XHTML Access Module

Module to enable generic document accessibility

W3C Working Group Note 16 December 2010

This version:
http://www.w3.org/TR/2010/NOTE-xhtml-access-20101216
Latest version:
http://www.w3.org/TR/xhtml-access
Previous version:
http://www.w3.org/TR/2008/WD-xhtml-access-20080526
Diff from previous version:
xhtml-access-diff.html
Editors:
Mark Birbeck, Sidewinder Labs mark.birbeck@sidewinderlabs.com
Shane McCarron, Applied Testing and Technology, Inc.
Steven Pemberton, CWI/W3C®
T. V. Raman, Google, Inc.
Richard Schwerdtfeger, IBM Corporation

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

The English version of this specification is the only normative version. Non-normative translations may also be available.

Copyright © 2007-2009 W3C® (MIT, ERCIM, Keio), All Rights Reserved. W3C liability, trademark and document use rules apply.

Abstract

The XHTML Access module defines an element that, when used in conjunction with other XHTML modules in XHTML Family Markup Languages, enables a more robust accessibility model than is presently possible.

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 Working Group Note. The XHTML2 Working Group's charter expired before it could complete work on this document. It is possible that the work will be continued by another group in the future.

This version is based upon comments received during the Candidate Recommendation period, upon work done in the definition of [XHTML2 [p.22]], and upon work done by the RDF-in-HTML Task Force, a joint task force of the Semantic Web Best Practices and Deployment Working Group and XHTML 2 Working Group. It is considered mature and stable by the working group. Comments on this document should be addressed to www-html-editor@w3.org. All comments sent to that address are available in a (public archive).

Publication as a Working Group Note 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 has been produced by the W3C XHTML 2 Working Group as part of the HTML Activity. The goals of the XHTML 2 Working Group are discussed in the XHTML 2 Working Group charter.

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.

Please report errors in this specification to www-html-editor@w3.org (archive). It is inappropriate to send discussion email to this address. Public discussion may take place on www-html@w3.org (archive).

Table of Contents

1. Introduction 5
2. Conformance Requirements 7
2.1. Document Conformance 7
2.2. Host Language Conformance 7
2.3. User Agent Conformance 8
3. XHTML Access Module 9
3.1. The `access` element 9
3.1.1. `activate` = (true | false*) 9

3.1.2. `key` = `Characters` 10
3.1.3. `media` = MediaDesc 11
3.1.4. `order` = (document* | list) 11
3.1.5. `targetid` = `IDREFs` 12
3.1.6. `targetrole` = `CURIEs` 12
3.2. Examples 12
A. Schema Implementation 15
A.1. Access Element Module 15
B. DTD Implementation 17
B.1. Qualified Names Module 17
B.2. Element Definition Module 18
C. References 21
C.1. Normative References 21
C.2. Other References 21
D. Acknowledgments 23

1. Introduction

This section is informative.

Most desktop applications provide a way for the user to navigate or activate specific functionality through the use of the keyboard alone, particularly by using keyboard shortcuts, which are a single key or combination of keys. Web pages and Web Applications have traditionally been less able to do so due to conflicting shortcuts within the operating system or browser itself, and due also to a lack of a coherent robust mechanism. Thus, Web resources have relied primarily on interaction via pointing devices, such as a mouse. This specification defines a way to assign character mappings (e.g. keyboard shortcuts, or keys combinations) to navigate to specific elements or sequential sets of elements, which may be activated by the user, or which may be activated immediately, based on the author's intent. It also addresses the need for end users to be able to remap these keys for personalizing the interaction, and describes how a browser might expose these character mappings to the user.

This document contains a single module designed to be used to help make more effective at supporting the needs of the Accessibility Community. It has been developed in conjunction with the W3C's Web Accessibility Initiative and other interested parties. The module herein contains functionality that is the logical successor to the `accesskey` attribute [HTML4 [p.21]], [XHTML1 [p.22]].

2. Conformance Requirements

This section is *normative*.

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119 [p.21]].

Note that all examples in this document are informative, and are not meant to be interpreted as normative requirements.

2.1. Document Conformance

XHTML Access is not a stand-alone document type. It is intended to be integrated into other XHTML Family Markup Languages. A conforming XHTML Access document is a document that requires only the facilities described as mandatory in this specification and the facilities described as mandatory in its host language. Such a document must meet all the following criteria:

1. The document MUST conform to the constraints expressed in its host language implementation.

2. If the host language *is* in the XHTML Namespace, there are no additional requirements. If the host language *is not* in the XHTML namespace, and the host language does not incorporate this module into its own namespace, then the document MUST contain an `xmlns:` declaration for the XHTML Access namespace [XMLNAMES [p.21]]. The namespace for XHTML Access Module is defined to be `http://www.w3.org/1999/xhtml`. An example start tag of a root element might look like:

   Example

   ```
   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" >
   ```

2.2. Host Language Conformance

When XHTML Access is included in a host language, all of the facilities required in this specification MUST be included in the host language. In addition, the element defined in this specification MUST be included in the content model of the host language. The element defined in this specification MAY be incorporated into the namespace of the host language, or it MAY remain in the XHTML namespace. Finally, XHTML Access requires the availability of the XHTML Role Attribute Module [XHTMLROLE [p.21]].

2.3. User Agent Conformance

A conforming user agent MUST support all of the features required in this specification.

3. XHTML Access Module

This section is *normative*.

This module defines the access element.

Element	Attributes	Minimal Content Model
access	activate, key, media, order, targetid, targetrole	EMPTY

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

Implementations: XML Schema [p.15] , XML DTD [p.17]

3.1. The `access` element

The access element assigns a mapping between "keys" or other events to elements within a document. Actuating the mapping results in the element gaining focus and potentially in additional events being activated.

The access element allows an author to specify a relationship between "key(s)" or other events and one or more elements within a document. When a mapped event is triggered, one of the specified target elements gains focus, and one or more other events (such as an 'activate' event) may also be dispatched to that element. The target elements are specified by means of the targetid or targetrole attributes, and these elements may each contain multiple targets, to allow sequential focus by successively triggering the associated key(s) or event(s).

If the access element's activate attribute has the value 'true', then in addition to receiving focus, the target element will be activated, if permitted by the user's settings. Additionally, using XML Events [XMLEVENTS [p.22]], one or more other events may also be dispatched to the element.

An access element must have either a targetrole or a targetid attribute specified. If neither a targetrole nor a targetid attribute are specified, or if there are no matching elements in the document, the user agent MUST NOT define a mapping for this element, nor change focus nor dispatch any events based on this element.

Attributes

3.1.1. `activate` = (true | false*)

The activate attribute indicates whether a target element should be activated or not once it obtains focus. The default value for this attribute is "false", indicating that the element will not be "activated". User agents MUST provide mechanisms for overriding the author setting with user-specified settings in order to ensure that the act of moving content focus does not cause the user agent to take any further action (as per Checkpoint 9.5 of UAAG 1.0 [UAAG1 [p.22]]).

User agents MUST provide keyboard mechanisms for "activating" any event associated with the focused element (as per Checkpoint 1.2 of UAAG 1.0). User agents SHOULD make available the list of events associated with the focused element (as per Checkpoint 9.6 of UAAG 1.0).

3.1.2. `key` = `Characters`

This attribute assigns one or more key mappings to an access shortcut. The value of is attribute is one or more single characters from the document character set.

The key attribute represents an abstraction. The use of the name "key" for this attribute is historical and does not mean that there is any association with a specific "key" on a keyboard, per se. It is up to the user agent to provide a mechanism for mapping the document character set value(s) of the attribute to the input methods available to the user agent. For instance, on some systems a user may have to press an "alt" or "cmd" key in addition to the access key. On other systems there may be voice commands, or gestures with a mouse, an eye tracking input device, etc. Regardless of the mechanism, the result is that it appears to the `access` element processor as if the defined *key* was entered.

A user entering any of the keys defined in an access element moves focus from its current position to the next element in navigation order that has one of the referenced role or id values (see activate for information on how the element may be activated). Note that it is possible to deliver alternate events via [XMLEVENTS [p.22]]. Note also that the concept of *navigation order* is a property of the Host Language, and is not defined in this specification.

User agents MUST provide mechanisms for overriding the author setting with user-specified settings in order to ensure that the act of moving content focus does not cause the user agent to take any further action, as required by UAAG 1.0, Checkpoint 9.5. [UAAG1 [p.22]] The character assigned to a key, and its relationship to a role or id attribute SHOULD be treated as an author suggestion. User agents MAY override any key assignment (e.g., if an assignment interferes with the operation of the user interface of the user agent, if the key is not available on a device, if a key is used by the operating environment). User agents MUST also allow users to override author assigned keys with their own key assignments (see Checkpoint 11.3 of UAAG 1.0). If a user chooses to change the key binding, the resultant user-defined remapping SHOULD persist across sessions.

If no key attribute is specified, the user agent SHOULD assign a key and alert the user to the key mapping. The resultant user agent assigned key mapping SHOULD persist across sessions.

It is common for user agents to provide a visual hint for accessing features via keyboard shortcuts, such as showing the shortcut next to a menu item, or underlining the shortcut character letter in a label. The rendering of such visual hints about access keys depends on the user agent. We recommend that authors include the access key character in label text or wherever the access key is to apply. If the user agent can recognize that the currently mapped access key character appears in the label text of the element to which it is mapped, then the user agent may render the character in such a way as to emphasize its role as the access key and distinguish it from other characters (e.g., by underlining it).

A conforming user agent SHOULD also provide a centralized view of the current access key assignments (see Checkpoint 11.1 and Checkpoint 11.2 of UAAG 1.0).

Authors MUST NOT assign the same key value to more than one *active* `access` element. Note that it is possible to have an `access` element be *inactive* through the use of the media attribute.

Authors are cautioned that not all characters are appropriate as access key values, since not all characters can be accessed directly from the keyboard. Some characters can only be generated when combined with base characters. Examples include: combining vowels or tone marks, such as are used in Arabic, Southeast Asian, or Indic scripts. These are more difficult to communicate to users because, while they can often be typed independently, they are not typically rendered independently and the user might not know which character is intended as the key mapping. Finally, authors are cautioned that any key available in one keyboard might not be available in a different keyboard layout.

3.1.3. `media` = MediaDesc

The value of this attribute is a comma-separated list of media descriptors for which this `access` element is intended. When the value of this attribute matches the current processing media, the associated `access` element is considered *active* and processed normally; otherwise it is *inactive* and ignored. The default value for this attribute is `all`.

3.1.4. `order` = (document* | list)

The order attribute indicates how this `access` element should determine the *navigation order* for its *matching elements*. The default value, `document`, indicates that the items MUST be traversed in document order. The alternate value, `list`, indicates that the *navigation order* of *matching elements* is determined by the author-defined order of items in the list of targetid or targetrole attribute values.

For the sake of determining *navigation order*, elements in the document that match the values in the targetid or targetrole attributes are called *matching elements*, and all elements that match the same value are members of an *element group*. Note: since the `id` of an element must be unique within a valid XML document, in such documents, each *element group* based on targetid values consist of no more than one *matching element*.

For each navigation operation, the *navigation order* for both *document-order* and *list-order* is sensitive to the current *focus* element. For document-order, if no element currently has *focus*, the first *matching element* in the document MUST be the *focus target*; if an element does have *focus*, the next *matching element* in the document MUST be the *focus target*. For list-order, the *focus target navigation order* is determined as follows:

- If no *matching element* of this `access` element currently has *focus*, the *focus target* MUST be the first element in document order that matches the first *element group*. If there is no such element, then the *focus target* is the first element that matches the second *element group*, and so on.
- If a *matching element* of this `access` element already has *focus*:

1. If there are additional *matching elements* of the same *element group* in document order, then *focus* MUST be sent to the next *matching element* of the same *element group*.
2. Otherwise, *focus* MUST go to the first *matching element* (in document order) of the next *element group*.
3. If there are no remaining elements groups, then the search MUST resume from the first *element group*.

The location of the next *matching element* MUST be determined each time the `access` element is triggered, since it is possible that between events the contents of the document will have changed.

A host language MUST define any circumstances under which an element would not qualify as a *matching element* (e.g., in XHTML 1.1 if a Form control were "disabled" it might not qualify.)

3.1.5. `targetid` = `IDREFs`

The targetid attribute specifies one or more space separated IDREFs related to target elements for the associated event (i.e., the node to which the event should be delivered).

3.1.6. `targetrole` = `CURIEs`

The targetrole attribute specifies a space separated list of `CURIE`s [CURIE [p.21]] that maps to an element with a role attribute with the same value.

If a targetid and a targetrole are both specified for an element, a user agent MUST only use the values from the the targetid attribute.

If the prefix is omitted from a CURIE, as the default value of `http://www.w3.org/1999/xhtml/vocab#` MUST be used. [XHTMLROLE [p.21]] *Many common accessibility roles are defined by the vocabulary at that URI. See [XHTMLVOCAB [p.22]] for more information.*

3.2. Examples

Access element that focuses into a field

```
<access key="s"
        title="Social Security Number"
        targetrole="ss:number" />
```

Accessing a table of contents

```
<access key="c"
        title="Table of Contents"
        targetrole="toc" />
```

Access that moves to the main content

```
<access key="m"
        title="Main content"
        targetrole="main" />
```

Access that moves among form controls in document order

```
<access key="i"
        title="Form Controls"
        order="document"
        targetrole="button checkbox option radio textbox" />
```

Access that moves among form controls in specific order

```
<access key="i"
        title="Form Controls"
        order="list"
        targetrole="button checkbox option radio textbox" />
```

Access element that goes to a specific element

```
<access key="u"
        title="Username"
        targetid="username" />
```

Access element with no specific key mapping

```
<access title="Navigation bar"
        targetrole="navigation" />
```

A. Schema Implementation

This appendix is *normative*.

The schema implementation of XHTML Access Module conforms to the requirements defined in [XHTMLMOD [p.21]]. It is included here as an example implementation.

A.1. Access Element Module

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified"
 xmlns:xh11d="http://www.w3.org/1999/xhtml/datatypes/"
>
    <xs:import namespace="http://www.w3.org/1999/xhtml/datatypes/"
               schemaLocation="xhtml-datatypes-1.xsd" />

    <xs:annotation>
        <xs:documentation>
      This is the XML Schema module for XHTML Access
      $Id: xhtml-access.ps,v 1.4 2010/12/16 16:48:31 plehegar Exp $
    </xs:documentation>
        <xs:documentation source="xhtml-copyright-1.xsd"/>
        <xs:documentation source="http://www.w3.org/TR/xhtml-role#A_role"/>
    </xs:annotation>
    <xs:attributeGroup name="xhtml.access.attlist">
        <xs:attributeGroup ref="xhtml.Common.attrib"/>
        <xs:attribute name="activate" default="no">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="yes"/>
                    <xs:enumeration value="no"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="key" type="xh11d:Character"/>
        <xs:attribute name="media" type="xh11d:MediaDesc"/>
        <xs:attribute name="order" default="document">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="document"/>
                    <xs:enumeration value="list"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="targetid">
            <xs:simpleType>
                <xs:list itemType="xs:IDREF"/>
            </xs:simpleType>
        </xs:attribute>
        <xs:attribute name="targetrole" type="xh11d:CURIEs"/>
    </xs:attributeGroup>
    <xs:group name="xhtml.access.content">
```

```
        <xs:sequence/>
    </xs:group>
    <xs:complexType name="xhtml.access.type">
        <xs:group ref="xhtml.access.content"/>
        <xs:attributeGroup ref="xhtml.access.attlist"/>
    </xs:complexType>
</xs:schema>
```

B. DTD Implementation

This appendix is *normative*.

The DTD implementation of XHTML Access Module conforms to the requirements defined in [XHTMLMOD [p.21]]. Consequently, it provides a Qualified Names sub-module, and a module file for the XHTML Access Module defined in this specification.

B.1. Qualified Names Module

```
<!-- ....................................................................... -->
<!-- XHTML Access Qname Module  ............................................ -->
<!-- file: xhtml-access-qname-1.mod

     This is XHTML Access - the Access Attribute Module for XHTML.

     Copyright 2007-2008 W3C (MIT, ERCIM, Keio), All Rights Reserved.

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

       PUBLIC "-//W3C//ENTITIES XHTML Access Attribute Qnames 1.0//EN"
       SYSTEM "http://www.w3.org/MarkUp/DTD/xhtml-access-qname-1.mod"

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

<!-- XHTML Access Attribute Qname (Qualified Name) Module

     This module is contained in two parts, labeled Section 'A' and 'B':

       Section A declares parameter entities to support namespace-
       qualified names, namespace declarations, and name prefixing
       for XHTML Access and extensions.

       Section B declares parameter entities used to provide
       namespace-qualified names for the XHTML access element:

         %XHTML.access.qname;   the xmlns-qualified name for access
         ...

     XHTML Access extensions would create a module similar to this one.
-->

<!-- Section A: XHTML Access Attribute XML Namespace Framework :::::::::::::: -->

<!-- 1. Declare a %XHTML.prefixed; conditional section keyword, used
        to activate namespace prefixing. The default value should
        inherit '%NS.prefixed;' from the DTD driver, so that unless
        overridden, the default behavior follows the overall DTD
        prefixing scheme.
-->
<!ENTITY % NS.prefixed "IGNORE" >
<!ENTITY % XHTML.prefixed "%NS.prefixed;" >
```

```
<!-- 2. Declare a parameter entity (eg., %XHTML.xmlns;) containing
        the URI reference used to identify the XHTML Access Attribute namespace
-->
<!ENTITY % XHTML.xmlns  "http://www.w3.org/1999/xhtml" >

<!-- 3. Declare parameter entities (eg., %XML.prefix;) containing
        the default namespace prefix string(s) to use when prefixing
        is enabled. This may be overridden in the DTD driver or the
        internal subset of an document instance. If no default prefix
        is desired, this may be declared as an empty string.

     NOTE: As specified in [XMLNAMES], the namespace prefix serves
     as a proxy for the URI reference, and is not in itself significant.
-->
<!ENTITY % XHTML.prefix  "" >

<!-- 4. Declare parameter entities (eg., %XHTML.pfx;) containing the
        colonized prefix(es) (eg., '%XHTML.prefix;:') used when
        prefixing is active, an empty string when it is not.
-->
<![%XHTML.prefixed;[
<!ENTITY % XHTML.pfx  "%XHTML.prefix;:" >
]]>
<!ENTITY % XHTML.pfx  "" >

<!-- declare qualified name extensions here ............ -->
<!ENTITY % xhtml-access-qname-extra.mod "" >
%xhtml-access-qname-extra.mod;

<!-- 5. The parameter entity %XHTML.xmlns.extra.attrib; may be
        redeclared to contain any non-XHTML Access namespace
        declaration attributes for namespaces embedded in XML. The default
        is an empty string.  XLink should be included here if used
        in the DTD.
-->
<!ENTITY % XHTML.xmlns.extra.attrib "" >


<!-- Section B: XML Qualified Names ::::::::::::::::::::::::::::: -->

<!-- 6. This section declares parameter entities used to provide
        namespace-qualified names for the XHTML Access element.
-->

<!ENTITY % XHTML.access.qname  "%XHTML.pfx;access" >

<!-- end of xhtml-access-qname-1.mod -->
```

B.2. Element Definition Module

```
<!-- .................................................................... -->
<!-- XHTML Access Module ................................................. -->
<!-- file: xhtml-access-1.mod

     This is XHTML Access - the Access Module for XHTML.
```

```
      Copyright 2007-2008 W3C (MIT, ERCIM, Keio), All Rights Reserved.

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

        PUBLIC "-//W3C//ELEMENTS XHTML Access Element 1.0//EN"
        SYSTEM "http://www.w3.org/MarkUp/DTD/xhtml-access-1.mod"

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


<!ENTITY % xhtml-datatypes.module "INCLUDE" >
<![%xhtml-datatypes.module;[
<!ENTITY % xhtml-datatypes.mod
     PUBLIC "-//W3C//ENTITIES XHTML Datatypes 1.0//EN"
            "http://www.w3.org/MarkUp/DTD/xhtml-datatypes-1.mod" >
%xhtml-datatypes.mod;]]>

<!ENTITY % XHTML.access.element  "INCLUDE" >
<![%XHTML.access.element;[
<!ENTITY % XHTML.access.content  "EMPTY" >
<!ENTITY % XHTML.access.qname  "access" >
<!ELEMENT %XHTML.access.qname;  %XHTML.access.content; >
<!-- end of XHTML.access.element -->]]>

<!ENTITY % XHTML.access.attlist  "INCLUDE" >
<![%XHTML.access.attlist;[
<!ATTLIST %XHTML.access.qname;
      %XHTML.access.Common.attrib;
      activate     ( true | false )         'false'
      order        ( document | list )      'document'
      key          %Character.datatype;     #IMPLIED
      media        %MediaDesc.datatype;     #IMPLIED
      targetid     %IDREFs.datatype;        #IMPLIED
      targetrole   %CURIEs.datatype;        #IMPLIED
>
<!-- end of XHTML.access.attlist -->]]>

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

C. References

This appendix is *normative*.

C.1. Normative References

[CURIE]
"*CURIE Syntax 1.0*", W3C Candidate Recommendation, M. Birbeck, S. McCarron, *ed.*, 16 January 2009.
Available at: http://www.w3.org/TR/2009/CR-curie-20090116/
The latest version is available at: http://www.w3.org/TR/curie

[DOM2EVENTS]
"*Document Object Model (DOM) Level 2 Events Specification*", W3C Recommendation, T. Pixley, *ed.*, 13 November 2000.
Available at: http://www.w3.org/TR/DOM-Level-2-Events/
The latest version is available at: http://www.w3.org/TR/DOM-Level-2-Events

[IRI]
"*Internationalized Resource Identifiers (IRI)*", RFC 3987, M.Duerst, M. Suignard January 2005.
Available at: http://www.ietf.org/rfc/rfc3987.txt

[RFC2119]
"*Key words for use in RFCs to indicate requirement levels*", RFC 2119, S. Bradner, March 1997.
Available at: http://www.rfc-editor.org/rfc/rfc2119.txt

[XHTMLMOD]
"*Modularization of XHTML™ 1.1*", W3C Recommendation, D. Austin *et al.*, *eds.*, 8 October 2008.
Available at: http://www.w3.org/TR/2008/REC-xhtml-modularization-20081008
The latest version is available at: http://www.w3.org/TR/xhtml-modularization

[XMLNAMES]
"*Namespaces in XML*", W3C Recommendation, T. Bray *et al.*, *eds.*, 14 January 1999.
Available at: http://www.w3.org/TR/1999/REC-xml-names-19990114
The latest version is available at: http://www.w3.org/TR/REC-xml-names

[XHTMLROLE]
"*XHTML Role Attribute Module*", W3C Working Draft, M. Birbeck *et al.*, *eds.*, 7 April 2008.
Available at: http://www.w3.org/TR/2008/WD-xhtml-role-20080407/
The latest version is available at: http://www.w3.org/TR/xhtml-role

C.2. Other References

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

[UAAG1]
"*User Agent Accessibility Guidelines 1.0*". Ian Jacobs *et al.*, 17 December 2002.
Available at: http://www.w3.org/TR/2002/REC-UAAG10-20021217
The latest version is available at: http://www.w3.org/TR/UAAG10
[XHTML1]
"*XHTML 1.0: The Extensible HyperText Markup Language (Second Edition)*", W3C Recommendation, S. Pemberton *et al.*, 26 January 2000, revised 1 August 2002.
Available at: http://www.w3.org/TR/2002/REC-xhtml1-20020801
[XHTML2]
"*XHTML™ 2.0*". J. Axelsson *et al.*, 26 July 2006.
Available at: http://www.w3.org/TR/2006/WD-xhtml2-20060726
The latest version is available at: http://www.w3.org/TR/xhtml2
[XHTMLVOCAB]
"XHTML Vocabulary", XHTML 2 Working Group.
Available at: http://www.w3.org/1999/xhtml/vocab
[XMLEVENTS]
"*XML Events*", W3C Recommendation, S. McCarron *et al.*, *eds.*, 14 October 2003.
Available at: http://www.w3.org/TR/2003/REC-xml-events-20031014

D. Acknowledgments

This section is informative.

This document was prepared by the W3C XHTML2 Working Group. The members at the time of publication of the Note were:

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