W3C WD-WAI-USERAGENT-19981112

WAI User Agent Guidelines

W3C Working Draft *12-Nov-1998*

This version:
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112
Latest version:
http://www.w3.org/TR/WD-WAI-USERAGENT
Previous version:
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19980703
Related documents:
Techniques for WAI User Agent Guidelines

Editors:
Jon Gunderson <jongund@uiuc.edu>
Ian Jacobs <ij@w3.org>

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

Abstract

This document provides guidelines to user agent manufacturers for making their products more accessible to people with disabilities and for increasing usability 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.

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 is intended as a draft of a Proposed Recommendation for how to improve user agent accessibility. 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/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.html
A plain text file:
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.txt,
HTML as a gzip'ed tar file:
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.tgz,
HTML as a zip file (this is a '.zip' file not an '.exe'):
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.zip,
A PostScript file:
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.ps,
A PDF file:
http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.pdf.

In case of a discrepancy between the various formats of the specification, http://www.w3.org/TR/1998/WD-WAI-USERAGENT-19981112/wai-useragent.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 General principles of accessible design 4
1.2 Priorities 5
2 Terms and definitions 5
2.1 Documents, Elements, and Attributes 5
2.2 Properties, Values, and Defaults 6
2.3 Views, Point of Regard, Selection, Focus, and Events 6
2.4 Alternative representations of content 8
2.5 Dependent vs. Independent user agents 8
3 The user agent must be accessible to all users 8
3.1 Ensure that the user interface is accessible 8
3.2 Ensure that user agent accessibility features are configurable . . .9
3.3 Ensure that users can disable features that might interfere with accessibility 9
3.4 Provide documentation about accessibility features and ensure the documentation is accessible 10
3.5 Provide summary information about keyboard access 10
3.6 Allow users to interact with the and document in a device-independent manner 11
4 The user agent must render information accessibly 11
4.1 Allow the user to control document styles 11
4.2 Provide access to alternative representations of content and control of its rendering 13
4.3 Allow the user to choose formatting solutions 14
5 The user agent must facilitate orientation, navigation, and searching on a page and between pages 15
5.1 Provide information about the content and structure of a document and the user interface 15
5.2 Provide information about document view and point of regard . . 16
5.3 Provide information about events that occur 17
5.4 Allow sequential keyboard navigation to structures within a document 17
5.5 Allow keyboard navigation of the document and views of the document 18
5.6 Allow the user to search for information on the page 18
6 The user agent must make information available to other technologies . . 19
6.1 Support language accessibility features 19
6.2 Use and provide accessible interfaces to other technologies . . . 19
7 Acknowledgments 20
8 References 20

1 Introduction

The guidelines, background information, and rationale presented in this document are meant to guide user agent developers and vendors as they design their products. Each guideline presents a principle that, when respected, will make user agents more accessible to users with disabilities and to users in general. The techniques listed in this document (and described in detail in the accompanying techniques document) describe specific ways to follow each of the guidelines. The techniques document may serve as an accessibility feature checklist for developers.

1.1 General principles of accessible design

This document organizes the guidelines according to the following general principles of accessible user agent design:

The user interface must be accessible to all users.

The user agent's interface agent must be made as intuitive and efficient to use as possible for a broad group of users having varied experience, knowledge, language skills and concentration levels. Software should be tested for usability by real users, including users with disabilities working in realistic settings. Where the interface is not accessible, inaccessible functionality must be made available through accessible means. User interaction with both the user agent and the document must be made accessible. Before and after user interaction, the user agent should provide useful feedback to the user.

All users must be able to access all functionalities of a user agent. Users access these functionalities through *controls* (menus, buttons, keyboard shortcuts, etc.). User interfaces must have redundant controls for the same functionality (e.g., menu, mouse, and keyboard equivalents) and must be operable without a pointing device. Redundancy includes offering visual as well as aural representations of a control. User agents must also allow users to customize and configure the controls to meet their needs.

The user agent must render information accessibly.

The user agent must be able to render information in accessible ways and allow users to control the style (colors, fonts, speech rate, speech volume, etc.) and format (e.g., linearization of the page, linearization of tables) of a document. Users must also be able to access both primary and alternative representations of content. [p. 8]

The user agent must facilitate orientation, navigation, and searching on a page and between pages.

The user agent must allow navigation at all times and through a variety of input devices (notably the keyboard). The user agent should provide Web page orientation information (the number of links on a page, the number of form controls, the number of images, etc.) so the user can quickly grasp content and context.

The user agent must make information available to assistive technology software.

Not all user agents satisfy the above principles natively, so they must make information available to complementary software (e.g., screen readers) for additional services. Also, specialized complementary software may offer superior service for some requirements, and so even those user agents that implement a feature natively must make information available to other software. Information must be made available through interoperable and standard means whenever possible.

Developers may believe that implementing accessibility features in their products is difficult or of limited use. In reality, considering accessibility during the design phase of a product leads to more flexible, manageable, and extensible software.

1.2 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.

Note. *In future versions of this document, guidelines will be assigned priorities as well to reflect their significance in ensuring or promoting accessibility. The User Agent Working Group welcomes comments from readers about guideline priorities.*

2 Terms and definitions

This section defines terms used in this document.

2.1 Documents, Elements, and Attributes

A document may be seen as a hierarchy of *elements.* Elements are defined by a language specification (e.g., HTML 4.0 or an XML application). Each element may have content, which generally contributes to the document's content. Elements may also have *attributes* that take values. Some attributes are integral to document accessibility (e.g., the "alt", "title", and "longdesc" attributes in HTML).

An element's *rendered content* is that which a user agent renders for the element. Often, this is what lies between the element's start and end tags. However, some elements cause external data to be rendered (e.g., the IMG element in HTML). At times, a browser may render the value of an attribute (e.g., "alt", "title" in HTML) in place of the element's content. Rendering is not limited to only visual displays, but can also include audio (speech and sound) and tactile displays (Braille and haptic displays).

2.2 Properties, Values, and Defaults

A user agent renders a document by applying formatting algorithms and style information to the document's elements. Formatting depends on a number of factors, including where the document is rendered: on screen, paper, through speakers, a braille device, a mobile device, etc. Style information (e.g., fonts, colors, voice inflection, etc.) may come from the elements themselves (e.g., certain style attributes in HTML), from style sheets, or from user agent settings. For the purposes of these guidelines, each formatting or style option is governed by a *property* and each property may take one value from a set of legal values.

The value given to a property by a user agent when it is started up is called the property's *default value.* User agents may allow users to change default values through a variety of mechanisms (e.g., the user interface, style sheets, initialization files, etc.).

Once the user agent is running, the value of a property for a given document or part of a document may be changed from the default value. The value of the property at a given moment is called its *current value.* Note that changes in the current value of a property do not change its default value.

Current values may come from documents, style sheets, scripts, or the user interface. Values that come from documents, their associated style sheets, or via a server are called *author styles.* Values that come from user interface settings, user style sheets, or other user interactions are called *user styles.*

2.3 Views, Point of Regard, Selection, Focus, and Events

Views and viewports

User agents may handle different types of source information: documents, sound objects, video objects, etc. The user perceives the information through a *viewport*, which may be a window, frame, a piece of paper, a speaker, a virtual magnifying glass, etc. A viewport may contain another viewport (e.g., nested frames, plug-ins, etc.).

User agents may render the same source information in a variety of ways; each rendering is called a *view.* For instance, a user agent may allow users to view a document in one window and a generated list of headers for the document in another.

The view is *how* source information is rendered and the viewport is *where* it is rendered.

Generally, viewports give users access to all rendered information, though not always at once. For example, a video player shows a certain number of frames per second, but allows the user to rewind and fast forward. A visual browser viewport generally features scrollbars or some other paging mechanism that allows the user to bring the rendered content into view.

Point of regard, Selection, and Focus

Some of the guidelines below involve tracking the user's *point of regard* in the view. The point of regard describes where the user is expected to interact with the rendered information. As the guidelines below state, user agents should warn users when the viewport is moved away from the point of regard unexpectedly, as this can disorient users.

Identifying the point of regard depends on the viewport. For paper, for example, it is difficult to identify the point of regard any more precisely than on the entire page. For sound and audio players (and linear devices in general), the point of regard designates the

information currently being rendered.

When the viewport gives the user access to information in more than one dimension (e.g., on the screen), there are several mechanisms generally offered by user agents that may be used to identify the point of regard:

The *insertion point.*

The insertion point is the location where document editing takes place. The insertion point may be set by the user (e.g., by a pointing device or the keyboard editing keys) or through an application programming interface (API). A view may have only one insertion point. When several views co-exist, each may have an insertion point, but only one is active, called the *current insertion point*

The insertion point is generally rendered specially (e.g., on the screen, by a vertical bar or similar cursor).

The *user focus.*

The user focus designates an *active component* in a document. In HTML documents, active components are defined to be links, form controls, elements with a value for the "longdesc" attribute, and elements with associated scripts. An element with the focus may be *activated* through any number of mechanisms, including the mouse, keyboard, an API, etc. The meaning of activation depends on the component. For instance, when a link is activated, the user agent generally retrieves the linked resource, which may be another Web page, program, etc. When a form control is activated, it may change state (e.g., check boxes) or may take user input (e.g., a text field). Activating a component with a script assigned for that particular activation mechanism (e.g., mouse down event, key press event, etc.) causes the script to be executed.

A view has only one focus. When several views co-exist, each may have a focus, but only one is active, called the *current focus.*

The current focus is generally highlighted.

The *user selection*

The user selection is defined as the part of a document (possibly spanning several elements) identified for user interaction other than with active components. For instance, the selection may be used for cut/copy/paste operations, to identify what a screen reader should read, etc. The user selection may be set by the user (e.g., by a pointing device or the keyboard) or through an application programming interface (API). A view may have only one user selection. When several views co-exist, each may have a user selection, but only one is active, called the *current user selection.*

The user selection is generally highlighted. On the screen, the selection may be highlighted using colors, fonts, graphics, or other mechanisms. Highlighted text is often used by third-party assistive technologies to indicate through speech or Braille output what the user wants to read. Most screen readers are sensitive to highlight colors. Third-party [p. 19] assistive technologies may provide alternative presentation of the selection through speech, enlargement, or dynamic Braille display.

The user selection may be used, for example, to identify the "current cell" of a table that the user is navigating.

Both the current focus and the current user selection must be in the same view, called the *current view.* The current view is generally highlighted when several views co-exist.

Which of the three mechanisms - insertion point, selection, and focus - is used to designate the point of regard depends on context. For example, for navigating among form controls, the focus determines which control has the point of regard. For navigating table cells, the selection determines which cell has the point of regard. When a technique involves the point of regard, it specifies which mechanism is used to designate it.

Events and scripting

When certain *events* occur (document loading or unloading events, mouse press or hover events, keyboard events, etc.), user agents often perform some task (e.g., execute a script). For instance, in most user agents, when a mouse button is released over a link, the link is activated and the linked resource retrieved. User agents may also execute author-defined scripts when certain events occur. The script bound to a particular event is called an *event handler.* **Note.** The interaction of HTML, style sheets, the Document Object Model [DOM1] [p. 20] and scripting is commonly referred to as "Dynamic HTML" or DHTML. However, as there is no W3C specification that formally defines DHTML, this document will only refer to event handlers and scripts.

2.4 Alternative representations of content

Certain types of content (e.g., images, audio, video, applets, etc.) may not be accessible to all users, so user agents must ensure that users have access to author-supplied *alternative representations of content.* The techniques document describes the different mechanisms authors may use to supply alternative representations of content.

2.5 Dependent vs. Independent user agents

An *independent* user agent only requires the source document (and associated style sheets, etc.) to render the document's content. It may use an accessibility API to retrieve information about a document.

A *dependent* user agent relies on the output of some other user agent in order to render a page. Dependent user agents include screen magnifiers and screen readers, both of which rely on the visual output of another user agent.

---

3 The user agent must be accessible to all users

Ideally, able-bodied as well as disabled users would be aware of user agent features that improve accessibility and experienced peers could share their knowledge with new users. In reality, most able-bodied peers do not know about features that improve accessibility (even though the same features are useful for mobile devices, slow network connections, etc.). Consequently, users with disabilities must find help information on their own to optimize the user agent to their preferences. This information must be readily available and in accessible formats.

3.1 Ensure that the user interface is accessible

[Technique: 3.1.1] [Priority 1]

Ensure that all functionalities offered by the user agent (for the user agent itself or for access to the document) are available through redundant input and output mechanisms (e.g. not mouse-only).

[Technique: 3.1.2] [Priority 1]
Implement user agent all windows, menus, controls and toolbars using the general principles of accessible design.
[Technique: 3.1.3] [Priority 1]
Ensure that the application installation procedure, including the interface, is accessible.

3.2 Ensure that user agent accessibility features are configurable

The organization and layout of accessibility feature controls has a major impact on helping users with disabilities find and learn about configuration options and the range of browsing options available to them. If the user agent interface does not offer coherent and consistent means for configuring the software, users with disabilities may not be able to optimize the user interface to their preferences.

[Technique: 3.2.1] [Priority 1]
Allow users to configure the user agent according to the conventions of the operating system.
[Technique: 3.2.2] [Priority 1]
Ensure that the Interface for configuring the software is accessible (e.g., no mouse-only configuration mechanisms).
[Technique: 3.2.3] [Priority 1]
Allow users to configure keyboard access (key combinations, distance between active keys, etc.).
[Technique: 3.2.4] [Priority 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).
[Technique: 3.2.5] [Priority 3]
Allow the user, through a keyboard command, to switch between user agent default values [p. 6] and the user profile.
[Technique: 3.2.6] [Priority 3]
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).
[Technique: 3.2.7] [Priority 3]
Furnish predefined profiles of user agent feature settings applicable to users with common disabilities.

3.3 Ensure that users can disable features that might interfere with accessibility

Users should be able to turn on and off certain features that may interfere with accessibility.

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

[Technique: 3.3.2] [Priority 1]
Allow the user to turn on and off video rendering.
[Technique: 3.3.3] [Priority 1]
Allow the user to turn on and off sound rendering.
[Technique: 3.3.4] [Priority 1]
Allow the user to turn on and off support for equivalent textual representation.
[Technique: 3.3.5] [Priority 1]
Allow the user to turn on and off blinking text for all cases when the user agent knows that text is blinking.
[Technique: 3.3.6] [Priority 1]
Allow the user to turn on and off blinking images and animations.
[Technique: 3.3.7] [Priority 1]
Allow the user to turn on and off support for scripts (including event handlers).
[Technique: 3.3.8] [Priority 1]
Allow the user to turn on and off support for style sheets.
[Technique: 3.3.9] [Priority 1]
Allow the user to turn on and off support for frames.

3.4 Provide documentation about accessibility features and ensure the documentation is accessible

Documentation must be available in accessible formats for people with print impairments. If users with visual impairments and learning disabilities and movement impairments that limit the use of paper or on-line documents, they may not know about user agent features or be able to use the user agent efficiently or at all.

[Technique: 3.4.1] [Priority 1]
Ensure that installation documentation is accessible.
[Technique: 3.4.2] [Priority 1]
Ensure that the online documentation is in an accessible format.
[Technique: 3.4.3] [Priority 1]
Ensure that the online documentation interface is accessible.
[Technique: 3.4.4] [Priority 2]
Provide a section on accessibility features in the online documentation.

3.5 Provide summary information about keyboard access

Keyboard access can be vital to users having any of several disabilities. The more apparent the keyboard commands are to **all** users, the more likely it is that new users with disabilities will find them. Readily available information about keyboard access is crucial to its effective use by users with visual impairments and some types of movement impairments, otherwise a user with disabilities may not think they can accomplish a particular task or may try to use a very inefficient technique with a pointing device, like a mouse.

[Technique: 3.5.1] [Priority 1]
Provide the user with information about keyboard bindings (organized by key or by topic).
[Technique: 3.5.2] [Priority 2]
Display keyboard navigation shortcut commands in customizable menus.

3.6 Allow users to interact with the and document in a device-independent manner

Users must be able to emulate events when their user agents do not allow them to trigger events in the anticipated way. For instance, users who cannot use a mouse pointing device must be able to trigger the same events that users with mice can trigger. Furthermore, some users may not realize when an event occurs because the event has an effect the user cannot perceive (e.g., a visual or auditory effect). The user agent must be able to inform users when events take place and to a certain extent, what has been the effect of the event.

[Technique: 3.6.1] [Priority 1]
Allow the user to follow links in a device-independent manner.
[Technique: 3.6.2] [Priority 1]
Allow the user to activate form controls in a device-independent manner.
[Technique: 3.6.3] [Priority 1]
Allow the user to trigger events in a device-independent manner.

4 The user agent must render information accessibly

In order to access a document, some users may require that it be rendered in a manner vastly different from that which the author intended. Users who are color-blind may not be able to perceive certain color combinations. Users with low vision may require large fonts. Blind users may require audio or tactile rendering. Deaf users may require textual equivalents of audio files.

User agents must therefore allow the user to control:

- The document's style (e.g., fonts, colors, aural parameters, etc.)
- The document's formatting: whether the document is presented textually, graphically, linearly, aurally, for tactile use, or some combination of these.
- The document's content. This means whether "normal" content or alternative representations of content or both are rendered.
- The user interface. Since authors may make changes to the user interface through scripting (e.g., by spawning new windows, causing dialog boxes to appear, etc.), users must be able to override changes that make the user agent or document inaccessible.

The following guidelines address user control over rendering.

4.1 Allow the user to control document styles

The user must be able to control the colors, fonts and other stylistic aspects of a document and to override author styles and user agent default styles. Otherwise, users who are blind, have visual impairments, some types of learning disabilities, or any user who cannot or has chosen not to view the primary representation of information will not know content of the information on the page.

Techniques for text:

[Technique: 4.1.1] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for font

family.

[Technique: 4.1.2] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for font size.

[Technique: 4.1.3] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for foreground color.

[Technique: 4.1.4] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for background color.

Techniques for the selection and focus:

[Technique: 4.1.5] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for selection foreground and background color.

[Technique: 4.1.6] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for focus foreground and background color.

Techniques for images, applets, and animations:

[Technique: 4.1.7] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for background images.

[Technique: 4.1.8] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for animations.

Techniques for video:

[Technique: 4.1.9] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for video frame rates.

Techniques for audio:

[Technique: 4.1.10] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for audio playback rate.

[Technique: 4.1.11] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for audio volume.

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

Techniques for speech:

[Technique: 4.1.13] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for speech playback rate.

[Technique: 4.1.14] [Priority 1]
Allow the user to override author styles [p. 6] and user agent defaults [p. 6] for speech volume.

Techniques for changes to the user interface:

[Technique: 4.1.15] [Priority 1]
When new windows or user interface components are spawned, allow the user to override author-designated changes to window size.
[Technique: 4.1.16] [Priority 1]
When new windows or user interface components are spawned, allow the user to override author-designated changes to window position.

4.2 Provide access to alternative representations of content and control of its rendering

User agents must give users access to author-supplied alternative representations of content [p. 8] (descriptions of images, video clips, sounds, applets, etc.) Some users cannot perceive the primary content due to a disability or a technological limitation (e.g., browser configured not to display images). If users cannot access alternative representations of content, users who are blind, have visual impairments, some types of learning disabilities, or any user who cannot or has chosen not to view the primary representation of information will not understand the intention or content of the page.

Techniques for images:

[Technique: 4.2.1] [Priority 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.
[Technique: 4.2.2] [Priority 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 at the same time as primary content.
[Technique: 4.2.3] [Priority 2]
When no alternative text representation is available, indicate what type of object is present.
[Technique: 4.2.4] [Priority 3]
When null alternative text has been defined, suppress the rendering of the alternative representation.

Techniques for audio and video:

[Technique: 4.2.5] [Priority 1]
Allow the user specify that textual equivalents for audio be rendered at the same time as the audio.
[Technique: 4.2.6] [Priority 1]
Allow the user specify that textual equivalents for video be rendered at the same time as the video.
[Technique: 4.2.7] [Priority 1]
Ensure that textual equivalents rendered at the same time as video not interfere visually with the video.

[Technique: 4.2.8] [Priority 1]
Allow the user specify that audio equivalents for video be rendered at the same time as the video.
[Technique: 4.2.9] [Priority 1]
For textual equivalents, allow the user to override author styles [p. 6] and user agent defaults [p. 6] for font family.
[Technique: 4.2.10] [Priority 1]
For textual equivalents, allow the user to override author styles [p. 6] and user agent defaults [p. 6] for font size.
[Technique: 4.2.11] [Priority 1]
For textual equivalents, allow the user to override author styles [p. 6] and user agent defaults [p. 6] for foreground color.
[Technique: 4.2.12] [Priority 1]
For textual equivalents, allow the user to override author styles [p. 6] and user agent defaults [p. 6] for background color.

Techniques for other types of objects

[Technique: 4.2.13] [Priority 1]
Allow the user to specify that alternatives to a script be rendered (e.g., in HTML, the content of NOSCRIPT).
[Technique: 4.2.14] [Priority 1]
Allow the user to specify that alternatives to a frame be rendered (e.g., in HTML, the content of NOFRAMES).
[Technique: 4.2.15] [Priority 1]
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).

4.3 Allow the user to choose formatting solutions

By offering several formatting solutions to users, user agents allow them to select the solution most adapted to their needs. For instance, users with screen readers will have difficulty understanding some tables whose cell contents exceed one line. In this case, if the independent user agent can linearize the table (i.e., render it one cell at a time), the screen reader’s output will not cause confusion. Similarly, some users may require that a user agent present an outline view [p. 6] of a document so they may have an easier time navigating the document.

[Technique: 4.3.1] [Priority 1]
Allow users to specify that a page be formatted linearly.
[Technique: 4.3.2] [Priority 1]
Allow users to specify that a table be formatted linearly.
[Technique: 4.3.3] [Priority 2]
Allow users to view a document outline constructed from its structural elements (e.g., from header and list elements in HTML).

5 The user agent must facilitate orientation, navigation, and searching on a page and between pages

Orientation - the sense of where one is within a document or series of documents - is fundamental to a successful Web experience. Authors may contribute to orientation through site maps, navigation bars, visual associations between documents using frames, etc. User agents also facilitate orientation with proportional scrollbars on viewports. [p. 6] Not all users can use visual orientation clues, however, so user agents must complement them by:

- Providing support for a point-of-regard [p. 6] for people relying on assistive technology.
- Providing summary information about certain elements (e.g., the number of links, headers, tables, etc.) in all or part of a document.
- Supporting, when appropriate for the target medium, speech and tactile identification of elements.

If users cannot orient themselves to the types of elements in a document, users who are blind, have visual impairments, some types of learning disabilities, or any user who cannot or has chosen not to view the author's representation of information will have incomplete knowledge of the content of the document.

Navigation and searching are closely related to orientation. There are different ways a user may want to navigate while browsing the Web, including:

- By changing the position of the viewport [p. 6] (e.g., by scrolling down the page).
- By shifting focus and/or selection from one active component to the next of the same type (e.g., from link to link).
- By navigating among documents.

For all of the navigation techniques described below, the user agent must allow the user to navigate via the keyboard.

5.1 Provide information about the content and structure of a document and the user interface

[Technique: 5.1.1] [Priority 1]
Provide the user with information about the number of links in a document.
[Technique: 5.1.2] [Priority 1]
Provide the user with information about the number of form controls in a document.
[Technique: 5.1.3] [Priority 1]
Provide the user with information about the number of tables in a document.
[Technique: 5.1.4] [Priority 2]
Provide the user with information about the number of viewports and how they may be distinguished.
[Technique: 5.1.5] [Priority 2]
When a document is loaded or when requested by the user, make available document summary information.

[Technique: 5.1.6] [Priority 2]
Provide a mechanism to indicate visually the presence of an "accesskey" attribute defined for a link.
[Technique: 5.1.7] [Priority 2]
Provide a mechanism to indicate visually the presence of an "accesskey" attribute defined for a form control.
[Technique: 5.1.8] [Priority 3]
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.
[Technique: 5.1.9] [Priority 3]
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.
[Technique: 5.1.10] [Priority 3]
Provide a mechanism to distinguish visited links from unvisited links.
[Technique: 5.1.11] [Priority 3]
Allow the user to specify that images used in links must have borders.

5.2 Provide information about document view and point of regard

Users that are viewing documents through linear channels of perception like speech (since speech is temporal in nature) and tactile displays (current tactile technology is limited in the amount of information that can be displayed) have difficultly maintaining a sense of their relative position in a document. The meaning of "relative position" depends on the situation. It may mean the distance from the beginning of the document to the point of regard [p. 6] (how much of the document has been read), it may mean the cell currently being examined in a table, or the position of the current document in a set of documents.

For people with visual impairments, it is important that the point of regard [p. 6] remain as stable as possible. For instance, when returning to a document previously viewed, the user's previous point of regard on the document should be restored. The user agent should not disturb the user's point of regard by shifting focus to a different frame or window when an event occurs without notifying the user of the change. Disturbing the user's point of regard may cause problems for users who have movement impairments, who are blind, who have visual impairments, who have certain types of learning disabilities, or for any user who cannot or has chosen not to view the authors representation of information.

[Technique: 5.2.1] [Priority 1]
Provide a mechanism for highlighting and identifying the current view.
[Technique: 5.2.2] [Priority 1]
Provide a mechanism for highlighting and identifying the user selection.
[Technique: 5.2.3] [Priority 1]
Provide a mechanism for highlighting and identifying the current focus.
[Technique: 5.2.4] [Priority 1]
Allow the user to specify that a view's focus should follow changes in the viewport.

[Technique: 5.2.5] [Priority 1]
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.
[Technique: 5.2.6] [Priority 2]
Provide the user with information about how much of the document has been viewed (i.e., the location of the point of regard).
[Technique: 5.2.7] [Priority 2]
Provide the user with information about which table cell is the current table cell. [p. 18]
[Technique: 5.2.8] [Priority 3]
Allow the user to turn on and off automatic page forwarding.

5.3 Provide information about events that occur

[Technique: 5.3.1] [Priority 2]
Alert the user when scripts are executed.
[Technique: 5.3.2] [Priority 3]
Provide information about document changes resulting from the execution of a script.
[Technique: 5.3.3] [Priority 2]
Allow users to be prompted before spawning a new window.

5.4 Allow sequential keyboard navigation to structures within a document

Often, users wish to navigate a linear sequence of elements within a document. If sequential navigation is not available users with some types of movement impairments, visual impairments and learning disabilities may not be able to access the links and controls in a document.

[Technique: 5.4.1] [Priority 1]
Allow the user to navigate sequentially among links.
[Technique: 5.4.2] [Priority 1]
Allow the user to navigate sequentially among elements with associated long descriptions.
[Technique: 5.4.3] [Priority 1]
Allow the user to navigate sequentially among form controls.
[Technique: 5.4.4] [Priority 1]
Allow the user to navigate sequentially among elements with associated event handlers.
[Technique: 5.4.5] [Priority 2]
Allow the user to navigate sequentially among headers.
[Technique: 5.4.6] [Priority 2]
Allow the user to navigate sequentially among block elements (e.g., paragraphs, lists and list items, etc.)

5.5 Allow keyboard navigation of the document and views of the document

Hierarchical navigation (through the document tree) is useful for efficiently navigating the major topics and sub-topics of a document. Topical navigation (section by section) conveys significant information about the organization of a document. If hierarchical navigation is not available users with some types of movement impairments, visual impairments and learning disabilities may not be able to access the links and controls on a page efficiently nor understand the topics and topic relationships within a document.

[Technique: 5.5.1] [Priority 1]
Allow the user to navigate views (notably those with frame viewports). Navigating into a view makes it the current view.
[Technique: 5.5.2] [Priority 2]
Allow the user to use the keyboard to navigate the document tree.

Tabular navigation is required by people with visual impairments and some types of learning disabilities to determine the content of a particular cell and spatial relationships between cells (which may convey information). If table navigation is not available users with some types of visual impairments and learning disabilities may not be able to understand the purpose of a table or table cell.

[Technique: 5.5.3] [Priority 1]
Provide a mechanism for designating the *current table* of a document.
[Technique: 5.5.4] [Priority 1]
Provide a mechanism for designating the *current cell* of a table.
[Technique: 5.5.5] [Priority 1]
Allow the user to navigate among tables in a document.
[Technique: 5.5.6] [Priority 1]
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 Allow the user to search for information on the page

For efficient browsing, users may want to navigate to a particular element of a certain type. For instance, in a document with many links, sequential navigation will require a great number of key strokes to reach a desired link. In addition to being wearisome, this may be physically difficult for some users. Without direct access (e.g, by element identifier or element position), users with some types of movement impairments, visual impairments and learning disabilities may not be able to access the links and controls and other elements in a document efficiently.

When a search matches, the point of regard [p. 6] is moved to the matched location.

[Technique: 5.6.1] [Priority 1]
Allow the user to search for a link in the current document based on its link text.
[Technique: 5.6.2] [Priority 1]
Allow the user to search for a link in the current document based on its attribute values.
[Technique: 5.6.3] [Priority 1]
Allow the user to search for a link in the current document based on its position.

[Technique: 5.6.4] [Priority 1]
Allow the user to search for a form control in the current document based on its text content.
[Technique: 5.6.5] [Priority 1]
Allow the user to search for a form control in the current document based on its attribute values.
[Technique: 5.6.6] [Priority 2]
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.
[Technique: 5.6.7] [Priority 2]
Allow the user to search for an element in the current document based on its text content.
[Technique: 5.6.8] [Priority 3]
Allow the user to search for a table cell based on its contents, row/column coordinates, or header information.

6 The user agent must make information available to other technologies

Many types of software: assistive technologies, scripting tools, automated test engines, etc., benefit from having access to user agent information.

6.1 Support language accessibility features

A user agent that supports a language should implement the accessibility features for the language. The techniques document lists the accessibility features of the following languages, defined by W3C specifications:

[Technique: 6.1.1] [Priority 1]
Support accessibility features of HTML.
[Technique: 6.1.2] [Priority 1]
Support accessibility features of CSS.
[Technique: 6.1.3] [Priority 1]
Support accessibility features of SMIL.

6.2 Use and provide accessible interfaces to other technologies

Whether or not a user agent chooses to implement a particular feature, it must make pertinent information available to other software in an interoperable manner. Even though a user agent may implement a feature, specialized assistive software may furnish a better implementation and must be given access to all pertinent information through standard interfaces.

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

[Technique: 6.2.2] [Priority 1]
For information that can be exchanged through an interface defined by a W3C specification, user agents should implement that specification.
[Technique: 6.2.3] [Priority 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.
[Technique: 6.2.4] [Priority 2]
Otherwise, standard interfaces defined for the operating system should be used.
[Technique: 6.2.5] [Priority 2]
Make use of operating system accessibility flags and interfaces.

7 Acknowledgments

Many thanks to the following people who have contributed through review and comment: Paul Adelson, James Alan, Denis Anson, Kitch Barnicle, Harvey Bingham, Judy Brewer, Kevin Carey, Wendy Chisholm, David Clark, Chetz Colwell, Wilson Craig, Daniel Dardailler, Neal Ewers, Geoff Freed, Larry Goldberg, Markku Hakkinen, Earle Harrison, Chris Hasser, Kathy Hewitt, Phill Jenkins, Leonard Kasday, George Kerscher, Marja-Riitta Koivunen, Josh Krieger, Catherine Laws, Greg Lowney, Scott Luebking, William Loughborough, Napoleon Maou, Charles McCathieNevile, Masafumi Nakane, Charles Oppermann, Mike Paciello, David Pawson, Michael Pederson, Helen Petrie, David Poehlman, Michael Pieper, Jan Richards, Hans Riesebos, Joe Roeder, 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.

8 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.