W3C WD-WAI-USERAGENT-TECH-19981030

Techniques for WAI User Agent Guidelines

W3C Working Draft *30-Oct-1998*

This version:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech
Latest version:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-TECH
Previous version:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981019/wai-useragent-tech
Current version of "WAI User Agent Guidelines":
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030
Latest "WAI User Agent Guidelines":
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT
Latest public version of "WAI User Agent Guidelines":
http://www.w3.org/TR/WD-WAI-USERAGENT
Editors:
Jon Gunderson <jongund@uiuc.edu>
Ian Jacobs <ij@w3.org>

Copyright © 1998 W3C (MIT, INRIA, Keio), All Rights Reserved.

Abstract

This document is a list of techniques that implement the guidelines described in "WAI User Agent Guidelines."

While "WAI User Agent Guidelines" strives to be a stable document (as a W3C Recommendation), the current document will undoubtedly evolve as technologies change and user agent manufacturers discover more effective techniques for designing accessible user agents.

This document is part of a series of accessibility documents published by the W3C Web Accessibility Initiative.

Status of this document

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

This document has been produced as part of the W3C WAI Activity and intends to improve user agent accessibility for all users. User agents include browsers (graphic, text, voice, etc.), multimedia players, and assistive technologies such as screen readers, screen

magnifiers, and voice input software.

The goals of the WAI-UA Working Group are discussed in the WAI UA charter. A list of the current Working Group members is available.

Available formats

This document is available in the following formats:

HTML:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.html
A plain text file:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.txt,
HTML as a gzip'ed tar file:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.tgz,
HTML as a zip file (this is a '.zip' file not an '.exe'):
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.zip,
A PostScript file:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.ps,
A PDF file:
http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.pdf.

In case of a discrepancy between the various formats of the specification, http://www.w3.org/WAI/UA/WD-WAI-USERAGENT-19981030/wai-useragent-tech.html is considered the definitive version.

Comments

Please send comments about this document to the public mailing list: w3c-wai-ua@w3.org.

Table of Contents

1 Introduction 4
1.1 Media dependencies 4
1.2 Native vs assistive support 4
1.3 Technique Priorities 5
2 The User Agent and the Operating System 5
3 The User Agent and the Document 6
3.1 Identification 6
3.2 Rendering alternative representations of content 7
3.3 Styles and formatting 8
4 Functionality offered by the User Agent to allow the User to work with the Document 9
4.1 Access to all functionality 9
4.2 Selection and focus 9
4.3 Navigation 10
4.4 Searching 10
4.5 Querying 11
4.6 Activation 11
5 The User Agent and the User 11
5.1 Visibility of accessibility features 11
5.2 Configuration 12
5.3 Notification of events and changes 14
6 The User Agent and Assistive Technologies 14
7 Support for accessibility features of certain languages 14
7.1 HTML 14
7.2 CSS 15
7.3 SMIL 15
7.4 MathML 15
8 Appendix: Accessibility features of some operating systems 16
8.1 Microsoft Windows 95 and Windows NT 4.0 16
8.2 Apple Macintosh 16
8.3 AccessX/The X Window System 16
9 Appendix: Techniques organized by HTML topics 17
10 Appendix: Techniques sorted by priority 17
11 Acknowledgments 24
12 References 25

1 Introduction

This document intends to serve as an accessibility checklist for user agent developers. For this reason, it organizes the techniques listed in the guidelines document according to different interfaces, primarily those between the user agent and some other entity: the operating system, the document, the user, and assistive technologies.

Note that each technique in this document is numbered according to its number in the guidelines document.

1.1 Media dependencies

[Ed. Must define, for each technique and when appropriate, what is considered the point of regard.]

Not all user agents are capable of rendering natively all media types (e.g., audio, video, etc.) or all formats within a given media type. Similarly, not all user agents are capable of rendering to all target devices (e.g., screen, paper, braille device, etc.) The following techniques only apply to those user agents enabled natively for a specific tasks. Thus, for example, if a user agent does not render video natively, it is not required to implement the techniques relating to video. Or, if a user agent does not render to a visual device, it is not required to satisfy the font or color techniques.

In this document, we refer to a user agent that is capable of performing a specific task natively as an *enabled user agent*.

An enabled user agent must implement a Priority 1 technique when it is enabled for that technique.

1.2 Native vs assistive support

While the purpose of this document is to provide developers with an accessibility checklist, the underlying motivation for implementing them remains user accessibility. Certain techniques are considered so fundamental to accessibility that all enabled user agents are required to implement them natively. The word "native" appears after such techniques.

Other techniques may be considered so specialized that not all user agents are required to implement them. Indeed, specialized user agents generally address these tasks more effectively due to developers' specialized knowledge or experience. When a user agent is not required to implement a technique natively, the technique is identified by the word "assistive". This means that:

- All user agents may implement the technique.
- If a user agent elects not to implement the technique, it must make available the information and interface necessary to allow assistive technologies to implement the technique. The exchange of information between one user agent and another must take place according to the techniques relating to the user agent and assistive technologies [p. 14] .

[Ed. Since this concept is new, this draft of the document does not distinguish between native and assistive techniques.

1.3 Technique Priorities

Each technique in this document is assigned a priority:

[Priority 1]
: This technique **must** be implemented by user agents as a native feature or through compatibility with assistive technology, otherwise one or more groups of users with disabilities will find it impossible to access information. Implementing this technique is a basic requirement for some individuals to be able to use the Web.

[Priority 2]
: This technique **should** be implemented by user agents as a native feature or through compatibility with assistive technology, otherwise one or more groups of users will find it difficult to access information. Implementing this technique will significantly improve access to the Web for some individuals.

[Priority 3]
: This technique **may** be implemented by user agents as a native feature or through compatibility with assistive technology, to make it easier for one or more groups of users to access information. Implementing this technique will improve access to the Web for some individuals.

2 The User Agent and the Operating System

[Technique: 6.2.1] Use operating system application programming interfaces (APIs) that support accessibility. [Priority 1]

The operating system application programming interfaces (APIs) that support accessibility are designed to provide a bridge between the standard user interface supported by the operating system and alternative user interfaces developed by third-party assistive technology vendors to provide access to persons with disabilities. Applications supporting these APIs are therefore generally more compatible with third-party assistive technology.

The WAI Working Group strongly recommends using and supporting APIs that improve accessibility and compatibility with 3rd party assistive technology. Third-party assistive technology can use the accessibility information provided by the APIs to provide an alternative user interface for various disabilities.

The following is an informative list of currently public APIs that support accessibility:

- Microsoft Active Accessibility in Windows 95/NT versions. Information on active accessibility can be found at the Microsoft WWW site on Active Accessibility.
- Sun Microsystems Java Accessibility API in Java Code. Information on Java Accessibility API can be found at Java Accessibility Utilities.

[Technique: 6.2.5] Make use of operating system accessibility flags and interfaces. [Priority 2]

Many major operating systems have built-in accessibility features for improving the usability of the standard operating system by persons with disabilities. When designing software that runs above an underlying operating system, developers should ensure that the application:

1. Makes use of operating system level features. See the appendix of accessibility features [p. 16] for some comment operating systems.
2. Inherits operating system settings related to accessibility. Pertinent settings include

font and color information as well as other pieces of information discussed in this document.

[Ed. We Should put a more exhaustive list here.]
[Technique: 3.1.1] Installation should be done, when possible, according to the conventions of the operating system. [Priority 1]
[Technique: 3.4.1] Ensure that installation documentation is accessible. [Priority 1]
[Technique: 3.2.2] Allow users to configure the user agent according to the conventions of the operating system. [Priority 1]

3 The User Agent and the Document

3.1 Identification

In order for the user agent to satisfy many of the techniques in this document (related to rendering, notification, navigation, etc.), it must be able to identify certain types of information in addition to recognizing object types such as images, video, sound, and elements of the document language.

Sources of alternative representations of information

User agents must be able to recognize sources of alternative representations of content.

In HTML

- For the IMG element: The "alt", "title", and "longdesc" attributes
- For the OBJECT element: The content of the element and the "title" attribute.
- For the APPLET element: The "alt" attribute.
- For the AREA element: The "alt" attribute.
- For the INPUT element: The "alt" attribute.
- For the ACRONYM and ABBR elements: The "title" attribute may be used for the acronym or abbreviation expansion.
- For the TABLE element, the "summary" attribute
- For frames, the NOFRAMES element and the "longdesc" attribute on FRAME and IFRAME.
- For scripts, the NOSCRIPT element.

In SMIL

[Ed. To be completed.]

[Ed. Other formats? What about captions?]

[Ed. To be completed.]

Sources of blinking text and animation

User agents must be able to recognize the following sources of blinking text and animations.

In HTML

- The BLINK element. **Note.** The BLINK element is not defined by a W3C specification.
- The MARQUIS element. **Note.** The BLINK element is not defined by a W3C specification.

In CSS

- The 'blink' value of the 'text-decoration' property.

In GIF animated images

[Ed. To be completed.]

[Ed. Other sources?]

[Ed. To be completed.]

Sources of event handlers

Certain elements of the document language may have associated event handlers that are triggered when certain events occur. User agents must be able to identify those elements with event handlers statically associated (i.e., associated in the document source, not in a script).

In HTML

All of the attributes beginning with the prefix "on": "onblur", "onchange", "onclick", "ondblclick", "onkeydown", "onkeypress", "onkeyup", "onload", "onmousedown", "onmousemove", "onmouseout", "onmouseover", "onmouseup", "onreset", "onselect", "onsubmit", and "onunload".

[Ed. Other sources?]

[Ed. To be completed.]

Information about the document

User agents must be able to compute the following information about a document:

- Has the document been fully loaded?
- What is the size in bytes of the document?
- How many active components are in the document?
- Where is the point of regard?

3.2 Rendering alternative representations of content

Alternative representations of information for images, video, applets

See the section on sources of alternative information [p. 6]

[Technique: 4.2.1] Allow the user to specify that alternative representations of content (e.g., the value of "alt" in HTML or SMIL, the resource designated by "longdesc", or the content of OBJECT in HTML 4.0) be rendered in place of primary content. [Priority 1]

[Technique: 4.2.2] Allow the user to specify that alternative representations of content (e.g., the value of "alt" in HTML or SMIL, the resource designated by "longdesc", or the content of OBJECT in HTML 4.0) be rendered at the same time as primary content. [Priority 1]

[Technique: 4.2.3] When no alternative text representation is available, indicate what type of object is present. [Priority 2]

[Technique: 4.2.4] When null alternative text has been defined, suppress the rendering of the alternative representation. [Priority 3]

Textual equivalents for audio and video

[Technique: 4.2.5] Allow the user specify that textual equivalents for audio be rendered at the same time as the audio. [Priority 1]

[Technique: 4.2.6] Allow the user specify that textual equivalents for video be rendered at the same time as the video. [Priority 1]

[Technique: 4.2.7] Ensure that textual equivalents rendered at the same time as video not interfere visually with the video. [Priority 1]

Audio equivalents for video

[Technique: 4.2.8] Allow the user specify that audio equivalents for video be rendered at the same time as the video. [Priority 1]

Alternative representations for scripts

[Technique: 4.2.13] Allow the user to specify that alternatives to a script be rendered (e.g., in HTML, the content of NOSCRIPT). [Priority 1]

In HTML, content of NOSCRIPT.

Alternative representations for frames

[Technique: 4.2.14] Allow the user to specify that alternatives to a frame be rendered (e.g., in HTML, the content of NOFRAMES). [Priority 1]

1. In HTML, content of NOFRAMES.
2. Otherwise, list of frames. Use "title" as name of frame, otherwise "name".

Alternative representations for tables

[Technique: 4.2.15] Allow the user to specify that alternatives to a table be rendered (e.g., the value of the "summary" attribute on TABLE in HTML 4.0). [Priority 1]

Generated content

Generated content specified through style sheets can help orient the user. For instance, if, through style sheets, the user can insert the word "Link" before every link, this text may be used by a speech synthesizer or Braille devices. See the section on the support for CSS2 accessibility features [p. 15] below.

3.3 Styles and formatting

Linearization

[Technique: 4.3.2] Allow users to specify that a table be formatted linearly. [Priority 1]

[Ed. Can be done by third-party assistive technologies?]

[Technique: 4.3.1] Allow users to specify that a page be formatted linearly. [Priority 1]

This helps current screen readers and people with cognitive disabilities. A linear page means:

- No frames
- Linearized tables
- Linearized columns
- Text only

[Ed. What else? Should alt content be used in this rendering, e.g., for images? Probably for some cases, such as images used as links.]

Links

[Technique: 5.1.9] Provide a mechanism to distinguish visited links from unvisited links. [Priority 3]

[Technique: 5.1.10] Allow the user to specify that images used in links must have borders. [Priority 3]

[Technique: 5.1.5] Provide a mechanism to indicate visually the presence of an "accesskey" attribute defined for a link. [Priority 2]

Forms

[Technique: 5.1.6] Provide a mechanism to indicate visually the presence of an "accesskey" attribute defined for a form control. [Priority 2]

Outline view

[Technique: 4.3.3] Allow users to view a document outline constructed from its structural elements (e.g., from header and list elements in HTML). [Priority 2]

Highlighting

[Technique: 5.2.1] Provide a mechanism for highlighting and identifying the current view. [Priority 1]

[Technique: 5.2.2] Provide a mechanism for highlighting and identifying the user selection. [Priority 1]

[Technique: 5.2.3] Provide a mechanism for highlighting and identifying the current focus. [Priority 1]

4 Functionality offered by the User Agent to allow the User to work with the Document

4.1 Access to all functionality

In general, this requirement may be met by providing keyboard access to all functionality.

[Technique: 3.6.1] All functionalities offered by the user agent (for the user agent itself or for access to the document) must available through means other than a pointing devices. [Priority 1]

4.2 Selection and focus

[Technique: 5.2.4] Allow the user to specify that a view’s focus should follow changes in the viewport. [Priority 1]

[Technique: 5.2.5] Keep track of the user’s point of regard in each view and put it within the viewport when the user returns to the view. [Priority 1]

[Technique: 5.5.3] Provide a mechanism for designating the *current table* of a document. [Priority 1]

[Technique: 5.5.4] Provide a mechanism for designating the *current cell* of a table. [Priority 1]

4.3 Navigation

Document navigation

[Technique: 5.5.1] Allow the user to navigate views (notably those with frame viewports). Navigating into a view makes it the current view. [Priority 1]

[Technique: 5.5.2] Allow the user to use the keyboard to navigate the document tree. [Priority 2]

Document object navigation

[Ed. Discuss tabbing navigation here. How tabbing navigation is specified in HTML: "tabindex".]

[Technique: 5.4.1] Allow the user to navigate sequentially among links. [Priority 1]

[Technique: 5.4.3] Allow the user to navigate sequentially among form controls. [Priority 1]

[Technique: 5.4.4] Allow the user to navigate sequentially among elements with associated event handlers. [Priority 1]

[Technique: 5.4.5] Allow the user to navigate sequentially among headers. [Priority 2]

[Technique: 5.4.6] Allow the user to navigate sequentially among block elements (e.g., paragraphs, lists and list items, etc.) [Priority 2]

[Technique: 5.4.2] Allow the user to navigate sequentially among elements with associated long descriptions. [Priority 1]

[Technique: 5.5.5] Allow the user to navigate among tables in a document. [Priority 1]

[Technique: 5.5.6] Allow the user to navigate among table cells of the current table (notably left/right within a row and up/down within a column). [Priority 1]

4.4 Searching

[Technique: 5.6.1] Allow the user to search for a link in the current document based on its link text. [Priority 1]

[Technique: 5.6.2] Allow the user to search for a link in the current document based on its attribute values. [Priority 1]

[Technique: 5.6.3] Allow the user to search for a link in the current document based on its position. [Priority 1]

[Technique: 5.6.4] Allow the user to search for a form control in the current document based on its text content. [Priority 1]

[Technique: 5.6.5] Allow the user to search for a form control in the current document based on its attribute values. [Priority 1]

[Technique: 5.6.7] Allow the user to search for an element in the current document based on its text content. [Priority 2]

[Technique: 5.6.6] Allow the user to search the long description text of any element with an identifiable long description (e.g., via the "longdesc" attribute). If a match occurs, the point of regard should be moved to the link to the long description in the main document. [Priority 2]

[Technique: 5.6.8] Allow the user to search for a table cell based on its contents, row/column coordinates, or header information. [Priority 3]

4.5 Querying

[Technique: 5.1.1] Provide the user with information about the number of links in a document. [Priority 1]

[Technique: 5.1.2] Provide the user with information about the number of form controls in a document. [Priority 1]

[Technique: 5.1.3] Provide the user with information about the number of tables in a document. [Priority 1]

[Technique: 5.1.4] When a document is loaded or when requested by the user, make available document summary information. [Priority 2]

[Ed. What does summary information include?]

[Technique: 5.2.6] Provide the user with information about how much of the document has been viewed (i.e., the location of the point of regard). [Priority 2]

[Technique: 5.2.7] Provide the user with information about which table cell is the current table cell. [Priority 2]

[Technique: 3.5.1] Provide the user with information about keyboard bindings (organized by key or by topic). [Priority 1]

4.6 Activation

[Technique: 3.6.2] Allow the user to follow links in a device-independent manner. [Priority 1]

[Technique: 3.6.3] Allow the user to activate form controls in a device-independent manner. [Priority 1]

[Technique: 3.6.4] Allow the user to trigger events in a device-independent manner. [Priority 1]

5 The User Agent and the User

5.1 Visibility of accessibility features

Universal design means that access to features that help accessibility should be integrated into normal menus. User agents should avoid regrouping access to accessibility features into specialized menus. However, documentation must include a section or chapter on accessibility where all available functionalities and mechanisms are described.

[Technique: 3.5.2] Display keyboard navigation shortcut commands in customizable menus. [Priority 2]

[Technique: 3.4.4] Provide a section on accessibility features in the online documentation. [Priority 2]

[Technique: 3.4.2] Ensure that the online documentation is in an accessible format. [Priority 1]

Documentation includes installation documentation. Users must have access to installation information, either in electronic form (diskette, over the Web), by fax, or by telephone.

[Technique: 3.4.3] Ensure that the online documentation interface is accessible. [Priority 1]

5.2 Configuration

Accessibility

[Technique: 3.2.1] Ensure that the Interface for configuring the software is accessible (e.g., no mouse-only configuration mechanisms). [Priority 1]

User control of style

To ensure accessibility, users must have **final control** over certain renderings.

For text:

[Technique: 4.1.1] Allow the user to override author styles and user agent defaults for font family. [Priority 1]

[Technique: 4.1.2] Allow the user to override author styles and user agent defaults for font size. [Priority 1]

[Technique: 4.1.3] Allow the user to override author styles and user agent defaults for foreground color. [Priority 1]

[Technique: 4.1.4] Allow the user to override author styles and user agent defaults for background color. [Priority 1]

[Technique: 4.1.5] Allow the user to override author styles and user agent defaults for selection foreground and background color. [Priority 1]

For the selection and focus:

[Technique: 4.1.5] Allow the user to override author styles and user agent defaults for selection foreground and background color. [Priority 1]

[Technique: 4.1.6] Allow the user to override author styles and user agent defaults for focus foreground and background color. [Priority 1]

[Ed. These may be rendered in a variety of ways. How do we specify rendering?

For images, applets, and animations:

[Technique: 4.1.7] Allow the user to override author styles and user agent defaults for background images. [Priority 1]

[Technique: 4.1.8] Allow the user to override author styles and user agent defaults for animations. [Priority 1]

For user agents rendering video:

[Technique: 4.1.9] Allow the user to override author styles and user agent defaults for video frame rates. [Priority 1]

For user agents rendering audio:

[Technique: 4.1.10] Allow the user to override author styles and user agent defaults for audio playback rate. [Priority 1]

[Technique: 4.1.11] Allow the user to override author styles and user agent defaults for audio volume. [Priority 1]

[Technique: 4.1.12] Allow the user to select a specific audio track when several are available. [Priority 1]

For user agents rendering speech:

[Technique: 4.1.14] Allow the user to override author styles and user agent defaults for speech volume. [Priority 1]

[Technique: 4.1.13] Allow the user to override author styles and user agent defaults for speech playback rate. [Priority 1]

For user agents rendering textual equivalents of video and sound:

[Technique: 4.2.9] For textual equivalents, allow the user to override author styles and user agent defaults for font family. [Priority 1]

[Technique: 4.2.10] For textual equivalents, allow the user to override author styles and user agent defaults for font size. [Priority 1]

[Technique: 4.2.11] For textual equivalents, allow the user to override author styles and user agent defaults for foreground color. [Priority 1]

[Technique: 4.2.12] For textual equivalents, allow the user to override author styles and user agent defaults for background color. [Priority 1]

User profiles

[Ed. Define profile here.]

[Technique: 3.2.4] Allow users to configure the user agent in profiles that may be shared (by other users or software). Furthermore, for convenience, users should be able to name groups of settings and be able to apply them all at once (e.g., by selecting a group by name from a menu). [Priority 2]

The user should be able to easily transfer their profile between installations of the same user agent.

[Technique: 3.2.5] Allow the user, through a keyboard command, to switch between user agent default values and the user profile. [Priority 3]

This is important for people when they are receiving help from an able-bodied user who may not recognize the information being displayed using the users profile.

[Technique: 3.2.6] Allow the user to name a group of settings and to apply them all at once (e.g., by selecting a group by name from a menu). [Priority 3]

[Technique: 3.2.7] Furnish predefined accessibility profiles for common disabilities. [Priority 3]

Feature control

[Technique: 3.3.1] Allow the user to turn on and off image rendering. [Priority 1]

[Technique: 3.3.2] Allow the user to turn on and off video rendering. [Priority 1]

[Technique: 3.3.3] Allow the user to turn on and off sound rendering. [Priority 1]

[Technique: 3.3.4] Allow the user to turn on and off support for equivalent textual representation. [Priority 1]

[Technique: 3.3.5] Allow the user to turn on and off blinking text for all cases when the user agent knows that text is blinking. [Priority 1]

[Technique: 3.3.6] Allow the user to turn on and off blinking images and animations. [Priority 1]

[Technique: 3.3.7] Allow the user to turn on and off support for scripts (including event handlers). [Priority 1]

[Technique: 3.3.8] Allow the user to turn on and off support for style sheets. [Priority 1]

[Technique: 3.3.9] Allow the user to turn on and off support for frames. [Priority 1]

[Technique: 5.2.8] Allow the user to turn on and off automatic page forwarding. [Ed. How is this identified?] [Priority 3]

Keyboard configuration

[Technique: 3.2.3] Allow users to configure keyboard access (key combinations, distance between active keys, etc.). [Priority 1]

[Ed. Discuss keyboard access here. How access is specified in HTML: "accesskey".]

[Ed. New section. Users must be allowed to control keyboard configuration based on specific needs. For poor motor control, keys far apart. For poor mobility, keys close together. General principle: fewest keystrokes, short distance to move.

5.3 Notification of events and changes

[Technique: 5.1.7] Provide the user with visual feedback about document loading information. Such information includes whether loading has stalled, whether enough of the page has loaded to begin navigating, whether following a link involves a fee, etc. [Priority 3]

[Technique: 5.1.8] Provide the user with audio feedback about document loading information. Such information includes whether loading has stalled, whether enough of the page has loaded to begin navigating, whether following a link involves a fee, etc. [Priority 3]

[Technique: 5.3.1] Provide information when a script is executed. [Priority 2]

[Technique: 5.3.2] Provide information about document changes resulting from the execution of a script. [Priority 3]

[Technique: 5.3.3] Allow users to be prompted before spawning a new window. [Priority 2]

6 The User Agent and Assistive Technologies

[Technique: 6.2.2] For information that can be exchanged through an interface defined by a W3C specification, user agents should implement that specification. [Priority 1]

Use DOM Level 1 [DOM1] [p. 25] for access to HTML and XML document information. However, as the DOM specification indicates:

> The DOM Level 1 does not include mechanisms to access and modify style specified through CSS 1. Furthermore, it does not define an event model for HTML documents. This functionality is planned to be specified in a future Level of this specification.

It is important for user agents to provide access to style and scripting information in a document, so the following techniques should be used to achieve this.

[Technique: 6.2.3] Otherwise, if an accessible application programming interface (API) is available for the exchange, user agents should implement that interface agents should implement that specification. [Priority 2]

[Technique: 6.2.4] Otherwise, standard interfaces defined for the operating system should be used. [Priority 2]

Use standard rather than custom controls when designing user agents. Third-party assistive technology developers are more likely able to access standard controls than custom controls. If you must use custom controls, review them for accessibility and compatibility with third-party assistive technology.

7 Support for accessibility features of certain languages

7.1 HTML

[Technique: 6.1.1] Support accessibility features of HTML. [Priority 1]

- Support the "longdesc"] attribute defined for IMG elements ([HTML 4.0] [p. 25] , section 13.2). This attribute may be used to attach additional descriptive information to images. [Priority 1]

- Support the CAPTION element ([HTML40] [p. 25] , section 11.2.2) for rich table captions. [Priority 2]
- Support the ACRONYM and ABBR elements ([HTML40] [p. 25] , section 9.2.1) for acronyms and abbreviations. [Priority 2]
- Support the "summary" attribute for TABLE ([HTML40] [p. 25] , section 11.2.1) for table summary information. [Priority 2]
- Support the NOSCRIPT element ([HTML40] [p. 25] , sections 18.3.1 and 16.4.1) for accessible alternatives to scripts. [Priority 2]
- Support the NOFRAMES element ([HTML40] [p. 25] , sections 18.3.1 and 16.4.1) for accessible alternatives to frames. [Priority 2]
- Support the "lang" attribute ([HTML40] [p. 25] , section 8.1). [Priority 3]
- Support the "tabindex" attribute ([HTML40] [p. 25] , section 17.11.1) for assigning the order of keyboard navigation within a document. [Priority 3]
- Support the "accesskey" attribute ([HTML40] [p. 25] , section 17.11.2) for assigning keyboard commands to active components such as links, and form controls. [Priority 3]

7.2 CSS

[Technique: 6.1.2] Support accessibility features of CSS. [Priority 1]

The following techniques apply to user agents that implement Cascading Style Sheets (see CSS, level 1 [p. 25] and CSS, level 2 [p. 25]). Cascading Style Sheets may be part of a source document or linked externally.

Stand-alone style sheets are useful for implementing *user profiles* in public access computer environments where several people use the same computer. User profiles allow for convenient customization and may be shared by a group.

- Completely implement Cascading Style Sheets, level 1 [Priority 1]
- Allow the user to turn off author styles represented by author style sheets. [Priority 1]
- Allow the user to adjust default values represented by user agent style sheets. [Priority 1]
- Support the :before and :after pseudo-elements as defined in CSS2 ([CSS2] [p. 25] , section 12.1). [Priority 1]
- Support the ’outline’ property as defined in CSS2 ([CSS2] [p. 25] , section 18.4). Outlines may be used to customize the focus display. [Priority 1]
- Allow the user to specify user styles through style sheets (see [CSS2] [p. 25] , section 6.4). [Priority 2]
- Implement the !important rule as defined in CSS2 ([CSS2] [p. 25] , section 6.4.2). These rules offer a way for users to override author styles and user agent defaults [Priority 2]
- Support aural cascading style sheets (see [CSS2] [p. 25] , chapter 19) for the auditory presentation of documents. [Priority 2]

7.3 SMIL

[Technique: 6.1.3] Support accessibility features of SMIL. [Priority 1]

8 Appendix: Accessibility features of some operating systems

8.1 Microsoft Windows 95 and Windows NT 4.0

The accessibility options can be adjusted from the control panel.

- Sticky Keys: Allows user to temporarily hold down the shift, alt, and control keys to allow one finger typing.
- Filter Keys: Allows user to change keyboard timing to accept a keypress and repeat rate.
- Toggle Keys: Provides auditory feedback lock keys are pressed (Caps Lock, Scroll Lock, Num Lock).
- Mouse Keys: Allows the user to emulate pointer movement and button press operations using the numeric keypad.
- High Contrast Display Mode: Changes the colors used on the display to a high-contrast color combination.
- Sound Sentry: Monitors the system sounds and flashes the menu bar when sound output is detected. Used by the hearing impaired to know when sound is being generated by the system.
- Show Sounds: A flag that is available to applications (including user agents) to notify them that audio information should be presented in a visual form for the hearing impaired. This requires the application program to provide the alternative format of information. This can include closed captioning information on animations and video clips.

8.2 Apple Macintosh

The accessibility options can be adjusted from the control panels through the Easy Access option and the Closeview option.

- Sticky Keys: Allows user to temporarily hold down the shift, open apple, option (alt) and control keys to allow one finger typing.
- Slow Keys: Allows user to change keyboard timing to accept a keypress and repeat rate.
- Mouse Keys: Allows the user to emulate pointer movement and button press operations using the numeric keypad.
- Closeview: Closeview is a screen enlargement and enhancement program used by persons with low vision to magnify the information on the visual display and change the colors used by the system.

8.3 AccessX/The X Window System

Disability access server features, known as AccessX, provide basic workstation accessibility, typically used by people with mobility impairments. AccessX became a supported part of the X Windows server in version X11/R6. The built-in server level access features include:

- StickyKeys: Provides locking or latching of modifier keys (Shift, Control, etc.) so that they can be used without simultaneously pressing the keys being modified. This allows single finger operation of multiple key combinations.
- RepeatKeys: Delays the onset of key repeat, allowing users with limited coordination time to release keys before multiple characters are sent.
- SlowKeys: Requires a key to be held down for a set period before keypress acceptance. This allows users with limited coordination to accidentally press keys without sending keypress events.
- MouseKeys: An alternative to the mouse that provides keyboard-based explicit control of cursor movement and all mouse button press/release events
- ToggleKeys: Indicates locking key state with a tone when pressed, (e.g., Caps Lock, Num Lock, Scroll Lock..).
- BounceKeys: Requires a delay between keystrokes before accepting the next keypress so users with tremors can prevent the system from accepting inadvertent keypresses.

9 Appendix: Techniques organized by HTML topics

Document structure and metadata
Language information
Text markup
Lists
Tables
Links
Images and image maps
Applets and other objects
Audio and video
Style and style sheets
Frames
Forms
Scripts

10 Appendix: Techniques sorted by priority

Priority 1

Technique	Guideline	Text
3.1.1 [p. 6]	3.1	Installation should be done, when possible, according to the conventions of the operating system.
3.2.1 [p. 12]	3.2	Ensure that the Interface for configuring the software is accessible (e.g., no mouse-only configuration mechanisms).
3.2.2 [p. 6]	3.2	Allow users to configure the user agent according to the conventions of the operating system.
3.2.3 [p. 13]	3.2	Allow users to configure keyboard access (key combinations, distance between active keys, etc.).

3.3.1 [p. 13]	3.3	Allow the user to turn on and off image rendering.
3.3.2 [p. 13]	3.3	Allow the user to turn on and off video rendering.
3.3.3 [p. 13]	3.3	Allow the user to turn on and off sound rendering.
3.3.4 [p. 13]	3.3	Allow the user to turn on and off support for equivalent textual representation.
3.3.5 [p. 13]	3.3	Allow the user to turn on and off blinking text for all cases when the user agent knows that text is blinking.
3.3.6 [p. 13]	3.3	Allow the user to turn on and off blinking images and animations.
3.3.7 [p. 13]	3.3	Allow the user to turn on and off support for scripts (including event handlers).
3.3.8 [p. 13]	3.3	Allow the user to turn on and off support for style sheets.
3.3.9 [p. 13]	3.3	Allow the user to turn on and off support for frames.
3.4.1 [p. 6]	3.4	Ensure that installation documentation is accessible.
3.4.2 [p. 11]	3.4	Ensure that the online documentation is in an accessible format.
3.4.3 [p. 11]	3.4	Ensure that the online documentation interface is accessible.
3.5.1 [p. 11]	3.5	Provide the user with information about keyboard bindings (organized by key or by topic).
3.6.1 [p. 9]	3.6	All functionalities offered by the user agent (for the user agent itself or for access to the document) must available through means other than a pointing devices.
3.6.2 [p. 11]	3.6	Allow the user to follow links in a device-independent manner.
3.6.3 [p. 11]	3.6	Allow the user to activate form controls in a device-independent manner.
3.6.4 [p. 11]	3.6	Allow the user to trigger events in a device-independent manner.
4.1.1 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for font family.
4.1.10 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for audio playback rate.

4.1.11 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for audio volume.
4.1.12 [p. 12]	4.1	Allow the user to select a specific audio track when several are available.
4.1.13 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for speech playback rate.
4.1.14 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for speech volume.
4.1.2 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for font size.
4.1.3 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for foreground color.
4.1.4 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for background color.
4.1.5 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for selection foreground and background color.
4.1.6 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for focus foreground and background color.
4.1.7 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for background images.
4.1.8 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for animations.
4.1.9 [p. 12]	4.1	Allow the user to override author styles and user agent defaults for video frame rates.
4.2.1 [p. 7]	4.2	Allow the user to specify that alternative representations of content (e.g., the value of "alt" in HTML or SMIL, the resource designated by "longdesc", or the content of OBJECT in HTML 4.0) be rendered in place of primary content.
4.2.10 [p. 13]	4.2	For textual equivalents, allow the user to override author styles and user agent defaults for font size.
4.2.11 [p. 13]	4.2	For textual equivalents, allow the user to override author styles and user agent defaults for foreground color.
4.2.12 [p. 13]	4.2	For textual equivalents, allow the user to override author styles and user agent defaults for background color.
4.2.13 [p. 8]	4.2	Allow the user to specify that alternatives to a script be rendered (e.g., in HTML, the content of NOSCRIPT).
4.2.14 [p. 8]	4.2	Allow the user to specify that alternatives to a frame be rendered (e.g., in HTML, the content of NOFRAMES).

4.2.15 [p. 8]	4.2	Allow the user to specify that alternatives to a table be rendered (e.g., the value of the "summary" attribute on TABLE in HTML 4.0).
4.2.2 [p. 7]	4.2	Allow the user to specify that alternative representations of content (e.g., the value of "alt" in HTML or SMIL, the resource designated by "longdesc", or the content of OBJECT in HTML 4.0) be rendered at the same time as primary content.
4.2.5 [p. 8]	4.2	Allow the user specify that textual equivalents for audio be rendered at the same time as the audio.
4.2.6 [p. 8]	4.2	Allow the user specify that textual equivalents for video be rendered at the same time as the video.
4.2.7 [p. 8]	4.2	Ensure that textual equivalents rendered at the same time as video not interfere visually with the video.
4.2.8 [p. 8]	4.2	Allow the user specify that audio equivalents for video be rendered at the same time as the video.
4.2.9 [p. 12]	4.2	For textual equivalents, allow the user to override author styles and user agent defaults for font family.
4.3.1 [p. 8]	4.3	Allow users to specify that a page be formatted linearly.
4.3.2 [p. 8]	4.3	Allow users to specify that a table be formatted linearly.
5.1.1 [p. 11]	5.1	Provide the user with information about the number of links in a document.
5.1.2 [p. 11]	5.1	Provide the user with information about the number of form controls in a document.
5.1.3 [p. 11]	5.1	Provide the user with information about the number of tables in a document.
5.2.1 [p. 9]	5.2	Provide a mechanism for highlighting and identifying the current view.
5.2.2 [p. 9]	5.2	Provide a mechanism for highlighting and identifying the user selection.
5.2.3 [p. 9]	5.2	Provide a mechanism for highlighting and identifying the current focus.
5.2.4 [p. 9]	5.2	Allow the user to specify that a view’s focus should follow changes in the viewport.
5.2.5 [p. 9]	5.2	Keep track of the user’s point of regard in each view and put it within the viewport when the user returns to the view.
5.4.1 [p. 10]	5.4	Allow the user to navigate sequentially among links.

5.4.2 [p. 10]	5.4	Allow the user to navigate sequentially among elements with associated long descriptions.
5.4.3 [p. 10]	5.4	Allow the user to navigate sequentially among form controls.
5.4.4 [p. 10]	5.4	Allow the user to navigate sequentially among elements with associated event handlers.
5.5.1 [p. 10]	5.5	Allow the user to navigate views (notably those with frame viewports). Navigating into a view makes it the current view.
5.5.3 [p. 9]	5.5	Provide a mechanism for designating the *current table* of a document.
5.5.4 [p. 10]	5.5	Provide a mechanism for designating the *current cell* of a table.
5.5.5 [p. 10]	5.5	Allow the user to navigate among tables in a document.
5.5.6 [p. 10]	5.5	Allow the user to navigate among table cells of the current table (notably left/right within a row and up/down within a column).
5.6.1 [p. 10]	5.6	Allow the user to search for a link in the current document based on its link text.
5.6.2 [p. 10]	5.6	Allow the user to search for a link in the current document based on its attribute values.
5.6.3 [p. 10]	5.6	Allow the user to search for a link in the current document based on its position.
5.6.4 [p. 10]	5.6	Allow the user to search for a form control in the current document based on its text content.
5.6.5 [p. 10]	5.6	Allow the user to search for a form control in the current document based on its attribute values.
6.1.1 [p. 14]	6.1	Support accessibility features of HTML.
6.1.2 [p. 15]	6.1	Support accessibility features of CSS.
6.1.3 [p. 15]	6.1	Support accessibility features of SMIL.
6.2.1 [p. 5]	6.2	Use operating system application programming interfaces (APIs) that support accessibility.
6.2.2 [p. 14]	6.2	For information that can be exchanged through an interface defined by a W3C specification, user agents should implement that specification.

Priority 2

Technique	Guideline	Text
3.2.4 [p. 13]	3.2	Allow users to configure the user agent in profiles that may be shared (by other users or software). Furthermore, for convenience, users should be able to name groups of settings and be able to apply them all at once (e.g., by selecting a group by name from a menu).
3.4.4 [p. 11]	3.4	Provide a section on accessibility features in the online documentation.
3.5.2 [p. 11]	3.5	Display keyboard navigation shortcut commands in customizable menus.
4.2.3 [p. 7]	4.2	When no alternative text representation is available, indicate what type of object is present.
4.3.3 [p. 9]	4.3	Allow users to view a document outline constructed from its structural elements (e.g., from header and list elements in HTML).
5.1.4 [p. 11]	5.1	When a document is loaded or when requested by the user, make available document summary information.
5.1.5 [p. 9]	5.1	Provide a mechanism to indicate visually the presence of an "accesskey" attribute defined for a link.
5.1.6 [p. 9]	5.1	Provide a mechanism to indicate visually the presence of an "accesskey" attribute defined for a form control.
5.2.6 [p. 11]	5.2	Provide the user with information about how much of the document has been viewed (i.e., the location of the point of regard).
5.2.7 [p. 11]	5.2	Provide the user with information about which table cell is the current table cell.
5.3.1 [p. 14]	5.3	Provide information when a script is executed.
5.3.3 [p. 14]	5.3	Allow users to be prompted before spawning a new window.
5.4.5 [p. 10]	5.4	Allow the user to navigate sequentially among headers.
5.4.6 [p. 10]	5.4	Allow the user to navigate sequentially among block elements (e.g., paragraphs, lists and list items, etc.)
5.5.2 [p. 10]	5.5	Allow the user to use the keyboard to navigate the document tree.

5.6.6 [p. 10]	5.6	Allow the user to search the long description text of any element with an identifiable long description (e.g., via the "longdesc" attribute). If a match occurs, the point of regard should be moved to the link to the long description in the main document.
5.6.7 [p. 10]	5.6	Allow the user to search for an element in the current document based on its text content.
6.2.3 [p. 14]	6.2	Otherwise, if an accessible application programming interface (API) is available for the exchange, user agents should implement that interface agents should implement that specification.
6.2.4 [p. 14]	6.2	Otherwise, standard interfaces defined for the operating system should be used.
6.2.5 [p. 5]	6.2	Make use of operating system accessibility flags and interfaces.

Priority 3

Technique	Guideline	Text
3.2.5 [p. 13]	3.2	Allow the user, through a keyboard command, to switch between user agent default values and the user profile.
3.2.6 [p. 13]	3.2	Allow the user to name a group of settings and to apply them all at once (e.g., by selecting a group by name from a menu).
3.2.7 [p. 13]	3.2	Furnish predefined accessibility profiles for common disabilities.
4.2.4 [p. 7]	4.2	When null alternative text has been defined, suppress the rendering of the alternative representation.
5.1.10 [p. 9]	5.1	Allow the user to specify that images used in links must have borders.
5.1.7 [p. 14]	5.1	Provide the user with visual feedback about document loading information. Such information includes whether loading has stalled, whether enough of the page has loaded to begin navigating, whether following a link involves a fee, etc.
5.1.8 [p. 14]	5.1	Provide the user with audio feedback about document loading information. Such information includes whether loading has stalled, whether enough of the page has loaded to begin navigating, whether following a link involves a fee, etc.
5.1.9 [p. 9]	5.1	Provide a mechanism to distinguish visited links from unvisited links.
5.2.8 [p. 13]	5.2	Allow the user to turn on and off automatic page forwarding. [Ed. How is this identified?]
5.3.2 [p. 14]	5.3	Provide information about document changes resulting from the execution of a script.
5.6.8 [p. 10]	5.6	Allow the user to search for a table cell based on its contents, row/column coordinates, or header information.

11 Acknowledgments

Many thanks to the following people who have contributed through review and comment: Paul Adelson, James Allen, Denis Anson, Kitch Barnicle, Harvey Bingham, Judy Brewer, Kevin Carey, Wendy Chisholm, Chetz Colwell, Daniel Dardailler, Neal Ewers, Geoff Freed, Larry Goldberg, Markku Hakkinen, Chris Hasser, Kathy Hewitt, Phill Jenkins, Leonard Kasday, George Kerscher, Marja-Riitta Koivunen, Josh Krieger, Greg Lowney, Scott Luebking, William Loughborough, Napoleon Maou, Charles McCathieNevile, Masafumi Nakane, Charles Opperman, Mike Paciello, David Pawson, Helen Petrie, David Poehlman, Michael Pieper, Jan Richards, Greg Rosmaita, Liam Quinn, T.V. Raman, Robert Savellis, Constantine Stephanidis, Jim Thatcher, Jutta Treviranus, Claus Thøgersen, Steve Tyler, Gregg Vanderheiden, Jaap van Lelieveld, Jon S. von Tetzchner, Ben Weiss, Evan Wies, Chris Wilson, Henk Wittingen, and Tom Wlodkowski.

If you have contributed to the UA guidelines and your name does not appear please contact the editors to add your name to the list.

12 References

[CSS1]
"CSS, level 1 Recommendation", B. Bos, H. Wium Lie, eds. The CSS1 Recommendation is available at:
http://www.w3.org/TR/REC-CSS1.

[CSS2]
"CSS, level 2 Recommendation", B. Bos, H. Wium Lie, C. Lilley, and I. Jacobs, eds. The CSS2 Recommendation is available at:
http://www.w3.org/TR/REC-CSS2/.

[CSS2-WAI]
"WAI Resources: CSS2 Accessibility Improvements", I. Jacobs and J. Brewer, eds. This document, which describes accessibility features in CSS2, is available at:
http://www.w3.org/WAI/References/CSS2-access.

[DOM1]
"Document Object Model (DOM) Level 1 Specification", V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, A. Le Hors, G. Nicol, J. Robie, R. Sutor, C. Wilson, and L. Wood, eds. The DOM Level 1 Recommendation is available at:
http://www.w3.org/TR/REC-DOM-Level-1/.

[HTML40]
"HTML 4.0 Recommendation", D. Raggett, A. Le Hors, and I. Jacobs, eds. The HTML 4.0 Recommendation is available at:
http://www.w3.org/TR/REC-html 40/.

[HTML4-WAI]
"WAI Resources: HTML 4.0 Accessibility Improvements", I. Jacobs, J. Brewer, and D. Dardailler, eds. This document, which describes accessibility features in HTML 4.0, is available at:
http://www.w3.org/WAI/References/HTML4-access.

[SMIL]
"Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", P. Hoschka, editor. The SMIL 1.0 Recommendation is available at:
http://www.w3.org/TR/REC-smil/

[WAI-PAGEAUTH]
"WAI Accessibility Guidelines: Page Authoring", G. Vanderheiden, W. Chisholm, and I. Jacobs, eds. These guidelines for designing accessible documents are available at:
http://www.w3.org/TR/WD-WAI-PAGEAUTH.

[WAI-GL-TECHNIQUES]
Techniques for "WAI Accessibility Guidelines: Page Authoring", G. Vanderheiden, W. Chisholm, and I. Jacobs, eds. This evolving document is available at:
http://www.w3.org/WAI/GL/wai-gl-techniques.