XHTML™ Basic 1.1

W3C Working Draft 7 June 2006

This version:
http://www.w3.org/TR/2006/WD-xhtml-basic-20060607
Latest version:
http://www.w3.org/TR/xhtml-basic
Previous version:
http://www.w3.org/TR/2000/REC-xhtml-basic-20001219
Diff-marked version:
xhtml-basic-diff.html
Version 1.1 Editors:
Shane McCarron, Applied Testing and Technology, Inc.
Masayasu Ishikawa, W3C
Editors:
Mark Baker, Sun Microsystems
Masayasu Ishikawa, W3C
Shinichi Matsui, Panasonic
Peter Stark, Ericsson
Ted Wugofski, Openwave Systems
Toshihiko Yamakami, ACCESS Co., Ltd.

This document is also available in these non-normative formats: PostScript version, PDF version, ZIP archive, and Gzip'd TAR archive.

Abstract

The XHTML Basic document type includes the minimal set of modules required to be an XHTML host language document type, and in addition it includes images, forms, basic tables, and object support. It is designed for Web clients that do not support the full set of XHTML features; for example, Web clients such as mobile phones, PDAs, pagers, and settop boxes. The document type is rich enough for content authoring.

XHTML Basic is designed as a common base that may be extended. The goal of XHTML Basic is to serve as a common language supported by various kinds of user agents.

This revision, 1.1, supercedes version 1.0 as defined in http://www.w3.org/TR/2000/REC-xhtml-basic-20001219. In this revision, four new features have been incorporated into the language in order to better serve the small-device community that is this language's major user:

1. Intrinsic Events (defined in [XHTMLMOD [p.23]])
2. The target attribute (defined in [XHTMLMOD [p.23]])
3. The style element (defined in [XHTMLMOD [p.23]])
4. The inputmode attribute (defined in Section 5 [p.15] of this document)

The document type definition is implemented using XHTML modules as defined in "*XHTML Modularization*" [XHTMLMOD [p.23]].

Status of this Document

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

This document is a First Public Working Draft, and is developed in preparation for entering the W3C Last Call Announcement process. It reflects clarifications, corrections, and extensions as a result of over four years of use by the community. It reflects our intent to incorporate a few new modules into XHTML Basic. *Note that the DTD implementation with this draft does not reflect these changes.* It is hereby made available for review by W3C members and other interested parties. If this document is approved as a W3C Recommendation, it will supersede the 19 December 2000 version of the the XHTML Basic Recommendation.

This document has been produced by the W3C HTML Working Group (*members only*) as part of the W3C HTML Activity.

Public discussion of HTML takes place on www-html@w3.org (archive). To subscribe send an email to www-html-request@w3.org with the word *subscribe* in the subject line.

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

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

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

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential

Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

1. Introduction

1.1. XHTML for Small Information Appliances

HTML 4 is a powerful language for authoring Web content, but its design does not take into consideration issues pertinent to small devices, including the implementation cost (in power, memory, etc.) of the full feature set. Consumer devices with limited resources cannot generally afford to implement the full feature set of HTML 4. Requiring a full-fledged computer for access to the World Wide Web excludes a large portion of the population from consumer device access of online information and services.

Because there are many ways to subset HTML, there are many almost identical subsets defined by organizations and companies. Without a common base set of features, developing applications for a wide range of Web clients is difficult.

The motivation for XHTML Basic is to provide an XHTML document type that can be shared across communities (e.g. desktop, TV, and mobile phones), and that is rich enough to be used for simple content authoring. New community-wide document types can be defined by extending XHTML Basic in such a way that XHTML Basic documents are in the set of valid documents of the new document type. Thus an XHTML Basic document can be presented on the maximum number of Web clients.

The document type definition for XHTML Basic is implemented based on the XHTML modules defined in XHTML Modularization [XHTMLMOD [p.23]].

1.2. Background and Requirements

Information appliances are targeted for particular uses. They support the features they need for the functions they are designed to fulfill. The following are examples of different information appliances:

- Mobile phones
- Televisions
- PDAs
- Vending machines
- Pagers
- Car navigation systems
- Mobile game machines
- Digital book readers
- Smart watches

Existing subsets and variants of HTML for these clients include Compact HTML [CHTML [p.23]], the Wireless Markup Language [WML [p.24]], and the "HTML 4.0 Guidelines for Mobile Access" [GUIDELINES [p.23]]. The common features found in these document types include:

- Basic text (including headings, paragraphs, and lists)
- Hyperlinks and links to related documents
- Basic forms
- Basic tables
- Images
- Meta information

This set of HTML features has been the starting point for the design of XHTML Basic. Since many content developers are familiar with these HTML features, they comprise a useful host language that may be combined with markup modules from other languages according to the methods described in "*XHTML Modularization*" [XHTMLMOD [p.23]]. For example, XHTML Basic may be extended with a custom module to support richer markup semantics in specific environments.

It is not the intention of XHTML Basic to limit the functionality of future languages. But since the features in HTML 4 (frames, advanced tables, etc.) were developed for a desktop computer type of client, they have proved to be inappropriate for many non-desktop devices. XHTML Basic will be extended and built upon. Extending XHTML from a common and basic set of features, instead of almost identical subsets or the too-large set of functions in HTML 4, will be good for interoperability on the Web, as well as for scalability.

Compared to the rich functionality of HTML 4, XHTML Basic may look like one step back, but in fact, it is two steps forward for clients that do not need what is in HTML 4 and for content developers who get one XHTML subset instead of many.

1.3. Design Rationale

This section explains why certain HTML features are not part of XHTML Basic.

1.3.1. Presentation

Many simple Web clients cannot display fonts other than monospace. Bi-directional text, bold faced font, and other text extension elements are not supported.

It is recommended that style sheets be used to create a presentation that is appropriate for the device.

1.3.2. Forms

Basic XHTML forms ([XHTMLMOD [p.23]], section 5.5.1) are supported. Since only devices with a local file system can take advantage of file and image input types in forms, they are not included in the basic forms. Also, content developers should keep in mind that users may not be able to input many characters from some devices (e.g. from a mobile phone).

1.3.3. Tables

Basic XHTML tables ([XHTMLMOD [p.23]], section 5.6.1) are supported, but tables can be difficult to display on small devices. It is recommended that content developers follow the Web Content Accessibility Guidelines 1.0 for creating accessible tables ([WCAG10 [p.24]], Guideline 5). Note that in the Basic Tables Module, nesting of tables is prohibited.

1.3.4. Frames

Frames are not supported. Frames depend on a screen interface and may not be applicable to some small appliances like phones, pagers, and watches.

2. Conformance

This section is *normative.*

2.1. Document Conformance

A Conforming XHTML Basic document is a document that requires only the facilities described as mandatory in this specification. Such a document must meet all of the following criteria:

1. The document must conform to the constraints expressed in Appendix B [p.25] .
2. The root element of the document must be `<html>`.
3. The name of the default namespace on the root element must be the XHTML namespace name, `http://www.w3.org/1999/xhtml`.
4. There must be a DOCTYPE declaration in the document prior to the root element. If present, the public identifier included in the DOCTYPE declaration must reference the DTD found in Appendix B [p.25] using its Formal Public Identifier. The system identifier may be modified appropriately.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
```

5. The DTD subset must not be used to override any parameter entities in the DTD.

2.2. User Agent Conformance

The user agent must conform to the "User Agent Conformance" section of the XHTML 1.0 specification ([XHTML1 [p.23]], section 3.2).

3. The XHTML Basic Document Type

This section is *normative*.

The XHTML Basic document type is defined as a set of XHTML modules. All XHTML modules are defined in the "*XHTML Modularization*" specification [XHTMLMOD [p.23]].

XHTML Basic consists of the following XHTML modules:

Structure Module*
`body, head, html, title`
Text Module*
`abbr, acronym, address, blockquote, br, cite, code, dfn, div, em, h1, h2, h3, h4, h5, h6, kbd, p, pre, q, samp, span, strong, var`
Hypertext Module*
`a`
List Module*
`dl, dt, dd, ol, ul, li`
Basic Forms Module
`form, input, label, select, option, textarea`
Basic Tables Module
`caption, table, td, th, tr`
Image Module
`img`
Object Module
`object, param`
Metainformation Module
`meta`
Link Module
`link`
Base Module
`base`
Intrinsic Events module
Events attributes
Scripting module
`script` and `noscript` elements
Stylesheet module
`style` element
Target Module
`target` attribute

(*) = This module is a required XHTML Host Language module.

An XML 1.0 DTD is available in Appendix B. [p.25]

NOTE: Since the HTML event handler attributes are not included in XHTML Basic, form controls outside forms may not function as expected by the user.

4. How to Use XHTML Basic

Although XHTML Basic can be used as it is - a simple XHTML language with text, links, and images - the intention of its simple design is for use as a host language. A host language can contain a mix of vocabularies all rolled into one document type. It is natural that XHTML is the host language, since that is what most Web developers are used to.

When markup from other languages is added to XHTML Basic, the resulting document type will be an extension of XHTML Basic. Content developers can develop for XHTML Basic or take advantage of the extensions. The goal of XHTML Basic is to serve as a common language supported by various kinds of user agents.

5. XHTML inputmode Attribute Module

This section is *normative*.

This section was originally a component of XForms 1.0 [p.24] , and was written by Martin Duerst.

The inputmode Attribute Module defines the `inputmode` attribute.

inputmode = CDATA
    This attribute specifies style information for the current element.

The following table shows additional attributes for elements defined elsewhere when the inputmode module is selected.

Elements	Attributes	Notes
input&	inputmode (CDATA)	When the Basic Forms or Forms Module is selected.

The attribute `inputmode` provides a *hint* to the user agent to select an appropriate input mode for the text input expected in an associated form control. The input mode may be a keyboard configuration, an input method editor (also called front end processor) or any other setting affecting input on the device(s) used.

Using `inputmode`, the author can give hints to the agent that make form input easier for the user. Authors should provide `inputmode` attributes wherever possible, making sure that the values used cover a wide range of devices.

5.1 `inputmode` Attribute Value Syntax

The value of the `inputmode` attribute is a white space separated list of tokens. Tokens are either sequences of alphabetic letters or absolute URIs. The later can be distinguished from the former by noting that absolute URIs contain a ':'. Tokens are case-sensitive. All the tokens consisting of alphabetic letters only are defined in this specification, in **5.3 List of Tokens** [p.17] (or a successor of this specification).

This specification does not define any URIs for use as tokens, but allows others to define such URIs for extensibility. This may become necessary for devices with input modes that cannot be covered by the tokens provided here. The URI should dereference to a human-readable description of the input mode associated with the use of the URI as a token. This description should describe the input mode indicated by this token, and whether and how this token modifies other tokens or is modified by other tokens.

5.2 User Agent Behavior

Upon entering an empty form control with an `inputmode` attribute, the user agent should select the input mode indicated by the `inputmode` attribute value. User agents should not use the `inputmode` attribute to set the input mode when entering a form control with text already present. To set the appropriate input mode when entering a form control that already contains text, user agents should rely on platform-specific conventions.

User agents should make available all the input modes which are supported by the (operating) system/device(s) they run on/have access to, and which are installed for regular use by the user. This is typically only a small subset of the input modes that can be described with the tokens defined here.

Note:

Additional guidelines for user agent implementation are found at [UAAG 1.0] [p.23] .

The following simple algorithm is used to define how user agents match the values of an `inputmode` attribute to the input modes they can provide. This algorithm does not have to be implemented directly; user agents just have to behave as if they used it. The algorithm is not designed to produce "obvious" or "desirable" results for every possible combination of tokens, but to produce correct behavior for frequent token combinations and predictable behavior in all cases.

First, each of the input modes available is represented by one or more lists of tokens. An input mode may correspond to more than one list of tokens; as an example, on a system set up for a Greek user, both "greek upperCase" and "user upperCase" would correspond to the same input mode. No two lists will be the same.

Second, the `inputmode` attribute is scanned from front to back. For each token *t* in the `inputmode` attribute, if in the remaining list of tokens representing available input modes there is any list of tokens that contains *t*, then all lists of tokens representing available input modes that do not contain *t* are removed. If there is no remaining list of tokens that contains *t*, then *t* is ignored.

Third, if one or more lists of tokens are left, and they all correspond to the same input mode, then this input mode is chosen. If no list is left (meaning that there was none at the start) or if the remaining lists correspond to more than one input mode, then no input mode is chosen.

Example: Assume the list of lists of tokens representing the available input modes is: {"cyrillic upperCase", "cyrillic lowerCase", "cyrillic", "latin", "user upperCase", "user lowerCase"}, then the following `inputmode` values select the following input modes: "cyrillic title" selects "cyrillic", "cyrillic lowerCase" selects "cyrillic lowerCase", "lowerCase cyrillic" selects "cyrillic lowerCase", "latin upperCase" selects "latin", but "upperCase latin" does select "cyrillic upperCase" or "user upperCase" if they correspond to the same input mode, and does not select any input mode if "cyrillic upperCase" and "user upperCase" do not correspond to the same input mode.

5.3 List of Tokens

Tokens defined in this specification are separated into two categories: *Script tokens* and *modifiers*. In `inputmode` attributes, script tokens should always be listed before modifiers.

5.3.1 Script Tokens

Script tokens provide a general indication the set of characters that is covered by an input mode. In most cases, script tokens correspond directly to [Unicode Scripts] [p.23] . Some tokens correspond to the block names in Java class java.lang.Character.UnicodeBlock ([Java Unicode Blocks] [p.23]) or Unicode Block names. However, this neither means that an input mode has to allow input for all the characters in the script or block, nor that an input mode is limited to only characters from that specific script. As an example, a "latin" keyboard doesn't cover all the characters in the Latin script, and includes punctuation which is not assigned to the Latin script. The version of the Unicode Standard that these script names are taken from is 3.2.

Input Mode Token	Comments
arabic	Unicode script name
armenian	Unicode script name
bengali	Unicode script name
bopomofo	Unicode script name
braille	used to input braille patterns (not to indicate a braille input device)
buhid	Unicode script name
canadianAboriginal	Unicode script name
cherokee	Unicode script name
cyrillic	Unicode script name
deseret	Unicode script name
devanagari	Unicode script name
ethiopic	Unicode script name
georgian	Unicode script name
greek	Unicode script name
gothic	Unicode script name
gujarati	Unicode script name
gurmukhi	Unicode script name

Input Mode Token	Comments
han	Unicode script name
hangul	Unicode script name
hanja	Subset of ’han’ used in writing Korean
hanunoo	Unicode script name
hebrew	Unicode script name
hiragana	Unicode script name (may include other Japanese scripts produced by conversion from hiragana)
ipa	International Phonetic Alphabet
kanji	Subset of ’han’ used in writing Japanese
kannada	Unicode script name
katakana	Unicode script name (full-width, not half-width)
khmer	Unicode script name
lao	Unicode script name
latin	Unicode script name
malayalam	Unicode script name
math	mathematical symbols and related characters
mongolian	Unicode script name
myanmar	Unicode script name
ogham	Unicode script name
oldItalic	Unico de script name
oriya	Unicode script name
runic	Unicode script name
simplifiedHanzi	Subset of ’han’ used in writing Simplified Chinese
sinhala	Unicode script name
syriac	Unicode script name
tagalog	Unicode script name
tagbanwa	Unicode script name

Input Mode Token	Comments
tamil	Unicode script name
telugu	Unicode script name
thaana	Unicode script name
thai	Unicode script name
tibetan	Unicode script name
traditionalHanzi	Subset of ’han’ used in writing Traditional Chinese
user	Special value denoting the ’native’ input of the user (e.g. to input her name or text in her native language).
yi	Unicode script name

5.3.2 Modifier Tokens

Modifier tokens can be added to the scripts they apply in order to more closely specify the kind of characters expected in the form control. Traditional PC keyboards do not need most modifier tokens (indeed, users on such devices would be quite confused if the software decided to change case on its own; CAPS lock for upperCase may be an exception). However, modifier tokens can be very helpful to set input modes for small devices.

Input Mode Token	Comments
lowerCase	lowercase (for bicameral scripts)
upperCase	uppercase (for bicameral scripts)
titleCase	title case (for bicameral scripts): words start with an upper case letter
startUpper	start input with one uppercase letter, then continue with lowercase letters
digits	digits of a particular script (e.g. inputmode=’thai digits’)
symbols	symbols, punctuation (suitable for a particular script)
predictOn	text prediction switched on (e.g. for running text)
predictOff	text prediction switched off (e.g. for passwords)
halfWidth	half-width compatibility forms (e.g. Katakana; deprecated)

5.4 Relationship to XML Schema pattern facets

User agents may use information available in an XML Schema pattern facet to set the input mode. Note that a pattern facet is a hard restriction on the lexical value of an instance data node, and can specify different restrictions for different parts of the data item. Attribute `inputmode` is a soft hint about the kinds of characters that the user may most probably start to input into the form control. Attribute `inputmode` is provided in addition to pattern facets for the following reasons:

1. The set of allowable characters specified in a pattern may be so wide that it is not possible to deduce a reasonable input mode setting. Nevertheless, there frequently is a kind of characters that will be input by the user with high probability. In such a case, `inputmode` allows to set the input mode for the user's convenience.

2. In some cases, it would be possible to derive the input mode setting from the pattern because the set of characters allowed in the pattern closely corresponds to a set of characters covered by an `inputmode` attribute value. However, such a derivation would require a lot of data and calculations on the user agent.

3. Small devices may leave the checking of patterns to the server, but will easily be able to switch to those input modes that they support. Being able to make data entry for the user easier is of particular importance on small devices.

5.5 Examples

This is an example of a form for Japanese address input. It is shown in table form; it will be replaced by actual syntax in a later version of this specification.

Caption:	**`inputmode`**
Family name	hiragana
(in kana)	katakana
Given name	hiragana
(in kana)	katakana
Zip code	latin digits
Address	hiragana
(in kana)	katakana
Email	latin lowerCase
Telephone	latin digits
Comments	user predictOn

6. Acknowledgements

This specification was prepared by the W3C HTML Working Group. At the time of publication of the first edition, the members were:

- Steven Pemberton, CWI (HTML Working Group Chair)
- Robert Adams, Intel (until November 2000)
- Murray Altheim, Sun Microsystems
- Takuya Asada, W3C (until October 2000)
- Daniel Austin, Mozquito Technologies
- Mark Baker, Sun Microsystems
- Wayne Carr, Intel
- Tantek Çelik, Microsoft
- Andrew W. Donoho, IBM
- Herman Elenbaas, Philips Electronics
- Beth Epperson, Netscape/AOL
- Masayasu Ishikawa, W3C (HTML Activity Lead)
- Shin'ichi Matsui, Panasonic
- Shane McCarron, Applied Testing and Technology
- Ann Navarro, WebGeek, Inc.
- Dave Raggett, W3C/Openwave Systems
- Sebastian Schnitzenbaumer, Mozquito Technologies (until September 2000)
- Peter Stark, Ericsson
- Michel Suignard, Microsoft
- Markku Vartiainen, Openwave Systems
- Jeremy Wadsworth, Quark Inc.
- Malte Wedel, Mozquito Technologies
- Linda Welsh, Intel
- Ted Wugofski, Openwave Systems

At the time of publication of the second edition, the members were:

(insert the list here)

Thanks to Gary Adams (Sun Microsystems), Jonny Axelsson (Metastasis design), Peter Chen (Philips), Dan Connolly (W3C), John Cowan (Reuters), Martin J. Dürst (W3C), Johan Hjelm (Ericsson), Ian Jacobs (W3C), Susan Lesch (W3C), Louis Theran (Nokia), Quinton Zondervan (Lotus), members of the W3C Mobile Access Interest Group, the W3C Synchronized Multimedia Working Group, the W3C WAI Protocols and Formats Working Group, and the WAP Forum, for contributing, reviewing and commenting on this document.

A. References

A.1. Normative References

[HTML4]
"*HTML 4.01 Specification*", W3C Recommendation, D. Raggett, A. Le Hors, I. Jacobs, eds., 24 December 1999. Available at: http://www.w3.org/TR/1999/REC-html401-19991224
The latest version is available at: http://www.w3.org/TR/html4

[XHTML1]
"*XHTML 1.0: The Extensible HyperText Markup Language (Second Edition) - A Reformulation of HTML 4 in XML 1.0*", W3C Recommendation, Steven Pemberton et al., 26 January 2000, revised 1 August 2002. Available at: http://www.w3.org/TR/2002/REC-xhtml1-20020801
The latest version is available at: http://www.w3.org/TR/xhtml1

[XHTMLMOD]
"*XHTML Modularization 1.0*", W3C Recommendation, Shane McCarron, et. al. eds., 10 April 2001. Available at: http://www.w3.org/TR/2001/REC-xhtml-modularization-20010410
The latest version is available at: http://www.w3.org/TR/xhtml-modularization

[XML]
"*Extensible Markup Language (XML) 1.0 (Third Edition)*", W3C Recommendation, T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, F. Yergeau, eds., 4 February 2004. Available at: http://www.w3.org/TR/2004/REC-xml-20040204
The latest version is available at: http://www.w3.org/TR/REC-xml

A.2. Informative References

[CHTML]
"*Compact HTML for Small Information Appliances*", W3C Note, T. Kamada, 9 February 1998. Available at: http://www.w3.org/TR/1998/NOTE-compactHTML-19980209

[GUIDELINES]
"*HTML 4.0 Guidelines for Mobile Access*, W3C Note, T. Kamada, T. Asada, M. Ishikawa, S. Matsui, eds., 15 March 1999. Available at: http://www.w3.org/TR/1999/NOTE-html40-mobile-19990315
The latest version is available at: http://www.w3.org/TR/NOTE-html40-mobile

Java Unicode Blocks
Java 2 Platform, Standard Edition, v 1.4.0 API Specification; Class Character.UnicodeBlock, Sun Microsystems, Inc, 2002. Available at http://java.sun.com/j2se/1.4/docs/api/java/lang/Character.UnicodeBlock.html.

UAAG 1.0
User Agent Accessibility Guidelines 1.0, Ian Jacobs, Jon Gunderson, Eric Hansen, 2002. Working Draft available at http://www.w3.org/TR/UAAG10/.

Unicode Scripts
Script Names, Mark Davis, 2001. Unicode Technical Report #24 available at http://www.unicode.org/unicode/reports/tr24/.

[WCAG10]
"*Web Content Accessibility Guidelines 1.0*", W3C Recommendation, W. Chisholm, G. Vanderheiden, I. Jacobs, eds., 5 May 1999. Available at: http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505
The latest version is available at: http://www.w3.org/TR/WCAG10
[WML]
"*Wireless Markup Language Specification*", WAP Forum Ltd. The WAP Forum has consolidated into the Open Mobile Alliance (OMA). The specification work from WAP continues within OMA and can be found on the OMA Web site at: http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html
[XFORMS]
Need real reference text here

B. XHTML Basic Document Type Definition

This appendix is *normative*.

B.1. SGML Open Catalog Entry for XHTML Basic

This section contains the SGML Open Catalog-format definition of the public identifiers for XHTML Basic.

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

--  XHTML Basic Catalog Data File

    Revision:  $Id: xhtml-basic10.cat,v 2.4 2000/12/18 21:42:58 mimasa Exp $ SMI

    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/html/tr9401.html>
--

-- ..................................................................... --
-- SGML declaration associated with XML  ............................... --

OVERRIDE YES

SGMLDECL "xml1.dcl"

-- ::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::: --

-- XHTML Basic DTD modular driver file  ................................ --

PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"                    "xhtml-basic10.dtd"

-- XHTML Basic framework module ....................................... --

PUBLIC "-//W3C//ENTITIES XHTML Basic 1.0 Document Model 1.0//EN"  "xhtml-basic10-model-1.mod"


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

B.2. XHTML Basic Driver

This section contains the driver for the XHTML Basic document type implementation as an XML DTD. It relies upon XHTML module implementations defined in [XHTMLMOD [p.23]].

```
<!-- XHTML Basic 1.0 DTD  ................................................ -->
<!-- file: xhtml-basic10.dtd -->

<!-- XHTML Basic 1.0 DTD

     This is XHTML Basic, a proper subset of XHTML.

     The Extensible HyperText Markup Language (XHTML)
```

```
     Copyright 1998-2005 World Wide Web Consortium
        (Massachusetts Institute of Technology, European Research Consortium
         for Informatics and Mathematics, Keio University).
         All Rights Reserved.

     Permission to use, copy, modify and distribute the XHTML Basic 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.

        Editors:    Murray M. Altheim <mailto:altheim@eng.sun.com>
                    Peter Stark       <mailto:Peter.Stark@ecs.ericsson.se>
        Revision:   $Id: xhtml-basic10.dtd,v 2.13 2000/12/18 12:56:23 mimasa Exp $ SMI

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

     This DTD is identified by the PUBLIC and SYSTEM identifiers:

     PUBLIC: "-//W3C//DTD XHTML Basic 1.0//EN"
     SYSTEM: "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd"
-->
<!ENTITY % XHTML.version  "-//W3C//DTD XHTML Basic 1.0//EN" >

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

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

     See the Qualified Names module for information
     on the use of namespace prefixes in the DTD.
-->
<!ENTITY % NS.prefixed "IGNORE" >
<!ENTITY % XHTML.prefix  "" >

<!-- Reserved for use with the XLink namespace:
-->
<!ENTITY % XLINK.xmlns "" >
<!ENTITY % XLINK.xmlns.attrib "" >

<!-- For example, if you are using XHTML Basic 1.0 directly, use
     the public identifier in the DOCTYPE declaration, with the namespace
     declaration on the document element to identify the default namespace:

         <?xml version="1.0"?>
         <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
             "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd" >
         <html xmlns="http://www.w3.org/1999/xhtml"
               xml:lang="en" >
         ...
         </html>
-->

<!-- reserved for future use with document profiles -->
<!ENTITY % XHTML.profile  "" >
```

```
<!-- Bidirectional Text features
     This feature-test entity is used to declare elements
     and attributes used for bidirectional text support.
-->
<!ENTITY % XHTML.bidi  "IGNORE" >

<?doc type="doctype" role="title" { XHTML Basic 1.0 } ?>

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

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

<!ENTITY % xhtml-model.mod
     PUBLIC "-//W3C//ENTITIES XHTML Basic 1.0 Document Model 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-basic10-model-1.mod" >

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

<!ENTITY % pre.content
     "( #PCDATA
      | %InlStruct.class;
      %InlPhras.class;
      %Anchor.class;
      %Inline.extra; )*"
>

<!ENTITY % xhtml-text.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Text 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-text-1.mod" >
%xhtml-text.mod;

<!ENTITY % xhtml-hypertext.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Hypertext 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-hypertext-1.mod" >
%xhtml-hypertext.mod;

<!ENTITY % xhtml-list.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Lists 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-list-1.mod" >
%xhtml-list.mod;

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

<!-- Image Module  ............................................ -->
<!ENTITY % xhtml-image.module "INCLUDE" >
<![%xhtml-image.module;[
<!ENTITY % xhtml-image.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Images 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-image-1.mod" >
%xhtml-image.mod;]]>

<!-- Tables Module ............................................ -->
<!ENTITY % xhtml-table.module "INCLUDE" >
<![%xhtml-table.module;[
```

```
<!ENTITY % xhtml-table.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Basic Tables 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-basic-table-1.mod" >
%xhtml-table.mod;]]>

<!-- Forms Module  ..................................................... -->
<!ENTITY % xhtml-form.module "INCLUDE" >
<![%xhtml-form.module;[
<!ENTITY % xhtml-form.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Basic Forms 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-basic-form-1.mod" >
%xhtml-form.mod;]]>

<!-- Link Element Module  ............................................... -->
<!ENTITY % xhtml-link.module "INCLUDE" >
<![%xhtml-link.module;[
<!ENTITY % xhtml-link.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Link Element 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-link-1.mod" >
%xhtml-link.mod;]]>

<!-- Document Metainformation Module  ................................... -->
<!ENTITY % xhtml-meta.module "INCLUDE" >
<![%xhtml-meta.module;[
<!ENTITY % xhtml-meta.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Metainformation 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-meta-1.mod" >
%xhtml-meta.mod;]]>

<!-- Base Element Module  ............................................... -->
<!ENTITY % xhtml-base.module "INCLUDE" >
<![%xhtml-base.module;[
<!ENTITY % xhtml-base.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Base Element 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-base-1.mod" >
%xhtml-base.mod;]]>

<!-- Param Element Module  .............................................. -->
<!ENTITY % xhtml-param.module "INCLUDE" >
<![%xhtml-param.module;[
<!ENTITY % xhtml-param.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Param Element 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-param-1.mod" >
%xhtml-param.mod;]]>

<!-- Embedded Object Module  ............................................ -->
<!ENTITY % xhtml-object.module "INCLUDE" >
<![%xhtml-object.module;[
<!ENTITY % xhtml-object.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Embedded Object 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-object-1.mod" >
%xhtml-object.mod;]]>

<!-- Target Attribute Module  ........................................... -->
<!ENTITY % xhtml-target.module "INCLUDE" >
<![%xhtml-target.module;[
<!ENTITY % xhtml-target.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Target 1.0//EN"
```

```
            "http://www.w3.org/MarkUp/DTD/xhtml-target-1.mod" >
%xhtml-target.mod;]]>


<!ENTITY % xhtml-struct.mod
     PUBLIC "-//W3C//ELEMENTS XHTML Document Structure 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-struct-1.mod" >
%xhtml-struct.mod;

<!-- end of XHTML Basic 1.0 DTD  ..................................... -->
```

B.3. XHTML Basic Customizations

An XHTML Family Document Type (such as XHTML Basic) must define the content model that it uses. This is done through a separate content model module that is instantiated by the XHTML Modular Framework. The content model module and the XHTML Basic Driver (above) work together to customize the module implementations to the document type's specific requirements. The content model module for XHTML Basic is defined below:

```
<!-- ..................................................................... -->
<!-- XHTML Basic 1.0 Document Model Module  ............................... -->
<!-- file: xhtml-basic10-model-1.mod

     This is XHTML Basic, a proper subset of XHTML.
     Copyright 1998-2005 W3C (MIT, ERCIM, Keio), All Rights Reserved.
     Revision: $Id: xhtml-basic10-model-1.mod,v 2.8 2000/11/03 14:28:25 mimasa Exp $ SMI

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

       PUBLIC "-//W3C//ENTITIES XHTML Basic 1.0 Document Model 1.0//EN"
       SYSTEM "http://www.w3.org/MarkUp/DTD/xhtml-basic10-model-1.mod"

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

<!-- XHTML Basic Document Model

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

<!-- Optional Elements in head  .............. -->

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

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

<!ENTITY % Misc.class "" >

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

<!ENTITY % InlStruct.class "%br.qname; | %span.qname;" >

<!ENTITY % InlPhras.class
     "| %em.qname; | %strong.qname; | %dfn.qname; | %code.qname;
```

```
      | %samp.qname; | %kbd.qname; | %var.qname; | %cite.qname;
      | %abbr.qname; | %acronym.qname; | %q.qname;" >

<!ENTITY % InlPres.class "" >

<!ENTITY % I18n.class "" >

<!ENTITY % Anchor.class "| %a.qname;" >

<!ENTITY % InlSpecial.class "| %img.qname; | %object.qname;" >

<!ENTITY % InlForm.class
     "| %input.qname; | %select.qname; | %textarea.qname;
      | %label.qname;"
>

<!ENTITY % Inline.extra "" >

<!ENTITY % Inline.class
     "%InlStruct.class;
      %InlPhras.class;
      %Anchor.class;
      %InlSpecial.class;
      %InlForm.class;
      %Inline.extra;"
>

<!ENTITY % InlNoAnchor.class
     "%InlStruct.class;
      %InlPhras.class;
      %InlSpecial.class;
      %InlForm.class;
      %Inline.extra;"
>

<!ENTITY % InlNoAnchor.mix
     "%InlNoAnchor.class;
      %Misc.class;"
>

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

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

<!ENTITY % Heading.class
     "%h1.qname; | %h2.qname; | %h3.qname;
      | %h4.qname; | %h5.qname; | %h6.qname;"
>
<!ENTITY % List.class  "%ul.qname; | %ol.qname; | %dl.qname;" >

<!ENTITY % Table.class "| %table.qname;" >

<!ENTITY % Form.class  "| %form.qname;" >

<!ENTITY % BlkStruct.class "%p.qname; | %div.qname;" >

<!ENTITY % BlkPhras.class
```

```
     "| %pre.qname; | %blockquote.qname; | %address.qname;"
>

<!ENTITY % BlkPres.class "" >

<!ENTITY % BlkSpecial.class
     "%Table.class;
      %Form.class;"
>

<!ENTITY % Block.extra "" >

<!ENTITY % Block.class
     "%BlkStruct.class;
      %BlkPhras.class;
      %BlkSpecial.class;
      %Block.extra;"
>

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

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

<!-- declares all content except tables
-->
<!ENTITY % FlowNoTable.mix
     "%Heading.class;
      | %List.class;
      | %BlkStruct.class;
      %BlkPhras.class;
      %Form.class;
      %Block.extra;
      | %Inline.class;
      %Misc.class;"
>

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

<!-- end of xhtml-basic10-model-1.mod -->
```