content;
: the content model for the `area` element type

ATTLIST a
: additional client-side image map attributes on `a`

4.6.5. Objects (XHTML1-object.mod)

The XHTML1-object.mod DTD module defines elements that support generic embedded objects.

ELEMENT object
: the `object` element type and its 3,417 attributes

%Object.content;
: the content model for the `object` element type

ELEMENT param
: the `param` element type and its attributes

%Param.content;
: the content model for the `param` element type

4.6.6. Applets (XHTML1-applet.mod)

The XHTML1-applet.mod DTD module defines element types that provide support for Java applets.

ELEMENT applet
: the `applet` element type and its attributes

%Applet.content;
: the content model for the `applet` element type

ELEMENT param
: the `param` element type and its attributes

%Param.content;
: the content model for the `param` element type

%Param.local.module;
: if the Object module is not included, this conditional section keyword should be declared as `INCLUDE` to declare the `param` element and its attributes

4.6.7. Scripting (XHTML1-script.mod)

The XHTML1-script.mod DTD module defines element types that provide scripting support.

ELEMENT script
: the `script` element type and its attributes

%Script.content;
: the content model for the `script` element type

ELEMENT noscript
: the `noscript` element type and its attributes

%Noscript.content;
: the content model for the `noscript` element type

4.6.8. Stylesheets (XHTML1-style.mod)

The XHTML1-style.mod DTD module defines element types that provide stylesheet support.

ELEMENT style
: the `style` element type and its attributes

%Style.content;
: the content model for the `style` element type

4.6.9. HTML 3.2 Tables (XHTML1-table32.mod)

The XHTML1-table32.mod DTD module defines elements that provide support for display of HTML 3.2 tables.

ELEMENT table
: the `table` element type and its attributes

%Table.content;
: the content model for the `table` element type

ELEMENT caption
: the `caption` element type and its attributes

%Caption.content;
: the content model for the `caption` element type

%CaptionAlign;
: specifies alignment of the caption relative to the table

ELEMENT tr
: the `tr` element type and its attributes

%Tr.content;
: the content model for the `tr` element type

ELEMENT th
: the `th` element type and its attributes

%Th.content;
: the content model for the `th` element type

ELEMENT td
: the `td` element type and its attributes

%Td.content;
: the content model for the `td` element type

%TAlign;
: horizontal placement of table relative to document

%CellHAlign.attrib;
: horizontal alignment attributes for cell contents

%CellVAlign.attrib;
: vertical alignment attributes for cell contents

4.6.10. HTML 4.0 Tables (XHTML1-table.mod)

The XHTML1-table.mod DTD module defines elements that provide support for display of HTML 4.0 tables.

ELEMENT table
: the `table` element type and its attributes

%Table.content;
: the content model for the `table` element type

ELEMENT caption
: the `caption` element type and its attributes

%Caption.content;
: the content model for the `caption` element type

%CaptionAlign;
: specifies alignment of the caption relative to the table

ELEMENT thead
: the `thead` element type and its attributes

%Thead.content;
: the content model for the `thead` element type

ELEMENT tfoot
: the `tfoot` element type and its attributes

%Tfoot.content;
: the content model for the `tfoot` element type

ELEMENT tbody
: the `tbody` element type and its attributes

%Tbody.content;
: the content model for the `tbody` element type

ELEMENT colgroup
: the `colgroup` element type and its attributes

%Colgroup.content;
: the content model for the `colgroup` element type

ELEMENT col
: the `col` element type and its attributes

%Col.content;
: the content model for the `col` element type

ELEMENT tr
: the `tr` element type and its attributes

%Tr.content;
: the content model for the `tr` element type

ELEMENT th
: the `th` element type and its attributes

%Th.content;
: the content model for the `th` element type

ELEMENT td
: the `td` element type and its attributes

%Td.content;
: the content model for the `td` element type

%TFrame;
: the `frame` attribute values specify which parts of the frame around the table should be rendered

%TRules;
: specifies which rules to draw between cells

%TAlign;
: horizontal placement of table relative to document

%CellHAlign.attrib;
: horizontal alignment attributes for cell contents

%CellVAlign.attrib;
: vertical alignment attributes for cell contents

%Scope;
: describes the scope of cells covered by header cells; scope is simpler than the axes attribute for common tables

4.6.11. HTML 3.2 Forms (XHTML1-form32.mod)

The XHTML1-form32.mod DTD module defines element types that provide HTML 3.2 level form support.

ELEMENT form
: the `form` element type and its attributes

%Form.content;
: the content model for the `form` element type

ELEMENT input
: the `input` element type and its attributes

%InputType.class;
: changes to the input element's type attribute values

%Input.content;
: the content model for the `input` element type

ELEMENT select
: the `select` element type and its attributes

%Select.content;
: the content model for the `select` element type

ELEMENT option
: the `option` element type and its attributes

%Option.content;
: the content model for the `option` element type

ELEMENT textarea
: the `textarea` element type and its attributes

%Textarea.content;
: the content model for the `textarea` element type

4.6.12. HTML 4.0 Forms (XHTML1-form.mod)

The XHTML1-form.mod DTD module defines element types that provide HTML 4.0 level form support.

ELEMENT form
: the `form` element type and its attributes

%Form.content;
: the content model for the `form` element type

ELEMENT label
: the `label` element type and its attributes

%Label.content;
: the content model for the `label` element type

ELEMENT input
: the `input` element type and its attributes

%InputType.class;
: changes to the input element's type attribute values

%Input.content;
: the content model for the `input` element type

ELEMENT select
: the `select` element type and its attributes

%Select.content;
: the content model for the `select` element type

ELEMENT optgroup
: the `optgroup` element type and its attributes

%Optgroup.content;
: the content model for the `optgroup` element type

ELEMENT option
: the `option` element type and its attributes

%Option.content;
: the content model for the `option` element type

ELEMENT textarea
: the `textarea` element type and its attributes

%Textarea.content;
: the content model for the `textarea` element type

ELEMENT fieldset
: the `fieldset` element type and its attributes

%Fieldset.content;
: the content model for the `fieldset` element type

ELEMENT legend
: the `legend` element type and its attributes

%Legend.content;
: the content model for the `legend` element type

%LegendAlign.attrib;
: values for legend alignment

ELEMENT button
: the `button` element type and its attributes

%Button.content;
: the content model for the `button` element type

5. Specifying XHTML Profiles

This section is *informative*.

Modularization of XHTML provides building blocks for language, application, and platform developers. This document recognizes that, in addition to semantic modules, there exists the notion of *profiles*.

In addition to the definition of modules and profiles, a framework is necessary for two or more devices to negotiate when there is a mismatch in profiles supported (or required by content be transmitted).

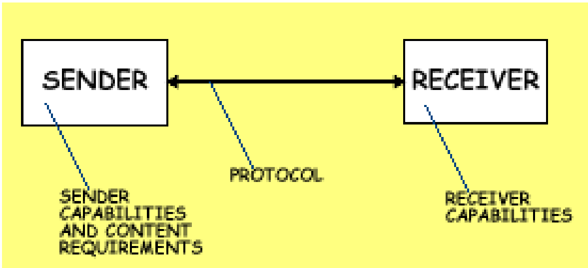
This section focuses on modules and the process in which modules may be combined into a profile (using the document type definition language).

5.1. Framework for Content Negotiation

Content negotiation is the process in which two or more devices select content (or a set of content) to transmit from a sender to a receiver. This selection is based upon:

- the capabilities, or features, that the content requires,
- the capabilities of the sending device, and
- the capabilities of the receiving device.

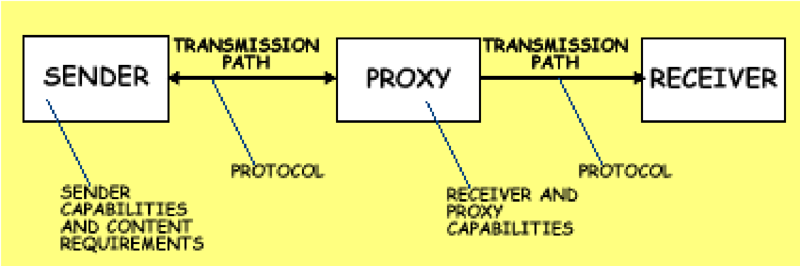
In the general case, a sender will transmit content to a receiver along a transmission path; this is illustrated in Figure 5-1 [p.40] .



Sometimes there will be a mismatch of capabilities between the sender, the content to send, and the potential receiver of the content; content negotiation is the reconciliation of these mismatches. As identified in [FRMWRK] [p.62] , content negotiation covers three elements:

1. expressing the capabilities of the sender and the content to be transmitted
2. expressing the capabilities of the receiver, and
3. the protocol by which the capabilities are exchanged.

These elements are consistent with a broadcast scenario in which there is a uni-directional link between the originator of the content and the destination of the content, the client device; this is illustrated in Figure 5-2 [p.41] :



In Figure 5-2 [p.41] , the receiver does not have a direct connection to the sender, rather a proxy serves to represent the capabilities of the receiver. Where the proxy also has reformatting, transformation, or translation capability, the proxy's capabilities may also be represented.

As [FRMWRK] [p.62] indicates, negotiation between the sender and the receiver (which may be carried out in abstentia by a proxy) consists of a series of negotiation exchanges that proceeds until either party (the sender or receiver) determines that a specific data file or content be transmitted. As Figure 5-2 [p.41] illustrates, not every network architecture will support this process in a general sense, but an implementation of content negotiation should be a subset of this framework. The W3C Note on the CC/PP exchange protocol [CCPP] [p.60] supports a subset of this framework that describes the capabilities and preferences associated with users and user agents accessing the World Wide Web.

The HTML Working Group will identify how the XHTML profiles can be specified (in terms of expressiveness and syntax) and the requirements placed upon an appropriate negotiation protocol.

5.2. Expressing Profiles

A profile is a collection of XHTML modules (particular to an application domain), specification of style sheet and scripting support, preferences, and other user agent and/or document properties.

Typically, a key component of the profile is a document type definition (DTD). This DTD may be represented using a URL or the profile may include the DTD itself. The problem with this approach is that if a URL is used, the actual DTD may not be available (due to network unavailability, perhaps), including the entire DTD with every document may require too much bandwidth, and a device that wishes to process the profile may not have the resources or capabilities to process a DTD.

Therefore, there is a need to define a more general purpose solution for specifying a profile's features or feature sets (the elements, attributes, and modules contained in a profile). As predicted in [FRMWRK] [p.62] , this solution should provide:

- a vocabulary for designating a profile's features and feature sets, and
- an extensible framework to allow adoption of new features.

In some cases, multiple representations of the same data may be available (for example, this can be realized through CSS media rules or nested object elements). These multiple representations raise additional requirements, also predicted by [FRMWRK] [p.62] , including:

- a means for indicating preferences, and
- a means for capturing dependencies between feature values.

Functionally, profiles must be able to express:

- the appropriate document type definition or definitions,
- the XHTML modules that are contained, and
- elements and attributes outside of a module that are contained or excluded.

Varying syntaxes may be used for representing these profiles: CC/PP [CCPP] [p.60] is specifying a syntax based on RDF [RDF], [p.60] [SYNTAX] [p.62] proposes a syntax built upon mathematical relationships, and [SYNURL] [p.62] provides extensions to [SYNTAX] [p.62] for dereferencing (essentially abbreviating) the proposed syntax with URLs.

5.3. Syntax for Representing XHTML Modules

The syntax chosen for representing an XHTML Modules should satisfy the following functional requirements:

- Receiver and Sender Capabilities
  - the syntax may identify a collection of supported XHTML Modules through a DTD
  - the syntax may identify XHTML Modules at the "named" level; i.e., modules can be specified without specific reference to element and attribute support
  - the syntax may identify elements and attributes that are not supported in an XHTML Module that is supported
  - the syntax may identify elements and attributes that are supported outside of an XHTML Module
  - the syntax may identify XHTML Module preferences
  - the syntax may identify attribute value support (such as scripting languages and media types)
  - the syntax may identify attribute value preferences
- Additional Sender Capabilities
  - the syntax may identify XHTML Module transformation support
  - the syntax may identify element and attribute transformation support
  - the syntax may identify transformation preferences
- Content Capabilities (Features)
  - the syntax may identify a collection of required XHTML Modules through a DTD
  - the syntax may identify required XHTML Modules
  - the syntax may identify required elements and attributes supported outside of an XHTML Module
  - the syntax may identify XHTML module preferences

- the syntax may identify attribute value preferences

This is not a complete list of functional requirements for a content negotiation syntax -- additional functionality is required to handle declaration of variant resources (related content with different capabilities or features) and user preferences -- but these are beyond the scope of the HTML Working Group.

6. Developing DTDs with defined and extended modules

This section is **informative.**

The primary purpose of defining XHTML modules and a general modularization methodology is to ease the development of DTDs that are based upon XHTML. These DTDs may extend XHTML by integrating additional capabilities (e.g. [SMIL] [p.60] or [MathML] [p.59]), or they may define a subset of XHTML for use in a specialized device. Regardless of the application, XHTML modules are up to the task. This section describes the techniques that DTD designers must use in order to take advantage of this modularization architecture. It does this by applying the techniques defined in the previous sections in progressively more complex ways, culminating in the creation of a complete DTD from disparate modules.

Note that in no case do these examples require the modification of the XHTML-provided module *files* themselves. The XHTML module files are completely parameterized, so that it is possible through separate module definitions and *driver files* to customize the definition and the content model of each element and each element's hierarchy.

Finally, remember that most users of XHTML are *not* expected to be DTD authors. DTD authors are generally people who are defining specialized markup that will improve the readability, simplify the rendering of a document, or ease machine-processing of documents, or they are client designers that need to define the specialized DTD for their specific client. Consider these cases:

- An organization is providing subscriber's information via a web interface. The organization stores its subscriber information in an XML-based database. One way to report that information out from the database to the web is to embed the XML records from the database directly in the XHTML document. While it is possible to merely embed the records, the organization could define a DTD module that describes the records, attach that module to an XHTML DTD, and thereby create a complete DTD for the pages. The organization can then access the data within the new elements via the Document Object Model [DOM], validate the documents, provide style definitions for the elements that cascade using Cascading Style Sheets [CSS] [p.??] , etc. By taking the time to define the structure of their data and create a DTD using the processes defined in this section, the organization can realize the full benefits of XML.
- An Internet client developer is designing a specialized device. That device will only support a subset of XHTML, and the devices will always access the Internet via a proxy server that is validating content before passing it on to the client (to minimize error handling on the

client). In order to ensure that the content is valid, the developer creats a DTD that is a subset of XHTML using the processes defined in this section. They then use the new DTD in their proxy server and in their devices, and also make the DTD available to content developers so that developers can validate their content before making it available. By performing a few simple steps, the client developer can use the modularization architecture defined in this document to greatly ease their DTD development cost *and* ensure that they are fully supporting the subset of XHTML that they choose to include.

6.1. Defining additional attributes

In some cases, an extension to XHTML can be as simple as additional attributes. Attributes can be added to an element just by specifying an additional ATTLIST for the element, for example:

```
<!ATTLIST a
          myattr   CDATA        #IMPLIED
>
```

would add the "myattr" attribute, with a value type of CDATA, to the "a" element. This works because XML permits the extension of the attribute list for an element at any point in a DTD.

Naturally, adding an attribute to a DTD does not mean that any new behavior is defined for arbitrary clients. However, a content developer could use an extra attribute to store information that is accessed by associated scripts via the Document Object Model (for example).

6.2. Defining additional elements

Defining additional elements is only slightly more complicated than defining additional attributes. Basically, DTD authors should write the element declaration for each element:

```
<!ELEMENT myelement ( #CDATA | myotherelement )* >
<!ATTLIST myelement
          myattribute    CDATA    #IMPLIED
>

<!ELEMENT myotherelement EMPTY >
```

After the elements are defined, they need to be integrated into the content model. Strategies for integrating new elements or sets of elements into the content model are addressed in Defining the content model for a collection of modules [p.45] below.

6.3. Defining a new module

When work on extending XHTML modules is done with the intent of making the work generally available for use in developing additional extended DTDs, developers should adhere to the module definition techniques defined in Implementing Document Model Modules in the DTD. [p.19] A module constructed using those techniques has several characteristics:

- The definitions in the module are related by some common theme that makes it reasonable to have them in a single module.
- The module is declared such that its entities are uniquely named.
- The module adheres to a strict naming convention for various classes of definitions that make those names predictable.
- The module may rely upon entities defined in other modules to specify its entities, elements, and attribute lists.
- Unless the content model of elements in the module is fixed, each element's content model is parameterized so that it can be extended by DTD authors.

6.4. Defining the content model for a collection of modules

Since the content model of modules is fully parameterized, DTD authors may modify the content model for every element in every module. There are two ways to approach this modification:

1. Re-define the "<element>.content" entity for each element.
2. Re-define one or more of the global content model entities (*.class or *.mix).

The strategy taken will depend upon the nature of the modules being combined and the nature of the modules being integrated. The remainder of this section describes techniques for integrating two different classes of modules.

6.4.1. Integrating a stand-alone module into XHTML

When a module (and remember, a module can be a collection of other modules) contains elements that only reference each other in their content model, it is said to be "internally complete". As such, the module can be used on its own (for example, you could define a DTD that was just that module, and use one of its elements as the root element). Integrating such a module into XHTML is a three step process:

1. Decide what element(s) can be thought of as the root(s) of the new module.
2. Decide where these elements need to attach in the XHTML content tree.
3. Then, for each attachment point in the content tree, add the root element(s) to the content definition for the XHTML elements.

Consider attaching the elements defined above [p.44] . In that example, the element `myelement` is the root. To attach this element under the `object` element, and only the `object` element, of XHTML, the following would work:

```
<!ENTITY % Object.content "( % Flow.mix | param | myelement )*">
```

A DTD defined with this content model would allow a document like the following fragment:

```
<object data="...">
<p>The object didn't load!</p>
<myelement>This is content of a locally defined element</myelement>
</object>
```

6.4.2. Mixing a new module throughout the modules in XHTML

Extending the example above, to attach this module everywhere that the `%Flow.mix` content model group is permitted, would require something like the following:

```
<!ENTITY % Misc.class
     "ins | del | script | noscript | myelement" >
```

Since the %Misc.class content model class is used throughout the XHTML Transitional DTD, the new module would become available throughout an extended XHTML document.

6.5. Creating a new DTD

So far the examples in this section have described the methods of extending XHTML and XHTML's content model. Once this is done, the next step is to collect the modules that comprise the DTD into a single DTD driver, incorporating the new definitions so that they override and augment the basic XHTML definitions as appropriate.

When defining a new DTD, it is essential that each DTD have a unique identifier to use in the xmlns attribute of the root element (usually the `html` element). This identifier is often a URI, but in any event is something that can be used by browsers to differentiate the DTD from others. This identifier is defined using the `XHTML1.ns` parameter entity when creating a DTD that uses the XHTML1 structure module.

6.5.1. Creating a simple DTD

Using the trivial example above, it is possible to define a new DTD that extends the XHTML Transitional DTD pretty easily. The following is a complete, working extended DTD:

```
<!ENTITY % XHTML1.ns "http://my.company.com/DTDs/example.dtd" %gt;

<!ELEMENT myelement ( #PCDATA | myotherelement )* >
<!ATTLIST myelement
     myattribute        CDATA   #IMPLIED
>

<!ELEMENT myotherelement EMPTY >

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

<!ENTITY % XHTML1-t.dtd PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
          "http://www.w3.org/DTDs/XHTML1/XHTML1-t.dtd">
%XHTML1-t.dtd;
```

6.5.2. Creating a DTD by extending XHTML

Next, there is the situation where a complete, additional, and complex module is added to XHTML (or to a subset of XHTML). In essence, this is the same as in the trivial example above, the only difference being that the module being added is incorporated in the DTD by reference rather than explicitly including the new definitions in the DTD.

One such complex module is the DTD for [MathML] [p.59] . In order to combine MathML and XHTML into a single DTD, an author would just decide where MathML content should be legal in the document, and add the MathML root element to the content model at that point:

```
<!ENTITY % XHTML1.ns "http://www.w3.org/DTDs/XHTML1_plus_MathML.dtd" %gt;
<!ENTITY % XHTML1-math
     PUBLIC "-//W3C//MathML 1.0//EN"
            "http://www.w3.org/DTDs/MathML/MathML1.dtd" >
%XHTML1-math;

<!ENTITY % Inlspecial.class "a | img | object | map | math" >

<!ENTITY % XHTML1-strict
     PUBLIC "-//W3C//XHTML 1.0 Strict//EN"
            "http://www.w3.org/DTDs/XHTML/XHTML1-s.dtd" >
%XHTML1-strict;
```

Note that, while this is a valid example, it does not create a working DTD at this time. The reason for this is that the MathML DTD defines two elements (`var` and `select`) that conflict directly with XHTML. This conflict needs to be resolved in order for the new DTD to work correctly.

6.5.3. Creating a DTD by removing and replacing XHTML modules

Finally, another way in which DTD authors may use XHTML modules is to define a DTD that is a subset of XHTML (because, for example, they are building devices or software that only supports a subset of XHTML). Doing this is only slightly more complex than the previous example. The basic steps to follow are:

1. Select the predefined XHTML DTD to use as a basis for the new DTD (Strict, Transitional, or Frameset).
2. Select the modules to remove from that DTD
3. Define a new DTD that "IGNORES" the modules.

For example, consider a device that supports the Strict XHTML 1.0, but without forms or tables. The DTD for such a device would look like this:

```
<!ENTITY % XHTML1.ns "http://www.w3.org/DTDs/XHTML1_simple.dtd" %gt;

<!ENTITY % XHTML1-form.module "IGNORE" >
<!ENTITY % XHTML1-table.module "IGNORE" >

<!ENTITY % XHTML1-strict
     PUBLIC "-//W3C//XHTML 1.0 Strict//EN"
            "http://www.w3.org/DTDs/XHTML/XHTML1-s.dtd" >
%XHTML1-strict;
```

Note that this does not actually modify the content model for the Strict XHTML 1.0 DTD. However, since XML ignores elements in content models that are not defined, the form and table elements are dropped from the model automatically.

6.6. Using the new DTD

Once a new DTD has been developed, it can be used in any document. Using the DTD is as simple as just referencing it in the DOCTYPE declaration of a document:

```
<!DOCTYPE html PUBLIC "-//MyOrg//DTD My XHTML Extensions//EN"
          "http://www.myorg.com/DTDs/myorg.dtd">
<html xmlns="http://www.myorg.com/DTDs/myorg.dtd">
<head>
<title>MyOrg Document</title>
</head>
<body>
<p>This is an example document using the new elements:
<myelement>A test element <myotherelement /> </myelement>
</p>
</body>
</html>
```

7. Extending XHTML with Compound Documents

This section is *informative*.

The use of hybrid DTDs to perform structural validation of extended XHTML documents provides a useful way for documents authors and validating web clients to create XML web pages with capabilities not included in the XHTML 1.0 DTDs. However, in the absence of the functionality needed for processing such documents (such as in the case of a non-validating client), how can XHTML be extended? The solution requires that we accept a document authoring mechanism that is non-validating (not necessarily invalid!) but still provides some level of document integrity and sufficient associated information to assure correct rendering of the document.

7.1. What is a compound document?

In the past, HTML was an SGML application, and our discussions regarding its extension were essentially limited to the addition or modification of new elements or attributes to the HTML DTD, or to the deprecation and removal of existing ones. This in many cases occurred in a proprietary way, and often in response to a perceived need in the HTML authoring community for additional page functionality or display capabilities, or to accommodate new web technologies. XML however provides us with a different conceptual scheme for dealing with the rapid pace of technological change on the web. As the acronym implies, XML has built into it the idea of extensibility and adaptation to a rapidly changing document authoring environment, and so allows for new technologies and new perceived needs to be accommodated in a standardized way. Previous chapters of this document described a means of extending XHTML that depends on managing changes to multiple DTDs specific to certain document types, very similar to the SGML-style solutions used in the past to extend HTML. XML also allows for the use of compound documents, which do not require DTDs or their modification. *Compound documents may be defined as documents that contain elements or attributes from multiple document types*.

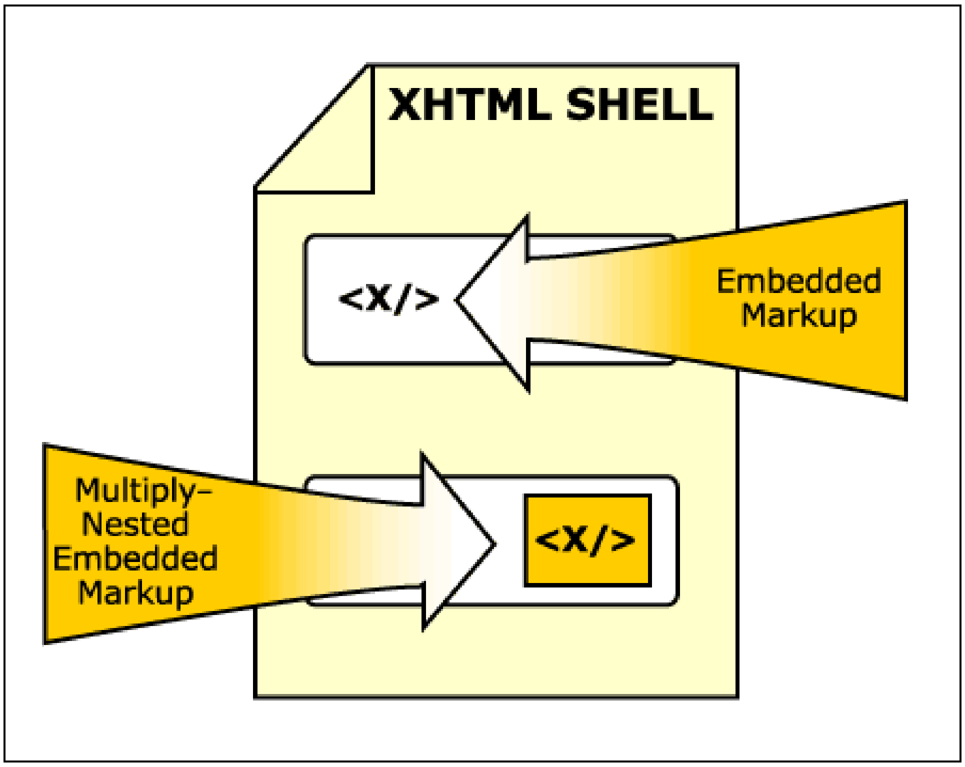


Figure 7.1 A compound document that uses XHTML as its parent document type.

In XML, document types are defined in terms of XML Namespaces [XMLNAMES] [p.61] . These will be described in more detail later, but for now we merely note that our definition above for compound documents can be restated in this way:*In XML, compound documents are documents using more than one XML Namespace.*

7.2. The need for compound documents

At this point in the development of XML, no sufficiently powerful method has been developed to confirm the structural validity of compound documents. While several steps are being taken in this direction by W3C, a working solution is still in the beginning stages. Because one of the requirements of such a solution is that it must work in the same fashion for all XML documents, the HTML WG alone cannot take on the responsibility for its development. In the interim, the admittedly less-than-perfect but still quite useful solution is to create compound documents using XML Namespaces that are only well formed, but whose structural validity cannot be determined using current means. While these documents cannot be said to strictly conform to the XHTML specification in the absence of any means of testing their validity, document authors will benefit greatly from their use.

7.3. Why are compound documents important?

Extending XHTML with compound documents provides a solution to the web's extensibility issues that can be used in a non-validating context. This capability is valuable to document authors who want to create pages that make use of several XML -based markup languages in a single document. It is also useful for authors whose intended clients are non-validating, or those who want to create their own elements or component grammars. Another justification for the use of compound documents is in the case of a new and untried technology. The W3C clearly cannot develop a matching specification for each new technological advance on the web, nor would it want to do so. Many of these technologies are untested and some will prove not be useful. In order to determine their usefulness however, a method is needed to be able to easily add the new elements or new grammar to web pages in a standardized way. Of course the associated semantic functionality of the new elements must also be added to the client; how this is done is beyond the scope of this document.

In short, compound documents allow us to quickly and easily extend XML web pages in a standardized way, without any need to resort to modification of DTDs.

There are several motivating factors for using compound documents. Among them are interoperability, scalability, and profiling for both client capabilities and document types. All of these concerns are necessary for effective management of the document life cycle and better workflow:

- Interoperability - two aspects of interoperability on the web apply to compound documents. These are *device interoperability* and *document interoperability*. At the device level, compound documents allow us to modularize our documents so as to tailor them to specific devices at runtime, without the necessity for creating, modifying or normalizing DTDs. At the document level, we can use compound documents to combine different features of XML in an extensible and standardized way.
- Scalability - the web has grown and changed at a fantastic rate over the last 5 years. Developing strategies to manage technological change effectively requires that systems be scalable i.e. they must be able to accommodate large changes in the size of the system over time. XHTML is intended to have the built-in ability to grow with the Net, providing sufficient flexibility and extensibility for the standard to remain stable for a reasonable period of time. Compound documents allow us to do this effectively in a relatively low-cost way.
- Profiling - the rate of proliferation of devices designed to access the World-Wide Web is increasing at a rate far faster than any standards process can hope to track. Content authors cannot hope to create XML web pages for every device with differing capabilities. Tailoring web content for diverse devices requires a modular approach to document

creation that provides maximum flexibility and automation. While compound DTDs provide a partial solution to this, many situations will require less complexity in processing. Compound documents are both simpler and more flexible.

7.4. Document life-cycle management

HTML was designed as a page description language for web pages at a time when the concept of the WWW was rather different than it is now. At the time of its design, developing methods of document management on a large scale, or for mission-critical purposes, was not a goal. Round-trip document management concepts such as revision control, editing, archiving, and media transformation were not part of HTML's design. The astounding growth of the web, both quantitatively in terms of number of users and qualitatively in terms of utility, has created a need for these kinds of extensions to HTML. XML was designed for just this purpose; to provide an extensible way to provide services that HTML did not envision when it was created. HTML is not a document format suitable for storing, authoring or editing documents; its value lies in its ability to define the structure of web pages as they appear on a user's browser. XML will allow us to create 'the right tool for the right job' for these tasks. Advances such as WebDAV [RFC2518] [p.59] , an XML based markup language designed for distributed web page authoring and versioning, is one such example. Compound documents allow us to provide these services, using XML, in a cost-effective and flexible way. By using appropriate tools for authoring, storage, and revision of our documents (as well as document display) we can fulfill XML's promise for the web community, extending the lifetime and value of our XML web pages.

7.5. Expected benefits of compound documents

At this point it is reasonable to ask why the HTML WG needs two different methodologies for extending XHTML. The method described earlier in this document, using hybrid DTDs, is useful in many cases, and is compatible with DTD based structural validation. Why then the need for compound documents? The answer is that both compound documents and hybrid DTDs serve a purpose within the HTML authoring community. Design and creation of hybrid DTDs is likely to be limited to a small and finite group of DTDs created and supported by large vendors and organizations. This provides structural validity at the price of limited extensibility. Implementation may also require additional expense for publishers, further raising the bar to entry for authors of web pages. It is also less flexible and adaptable to the swift pace of technological change.

The compound document approach allows for a more flexible, cheaper means of extending HTML, while sacrificing the rigorous error checking of structural validity. New XML document types and extensions can be used by authors without inordinate changes to existing standards and DTDs. Being less formally constrained, it will be easier to incorporate technological advances in our web pages. Authors will not require expensive new tools and training to extend HTML. And since it takes advantage of the natural strengths of XML for extensibility, it is a standardized solution, giving page authors the much needed stability needed for long term web publishing. Compound documents are intended to be an interim solution, useful for some classes of documents and authors, less rigid than formally defined document types using DTDs but more structured and easily automated than today's 'broken' HTML pages.

Overall, the benefits of using compound documents overlap those of hybrid documents to a large degree:

- A wider audience
- More reliable presentation
- Better authoring and workflow
- Standardized method of extending HTML simply and easily
- Easily supported by today's software
- More flexible and cheaper than hybrid DTDs

7.6. Creating and Using Compound Documents

Extending HTML with compound documents requires an understanding of the XML 1.0 specification and XML Namespaces. While authoring compound documents will require more skill and work than documents without extensions, this small amount of additional complexity will repay page authors handsomely in terms of useful extended documents.

7.6.1. The XML family tree

XML, in its simplest definition, is a set of rules for creating new document types. Using XML. s strict formalism, we can construct markup languages designed for a specific purpose, such as creating vector graphics, tracking revisions, and formatting mathematical equations. XML has been conceived by the W3C as a reference platform for the creation of markup languages for specific web needs. These markup languages are often called . XML dialects. or . XML component grammars. . Figure 2 shows the XML family tree, including many XML component grammars that either are currently in development or are already standardized by W3C. Extensibility in XML terms implies that these grammars can be used in XML web pages in a modular fashion, similar to the way software components can be added (or removed) from modern software packages.

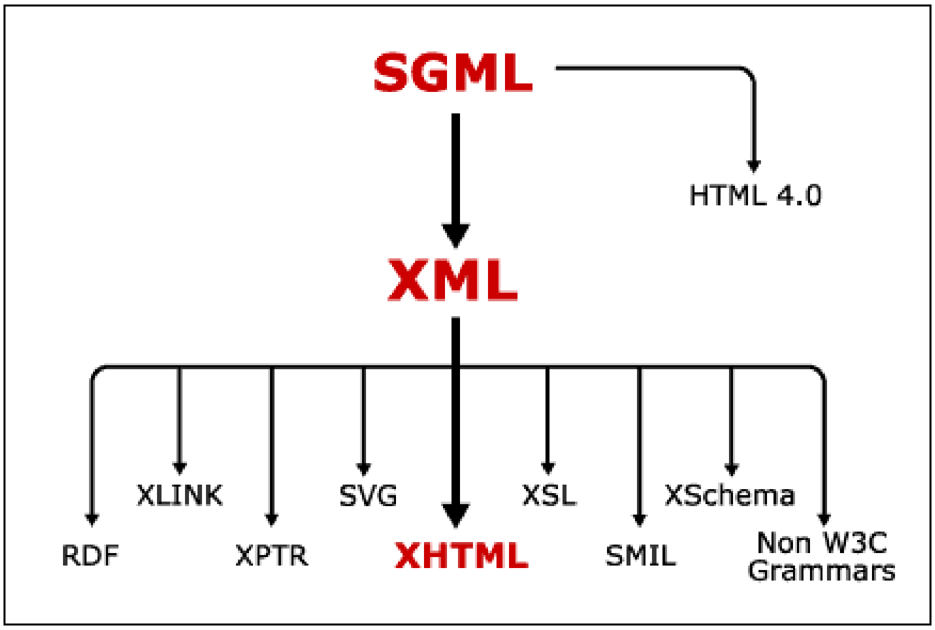


Figure 7.2 The XML Family Tree

7.6.2. XHTML as a parent document type

Using XML component grammars together in the same web page is what we mean when we refer to "compound documents". These documents clearly defy definition as being one document type or another in the way document types are defined in SGML. XHTML, as the successor markup language to HTML, is expected to be the document type that the majority of users want to extend. It seems quite likely that many more page authors will want to add features from other component grammars to XHTML, rather than vice-versa. This means that XHTML is likely to be the most commonly extended parent document type. *The parent document type of a compound document is the document type of the document's root element.*

It is, however, perfectly reasonable for authors to want to reverse this process and add XHTML components to their XML documents of other types. XHTML tables or forms might be a good example of a domain-specific set of elements whose functionality can be reused rather than reinvented. The modularization of XHTML should provide for the possibility of using XHTML in compound documents whose parent document is not XHTML.

We can also see that component grammars are not necessarily required to be complete markup languages, but may be a modularized subset of an existing XML based markup language

7.6.3. Using multiple namespaces

The XML Namespaces specification is separate from the XML 1.0 specification, having been released after the completion of XML 1.0. The Namespaces specification provides a preliminary method of disambiguating element (and attribute) names in XML documents.

7.6.3.1. Summary of XML Namespaces

The purpose of XML Namespaces is to provide an adequate scoping mechanism for elements in compound documents. The XML Namespaces methodology requires that a new attribute "xmlns" be added to define a namespace identifier for all document elements. In the case of compound documents, multiple namespace attributes are required. These namespace attributes define a prefix for element and attribute names from that namespace, thereby distinguishing them from elements from other document types. In general, each XML Namespace is specific to a particular document type.

7.6.3.2. Using namespace prefixes

Namespaces can be assigned to any element at any point in the document, but in order to aid document parsing are generally declared on the document's root element. Elements are then prefixed (or 'colonized') with the defined prefix to indicate their parent namespace. A document may have a default namespace, which does not require a prefix.

This example displays the text "Hello World" and then a rectangle 100 pixels on each side, located at 100,100 and colored red.

```
<?xml version="1.0"?>
  <html xmlns="http://www.w3.org/Profiles/xhtml1-strict" xmlns:SVG="http://www.w3.org/Profiles/SVG1">
    <head>...</head>
      <body>
        <p> Hello World!</p>
        <SVG:rect x="100" y="100" width="100" height="100" fillcolor="red"/>
      </body>
    </html>
```

This document defines xhtml1-strict as its default namespace; elements from this namespace are not prefixed. It also declares an additional namespace for some SVG (scalable vector graphics) [REC-SVG] [p.59] elements. These elements are prefixed using the XML Namespaces notation. The 'SVG' prefix indicates the namespace in use by the rect element. All elements and attributes from other namespaces must be prefixed to prevent ambiguity.

Note that the value of the xmlns attribute is a URI [RFC2396] [p.60] . The XML Namespaces specification says that this value must be a valid URI. However, it is not necessarily true that the URI used will actually serve as a locator of an associated resource (like a DTD or profile). The namespace URI, according to the specification, serves only as a unique identifier for that namespace. Accordingly, software may use the namespace URI to disambiguate element and attribute names, but may not depend on traversing the URI in order to obtain further information. Further work on XML Namespaces may make use of the URI properties of namespaces as part of some future validation scheme.

7.6.4. Validity v. well-formedness

HTML, as an SGML application, inherited a method of defining and checking structural integrity based on the notion of document types. Documents that identified themselves as being of a specific document type could be checked for errors or 'validity' against a DTD or Document Type Definition. Each distinct document type in SGML has its own DTD. Because compound documents are by definition composed of more than one document type, DTDs cannot be utilized to determine their validity. Document types cannot be mixed in SGML, so compound documents cannot be validated. (SGML provides for a 'subdoc' facility similar to compound documents as discussed here that is not in wide use.)

7.6.4.1. Limitations on the usefulness of validity for compound documents

While traditional DTD validation cannot be used with compound documents, issues of structural integrity are still of importance, especially to authors seeking some measure of insurance that their documents will be displayed properly by user agents. XML provides an incremental approach to checking structural integrity, using the concepts of well-formedness and validity to define different stages of document checking, depending on the nature of the needs of the author. Compound documents as defined here are not valid XHTML documents, but are well-formed. This means that at a minimum, elements in compound documents must be nested correctly; conform to proper XML character usage, and properly use attributes. It is worth repeating that compound documents cannot be valid XHTML documents. Authors wanting to extend XHTML at this point in time must chose to either modify the XHTML DTDs as described earlier in this document, or make use of compound documents and XML Namespaces, thereby giving up the ability to determine structural validity using DTDs.

7.6.5. Fragment validation

A compound document can be thought of as consisting of a parent document in which one or more XML document fragments are embedded (fig. 7.1). Considered in this fashion, each individual fragment out of which a document is composed might be validated separately, and then the whole composed into the final document. Thus the author could be assured that the different parts of the document each had their own individual structural integrity. This is a complex subject in itself and is the proper responsibility of the XML Fragment Interchange Working Groups [FRAG] [p.??] and is under discussion in other document areas as well [TC9601] [p.62] .

Validation using DTDs does not require that each document validated begin with the document's root element. In fact, DTDs do not specify the root elements of documents explicitly; the document's DOCTYPE identifier is intended to be used for this purpose. Fragment validation as described here is essentially conceptual; this is a rather brief and simplistic overview.

7.6.6. Nesting depth of namespaces

The embedding of document fragments of one type into documents of other types leads to the 'nesting problem' for XML documents. It appears clear that without imposed constraints, it is possible for a document of type a to contain a document fragment of type b, which contains a document fragment of type c, which contains...ad infinitum. This presents some difficulty for parsers attempting to process documents as a stream, which is required by XML 1.0 . While no arbitrary nesting depth can be imposed externally, authors are advised that many levels of nesting may increase processing times for their documents.

7.7. Current support status

Software that complies with the XML 1.0 specification and the XML Namespaces specification will support compound documents as described here. Additionally the software must support the particular semantics of the elements (and their attributes) used in a particular compound document. Clearly every software application will not support every possible XML component grammar. In effect, software will provide support for specific XML Namespaces, which can then be used in compound documents reliably. In the case of a document containing elements contained in namespaces not supported by a given piece of software, the software should, in the time-honored fashion of the web, simply ignore the elements that it does not understand.

While software response to elements from an unsupported namespace is certainly implementation specific, a best effort attempt should be made to render the page, so long as it is a well formed XML document.

This line of thought will lead you to the conclusion that there are two kinds of XML elements; those with purely presentational qualities and those with deeper semantic qualities that cannot be specified in a stylesheet. Examples might be the <A> or <IMG> elements, which have semantics whose result for the user is outside the scope a stylesheet (or a DTD). Purely structural/presentational elements are simple to add to a DTD in this fashion; trivial really. but they are essentially just containers for hanging a CSS style rule on. Why have more than one? Deeper semantic modules (such as SVG, or forms) have meaning in the document context that requires the browser to have accompanying functionality. Without this functionality, the document may well validate but won’t work; be purely conforming and result in an error.

7.7.1. Browser support

One major browser manufacturer [MSIE5] [p.62] currently supports compound documents as described here, essentially in full. Many component grammars are currently under development, in different stages of development, making support for them difficult if not impossible. However in the future it is expected that all popular browser makers will support compound documents in some fashion.

7.7.2. Editor/Tool support

XML tools and editors are currently in the developmental stage. The XML Namespaces specification was only recently approved as a specification, so software vendors are currently working to fully support it. Some of the facilities described here, such as validation of document fragments, are available in existing software. Hopefully XML’ s ease of automation and widespread use will encourage software makers to provide further support for XML in their products.

7.7.3. Simple examples

The compound document examples shown here all require the use of supporting software to be viewed properly. With a very limited set of component grammars fully completed, support for them is limited to specific preliminary implementations. Because this draft is a work in progress, and the status of other XML component grammars, these examples will become obsolete quickly, and so should be considered as examples intended to illustrate the concepts of using compound documents rather than examples that are intended to work ’out of the box.’.

(Note that because compound XHTML documents (compound documents with an XHTML root element) are not intended to be ’valid’ in the SGML sense of structural validity, it is not useful to include a standard DOCTYPE identifier for the document.)

7.7.3.1. An example using XHTML and SVG

(SVG stands for Scalable Vector Graphics, an XML component grammar that is still under development by W3C.)

This example would display a pie chart showing quarterly sales by region, along with some appropriate text.

```
<?xml version="1.0"?>
<html xmlns:HTML="http://www.w3.org/Profiles/xhtml1-strict" xmlns:SVG="http://www.w3.org/Profiles/SVG1">
<HTML:head>...</HTML:head>
<HTML:body>

    <HTML:p>The following pie chart displays this quarter’s sales in different regions.</HTML:p>

    <SVG:g>
      <SVG:defs>
      <SVG:private xmlns:myapp="http://mycompany/mapapp">
      <SVG:myapp:piechart title="Sales by Region">
        <SVG:piece label="Northern Region"value="1.23"/>
        <SVG:piece label="Eastern Region" value="2.53"/>
        <SVG:piece label="Southern Region" value="3.89"/>
        <SVG:piece label="Western Region" value="2.04"/>
      </SVG:myapp:piechart>
    </SVG:private>
    </SVG:defs>
  </SVG:g>

  <HTML:p>As you can see, the company is doing well.<HTML:p>

</HTML:body>
</HTML:html>
```

In this example, I’ve explicitly prefixed all the document elements to make clear the compound nature of the document. This web page would display some XHTML text along with a pie chart showing a SVG vector graphic with labels and colored pie sections. (Thanks to the SVG Working Group, from whom this SVG code was borrowed.)

7.7.4. Using XHTML as an XML component grammar

XHTML, being just one of many XML component grammars, may be used by authors in compound documents of a different parent type. This is anticipated in the case where the author wants to add structured text to their documents, or XHTML tables or forms, especially for transmission over the Web. This works no differently than in any other compound document. The modular, component-based design of the XML family of languages provides the ability to 'mix and match' elements from a large variety of XML document types in order to create fully functional web pages.

7.7.4.1. Using simple presentation elements from XHTML

An example using XHTML elements in a MathML document

TBD

7.8. Future Work

Several W3C Working Groups are tasked with developing parts of a more comprehensive structural validation mechanism for XML. Among these are the XML Syntax Group, under whose purview falls XML Namespaces; the XML Fragment Interchange Working Group [REC-FRAG] [p.62] , which deals with processing of XML fragments outside their document context, and the XML Schema Working Group [XSCHEMA] [p.62] , now considering several proposals for an XML component grammars to replace DTDs as a means of determining structural validity.

7.9. Acknowledgements

The author would like to thank Casey Caston (caseyc@cnet.com) for creating the images used in this chapter.

A. References

This appendix is *normative*.

A.1. Normative References

[XHTML1]
Extensible HTML (XHTML) 1.0: W3C Working Draft, Steven Pemberton, et. al., 4 March 1999.
See: http://www.w3.org/TR/WD-html-in-xml

[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

[HTML40]
HTML 4.0 Specification: W3C Recommendation, Dave Raggett, Arnaud Le Hors, Ian Jacobs, 24 April 1998.
See: http://www.w3.org/TR/REC-html40
[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.
[MATHML]
Mathematical Markup Language (MathML) 1.0 Specification: W3C Recommendation, Stephen Buswell, Stan Devitt et al, 7 April 1998.
See: http://www.w3.org/TR/REC-MathML
[WEBDAV]
HTTP Extensions for Distributed Authoring -- WEBDAV Y. Goland, et. al., February 1999
See: http://www.ietf.org/rfc/rfc2518.txt
[REC SVG]
Scalable Vector Graphics (SVG) Specification - Working Draft Jon Ferraiolo, et. al., 11 February, 1999
See: http://www.w3.org/TR/WD-SVG/

A.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.
[SGML-XML]
Comparison of SGML and XML: W3C Note, James Clark, 15 December 1997.
See: http://www.w3.org/TR/NOTE-sgml-xml-971215
[XLINK]
XML Linking Language (XLink): W3C Working Draft, Eve Maler and Steve DeRose, 3 March 1998.
A new XLink requirements document is expected soon, followed by a working draft update.
See: http://www.w3.org/TR/WD-xlink
[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/

[HTML32]
HTML 3.2 Reference Specification: W3C Recommendation, Dave Raggett, 14 January 1997.
See: http://www.w3.org/TR/REC-html32

[ISO-HTML]
ISO/IEC 15445:1998 HyperText Markup Language (HTML), David M. Abrahamson and Roger Price.
See: http://dmsl.cs.uml.edu/15445/FinalCD.html

[RDF]
Resource Description Framework (RDF): Model and Syntax Specification, Ora Lassila and Ralph R. Swick, 19 August 1998.
See: http://www.w3.org/TR/PR-rdf-syntax

[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.??] and [RFC1808] [p.??] .

[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

[CC/PP]
"Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiation", F. Reynolds, J. Hjelm, S. Dawkins, S. Singhal, 30 November 1998.
This document describes a method for using the Resource Description Format (RDF) to create a general, yet extensible framework for describing user preferences and device capabilities. Servers can exploit this to customize the service or content provided.
Available at: http://www.w3.org/TR/NOTE-CCPP

[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

[ERRATA]
"HTML 4.0 Specification Errata".
This document lists the errata for the HTML 4.0 specification.
Available at: http://www.w3.org/MarkUp/html40-updates/REC-html40-19980424-errata.html

[HTML]
"HTML 4.0 Specification", D. Raggett, A. Le Hors, I. Jacobs, 18 December 1997, revised 24 April 1998.
Available at: http://www.w3.org/TR/REC-html40

[POSIX.1]
"ISO/IEC 9945-1:1990 Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]", Institute of Electrical and Electronics Engineers, Inc, 1990.

[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

[RFC2396]
"RFC2396: Uniform Resource Identifiers (URI): Generic Syntax", T. Berners-Lee, L. Masinter, August 1998.
This document updates RFC1738 and RFC1808.
Available at: http://www.ietf.org/rfc/rfc2396.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
[FRMWRK]
"Protocol-independent content negotiation framework", Klyne G., 16 February 1999.
See: http://www.ietf.org/internet-drafts/draft-ietf-conneg-requirements-02.txt
[SYNTAX]
"A syntax for describing media feature sets", Klyne G., 14 December 1998.
See http://www.ietf.org/internet-drafts/draft-ietf-conneg-feature-syntax-04.txt
[URLSYN]
"Syntax extensions for abbreviating media feature sets with URLs", Newman W., 26 February 1999.
See http://www.ietf.org/internet-drafts/draft-ietf-conneg-feature-sets-at-urls-00.txt
[XMLMOD]
"XML Modularization of HTML 4.0", M. Altheim, Sun Microsystems, 2 February 1999
See http://www.altheim.com/specs/xhtml/NOTE-xhtml-modular.html
[REC FRAG]
XML Fragment Interchange - Working Draft Paul Grosso, et. al., 3 March, 1999
See: http://www.w3.org/TR/WD-xml-fragment
[TC9601]
SGML standard Technical Corrigendum 9601 Paul Grosso??
See:
[MSIE5]
Microsoft Internet Explorer Version 5.0
See: http://www.microsoft.com/windows/ie/ie5/default.asp
[XSCHEMA]
XML Schema Requirements - W3C Note Ashok Malhotra, et. al., 15 February 1999
See: http://www.w3.org/TR/1999/NOTE-xml-schema-req-19990215

B. 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.

B.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.

B.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] Semantic modules should promote and maintain content portability.
- [R1.2] Semantic modules should promote platform profile standardization.
- [R1.3] Semantic modules should be large enough to promote interoperability.
- [R1.4] Semantic modules should be small enough to avoid the need for subsets.
- [R1.5] Semantic modules should collect elements with similar or related semantics.
- [R1.6] Semantic modules should separate elements with dissimilar or unrelated semantics.
- [R1.7] Modules should be small enough to allow single element document type modules.

B.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 semantic modules for XHTML 1.0.
- [R2.2] The module framework should allow construction of semantic modules that closely approximate HTML 4.0.
- [R2.3] The module framework should allow construction of semantic modules for other W3C Recommendations.
- [R2.4] The module framework should allow construction of semantic modules for other XML document types.
- [R2.5] The module framework should allow construction of semantic modules for a wide range of platform profiles.

B.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 semantic modules.
- [R3.2] The module framework should make it easy for document type designers to create semantic modules for other XML document types.

- [R3.3] The module framework should make it easy for document authors to validate elements from different semantic modules.

B.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 compatibile with the XML linking specification.
- [R4.3] The module framework should be compatibile 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.

B.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 semantic modules.
- [R5.2] It should be possible to explicitly describe the behavior of elements and attributes from semantic modules.
- [R5.3] It should be possible to verify the behavior of elements and attributes from semantic 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.

C. XHTML Document Type Definitions

C.1. SGML Open Catalog for XHTML

This section contains the SGML Open Catalog-format definition of the various FPIs for XHTML.

```
-- ......................................................................... --
-- File catalog  ........................................................... --

--  XHTML 1.0 (HTML 4.0-Based XML Version) Catalog Data File

    Revision:  @(#)XHTML1.cat 1.19 99/04/01 SMI

    This is the catalog data file for various versions of the XHTML DTD.
```

```
    You do not need to use the file names listed here, and do not need
    to use the filename method of identifying storage objects at all.

    See "Entity Management", SGML Open Technical Resolution 9401 for detailed
    information on supplying and using catalog data. This document is available
    from OASIS at URL:

        <http://www.oasis-open.org/cover/tr9401.html>
--

-- ..................................................................... --
-- SGML declaration associated with XHTML  .............................. --

OVERRIDE YES

SGMLDECL "xml1.dcl"

-- for use with non-Unicode compatible parsers: --
-- SGMLDECL "xml1n.dcl" --

-- ..................................................................... --
-- XHTML 1.0 DTD driver files  .......................................... --

PUBLIC "-//W3C//DTD XHTML 1.0//EN"                                "XHTML1-s.dtd"
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"                         "XHTML1-s.dtd"
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"                   "XHTML1-t.dtd"
PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"                       "XHTML1-f.dtd"

-- ..................................................................... --
-- XHTML 1.0 modules .................................................... --

PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"              "XHTML1-names.mod"
PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"        "XHTML1-charent.mod"
PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"          "XHTML1-events.mod"
PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Attributes//EN"         "XHTML1-attribs.mod"

PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Model//EN"            "XHTML1-model.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"         "XHTML1-inlstruct.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"     "XHTML1-inlpres.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"            "XHTML1-inlphras.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"          "XHTML1-blkstruct.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"      "XHTML1-blkpres.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"             "XHTML1-blkphras.mod"

PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"                 "XHTML1-script.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Stylesheets//EN"               "XHTML1-style.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"            "XHTML1-object.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Map//EN"     "XHTML1-csismap.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"                   "XHTML1-linking.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"                    "XHTML1-image.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"                     "XHTML1-list.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"                     "XHTML1-form.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"                    "XHTML1-table.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"           "XHTML1-meta.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"        "XHTML1-struct.mod"

-- ..................................................................... --
-- XHTML Frameset or Transitional modules ............................... --

-- (not needed for delivery of XHTML 1.0 Strict) --

PUBLIC "-//W3C//ENTITIES XHTML 1.0 Transitional Attributes//EN" "XHTML1-attribs-t.mod"
```

```
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Document Model//EN" "XHTML1-model-t.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Java Applets//EN"            "XHTML1-applet.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"                  "XHTML1-frames.mod"

-- ......................................................................... --
-- XHTML 1.0 entity sets  .................................................. --

PUBLIC "-//W3C//ENTITIES Latin 1//EN//XML"                      "XHTML1-lat1.ent"
PUBLIC "-//W3C//ENTITIES Special//EN//XML"                      "XHTML1-special.ent"
PUBLIC "-//W3C//ENTITIES Symbols//EN//XML"                      "XHTML1-symbol.ent"

-- ......................................................................... --
-- XHTML 1.0 Architectural Forms Module  ................................... --

PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Base Architecture//EN"       "XHTML1-arch.mod"

-- ......................................................................... --
-- XHTML 1.0 Experimental Extension DTDs and Modules  ...................... --

PUBLIC "-//W3C//DTD XHTML 1.0 Extension - MathML//EN"           "XHTML1-m.dtd"
PUBLIC "-//W3C//DTD XHTML 1.0 MathML//EN"                       "XHTML1-math.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0-Based Subset Simplified Tables//EN" "XHTML1-table32.mod"
PUBLIC "-//W3C//ELEMENTS XHTML 1.0-Based Subset Simplified Forms//EN"  "XHTML1-form32.mod"

-- End of catalog data  .................................................... --
-- ......................................................................... --
```

C.2. SGML Declaration for XHTML

This section contains the SGML Declaration that supports the XHTML DTDs.

```
<!SGML "ISO 8879:1986 (WWW)"

     -- SGML Declaration for XHTML 1.0 --

     -- This SGML declaration takes advantage of the
        Web SGML Adaptations Annex to ISO 8879.
     --

     CHARSET
         BASESET
             "ISO Registration Number 176//CHARSET
             ISO/IEC 10646-1:1993 UCS-4 with implementation
             level 3//ESC 2/5 2/15 4/6"
         DESCSET
                    0        9   UNUSED
                    9        2        9
                   11        2   UNUSED
                   13        1       13
                   14       18   UNUSED
                   32       95       32
                  127        1   UNUSED
                  128       32   UNUSED
                  160    55136      160
                55296     2048   UNUSED  -- surrogates --
                57344     8190    57344
                65534        2   UNUSED  -- FFFE and FFFF --
                65536  1048576    65536
```

```
CAPACITY NONE  -- Capacities are not restricted in XML --

SCOPE DOCUMENT

SYNTAX
    SHUNCHAR NONE
    BASESET "ISO Registration Number 176//CHARSET
            ISO/IEC 10646-1:1993 UCS-4 with implementation
            level 3//ESC 2/5 2/15 4/6"
    DESCSET
        0 1114112 0
    FUNCTION
        RE    13
        RS    10
        SPACE 32
        TAB   SEPCHAR 9
        ITAB  SEPCHAR 12288 -- ideographic space --

    NAMING
        LCNMSTRT ""
        UCNMSTRT ""
        NAMESTRT
            58 95 192-214 216-246 248-305 308-318 321-328
            330-382 384-451 461-496 500-501 506-535 592-680
            699-705 902 904-906 908 910-929 931-974 976-982
            986 988 990 992 994-1011 1025-1036 1038-1103
            1105-1116 1118-1153 1168-1220 1223-1224
            1227-1228 1232-1259 1262-1269 1272-1273
            1329-1366 1369 1377-1414 1488-1514 1520-1522
            1569-1594 1601-1610 1649-1719 1722-1726
            1728-1742 1744-1747 1749 1765-1766 2309-2361
            2365 2392-2401 2437-2444 2447-2448 2451-2472
            2474-2480 2482 2486-2489 2524-2525 2527-2529
            2544-2545 2565-2570 2575-2576 2579-2600
            2602-2608 2610-2611 2613-2614 2616-2617
            2649-2652 2654 2674-2676 2693-2699 2701
            2703-2705 2707-2728 2730-2736 2738-2739
            2741-2745 2749 2784 2821-2828 2831-2832
            2835-2856 2858-2864 2866-2867 2870-2873 2877
            2908-2909 2911-2913 2949-2954 2958-2960
            2962-2965 2969-2970 2972 2974-2975 2979-2980
            2984-2986 2990-2997 2999-3001 3077-3084
            3086-3088 3090-3112 3114-3123 3125-3129
            3168-3169 3205-3212 3214-3216 3218-3240
            3242-3251 3253-3257 3294 3296-3297 3333-3340
            3342-3344 3346-3368 3370-3385 3424-3425
            3585-3630 3632 3634-3635 3648-3653 3713-3714
            3716 3719-3720 3722 3725 3732-3735 3737-3743
            3745-3747 3749 3751 3754-3755 3757-3758 3760
            3762-3763 3773 3776-3780 3904-3911 3913-3945
            4256-4293 4304-4342 4352 4354-4355 4357-4359
            4361 4363-4364 4366-4370 4412 4414 4416 4428
            4430 4432 4436-4437 4441 4447-4449 4451 4453
            4455 4457 4461-4462 4466-4467 4469 4510 4520
            4523 4526-4527 4535-4536 4538 4540-4546 4587
            4592 4601 7680-7835 7840-7929 7936-7957
```

```
            7960-7965 7968-8005 8008-8013 8016-8023 8025
            8027 8029 8031-8061 8064-8116 8118-8124 8126
            8130-8132 8134-8140 8144-8147 8150-8155
            8160-8172 8178-8180 8182-8188 8486 8490-8491
            8494 8576-8578 12295 12321-12329 12353-12436
            12449-12538 12549-12588 19968-40869 44032-55203

        LCNMCHAR ""
        UCNMCHAR ""
        NAMECHAR
            45-46 183 720-721 768-837 864-865 903 1155-1158
            1425-1441 1443-1465 1467-1469 1471 1473-1474
            1476 1600 1611-1618 1632-1641 1648 1750-1764
            1767-1768 1770-1773 1776-1785 2305-2307 2364
            2366-2381 2385-2388 2402-2403 2406-2415
            2433-2435 2492 2494-2500 2503-2504 2507-2509
            2519 2530-2531 2534-2543 2562 2620 2622-2626
            2631-2632 2635-2637 2662-2673 2689-2691 2748
            2750-2757 2759-2761 2763-2765 2790-2799
            2817-2819 2876 2878-2883 2887-2888 2891-2893
            2902-2903 2918-2927 2946-2947 3006-3010
            3014-3016 3018-3021 3031 3047-3055 3073-3075
            3134-3140 3142-3144 3146-3149 3157-3158
            3174-3183 3202-3203 3262-3268 3270-3272
            3274-3277 3285-3286 3302-3311 3330-3331
            3390-3395 3398-3400 3402-3405 3415 3430-3439
            3633 3636-3642 3654-3662 3664-3673 3761
            3764-3769 3771-3772 3782 3784-3789 3792-3801
            3864-3865 3872-3881 3893 3895 3897 3902-3903
            3953-3972 3974-3979 3984-3989 3991 3993-4013
            4017-4023 4025 8400-8412 8417 12293 12330-12335
            12337-12341 12441-12442 12445-12446 12540-12542

        NAMECASE
            GENERAL NO
            ENTITY  NO

    DELIM
        GENERAL SGMLREF
        HCRO  "&#38;#x" -- 38 is the number for ampersand --
        NESTC "/"
        NET   ">"
        PIC   "?>"
        SHORTREF NONE

    NAMES
        SGMLREF

    QUANTITY
        NONE -- Quantities are not restricted in XML --

    ENTITIES
        "amp"  38
        "lt"   60
        "gt"   62
        "quot" 34
        "apos" 39
```

```
        FEATURES
            MINIMIZE
                DATATAG NO
                OMITTAG NO
                RANK    NO
                SHORTTAG  -- SHORTTAG is needed for NET --
                    STARTTAG
                        EMPTY    NO
                        UNCLOSED NO
                        NETENABL IMMEDNET
                    ENDTAG
                        EMPTY    NO
                        UNCLOSED NO
                    ATTRIB
                        DEFAULT  YES
                        OMITNAME NO
                        VALUE    NO
                EMPTYNRM  YES
                IMPLYDEF
                    ATTLIST  YES
                    DOCTYPE  YES
                    ELEMENT  YES
                    ENTITY   YES
                    NOTATION YES
            LINK
                SIMPLE   NO
                IMPLICIT NO
                EXPLICIT NO
            OTHER
                CONCUR   NO
                SUBDOC   NO
                FORMAL   NO
                URN      NO
                KEEPRSRE YES
                VALIDITY TYPE
                ENTITIES
                    REF      ANY
                    INTEGRAL YES
        APPINFO NONE
        SEEALSO "ISO 8879//NOTATION Application Requirements for XML//EN"
    >
    <!-- Revision: @(#)XHTML1.dcl 1.12 99/03/23 SMI -->
```

C.3. XHTML Character Entities

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

C.3.1. XHTML Latin 1 Character Entities

```
<!-- XML-compatible ISO Latin 1 Character Entity Set for XHTML 1.0

     Typical invocation:

       <!ENTITY % XHTML1-lat1
           PUBLIC "-//W3C//ENTITIES Latin 1//EN//XML"
                  "XHTML1-lat1.ent">
       %XHTML1-lat1;

     Revision:  @(#)XHTML1-lat1.ent 1.13 99/04/01 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 -->
```

C.3.2. XHTML Special Characters

```
<!-- XML-compatible ISO Special Character Entity Set for XHTML 1.0

     Typical invocation:

       <!ENTITY % XHTML1-special
           PUBLIC "-//W3C//ENTITIES Special//EN//XML"
                  "XHTML1-special.ent">
       %XHTML1-special;

     Revision:  @(#)XHTML1-special.ent 1.13 99/04/01 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 -->
```

C.3.3. XHTML Mathematical, Greek, and Symbolic Characters

```
<!-- XML-compatible ISO Mathematical, Greek and Symbolic
     Character Entity Set for XHTML 1.0

     Typical invocation:

       <!ENTITY % XHTML1-symbol
           PUBLIC "-//W3C//ENTITIES Symbols//EN//XML"
                  "XHTML1-symbol.ent">
       %XHTML1-symbol;

     Revision:  @(#)XHTML1-symbol.ent 1.13 99/04/01 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 -->
```

C.4. Common Declaration

In order to take advantage of the XHTML DTD Modules, DTD authors needs to define the content model for their DTD. XHTML provides a variety of tools to ease this effort. They are defined in the following support modules. See Extending XHTML [p.??] for more information on using these and the Modules in custom DTDs

C.4.1. XHTML Common Names

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Document Common Names Module  .............................. -->
<!-- file: XHTML1-names.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-names.mod 1.16 99/04/01 SMI

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

```
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
     SYSTEM "XHTML1-names.mod"

     Revisions:
# 1999-01-31  added URIs PE for multiple URI attribute values
     ..................................................................... -->

<!-- i. Common Names

     defines the following common names, many of these imported
     from other specifications and standards.
-->

<!-- ....  Imported Names  .... -->

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

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

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

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

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

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

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

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

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

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

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

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

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

<!-- style sheet data -->
<!ENTITY % StyleSheet "CDATA" >
```

```
<!ENTITY % Text "CDATA" >

<!-- Length defined in strict DTD for cellpadding/cellspacing -->

<!-- nn for pixels or nn% for percentage length -->
<!ENTITY % Length "CDATA" >

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

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

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

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

<!-- a color using sRGB: #RRGGBB as Hex values -->
<!ENTITY % Color "CDATA" >

<!-- end of XHTML1-names.mod -->
```

C.4.2. XHTML Common Attribute Definitions

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Common Attributes Module  .................................. -->
<!-- file: XHTML1-attribs.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-attribs.mod 1.14 99/04/01 SMI

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

     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Attributes//EN"
     SYSTEM "XHTML1-attribs.mod"

     Revisions:
# 1999-02-24  changed PE names for attribute classes to *.attrib;
     ....................................................................... -->

<!-- ii. Common Attributes

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

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

<!ENTITY % I18n.attrib
```

```
    "lang          %LanguageCode;          #IMPLIED
     xml:lang      %LanguageCode;          #IMPLIED
     dir           (ltr|rtl)               #IMPLIED"
>

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

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

<!ENTITY % Align.attrib "" >

<!ENTITY % XLink.attribs  "INCLUDE" >
<![%XLink.attribs;[
<!-- XLink attributes for a simple 'a' style link -->

<!ENTITY % Alink.attrib
     "xml:link     CDATA                   #FIXED   'simple'
      role         CDATA                   #IMPLIED
      inline       CDATA                   #FIXED   'true'
      content-role CDATA                   #IMPLIED
      content-title CDATA                  #IMPLIED
      show         CDATA                   #FIXED   'replace'
      activate     CDATA                   #FIXED   'user'
      behavior     CDATA                   #IMPLIED"
>
]]>
<!ENTITY % Alink.attrib  "" >

<!-- end of XHTML1-attribs.mod -->
```

C.4.3. XHTML Transitional Attribute Definitions

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Transitional Attributes Module  ............................ -->
<!-- file: XHTML1-attribs-t.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-attribs-t.mod 1.14 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Attributes//EN"
     SYSTEM "XHTML1-attribs-t.mod"

     Revisions:
# 1999-01-24  changed PE names for attribute classes to *.attrib;
     ....................................................................... -->

<!-- ii(t). Common Transitional Attributes

     This modules declares the same set of common attributes as the Strict
```

```
     version, but additionally includes ATTLIST declarations for the additional
     attribute specifications found in the Transitional DTD.
-->

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

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

<!-- HTML intrinsic event attributes declared previously -->

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

<!-- horizontal text alignment -->
<!ENTITY % Align.attrib
     "align   (left|center|right|justify)   #IMPLIED"
>

<!-- horizontal and vertical alignment -->
<!ENTITY % IAlign.attrib
     "align   (top|middle|bottom|left|right) #IMPLIED"
>

<!ENTITY % XLink.attribs  "INCLUDE" >
<![%XLink.attribs;[
<!-- XLink attributes for a simple anchor link -->

<!ENTITY % Alink.attrib
     "xml:link     CDATA                     #FIXED   'simple'
      role         CDATA                     #IMPLIED
      inline       CDATA                     #FIXED   'true'
      content-role CDATA                     #IMPLIED
      content-title CDATA                    #IMPLIED
      show         CDATA                     #FIXED   'replace'
      activate     CDATA                     #FIXED   'user'
      behavior     CDATA                     #IMPLIED"
>
]]>
<!ENTITY % Alink.attrib  "" >

<!-- end of XHTML1-attribs-t.mod -->
```

C.4.4. XHTML Strict Content Model

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Document Model Module  ..................................... -->
<!-- file: XHTML1-model.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-model.mod 1.12 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Model//EN"
     SYSTEM "XHTML1-model.mod"

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

<!-- iii. Document Model

     This modules declares entities describing all text flow
     elements, excluding Transitional elements. This module
     describes the groupings of elements that make up HTML's
     document model.

     HTML has two basic content models:

         %Inline.mix;  character-level elements
         %Block.mix;   block-like elements, eg., paragraphs and lists

     The reserved word '#PCDATA' (indicating a text string) is now
     included explicitly with each element declaration, as XML requires
     that the reserved word occur first in a content model specification..
-->

<!-- ................  Miscellaneous Elements  ................ -->

<!-- These elements are neither block nor inline, and can
     essentially be used anywhere in the document body -->

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

<!-- ...................  Inline Elements  ................... -->

<!ENTITY % Inlstruct.class
       "bdo | br | span" >

<!ENTITY % Inlpres.class  "tt | i | b | big | small | sub | sup" >

<!ENTITY % Inlphras.class
     "em | strong | dfn | code | samp | kbd | var | cite | abbr | acronym | q" >

<!ENTITY % Inlspecial.class  "a | img | object | map" >
```

```
<!ENTITY % Formctrl.class  "input | select | textarea | label | button" >

<!-- %Inline.class; includes all inline elements, used as a component in mixes -->

<!ENTITY % Inline.class
     "%Inlstruct.class;
      | %Inlpres.class;
      | %Inlphras.class;
      | %Inlspecial.class;
      | %Formctrl.class;"
>

<!-- %Inline.mix; includes all inline elements, including %Misc.class; -->

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

<!-- %Inline-noa.class; includes all non-anchor inlines,
     used as a component in mixes -->

<!ENTITY % Inline-noa.class
     "%Inlstruct.class;
      | %Inlpres.class;
      | %Inlphras.class;
      | img | object | map
      | %Formctrl.class;"
>

<!-- %Inline-noa.mix; includes all non-anchor inlines -->

<!ENTITY % Inline-noa.mix
     "%Inline-noa.class;
      | %Misc.class;"
>

<!-- ....................  Block Elements  ................... -->

<!-- In the HTML 4.0 DTD, heading and list elements were included
     in the %block; parameter entity. The %Heading.class; and
     %List.class; parameter entities must now be included explicitly
     on element declarations where desired.
-->

<!--  There are six levels of headings from H1 (the most important)
      to H6 (the least important).
  -->
<!ENTITY % Heading.class  "h1 | h2 | h3 | h4 | h5 | h6" >

<!ENTITY % List.class "ul | ol | dl" >

<!ENTITY % Blkstruct.class  "p | div" >

<!ENTITY % Blkpres.class "hr" >

<!ENTITY % Blkphras.class  "pre | blockquote | address" >
```

```
<!ENTITY % Blkform.class  "form | fieldset" >

<!ENTITY % Blkspecial.class  "table" >

<!-- %Block.class; includes all block elements,
     used as an component in mixes -->

<!ENTITY % Block.class
     "%Blkstruct.class;
      | %Blkpres.class;
      | %Blkphras.class;
      | %Blkform.class;
      | %Blkspecial.class;"
>

<!-- %Block.mix; includes all block elements plus %Misc.class; -->

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

<!-- %Block-noform.class; includes all non-form block elements,
     used as a component in mixes -->

<!ENTITY % Block-noform.class
     "%Blkstruct.class;
      | %Blkpres.class;
      | %Blkphras.class;
      | %Blkspecial.class;"
>

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

<!ENTITY % Block-noform.mix
     "%Block-noform.class;
      | %Misc.class;"
>

<!-- ................  All Content Elements  .................. -->

<!-- %Flow.mix; includes all text content, block and inline -->

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

<!-- end of XHTML1-model.mod -->
```

C.4.5. XHTML Transitional Content Model

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Transitional Text Markup Module  .......................... -->
<!-- file: XHTML1-model-t.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-model-t.mod 1.14 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Document Model//EN"
     SYSTEM "XHTML1-model-t.mod"

     Revisions:
#1999-01-14  rearranged forms and frames PEs, adding %Blkform.class;
     ....................................................................... -->

<!-- iii(t). Transitional Document Model

     This modules declares entities describing all text flow
     elements, including Transitional elements. This module
     describes the groupings of elements that make up HTML's
     document model.

     HTML has two basic content models:

         %Inline.mix;  character-level elements
         %Block.mix;   block-like elements, eg., paragraphs and lists

     The reserved word '#PCDATA' (indicating a text string) is now
     included explicitly with each element declaration, as XML requires
     that the reserved word occur first in a content model specification..
-->

<!-- .................  Miscellaneous Elements  ................ -->

<!-- These elements are neither block nor inline, and can
     essentially be used anywhere in the document body -->

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

<!-- ....................  Inline Elements  .................... -->

<!ENTITY % Inlstruct.class
       "bdo | br | span" >

<!ENTITY % Inlpres.class
       "tt | i | b | u | s | strike | big | small | font | basefont | sub | sup" >

<!ENTITY % Inlphras.class
       "em | strong | dfn | code | samp | kbd | var | cite | abbr | acronym | q" >

<![%XHTML1-frames.module;[
```

```
<!-- %Inlspecial.class; includes iframe in Frameset DTD version -->

<!ENTITY % Inlspecial.class  "a | img | applet | object | map | iframe">
]]>

<!ENTITY % Inlspecial.class  "a | img | applet | object | map">

<!ENTITY % Formctrl.class  "input | select | textarea | label | button">

<!-- %Inline.class; includes all inline elements, used as a component in mixes -->

<!ENTITY % Inline.class
     "%Inlstruct.class;
      | %Inlpres.class;
      | %Inlphras.class;
      | %Inlspecial.class;
      | %Formctrl.class;"
>

<!-- %Inline.mix; includes all inline elements, including %Misc.class; -->

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

<!-- %Inline-noa.class; includes all non-anchor inlines,
     used as a component in mixes -->

<!ENTITY % Inline-noa.class
     "%Inlstruct.class;
      | %Inlpres.class;
      | %Inlphras.class;
      | img | applet | object | map
      | %Formctrl.class;"
>

<!-- %Inline-noa.mix; includes all non-anchor inlines -->

<!ENTITY % Inline-noa.mix
     "%Inline-noa.class;
      | %Misc.class;"
>

<!-- ....................  Block Elements  ................... -->

<!-- In the HTML 4.0 DTD, heading and list elements were included
     in the %block; parameter entity. The %Heading.class; and
     %List.class; parameter entities must now be included explicitly
     on element declarations where desired.
-->

<!--  There are six levels of headings from h1 (the most important)
      to h6 (the least important).
-->
<!ENTITY % Heading.class  "h1 | h2 | h3 | h4 | h5 | h6">
```

```
<!ENTITY % List.class  "ul | ol | dl | menu | dir" >

<!ENTITY % Blkstruct.class  "p | div" >

<!ENTITY % Blkpres.class  "center | hr" >

<!ENTITY % Blkphras.class  "pre | blockquote | address" >

<!ENTITY % Blkform.class  "form | fieldset" >

<![%XHTML1-frames.module;[
<!-- Blkspecial.class includes noframes in Frameset DTD version -->

<!ENTITY % Blkspecial.class  "noframes | table" >
]]>

<!ENTITY % Blkspecial.class  "table" >

<!-- %Block.class; includes all block elements,
     used as an component in mixes -->

<!ENTITY % Block.class
     "%Blkstruct.class;
      | %Blkpres.class;
      | %Blkphras.class;
      | %Blkform.class;
      | %Blkspecial.class;"
>

<!-- %Block.mix; includes all block elements plus %Misc.class; -->

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

<!-- %Block-noform.class; includes all non-form block elements,
     used as a component in mixes -->

<!ENTITY % Block-noform.class
     "%Blkstruct.class;
      | %Blkpres.class;
      | %Blkphras.class;
      | %Blkspecial.class;"
>

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

<!ENTITY % Block-noform.mix
     "%Block-noform.class;
      | %Misc.class;"
>

<!-- ................  All Content Elements  .................. -->

<!-- %Flow.mix; includes all text content, block and inline -->
```

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

<!-- end of XHTML1-model-t.mod -->
```

C.4.6. Intrinsic Events

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Intrinsic Events Module  .................................. -->
<!-- file: XHTML1-events.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-events.mod 1.16 99/04/01 SMI

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

     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
     SYSTEM "XHTML1-events.mod"

     Revisions:
#1999-01-14  transferred onfocus and onblur ATTLIST for 'a' from link module
#1999-04-01  transferred remaining events attributes from other modules
     ....................................................................... -->

<!-- iv. Intrinsic Event Attributes

     These are the event attributes defined in HTML 4.0,
     Section 18.2.3 "Intrinsic Events"

    "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;                  #IMPLIED
      ondblclick   %Script;                  #IMPLIED
      onmousedown  %Script;                  #IMPLIED
      onmouseup    %Script;                  #IMPLIED
      onmouseover  %Script;                  #IMPLIED
      onmousemove  %Script;                  #IMPLIED
      onmouseout   %Script;                  #IMPLIED
      onkeypress   %Script;                  #IMPLIED
      onkeydown    %Script;                  #IMPLIED
      onkeyup      %Script;                  #IMPLIED"
>
```

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

<!ATTLIST a     onfocus      %Script;                  #IMPLIED
     onblur       %Script;                  #IMPLIED
>

<!-- additional attributes on form element -->

<!ATTLIST form      onsubmit     %Script;                  #IMPLIED
      onreset      %Script;                 #IMPLIED
>

<!-- additional attributes on label element -->

<!ATTLIST label      onfocus      %Script;                  #IMPLIED
      onblur       %Script;                 #IMPLIED
>

<!-- additional attributes on input element -->

<!ATTLIST input      onfocus      %Script;                  #IMPLIED
      onblur       %Script;                 #IMPLIED
      onselect     %Script;                 #IMPLIED
      onchange     %Script;                 #IMPLIED
>

<!-- additional attributes on select element -->

<!ATTLIST select      onfocus      %Script;                  #IMPLIED
      onblur       %Script;                 #IMPLIED
      onchange     %Script;                 #IMPLIED
>

<!-- additional attributes on textarea element -->

<!ATTLIST textarea      onfocus      %Script;                  #IMPLIED
      onblur       %Script;                 #IMPLIED
      onselect     %Script;                 #IMPLIED
      onchange     %Script;                 #IMPLIED
>

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

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

<!-- additional attributes on body element -->

<!ATTLIST body      onload       %Script;                  #IMPLIED
      onunload     %Script;                 #IMPLIED
>

<!-- additional attributes on area element -->

<!ATTLIST area
      onfocus      %Script;                 #IMPLIED
```

```
      onblur      %Script;                 #IMPLIED
>

<!ENTITY % XHTML1-frames.module "IGNORE" >
<![%XHTML1-frames.module;[
<!-- additional attributes on frameset element -->

<!ATTLIST frameset
      onload      %Script;                 #IMPLIED
      onunload    %Script;                 #IMPLIED
>
]]>

<!-- end of XHTML1-events.mod -->
```

C.4.7. XHTML Character Entities

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Character Entities Module  ................................. -->
<!-- file: XHTML1-charent.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-charent.mod 1.16 99/04/01 SMI

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

     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
     SYSTEM "XHTML1-charent.mod"

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

<!-- v. Character Entities for XHTML

     declares the set of character entities for XHTML, including Latin 1,
     symbol and special characters.
-->

<!-- to exclude character entity declarations from a normalized
     DTD, declare %XHTML1.ents; as "IGNORE" in the internal
     subset of the dummy XHTML file used for normalization.
-->
<!ENTITY % XHTML1.ents "INCLUDE" >

<![%XHTML1.ents;[
<!ENTITY % XHTML1-lat1
    PUBLIC "-//W3C//ENTITIES Latin 1//EN//XML"
           "XHTML1-lat1.ent" >
%XHTML1-lat1;

<!ENTITY % XHTML1-symbol
    PUBLIC "-//W3C//ENTITIES Symbols//EN//XML"
           "XHTML1-symbol.ent" >
%XHTML1-symbol;
```

```
<!ENTITY % XHTML1-special
    PUBLIC "-//W3C//ENTITIES Special//EN//XML"
           "XHTML1-special.ent" >
%XHTML1-special;
]]>

<!-- end of XHTML1-charent.mod -->
```

C.5. XHTML Document Structure Modules

This section contains the formal definition of each XHTML Module.

C.5.1. Structure

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Structure Module  .......................................... -->
<!-- file: XHTML1-struct.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-struct.mod 1.15 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
     SYSTEM "XHTML1-struct.mod"

     Revisions:
# 1998-10-27  content model on head changed to exclude multiple title or base
# 1998-11-11  ins and del inclusions on body removed, added to indiv. elements
# 1998-11-15  added head element version attribute (restoring from HTML 3.2)
# 1999-03-24  %Profile.attrib; unused, but reserved for future use
     ....................................................................... -->

<!-- a1. Document Structure

        body, head, html
-->

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

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

<!-- reserved for future use with document profiles -->
<!ENTITY % Profile.attrib
     "profile      %URI;                    #FIXED '%XHTML.profile;'" >

<!ELEMENT head  %Head.content; >
<!ATTLIST head
      %I18n.attrib;
      profile      %URI;                    #IMPLIED
>

<![%XHTML.Transitional;[
```

```
<!-- in Transitional, allow #PCDATA and inlines directly within body -->

<!ENTITY % Body.content  "( #PCDATA | %Flow.mix; )*" >
]]>
<!ENTITY % Body.content
     "( %Heading.class;
      | %List.class;
      | %Block.class;
      | %Misc.class; )+"
>

<!ELEMENT body  %Body.content; >
<!ATTLIST body      %Common.attrib;
>

<![%XHTML.Transitional;[
<!-- ....  additional Transitional attributes on body .... -->

<!-- There are also 16 widely known color names with their sRGB values:

    Black  = #000000   Maroon = #800000   Green  = #008000   Navy   = #000080
    Silver = #C0C0C0   Red    = #FF0000   Lime   = #00FF00   Blue   = #0000FF
    Gray   = #808080   Purple = #800080   Olive  = #808000   Teal   = #008080
    White  = #FFFFFF   Fuchsia= #FF00FF   Yellow = #FFFF00   Aqua   = #00FFFF
-->

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

<!ENTITY % Html.content  "( head, body )" >

<!-- version and namespace attribute values defined in driver -->
<!ENTITY % Version.attrib
     "version      CDATA                    #FIXED '%HTML.version;'" >
<!ENTITY % Ns.attrib
     "xmlns        %URI;                    #FIXED '%XHTML.ns;'" >

<!ELEMENT html  %Html.content; >
<!ATTLIST html      %I18n.attrib;
      %Version.attrib;
      %Ns.attrib;
>

<!-- end of XHTML1-struct.mod -->
```

C.5.2. Frameset

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Frames Module  ............................................. -->
<!-- file: XHTML1-frames.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-frames.mod 1.15 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
     SYSTEM "XHTML1-frames.mod"

     Revisions:
#1999-01-14  transferred 'target' attribute on 'a' from linking module
     ....................................................................... -->

<!-- a2. Frames

        frame, frameset, iframe, noframes
-->

<!-- The content model for HTML documents depends on whether the HEAD is
     followed by a FRAMESET or BODY element. The widespread omission of
     the BODY start tag makes it impractical to define the content model
     without the use of a conditional section.
-->

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

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

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

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

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

<!-- 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 )">
<!-- in HTML 4.0 was "( body ) -( noframes )" exclusion on body -->
<!ELEMENT noframes  %Noframes.content; >
<!ATTLIST noframes
      %Common.attrib;
>

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

<!-- end of XHTML1-frames.mod -->
```

C.5.3. Lists

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Lists Module  ............................................... -->
<!-- file: XHTML1-list.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-list.mod 1.13 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
     SYSTEM "XHTML1-list.mod"

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

```
<!-- a3. Lists

        dl, dt, dd, ol, ul, li

     A conditional section includes additional declarations for the Transitional DTD

        dir, menu
-->

<!-- definition lists - DT for term, DD for its definition -->

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

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

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

<!-- Ordered Lists (ol) numbered styles -->

<!ENTITY % Ol.content  "( li )+" >
<!ELEMENT ol  %Ol.content; >
<!ATTLIST ol      %Common.attrib;
>

<!-- Unordered Lists (ul) bullet styles -->

<!ENTITY % Ul.content  "( li )+" >
<!ELEMENT ul  %Ul.content; >
<!ATTLIST ul
      %Common.attrib;
>

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

<![%XHTML.Transitional;[
<!-- Ordered lists (ol) Numbering style

    1   arabic numbers      1, 2, 3, ...
    a   lower alpha         a, b, c, ...
    A   upper alpha         A, B, C, ...
    i   lower roman         i, ii, iii, ...
    I   upper roman         I, II, III, ...

    The style is applied to the sequence number which by default
    is reset to 1 for the first list item in an ordered list.
```

```
-->

<!ENTITY % OlStyle "CDATA" >

<!ATTLIST ol      type         %OlStyle;                #IMPLIED
      compact      (compact)                #IMPLIED
      start        %Number;                 #IMPLIED
>

<!-- Unordered Lists (ul) bullet styles -->
<!ENTITY % UlStyle "(disc|square|circle)" >

<!ATTLIST ul      type         %UlStyle;                #IMPLIED
      compact      (compact)                #IMPLIED
>

<!ENTITY % Dir.content  "( li )+" >
<!ELEMENT dir  %Dir.content; >
<!ATTLIST dir
      %Common.attrib;
      compact      (compact)                #IMPLIED
>

<!ENTITY % Menu.content  "( li )+" >
<!ELEMENT menu  %Menu.content; >
<!ATTLIST menu
      %Common.attrib;
      compact      (compact)                #IMPLIED
>
]]>

<!-- end of XHTML1-list.mod -->
```

C.6. XHTML Block Element Modules

This section contains the formal definition of the Block Element Modules.

C.6.1. Block Structural

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Block Structural Module  .................................. -->
<!-- file: XHTML1-blkstruct.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-blkstruct.mod 1.10 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
     SYSTEM "XHTML1-blkstruct.mod"

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

```
<!-- b1. Block Structural

        div, p
-->

<!ENTITY % Div.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT div  %Div.content; >
<!ATTLIST div      %Common.attrib;
      %Align.attrib;
>

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

<!-- end of XHTML1-blkstruct.mod -->
```

C.6.2. Block Phrasal

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Block Phrasal Module  ..................................... -->
<!-- file: XHTML1-blkphras.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-blkphras.mod 1.13 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
     SYSTEM "XHTML1-blkphras.mod"

     Revisions:
# 1998-11-10  removed pre exclusions - content model changed to mimic HTML 4.0
# 1999-01-29  moved div and p to block structural module
     ....................................................................... -->

<!-- b2. Block Phrasal

        address, blockquote, pre, h1, h2, h3, h4, h5, h6
-->

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

<![%XHTML.Transitional;[
<!ENTITY % Blockquote.content  "( %Flow.mix; )*" >
]]>
<!ENTITY % Blockquote.content
     "( %Heading.class;
      | %List.class;
```

```
      | %Block.mix; )+"
>

<!ELEMENT blockquote  %Blockquote.content; >
<!ATTLIST blockquote
      %Common.attrib;
      cite         %URI;                    #IMPLIED
>

<!ENTITY % Pre.content
     "( #PCDATA | tt | i | b
      | %Inlstruct.class; | %Inlphras.class;
      | a | script | map
      | %Formctrl.class; )*"
>

<!ELEMENT pre  %Pre.content; >
<!ATTLIST pre
      %Common.attrib;
      xml:space    CDATA                    #FIXED "preserve"
>

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

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

<!ELEMENT h1  %Heading.content; >
<!ATTLIST h1      %Common.attrib;
      %Align.attrib;
>

<!ELEMENT h2  %Heading.content; >
<!ATTLIST h2      %Common.attrib;
      %Align.attrib;
>

<!ELEMENT h3  %Heading.content; >
<!ATTLIST h3      %Common.attrib;
      %Align.attrib;
>

<!ELEMENT h4  %Heading.content; >
<!ATTLIST h4      %Common.attrib;
      %Align.attrib;
>

<!ELEMENT h5  %Heading.content; >
<!ATTLIST h5      %Common.attrib;
      %Align.attrib;
>

<!ELEMENT h6  %Heading.content; >
<!ATTLIST h6      %Common.attrib;
      %Align.attrib;
>

<!-- end of XHTML1-blkphras.mod -->
```

C.6.3. Block Presentational

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Block Presentational Module  ............................... -->
<!-- file: XHTML1-blkpres.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-blkpres.mod 1.15 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"
     SYSTEM "XHTML1-blkpres.mod"

     Revisions:
# 1999-01-31  added I18n.attrib to hr (errata)
     ....................................................................... -->

<!-- b3. Block Presentational

        hr

     A conditional section includes additional declarations for the Transitional DTD

        center
-->

<!ENTITY % Hr.content  "EMPTY" >
<!ELEMENT hr  %Hr.content; >
<!ATTLIST hr
      %Core.attrib;
      %I18n.attrib;
      %Events.attrib;
>

<![%XHTML.Transitional;[
<!ENTITY % Center.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT center  %Center.content; >
<!ATTLIST center
      %Common.attrib;
>

<!-- additional attributes on hr -->
<!ATTLIST hr
      align        (left|center|right)      #IMPLIED
      noshade      (noshade)                #IMPLIED
      size         %Pixels;                 #IMPLIED
      width        %Length;                 #IMPLIED
>
]]>

<!-- end of XHTML1-blkpres.mod -->
```

C.7. XHTML Inline Element Modules

This section contains the formal definition of each XHTML Inline Element Module.

C.7.1. Inline Structural

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Inline Phrasal Module  ..................................... -->
<!-- file: XHTML1-inlstruct.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-inlstruct.mod 1.10 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
     SYSTEM "XHTML1-inlstruct.mod"

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

<!-- c1. Inline Structural

        bdo, br, del, ins, span
-->

<!ENTITY % Bdo.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT bdo  %Bdo.content; >
<!ATTLIST bdo      %Core.attrib;
      lang         %LanguageCode;           #IMPLIED
      dir          (ltr|rtl)                #REQUIRED
>

<!ENTITY % Br.content  "EMPTY" >
<!ELEMENT br  %Br.content; >
<!ATTLIST br      %Core.attrib;
>

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

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

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

```
<!ELEMENT span  %Span.content; >
<!ATTLIST span      %Common.attrib;
>

<!-- end of XHTML1-inlstruct.mod -->
```

C.7.2. Inline Phrasal

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Inline Phrasal Module  ..................................... -->
<!-- file: XHTML1-inlphras.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-inlphras.mod 1.14 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
     SYSTEM "XHTML1-inlphras.mod"

     Revisions:
#1999-01-29  moved bdo, br, del, ins, span to inline structural module
     ...................................................................... -->

<!-- c2. Inline Phrasal

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

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

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

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

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

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

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

```
<!ELEMENT em  %Em.content; >
<!ATTLIST em      %Common.attrib;
>

<!ENTITY % Kbd.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT kbd  %Kbd.content; >
<!ATTLIST kbd      %Common.attrib;
>

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

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

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

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

<!-- end of XHTML1-inlphras.mod -->
```

C.7.3. Inline Presentational

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Inline Presentational Module  ............................. -->
<!-- file: XHTML1-inlpres.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-inlpres.mod 1.13 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"
     SYSTEM "XHTML1-inlpres.mod"

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

<!-- c3. Inline Presentational

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

```
     A conditional section includes additional declarations for the Transitional DTD

        basefont, font, s, strike, u
-->

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

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

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

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

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

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

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

<![%XHTML.Transitional;[

<!ENTITY % Basefont.content  "EMPTY" >
<!ELEMENT basefont  %Basefont.content; >
<!ATTLIST basefont      id           ID                       #IMPLIED
      size         CDATA                    #REQUIRED
      color        %Color;                  #IMPLIED
      face         CDATA                    #IMPLIED
>

<!ENTITY % Font.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT font  %Font.content; >
<!ATTLIST font      %Core.attrib;
      %I18n.attrib;
      size         CDATA                    #IMPLIED
      color        %Color;                  #IMPLIED
      face         CDATA                    #IMPLIED
```

```
>

<!ENTITY % S.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT s  %S.content; >
<!ATTLIST s      %Common.attrib;
>

<!ENTITY % Strike.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT strike  %Strike.content; >
<!ATTLIST strike      %Common.attrib;
>

<!ENTITY % U.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT u  %U.content; >
<!ATTLIST u      %Common.attrib;
>

]]>

<!-- end of XHTML1-inlpres.mod -->
```

C.8. XHTML Special Case Modules

This section contains the formal definition of each of the XHTML Special Case Modules.

C.8.1. Meta

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Document Metainformation Module  ........................... -->
<!-- file: XHTML1-meta.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-meta.mod 1.14 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
     SYSTEM "XHTML1-meta.mod"

     Revisions:
# 1998-11-11  title content model changed - exclusions no longer necessary
# 1999-02-01  removed isindex
     ....................................................................... -->

<!-- d1. Meta Information

        meta, 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.content  "( #PCDATA )" >
<!ELEMENT title  %Title.content; >
<!ATTLIST title      %I18n.attrib;
>

<!ENTITY % Meta.content  "EMPTY" >
<!ELEMENT meta  %Meta.content; >
<!ATTLIST meta      %I18n.attrib;
      http-equiv   NMTOKEN                  #IMPLIED
      name         NMTOKEN                  #IMPLIED
      content      CDATA                    #REQUIRED
      scheme       CDATA                    #IMPLIED
>

<!-- end of XHTML1-meta.mod -->
```

C.8.2. Linking

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Linking Module  ............................................ -->
<!-- file: XHTML1-linking.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-linking.mod 1.13 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
     SYSTEM "XHTML1-linking.mod"

     Revisions:
# 1998-10-27  exclusion on 'a' within 'a' removed for XML
# 1998-11-15  moved shape and coords attributes on 'a' to csismap module
# 1999-01-14  moved onfocus and onblur attributes on 'a' to events module
     ....................................................................... -->

<!-- d2. Linking

        a, base, link
-->

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

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

<!ENTITY % A.content  "( #PCDATA | %Inline-noa.mix; )*" >
<!ELEMENT a  %A.content; >
<!ATTLIST a      %Common.attrib;
      name         CDATA                    #IMPLIED
      href         %URI;                    #IMPLIED
      %Alink.attrib;
      charset      %Charset;                #IMPLIED
      type         %ContentType;            #IMPLIED
      hreflang     %LanguageCode;           #IMPLIED
```

```
      rel           %LinkTypes;                #IMPLIED
      rev           %LinkTypes;                #IMPLIED
      accesskey     %Character;                #IMPLIED
      tabindex      %Number;                   #IMPLIED
>

<!-- ...........  Base Element  ........... -->

<!ENTITY % Base.content  "EMPTY" >
<!ELEMENT base  %Base.content; >
<!ATTLIST base
      href          %URI;                      #REQUIRED
>

<!-- ...........  Link Element  ........... -->

<!-- Relationship values can be used in principle:

   a) for document specific toolbars/menus when used
      with the LINK element in document head 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)
-->

<!ENTITY % Link.content  "EMPTY" >
<!ELEMENT link  %Link.content; >
<!ATTLIST link      %Common.attrib;
      charset       %Charset;                  #IMPLIED
      href          %URI;                      #IMPLIED
      hreflang      %LanguageCode;             #IMPLIED
      type          %ContentType;              #IMPLIED
      rel           %LinkTypes;                #IMPLIED
      rev           %LinkTypes;                #IMPLIED
      media         %MediaDesc;                #IMPLIED
>

<!-- end of XHTML1-linking.mod -->
```

C.8.3. Image

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Images Module  ............................................. -->
<!-- file: XHTML1-image.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-image.mod 1.15 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
```

```
     SYSTEM "XHTML1-image.mod"

     Revisions:
# 1999-01-31  corrected transitional attributes (errata)
     ........................................................................ -->

<!-- d3.1. Images

        img
-->

<!-- 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 ALT, and avoid server-side image maps
-->

<!ENTITY % Img.content  "EMPTY" >
<!ELEMENT img  %Img.content; >
<!ATTLIST img      %Common.attrib;
      src          %URI;                     #REQUIRED
      alt          %Text;                    #REQUIRED
      longdesc     %URI;                     #IMPLIED
      height       %Length;                  #IMPLIED
      width        %Length;                  #IMPLIED
      usemap       %URI;                     #IMPLIED
      ismap        (ismap)                   #IMPLIED
>

<!-- USEMAP points to a MAP element which may be in this document or
     an external document, although the latter is not widely supported
-->

<![%XHTML.Transitional;[
<!-- additional Transitional attributes -->
<!ATTLIST img      %IAlign.attrib;
      border       %Pixels;                  #IMPLIED
      hspace       %Pixels;                  #IMPLIED
      vspace       %Pixels;                  #IMPLIED
>
]]>

<!-- end of XHTML1-image.mod -->
```

C.8.4. Client-side Image Map

```
<!-- ..................................................................... -->
<!-- XHTML 1.0 Client-side Image Map Module  ............................. -->
<!-- file: XHTML1-csismap.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-csismap.mod 1.15 99/04/01 SMI

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

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

```
     SYSTEM "XHTML1-csismap.mod"

     Revisions:
# 1999-01-31  fixed map content model (errata)
     ....................................................................... -->

<!-- d3.2. Client-side Image Maps

        area, map
-->

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

<!ENTITY % Map.content  "(( %Heading.class; | %List.class; | %Block.mix; ) | area)+" >
<!ELEMENT map  %Map.content; >
<!ATTLIST map      %Common.attrib;
      name         CDATA                    #REQUIRED
>

<!ENTITY % Area.content  "EMPTY" >
<!ELEMENT area  %Area.content; >
<!ATTLIST area      %Common.attrib;
      href         %URI;                    #IMPLIED
      shape        %Shape;                  'rect'
      coords       %Coords;                 #IMPLIED
      nohref       (nohref)                 #IMPLIED
      alt          %Text;                   #REQUIRED
      tabindex     %Number;                 #IMPLIED
      accesskey    %Character;              #IMPLIED
>

<!-- modify anchor (<a>) attribute definition list to
     allow for client-side image maps -->

<!ATTLIST a      shape        %Shape;                  'rect'
      coords       %Coords;                 #IMPLIED
>

<!-- end of XHTML1-csismap.mod -->
```

C.8.5. Object

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 External Inclusion Module  ................................. -->
<!-- file: XHTML1-object.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-object.mod 1.16 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
     SYSTEM "XHTML1-object.mod"

     Revisions:
```

```
# 1999-01-31  changed object's archive attr to allow for multiple URIs
# 1999-01-31  corrected transitional attributes (errata)
     ........................................................................ -->

<!-- d3.3. Objects

        object, param

     object is used to embed objects as part of HTML pages;
     param elements should precede other content.
-->

<!ENTITY % Object.content  "( %Flow.mix; | param )*" >
<!ELEMENT object  %Object.content; >
<!ATTLIST object      %Common.attrib;
      declare      (declare)                 #IMPLIED
      classid      %URI;                     #IMPLIED
      codebase     %URI;                     #IMPLIED
      data         %URI;                     #IMPLIED
      type         %ContentType;             #IMPLIED
      codetype     %ContentType;             #IMPLIED
      archive      %URIs;                    #IMPLIED
      standby      %Text;                    #IMPLIED
      height       %Length;                  #IMPLIED
      width        %Length;                  #IMPLIED
      usemap       %URI;                     #IMPLIED
      name         CDATA                     #IMPLIED
      tabindex     %Number;                  #IMPLIED
>

<![%XHTML.Transitional;[
<!-- additional Transitional attributes -->
<!ATTLIST object      %IAlign.attrib;
      border       %Pixels;                  #IMPLIED
      hspace       %Pixels;                  #IMPLIED
      vspace       %Pixels;                  #IMPLIED
>
]]>

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

<!-- end of XHTML1-object.mod -->
```

C.8.6. Applet

```
<!-- ....................................................................... -->
<!-- XHTML 1.0 Draft Document Java Applet Module  ......................... -->
<!-- file: XHTML1-applet.mod
```

```
     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-applet.mod 1.14 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML V1.0 Java Applets//EN"
     SYSTEM "XHTML1-applet.mod"

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

<!-- d4. Scripting

        applet
-->

<!-- One of code or object attributes must be present.
     Place param elements before other content.
-->

<!ENTITY % Applet.content  "( param | %Flow.mix; )*">
<!ELEMENT applet  %Applet.content; >
<!ATTLIST applet
      %Core.attrib;
      codebase     %URI;                    #IMPLIED
      archive      CDATA                    #IMPLIED
      code         CDATA                    #IMPLIED
      object       CDATA                    #IMPLIED
      alt          %Text;                   #IMPLIED
      name         CDATA                    #IMPLIED
      width        %Length;                 #REQUIRED
      height       %Length;                 #REQUIRED
      %IAlign.attrib;
      hspace       %Pixels;                 #IMPLIED
      vspace       %Pixels;                 #IMPLIED
>

<!-- If the Object module that supplies the param element declarations
     is not used, redeclare %Param.local.module; as 'INCLUDE':  -->
<!ENTITY % Param.local.module  "IGNORE" >
<![%Param.local.module;[
<!ENTITY % Param.content  "EMPTY">
<!ELEMENT param  %Param.content; >
<!ATTLIST param
      id           ID                       #IMPLIED
      name         CDATA                    #REQUIRED
      value        CDATA                    #IMPLIED
      valuetype    (data|ref|object)        'data'
      type         %ContentType;            #IMPLIED
>
]]>

<!-- end of XHTML1-applet.mod -->
```

C.8.7. Scripting

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Document Scripting Module  ................................. -->
<!-- file: XHTML1-script.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-script.mod 1.13 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
     SYSTEM "XHTML1-script.mod"

     Revisions:
# 1999-01-30  added xml:space to script
# 1999-02-01  removed for and event attributes from script
     ....................................................................... -->

<!-- d4. Scripting

        script, noscript
-->

<!ENTITY % Script.content  "( #PCDATA )" >
<!ELEMENT script  %Script.content; >
<!ATTLIST script      charset      %Charset;                #IMPLIED
      type         %ContentType;            #REQUIRED
      src          %URI;                    #IMPLIED
      defer        (defer)                  #IMPLIED
      xml:space    CDATA                    #FIXED 'preserve'
>

<!ENTITY % Noscript.content
     "( %Heading.class;
      | %List.class;
      | %Block.mix; )+"
>
<!ELEMENT noscript  %Noscript.content; >
<!ATTLIST noscript      %Common.attrib;
>

<!-- end of XHTML1-script.mod -->
```

C.8.8. Stylesheets

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Document Stylesheet Module  ................................ -->
<!-- file: XHTML1-style.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-style.mod 1.13 99/04/01 SMI

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

```
     PUBLIC "-//W3C//DTD XHTML 1.0 Stylesheets//EN"
     SYSTEM "XHTML1-style.mod"

     Revisions:
# 1999-01-30  added xml:space to style
     ....................................................................... -->

<!-- d5. Stylesheets

        style
-->

<!ENTITY % Style.content  "( #PCDATA )" >
<!ELEMENT style  %Style.content; >
<!ATTLIST style      %I18n.attrib;
      type         %ContentType;            #REQUIRED
      media        %MediaDesc;              #IMPLIED
      title        %Text;                   #IMPLIED
      xml:space    CDATA                    #FIXED 'preserve'
>

<!-- end of XHTML1-style.mod -->
```

C.8.9. HTML 3.2 Tables

```
<!-- ....................................................................... -->
<!-- XHTML 1.0 Simple Table Module  ....................................... -->
<!-- file: XHTML1-table32.mod

     This module subsets XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-table32.mod 1.3 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0-Based Subset Simplified Tables//EN"
     SYSTEM "XHTML1-table32.mod"

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

<!-- Simplified (HTML 3.2) Tables

     This table module is based on the HTML 3.2 table model, itself based
     on a subset of the CALS table model as described in the IETF HTML
     table specification [see IETF RFC 1942]. 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 attributes.

        table, caption, tr, th, td
-->

<!-- 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"
>

<!ENTITY % CaptionAlign  "(top|bottom)">

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

<!ENTITY % Table.content  "( caption?, tr+ )" >
<!ELEMENT table  %Table.content; >
<!ATTLIST table      %Common.attrib;
      width        %Length;                 #IMPLIED
      border       %Pixels;                 #IMPLIED
      cellspacing  %Length;                 #IMPLIED
      cellpadding  %Length;                 #IMPLIED
>

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

<!ENTITY % Tr.content  "( th | td )+" >
<!ELEMENT tr  %Tr.content; >
<!ATTLIST tr      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

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

<!ENTITY % Th.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT th  %Th.content; >
<!ATTLIST th      %Common.attrib;
      rowspan      %Number;                 '1'
      colspan      %Number;                 '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Td.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT td  %Td.content; >
<!ATTLIST td      %Common.attrib;
      rowspan      %Number;                 '1'
      colspan      %Number;                 '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<![%XHTML.Transitional;[
<!-- additional Transitional attributes for XHTML tables -->
```

```
<!-- horizontal placement of table relative to document -->
<!ENTITY % TAlign  "(left|center|right)" >

<!ATTLIST table      align       %TAlign;                  #IMPLIED
      bgcolor     %Color;                    #IMPLIED
>

<!ATTLIST caption      align       %CaptionAlign;            #IMPLIED
>

<!ATTLIST tr      bgcolor     %Color;                    #IMPLIED
>

<!-- note that 'nowrap' in XML must be expressed as nowrap="nowrap" -->

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

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

<!-- end of XHTML1-table.mod -->
```

C.8.10. HTML 4.0 Tables

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Table Module  .............................................. -->
<!-- file: XHTML1-table.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-table.mod 1.15 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
     SYSTEM "XHTML1-table.mod"

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

<!-- d6. HTML 4.0 Tables

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

     A conditional section includes additional declarations for the Transitional DTD
-->
```

```
<!-- IETF HTML table standard, see [RFC1942] -->

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

     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.

     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.
-->

<!ENTITY % TFrame "(void|above|below|hsides|lhs|rhs|vsides|box|border)">

<!-- 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 "(none | groups | rows | cols | all)">

<!-- horizontal placement of table relative to document -->
<!ENTITY % TAlign "(left|center|right)">

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

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

<!ENTITY % CaptionAlign "(top|bottom|left|right)">

<!-- Scope is simpler than axes attribute for common tables -->

<!ENTITY % Scope "(row|col|rowgroup|colgroup)" >

<!ENTITY % Table.content
      "( caption?, ( col* | colgroup* ), (( thead?, tfoot?, tbody+ ) | ( tr+ )))"
>
<!ELEMENT table  %Table.content; >
<!ATTLIST table      %Common.attrib;
      summary      %Text;                   #IMPLIED
      width        %Length;                 #IMPLIED
      border       %Pixels;                 #IMPLIED
```

```
      frame        %TFrame;                 #IMPLIED
      rules        %TRules;                 #IMPLIED
      cellspacing  %Length;                 #IMPLIED
      cellpadding  %Length;                 #IMPLIED
      datapagesize CDATA                    #IMPLIED
>

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

<!ENTITY % Thead.content  "( tr )+" >
<!ELEMENT thead  %Thead.content; >
<!-- Use thead to duplicate headers when breaking table
     across page boundaries, or for static headers when
     TBODY sections are rendered in scrolling panel.

     Use tfoot to duplicate footers when breaking table
     across page boundaries, or for static footers when
     TBODY sections are rendered in scrolling panel.

     Use multiple tbody sections when rules are needed
     between groups of table rows.
-->
<!ATTLIST thead      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Tfoot.content  "( tr )+" >
<!ELEMENT tfoot  %Tfoot.content; >
<!ATTLIST tfoot      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Tbody.content  "( tr )+" >
<!ELEMENT tbody  %Tbody.content; >
<!ATTLIST tbody      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Colgroup.content  "( col )*" >
<!ELEMENT colgroup  %Colgroup.content; >
<!-- colgroup groups a set of col elements. It allows you to group
     several semantically related columns together.
-->
<!ATTLIST colgroup      %Common.attrib;
      span         %Number;                 '1'
      width        %MultiLength;            #IMPLIED
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Col.content  "EMPTY" >
```

```
<!ELEMENT col  %Col.content; >
<!-- 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.
  -->
<!ATTLIST col      %Common.attrib;
      span         %Number;                  '1'
      width        %MultiLength;             #IMPLIED
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Tr.content  "( th | td )+" >
<!ELEMENT tr  %Tr.content; >
<!ATTLIST tr      %Common.attrib;
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

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

<!ENTITY % Th.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT th  %Th.content; >
<!ATTLIST th      %Common.attrib;
      abbr         %Text;                    #IMPLIED
      axis         CDATA                     #IMPLIED
      headers      IDREFS                    #IMPLIED
      scope        %Scope;                   #IMPLIED
      rowspan      %Number;                  '1'
      colspan      %Number;                  '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<!ENTITY % Td.content  "( #PCDATA | %Flow.mix; )*" >
<!ELEMENT td  %Td.content; >
<!ATTLIST td      %Common.attrib;
      abbr         %Text;                    #IMPLIED
      axis         CDATA                     #IMPLIED
      headers      IDREFS                    #IMPLIED
      scope        %Scope;                   #IMPLIED
      rowspan      %Number;                  '1'
      colspan      %Number;                  '1'
      %CellHAlign.attrib;
      %CellVAlign.attrib;
>

<![%XHTML.Transitional;[
<!-- additional Transitional attributes for XHTML tables:
     (in XML, multiple ATTLIST declarations are merged)
```

```
-->

<!ATTLIST table      align       %TAlign;                  #IMPLIED
      bgcolor     %Color;                   #IMPLIED
>

<!ATTLIST caption      align       %CaptionAlign;            #IMPLIED
>

<!ATTLIST tr      bgcolor     %Color;                   #IMPLIED
>

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

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

<!-- end of XHTML1-table.mod -->
```

C.8.11. HTML 3.2 Forms

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Simplified Forms Module  ................................... -->
<!-- file: XHTML1-form32.mod

     This module subsets XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-form32.mod 1.3 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0-Based Subset Simplified Forms//EN"
     SYSTEM "XHTML1-form32.mod"

     Revisions:
# 1998-10-27  exclusion on form within form removed for XML
# 1998-11-10  changed button content model to mirror exclusions
# 1999-01-31  added 'accept' attribute on form (errata)
     ....................................................................... -->

<!-- N O T E :

     this module is a work-in-progress, and is currently only a placeholder
-->

<!-- d7(x). Simplified (HTML 3.2) Forms

     This forms module is based on the HTML 3.2 forms model. While this
```

```
     module mimics the content model and attributes of HTML 3.2 forms,
     the element types declared herein also includes all HTML 4.0
     common attributes.

        form, input, select, option, textarea
-->

<![%XHTML.Transitional;[
<!ENTITY % Form.content
     "( %Heading.class;
      | %List.class;
      | %Inline.class;
      | %Block-noform.mix; )*"
>
]]>
<!ENTITY % Form.content
     "( %Heading.class;
      | %List.class;
      | %Block-noform.mix; )+"
>

<!ELEMENT form  %Form.content; >
<!ATTLIST form      %Common.attrib;
      action       %URI;                    #REQUIRED
      method       (get|post)               'get'
      enctype      %ContentType;            'application/x-www-form-urlencoded'
>

<!ENTITY % InputType.class
     "( text | password | checkbox | radio | submit
      | reset | file | hidden | image )"
>

<!-- attribute name required for all but submit & reset -->

<!ENTITY % Input.content  "EMPTY" >
<!ELEMENT input  %Input.content; >
<!ATTLIST input      %Common.attrib;
      type         %InputType.class;        'text'
      name         CDATA                    #IMPLIED
      value        CDATA                    #IMPLIED
      checked      (checked)                #IMPLIED
      size         CDATA                    #IMPLIED
      maxlength    %Number;                 #IMPLIED
      src          %URI;                    #IMPLIED
>
<![%XHTML.Transitional;[
<!ATTLIST input      align  (top|middle|bottom|left|right) top
>
]]>

<!ENTITY % Select.content  "( option )+" >
<!ELEMENT select  %Select.content; >
<!ATTLIST select      %Common.attrib;
      name         CDATA                    #IMPLIED
      size         %Number;                 #IMPLIED
      multiple     (multiple)               #IMPLIED
```

```
>

<!ENTITY % Option.content  "( #PCDATA )" >
<!ELEMENT option  %Option.content; >
<!ATTLIST option      %Common.attrib;
      selected     (selected)                #IMPLIED
      value        CDATA                     #IMPLIED
>

<!ENTITY % Textarea.content  "( #PCDATA )" >
<!ELEMENT textarea  %Textarea.content; >
<!ATTLIST textarea      %Common.attrib;
      name         CDATA                     #IMPLIED
      rows         %Number;                  #REQUIRED
      cols         %Number;                  #REQUIRED
>

<!-- end of forms.mod -->
```

C.8.12. HTML 4.0 Forms

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Forms Module  .............................................. -->
<!-- file: XHTML1-form.mod

     This is XHTML 1.0, an XML reformulation of HTML 4.0.
     Copyright 1998-1999 W3C (MIT, INRIA, Keio), All Rights Reserved.
     Revision: @(#)XHTML1-form.mod 1.18 99/04/01 SMI

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

     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
     SYSTEM "XHTML1-form.mod"

     Revisions:
# 1998-10-27  exclusion on form within form removed for XML
# 1998-11-10  changed button content model to mirror exclusions
# 1999-01-31  added 'accept' attribute on form (errata)
     ....................................................................... -->

<!-- d7. Forms

        form, label, input, select, optgroup, option, textarea,
        fieldset, legend, button
-->

<![%XHTML.Transitional;[
<!ENTITY % Form.content
     "( %Heading.class;
      | %List.class;
      | %Inline.class;
      | %Block-noform.mix;
      | fieldset )*"
>
]]>
<!ENTITY % Form.content
```

```
     "( %Heading.class;
      | %List.class;
      | %Block-noform.mix;
      | fieldset )+"
>

<!ELEMENT form  %Form.content; >
<!ATTLIST form      %Common.attrib;
      action       %URI;                  #REQUIRED
      method       (get|post)             'get'
      enctype      %ContentType;          'application/x-www-form-urlencoded'
      accept-charset %Charsets;           #IMPLIED
      accept       %ContentTypes;         #IMPLIED
>

<!-- Each label must not contain more than ONE field -->

<!ENTITY % Label.content
     "( #PCDATA
      | %Inlstruct.class;
      | %Inlpres.class;
      | %Inlphras.class;
      | %Inlspecial.class;
      | input | select | textarea | button
      | %Misc.class; )*"
>
<!ELEMENT label  %Label.content; >
<!ATTLIST label      %Common.attrib;
      for          IDREF                  #IMPLIED
      accesskey    %Character;            #IMPLIED
>

<!ENTITY % InputType.class
     "( text | password | checkbox | radio | submit
      | reset | file | hidden | image | button )"
>

<!-- attribute name required for all but submit & reset -->

<!ENTITY % Input.content  "EMPTY" >
<!ELEMENT input  %Input.content; >
<!ATTLIST input      %Common.attrib;
      type         %InputType.class;      'text'
      name         CDATA                  #IMPLIED
      value        CDATA                  #IMPLIED
      checked      (checked)              #IMPLIED
      disabled     (disabled)             #IMPLIED
      readonly     (readonly)             #IMPLIED
      size         CDATA                  #IMPLIED
      maxlength    %Number;               #IMPLIED
      src          %URI;                  #IMPLIED
      alt          CDATA                  #IMPLIED
      usemap       %URI;                  #IMPLIED
      tabindex     %Number;               #IMPLIED
      accesskey    %Character;            #IMPLIED
      accept       %ContentTypes;         #IMPLIED
>
```

```
<!ENTITY % Select.content  "( optgroup | option )+" >
<!ELEMENT select  %Select.content; >
<!ATTLIST select      %Common.attrib;
      name         CDATA                    #IMPLIED
      size         %Number;                 #IMPLIED
      multiple     (multiple)               #IMPLIED
      disabled     (disabled)               #IMPLIED
      tabindex     %Number;                 #IMPLIED
>

<!ENTITY % Optgroup.content  "( option )+" >
<!ELEMENT optgroup  %Optgroup.content; >
<!ATTLIST optgroup      %Common.attrib;
      disabled     (disabled)               #IMPLIED
      label        %Text;                   #REQUIRED
>

<!ENTITY % Option.content  "( #PCDATA )" >
<!ELEMENT option  %Option.content; >
<!ATTLIST option      %Common.attrib;
      selected     (selected)               #IMPLIED
      disabled     (disabled)               #IMPLIED
      label        %Text;                   #IMPLIED
      value        CDATA                    #IMPLIED
>

<!ENTITY % Textarea.content  "( #PCDATA )" >
<!ELEMENT textarea  %Textarea.content; >
<!ATTLIST textarea      %Common.attrib;
      name         CDATA                    #IMPLIED
      rows         %Number;                 #REQUIRED
      cols         %Number;                 #REQUIRED
      disabled     (disabled)               #IMPLIED
      readonly     (readonly)               #IMPLIED
      tabindex     %Number;                 #IMPLIED
      accesskey    %Character;              #IMPLIED
>

<!-- #PCDATA is to solve the mixed content problem, per
     specification only whitespace is allowed there!
 -->

<!ENTITY % Fieldset.content  "( #PCDATA | legend | %Flow.mix; )*" >
<!ELEMENT fieldset  %Fieldset.content; >
<!ATTLIST fieldset      %Common.attrib;
>

<![%XHTML.Transitional;[
<!ENTITY % LegendAlign.attrib
     "align        (top|bottom|left|right)  #IMPLIED" >
]]>
<!ENTITY % LegendAlign.attrib  "" >

<!ENTITY % Legend.content  "( #PCDATA | %Inline.mix; )*" >
<!ELEMENT legend  %Legend.content; >
<!ATTLIST legend      %Common.attrib;
```

```
      accesskey    %Character;              #IMPLIED
      %LegendAlign.attrib;
>

<!ENTITY % Button.content
     "( #PCDATA
      | %Heading.class;
      | %List.class;
      | %Inlpres.class;
      | %Inlphras.class;
      | %Block-noform.mix;
      | img | object | map )*"
>
<!ELEMENT button  %Button.content; >
<!ATTLIST button      %Common.attrib;
      name         CDATA                    #IMPLIED
      value        CDATA                    #IMPLIED
      type         (button|submit|reset)    'submit'
      disabled     (disabled)               #IMPLIED
      tabindex     %Number;                 #IMPLIED
      accesskey    %Character;              #IMPLIED
>

<!-- end of forms.mod -->
```

C.9. Reformulated XHTML 1.0 DTDs

This section defines the three XHTML 1.0 DTDs based upon the modules defined above. The content models of these DTD do not differ from their predecessors in XHTML 1.0 [XHTML1] [p.58] .

C.9.1. XHTML 1.0 Strict

```
<!-- ...................................................................... -->
<!-- XHTML 1.0 Strict DTD  ................................................ -->
<!-- file: XHTML1-s.dtd
-->

<!--  XHTML 1.0 Strict DTD

      This is XHTML 1.0, an XML reformulation of HTML 4.0.

      Copyright 1998-1999 World Wide Web Consortium
         (Massachusetts Institute of Technology, Institut National de
          Recherche en Informatique et en Automatique, Keio University).
          All Rights Reserved.

      Permission to use, copy, modify and distribute the XHTML 1.0 DTD and
      its accompanying documentation for any purpose and without fee is
      hereby granted in perpetuity, provided that the above copyright notice
      and this paragraph appear in all copies.  The copyright holders make
      no representation about the suitability of the DTD for any purpose.

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

```
     Author:     Murray M. Altheim <altheim@eng.sun.com>
     Revision:   @(#)XHTML1-s.dtd 1.14 99/04/01 SMI

   The XHTML 1.0 DTD is an XML variant based on the W3C HTML 4.0 DTD:

     Draft:      $Date: 1999/04/06 21:40:14 $

     Authors:    Dave Raggett <dsr@w3.org>
                 Arnaud Le Hors <lehors@w3.org>
                 Ian Jacobs <ij@w3.org>

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

      Please use this formal public identifier to identify it:

          "-//W3C//DTD XHTML 1.0 Strict//EN"

      Please use this URI to identify the default namespace:

          "http://www.w3.org/TR/1999/REC-html-in-xml"

      For example, if you are using XHTML 1.0 directly, use the FPI
      in the DOCTYPE declaration, with the xmlns attribute on the
      document element to identify the default namespace:

        <?xml version="1.0" ?>
        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
                              "XHTML1-s.dtd" >
        <html xmlns="http://www.w3.org/TR/1999/REC-html-in-xml"
              xml:lang="en" lang="en" >
        ...
        </html>
-->

<!-- The version attribute has historically been a container for the
     DTD's public identifier (an FPI), but is unused in Strict:  -->
<!ENTITY % HTML.version  "" >
<!ENTITY % Version.attrib "" >

<!-- The xmlns attribute on <html> identifies the
     default namespace to namespace-aware applications:  -->
<!ENTITY % XHTML.ns  "http://www.w3.org/TR/1999/REC-html-in-xml" >

<!-- reserved for future use with document profiles -->
<!ENTITY % XHTML.profile  "" >

<!-- used to ignore Transitional features within modules -->
<!ENTITY % XHTML.Transitional    "IGNORE" >


<!-- XHTML Base Architecture Module (optional) ......... -->
<!ENTITY % XHTML1-arch.module "IGNORE" >
<![%XHTML1-arch.module;[
<!ENTITY % XHTML1-arch.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Base Architecture//EN"
```

```
            "XHTML1-arch.mod" >
%XHTML1-arch.mod;
]]>

<!-- Common Names Module ............................. -->
<!ENTITY % XHTML1-names.module "INCLUDE" >
<![%XHTML1-names.module;[
<!ENTITY % XHTML1-names.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
            "XHTML1-names.mod" >
%XHTML1-names.mod;
]]>

<!-- Character Entities Module ........................ -->
<!ENTITY % XHTML1-charent.module "INCLUDE" >
<![%XHTML1-charent.module;[
<!ENTITY % XHTML1-charent.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
            "XHTML1-charent.mod" >
%XHTML1-charent.mod;
]]>

<!-- Intrinsic Events Module .......................... -->
<!ENTITY % XHTML1-events.module "INCLUDE" >
<![%XHTML1-events.module;[
<!ENTITY % XHTML1-events.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
            "XHTML1-events.mod" >
%XHTML1-events.mod;
]]>

<!-- Common Attributes Module ......................... -->
<!ENTITY % XHTML1-attribs.module "INCLUDE" >
<![%XHTML1-attribs.module;[
<!ENTITY % align "" >
<!ENTITY % XHTML1-attribs.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Attributes//EN"
            "XHTML1-attribs.mod" >
%XHTML1-attribs.mod;
]]>

<!-- Document Model Module ........................... -->
<!ENTITY % XHTML1-model.module "INCLUDE" >
<![%XHTML1-model.module;[
<!ENTITY % XHTML1-model.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Model//EN"
            "XHTML1-model.mod" >
%XHTML1-model.mod;
]]>

<!-- Inline Structural Module ......................... -->
<!ENTITY % XHTML1-inlstruct.module "INCLUDE" >
<![%XHTML1-inlstruct.module;[
<!ENTITY % XHTML1-inlstruct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
            "XHTML1-inlstruct.mod" >
%XHTML1-inlstruct.mod;
```

```
]]>

<!-- Inline Presentational Module ..................... -->
<!ENTITY % XHTML1-inlpres.module "INCLUDE" >
<![%XHTML1-inlpres.module;[
<!ENTITY % XHTML1-inlpres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"
            "XHTML1-inlpres.mod" >
%XHTML1-inlpres.mod;
]]>

<!-- Inline Phrasal Module ........................... -->
<!ENTITY % XHTML1-inlphras.module "INCLUDE" >
<![%XHTML1-inlphras.module;[
<!ENTITY % XHTML1-inlphras.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
            "XHTML1-inlphras.mod" >
%XHTML1-inlphras.mod;
]]>

<!-- Block Structural Module ......................... -->
<!ENTITY % XHTML1-blkstruct.module "INCLUDE" >
<![%XHTML1-blkstruct.module;[
<!ENTITY % XHTML1-blkstruct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
            "XHTML1-blkstruct.mod" >
%XHTML1-blkstruct.mod;
]]>

<!-- Block Presentational Module ..................... -->
<!ENTITY % XHTML1-blkpres.module "INCLUDE" >
<![%XHTML1-blkpres.module;[
<!ENTITY % XHTML1-blkpres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"
            "XHTML1-blkpres.mod" >
%XHTML1-blkpres.mod;
]]>

<!-- Block Phrasal Module ............................ -->
<!ENTITY % XHTML1-blkphras.module "INCLUDE" >
<![%XHTML1-blkphras.module;[
<!ENTITY % XHTML1-blkphras.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
            "XHTML1-blkphras.mod" >
%XHTML1-blkphras.mod;
]]>

<!-- Scripting Module ................................ -->
<!ENTITY % XHTML1-script.module "INCLUDE" >
<![%XHTML1-script.module;[
<!ENTITY % XHTML1-script.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
            "XHTML1-script.mod" >
%XHTML1-script.mod;
]]>

<!-- Stylesheets Module .............................. -->
```

```
<!ENTITY % XHTML1-style.module "INCLUDE" >
<![%XHTML1-style.module;[
<!ENTITY % XHTML1-style.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Stylesheets//EN"
            "XHTML1-style.mod" >
%XHTML1-style.mod;
]]>

<!-- Image Module ..................................... -->
<!ENTITY % XHTML1-image.module "INCLUDE" >
<![%XHTML1-image.module;[
<!ENTITY % XHTML1-image.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
            "XHTML1-image.mod" >
%XHTML1-image.mod;
]]>

<!-- Frames Module .................................... -->
<!ENTITY % XHTML1-frames.module  "IGNORE" >
<![%XHTML1-frames.module;[
<!ENTITY % XHTML1-frames.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
            "XHTML1-frames.mod" >
%XHTML1-frames.mod;
]]>

<!-- Linking Module ................................... -->
<!ENTITY % XHTML1-linking.module "INCLUDE" >
<![%XHTML1-linking.module;[
<!ENTITY % XHTML1-linking.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
            "XHTML1-linking.mod" >
%XHTML1-linking.mod;
]]>

<!-- Client-side Image Map Module ..................... -->
<!ENTITY % XHTML1-csismap.module "INCLUDE" >
<![%XHTML1-csismap.module;[
<!ENTITY % XHTML1-csismap.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Map//EN"
            "XHTML1-csismap.mod" >
%XHTML1-csismap.mod;
]]>

<!-- Object Element Module ............................ -->
<!ENTITY % XHTML1-object.module "INCLUDE" >
<![%XHTML1-object.module;[
<!ENTITY % XHTML1-object.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
            "XHTML1-object.mod" >
%XHTML1-object.mod;
]]>

<!-- Lists Module ..................................... -->
<!ENTITY % XHTML1-list.module "INCLUDE" >
<![%XHTML1-list.module;[
<!ENTITY % XHTML1-list.mod
```

```
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
            "XHTML1-list.mod" >
%XHTML1-list.mod;
]]>

<!-- Forms Module ..................................... -->
<!ENTITY % XHTML1-form.module "INCLUDE" >
<![%XHTML1-form.module;[
<!ENTITY % XHTML1-form.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
            "XHTML1-form.mod" >
%XHTML1-form.mod;
]]>

<!-- Tables Module .................................... -->
<!ENTITY % XHTML1-table.module "INCLUDE" >
<![%XHTML1-table.module;[
<!ENTITY % XHTML1-table.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
            "XHTML1-table.mod" >
%XHTML1-table.mod;
]]>

<!-- Document Metainformation Module ................... -->
<!ENTITY % XHTML1-meta.module "INCLUDE" >
<![%XHTML1-meta.module;[
<!ENTITY % XHTML1-meta.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
            "XHTML1-meta.mod" >
%XHTML1-meta.mod;
]]>

<!-- Document Structure Module ......................... -->
<!ENTITY % XHTML1-struct.module "INCLUDE" >
<![%XHTML1-struct.module;[
<!ENTITY % XHTML1-struct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
            "XHTML1-struct.mod" >
%XHTML1-struct.mod;
]]>

<!-- end of XHTML 1.0 Strict DTD  ......................................... -->
<!-- ....................................................................... -->
```

C.9.2. XHTML 1.0 Transitional

```
<!-- ....................................................................... -->
<!-- XHTML 1.0 Transitional DTD  .......................................... -->
<!-- file: XHTML1-t.dtd
-->

<!--  XHTML 1.0 Transitional DTD

      This is XHTML 1.0, an XML reformulation of HTML 4.0.

      Copyright 1998-1999 World Wide Web Consortium
```

```
             (Massachusetts Institute of Technology, Institut National de
              Recherche en Informatique et en Automatique, Keio University).
              All Rights Reserved.

         Permission to use, copy, modify and distribute the XHTML 1.0 DTD and
         its accompanying documentation for any purpose and without fee is
         hereby granted in perpetuity, provided that the above copyright notice
         and this paragraph appear in all copies.  The copyright holders make
         no representation about the suitability of the DTD for any purpose.

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

           Author:     Murray M. Altheim <altheim@eng.sun.com>
           Revision:   @(#)XHTML1-t.dtd 1.14 99/04/01 SMI

         The XHTML 1.0 DTD is an XML variant based on the W3C HTML 4.0 DTD:

           Draft:      $Date: 1999/04/06 21:40:14 $

           Authors:    Dave Raggett <dsr@w3.org>
                       Arnaud Le Hors <lehors@w3.org>
                       Ian Jacobs <ij@w3.org>

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

         Please use this formal public identifier to identify it:

             "-//W3C//DTD XHTML 1.0 Transitional//EN"

         Please use this URI to identify the default namespace:

             "http://www.w3.org/TR/1999/REC-html-in-xml"

         For example, if you are using XHTML 1.0 directly, use the FPI
         in the DOCTYPE declaration, with the xmlns attribute on the
         document element to identify the default namespace:

           <?xml version="1.0" ?>
           <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
                                 "XHTML1-t.dtd" >
           <html xmlns="http://www.w3.org/TR/1999/REC-html-in-xml"
                 xml:lang="en" lang="en" >
           ...
           </html>
   -->

   <!-- The version attribute has historically been a container
        for the DTD's public identifier (an FPI):  -->
   <!ENTITY % HTML.version  "-//W3C//DTD XHTML 1.0 Transitional//EN" >

   <!-- The xmlns attribute on <html> identifies the
        default namespace to namespace-aware applications:  -->
   <!ENTITY % XHTML.ns  "http://www.w3.org/TR/1999/REC-html-in-xml" >

   <!-- reserved for future use with document profiles -->
   <!ENTITY % XHTML.profile  "" >
```

```
<!ENTITY % XHTML1-frames.module  "IGNORE" >
<!ENTITY % XHTML.Transitional    "INCLUDE" >


<!-- XHTML Base Architecture Module (optional) ......... -->
<!ENTITY % XHTML1-arch.module "IGNORE" >
<![%XHTML1-arch.module;[
<!ENTITY % XHTML1-arch.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Base Architecture//EN"
            "XHTML1-arch.mod" >
%XHTML1-arch.mod;
]]>

<!-- Common Names Module .............................. -->
<!ENTITY % XHTML1-names.module "INCLUDE" >
<![%XHTML1-names.module;[
<!ENTITY % XHTML1-names.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Common Names//EN"
            "XHTML1-names.mod" >
%XHTML1-names.mod;
]]>

<!-- Character Entities Module ........................ -->
<!ENTITY % XHTML1-charent.module "INCLUDE" >
<![%XHTML1-charent.module;[
<!ENTITY % XHTML1-charent.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Character Entities//EN"
            "XHTML1-charent.mod" >
%XHTML1-charent.mod;
]]>

<!-- Intrinsic Events Module .......................... -->
<!ENTITY % XHTML1-events.module "INCLUDE" >
<![%XHTML1-events.module;[
<!ENTITY % XHTML1-events.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Intrinsic Events//EN"
            "XHTML1-events.mod" >
%XHTML1-events.mod;
]]>

<!-- Transitional Attributes Module ................... -->
<!ENTITY % XHTML1-attribs-t.module "INCLUDE" >
<![%XHTML1-attribs-t.module;[
<!ENTITY % XHTML1-attribs-t.mod
     PUBLIC "-//W3C//ENTITIES XHTML 1.0 Transitional Attributes//EN"
            "XHTML1-attribs-t.mod" >
%XHTML1-attribs-t.mod;
]]>

<!-- Transitional Document Model Module ............... -->
<!ENTITY % XHTML1-model-t.module "INCLUDE" >
<![%XHTML1-model-t.module;[
<!ENTITY % XHTML1-model-t.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Transitional Document Model//EN"
            "XHTML1-model-t.mod" >
%XHTML1-model-t.mod;
```

```
]]>

<!-- Inline Structural Module ......................... -->
<!ENTITY % XHTML1-inlstruct.module "INCLUDE" >
<![%XHTML1-inlstruct.module;[
<!ENTITY % XHTML1-inlstruct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Structural//EN"
            "XHTML1-inlstruct.mod" >
%XHTML1-inlstruct.mod;
]]>

<!-- Inline Presentational Module ..................... -->
<!ENTITY % XHTML1-inlpres.module "INCLUDE" >
<![%XHTML1-inlpres.module;[
<!ENTITY % XHTML1-inlpres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Presentational//EN"
            "XHTML1-inlpres.mod" >
%XHTML1-inlpres.mod;
]]>

<!-- Inline Phrasal Module ............................ -->
<!ENTITY % XHTML1-inlphras.module "INCLUDE" >
<![%XHTML1-inlphras.module;[
<!ENTITY % XHTML1-inlphras.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Inline Phrasal//EN"
            "XHTML1-inlphras.mod" >
%XHTML1-inlphras.mod;
]]>

<!-- Block Structural Module .......................... -->
<!ENTITY % XHTML1-blkstruct.module "INCLUDE" >
<![%XHTML1-blkstruct.module;[
<!ENTITY % XHTML1-blkstruct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Structural//EN"
            "XHTML1-blkstruct.mod" >
%XHTML1-blkstruct.mod;
]]>

<!-- Block Presentational Module ...................... -->
<!ENTITY % XHTML1-blkpres.module "INCLUDE" >
<![%XHTML1-blkpres.module;[
<!ENTITY % XHTML1-blkpres.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Presentational//EN"
            "XHTML1-blkpres.mod" >
%XHTML1-blkpres.mod;
]]>

<!-- Block Phrasal Module ............................. -->
<!ENTITY % XHTML1-blkphras.module "INCLUDE" >
<![%XHTML1-blkphras.module;[
<!ENTITY % XHTML1-blkphras.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Block Phrasal//EN"
            "XHTML1-blkphras.mod" >
%XHTML1-blkphras.mod;
]]>

<!-- Scripting Module ................................. -->
```

```
<!ENTITY % XHTML1-script.module "INCLUDE" >
<![%XHTML1-script.module;[
<!ENTITY % XHTML1-script.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Scripting//EN"
            "XHTML1-script.mod" >
%XHTML1-script.mod;
]]>

<!-- Stylesheets Module ............................... -->
<!ENTITY % XHTML1-style.module "INCLUDE" >
<![%XHTML1-style.module;[
<!ENTITY % XHTML1-style.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Stylesheets//EN"
            "XHTML1-style.mod" >
%XHTML1-style.mod;
]]>

<!-- Image Module ..................................... -->
<!ENTITY % XHTML1-image.module "INCLUDE" >
<![%XHTML1-image.module;[
<!ENTITY % XHTML1-image.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Images//EN"
            "XHTML1-image.mod" >
%XHTML1-image.mod;
]]>

<!-- Frames Module .................................... -->
<![%XHTML1-frames.module;[
<!ENTITY % XHTML1-frames.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Frames//EN"
            "XHTML1-frames.mod" >
%XHTML1-frames.mod;
]]>

<!-- Linking Module ................................... -->
<!ENTITY % XHTML1-linking.module "INCLUDE" >
<![%XHTML1-linking.module;[
<!ENTITY % XHTML1-linking.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Linking//EN"
            "XHTML1-linking.mod" >
%XHTML1-linking.mod;
]]>

<!-- Client-side Image Map Module ..................... -->
<!ENTITY % XHTML1-csismap.module "INCLUDE" >
<![%XHTML1-csismap.module;[
<!ENTITY % XHTML1-csismap.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Client-side Image Map//EN"
            "XHTML1-csismap.mod" >
%XHTML1-csismap.mod;
]]>

<!-- Object Element Module ............................ -->
<!ENTITY % XHTML1-object.module "INCLUDE" >
<![%XHTML1-object.module;[
<!ENTITY % XHTML1-object.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Object Element//EN"
```

```
              "XHTML1-object.mod" >
%XHTML1-object.mod;
]]>

<!-- Java Applet Element Module ....................... -->
<!ENTITY % XHTML1-applet.module "INCLUDE" >
<![%XHTML1-applet.module;[
<!ENTITY % XHTML1-applet.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Java Applets//EN"
              "XHTML1-applet.mod" >
%XHTML1-applet.mod;
]]>

<!-- Lists Module ..................................... -->
<!ENTITY % XHTML1-list.module "INCLUDE" >
<![%XHTML1-list.module;[
<!ENTITY % XHTML1-list.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Lists//EN"
              "XHTML1-list.mod" >
%XHTML1-list.mod;
]]>

<!-- Forms Module ..................................... -->
<!ENTITY % XHTML1-form.module "INCLUDE" >
<![%XHTML1-form.module;[
<!ENTITY % XHTML1-form.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Forms//EN"
              "XHTML1-form.mod" >
%XHTML1-form.mod;
]]>

<!-- Tables Module .................................... -->
<!ENTITY % XHTML1-table.module "INCLUDE" >
<![%XHTML1-table.module;[
<!ENTITY % XHTML1-table.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Tables//EN"
              "XHTML1-table.mod" >
%XHTML1-table.mod;
]]>

<!-- Document Metainformation Module .................. -->
<!ENTITY % XHTML1-meta.module "INCLUDE" >
<![%XHTML1-meta.module;[
<!ENTITY % XHTML1-meta.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Metainformation//EN"
              "XHTML1-meta.mod" >
%XHTML1-meta.mod;
]]>

<!-- Document Structure Module ........................ -->
<!ENTITY % XHTML1-struct.module "INCLUDE" >
<![%XHTML1-struct.module;[
<!ENTITY % XHTML1-struct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML 1.0 Document Structure//EN"
              "XHTML1-struct.mod" >
%XHTML1-struct.mod;
```

```
]]>

<!-- end of XHTML 1.0 Transitional DTD  ................................... -->
<!-- ....................................................................... -->
```

C.9.3. XHTML 1.0 Frameset

```
<!-- ....................................................................... -->
<!-- XHTML 1.0 Frameset DTD  ............................................... -->
<!-- file: XHTMl1-f.dtd
-->

<!--  XHTML 1.0 Frameset DTD

      This is XHTML 1.0, an XML reformulation of HTML 4.0.

      Copyright 1998-1999 World Wide Web Consortium
         (Massachusetts Institute of Technology, Institut National de
          Recherche en Informatique et en Automatique, Keio University).
          All Rights Reserved.

      Permission to use, copy, modify and distribute the XHTML 1.0 DTD and
      its accompanying documentation for any purpose and without fee is
      hereby granted in perpetuity, provided that the above copyright notice
      and this paragraph appear in all copies.  The copyright holders make
      no representation about the suitability of the DTD for any purpose.

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

        Author:     Murray M. Altheim <altheim@eng.sun.com>
        Revision:   @(#)XHTML1-f.dtd 1.17 99/04/01 SMI

      The XHTML 1.0 DTD is an XML variant based on the W3C HTML 4.0 DTD:

        Draft:      $Date: 1999/04/06 21:40:14 $

        Authors:    Dave Raggett <dsr@w3.org>
                    Arnaud Le Hors <lehors@w3.org>
                    Ian Jacobs <ij@w3.org>

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

      Please use this formal public identifier to identify it:

          "-//W3C//DTD XHTML 1.0 Frameset//EN"

      Please use this URI to identify the default namespace:

          "http://www.w3.org/TR/1999/REC-html-in-xml"

      For example, if you are using XHTML 1.0 directly, use the FPI
      in the DOCTYPE declaration, with the xmlns attribute on the
      document element to identify the default namespace:

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

```
        <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
                              "XHTML1-f.dtd" >
        <html xmlns="http://www.w3.org/TR/1999/REC-html-in-xml"
              xml:lang="en" lang="en" >
        ...
        </html>
-->

<!-- The version attribute has historically been a container
     for the DTD's public identifier (an FPI):  -->
<!ENTITY % HTML.version  "-//W3C//DTD XHTML 1.0 Frameset//EN" >

<!-- The xmlns attribute on <html> identifies the
     default namespace to namespace-aware applications:  -->
<!ENTITY % XHTML.ns  "http://www.w3.org/TR/1999/REC-html-in-xml" >

<!-- reserved for future use with document profiles -->
<!ENTITY % XHTML.profile  "" >

<!ENTITY % XHTML1-frames.module  "INCLUDE" >
<!ENTITY % XHTML.Transitional    "INCLUDE" >


<!-- declare and instantiate the XHTML Transitional DTD  -->
<!ENTITY % XHTML1-t.dtd
     PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
            "XHTML1-t.dtd" >
%XHTML1-t.dtd;

<!-- end of XHTML 1.0 Frameset DTD  ....................................... -->
<!-- ..................................................................... -->
```