Authoring Tool Accessibility Guidelines 1.0 (working draft)

W3C Working Draft 28-July-1999

This version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS-19990728
(plain text, HTML gzip tar archive, HTML zip archive, PostScript, PDF)
Latest version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS
Previous version:
http://www.w3.org/WAI/AU/WAI-AUTOOLS-19990709

Editors:
Jutta Treviranus <jutta.treviranus@utoronto.ca>
Jan Richards <jan.richards@utoronto.ca>
Ian Jacobs <ij@w3.org>
Charles McCathieNevile <charles@w3.org>

Copyright © 1999 W3C (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

This document provides guidelines for Web authoring tool developers. Its purpose is two-fold: to assist developers in designing authoring tools that generate accessible Web content and to assist developers in creating an accessible authoring interface.

Accessible Web content is achieved by encouraging authoring tool users ("authors") to create accessible Web content through mechanisms such as prompts, alerts, checking and repair functions, help files and automated tools. It is equally important that all people can be the authors of Web content, rather than merely recipients. It is therefore of critical importance that the tools used to create this content are themselves accessible. Adoption of these guidelines will result in the proliferation of Web pages that can be read by a broader range of readers and in authoring tools that can be used by a broader range of authors.

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

Status of this document

This is a Working Draft of the Authoring Tool AccessibilityGuidelines. It is a draft document and may be updated, replaced or rendered obsolete 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 either W3C or members of the WAI Authoring Tool (AU) Working Group. It is expected that a new working draft will render this draft obsolete in early August 1999.

This draft follows the working group meeting on July 28 1999. A log of changes between successive working drafts is available.

The Techniques given in the linked "Techniques" document are intended to be informative only. They will not be present in the "normative" version, although there will still be a link to the Techniques document. This will enable them to be updated more easily than the Guidelines themselves.

The goals of the WAI AU Working Group are discussed in the WAI AU charter.

Please send comments about this document to the public mailing list: w3c-wai-au@w3.org, archived at http://lists.w3.org/Archives/Public/w3c-wai-au

A list of the current AU Working Group members is available.

Table of Contents

Abstract 1
Status of this document 2
1. Introduction 4
1.1 Scope of these Guidelines 4
1.2 How the Guidelines are organized. 4
1.3 Checkpoint priorities 5
1.4 Conformance to these Guidelines 6
2. Guidelines 6
1. Ensure that the Authoring Tool is Accessible to Authors with Disabilities 6
2. Generate standard markup 7
3. Support accessible authoring practices 7
4. Ensure that no accessibility content is missing 8
5. Integrate accessibility solutions into the overall "look and feel" . . .9
6. Provide methods of checking and correcting inaccessible content . .9
7. Promote accessibility in help and documentation 10
3. Terms and Definitions 11
4. Acknowledgments 14
5. References 15

1. Introduction

The various authoring tools used to generate Web content play a critical role in determining the form and accessibility of the Web. Authoring tools are used to automate the mechanical tasks that are part of producing Web pages. The power of this automation can enhance the accessibility of the Web if it is used to ensure that the code produced promotes accessibility, and frees the author to concentrate on the higher level problems of overall design, content, description, etc. It is imperative that authoring tools generate content that is accessible, and that they are accessible themselves, to allow people to be consumers and producers of Web content on an equal footing, regardless of disability.

The accessibility of authoring tools encompasses some general principles of software accessibility, and some features that are specific requirements for authoring. The accessibility of the content produced depends on the ability of the tool to be used in producing accessible markup, and on the user interface of the tool enabling, informing, and encouraging the use of accessible markup authoring practices. These Guidelines refer extensively to the Web Content Accessibility Guidelines, that details accessibility requirements for markup itself, and include checkpoints that are basic requirements for the accessibility of the tool and its output. In addition, there are guidelines and checkpoints that are uniquely relevant to the role authoring tools play in guiding the author to produce accessible content.

1.1 Scope of these Guidelines

These guidelines are intended to be used by developers of all tools used to produce content for the Web. These include:

- Editing tools specifically designed to produce Web content (e.g., WYSIWYG HTML editors, SMIL authoring packages);
- Tools that offer the option of saving material in a Web format (e.g., word processors or desktop publishing packages);
- Tools that translate documents into Web formats (e.g., filters to translate desktop publishing formats to HTML);
- Tools that produce multimedia, especially where it is intended for use on the Web (e.g., video production and editing suites);
- Tools for site management or site publication, including on-the-fly conversion and Web site publishing tools;
- Tools for management of layout (e.g., CSS formatting tools).

1.2 How the Guidelines are organized.

This document includes seven guidelines, or general principles of accessible design. Each guideline includes:

- The guideline number.
- The statement of the guideline.
- The rationale behind the guideline.
- A list of checkpoint definitions.

The checkpoint definitions in each guideline specify requirements for authoring tools to follow the guideline. Each checkpoint definition includes:

- The checkpoint number.
- The statement of the checkpoint.
- The priority of the checkpoint.
- In some cases informative notes, clarifying examples, or cross references to related guidelines or checkpoints.
- A link to a section of the Techniques Document ([WAI-AUTOOLS-TECH] [p. 15]) where implementations and examples of the checkpoint are discussed.

Each checkpoint is intended to be specific enough that someone may verify that the checkpoint has been satisfied, while being sufficiently general to allow developers the freedom to use the most appropriate strategies to meet the checkpoint.

The Techniques provided in the techniques document are suggestions for how implementation might be done, or where further information can be found. They are informative only, and other strategies may be used to meet the checkpoint as well as, or in place of, those discussed.

1.3 Checkpoint priorities

There are four goals:

1. The authoring tool is accessible
2. The authoring tool generates accessible content by default
3. The authoring tool is user configurable
4. The authoring tool encourages the creation of accessible content

Checkpoints are assigned priority according to how important they are to meeting those goals:

[Priority 1]
: Essential to meeting those goals

[Priority 2]
: Important to meeting those goals

[Priority 3]
: Beneficial to meeting those goals

1.4 Conformance to these Guidelines

This section defines three levels of conformance to this document:

- Conformance Level "A": all Priority 1 checkpoints are satisfied;
- Conformance Level "Double-A": all Priority 1 and 2 checkpoints are satisfied;
- Conformance Level "Triple-A": all Priority 1, 2, and 3 checkpoints are satisfied;

Note. Conformance levels are spelled out in text so they may be understood when rendered to speech.

Claims of conformance to this document must use one of the following two forms.

Form 1: Specify:

- The guidelines' title: "Authoring Tool Accessibility Guidelines 1.0 (working draft)"
- The guidelines' URI: http://www.w3.org/WAI/AU/WAI-AUTOOLS-19990728
- The conformance level satisfied: "A", "Double-A", or "Triple-A".
- The scope covered by the claim (e.g., tool name and version number, upgrades or plugins required).

Example of Form 1: "MyAuthoringTool version 2.3 conforms to W3C's "Authoring Tool Accessibility Guidelines 1.0 (working draft)", available at http://www.w3.org/WAI/AU/WAI-AUTOOLS-19990728, level Double-A."

Form 2: Include, on each statement of conformance, one of three icons provided by W3C and link the icon to the appropriate W3C explanation of the claim.

[Editors' note: In the event this document becomes a Recommendation, by that date WAI will provide a set of three icons, for "A", "Double-A", or "Triple-A" conformance levels of "Authoring Tool Accessibility Guidelines 1.0 (working draft)", together with a stable URI to the W3C Web site for linking the icons to the W3C explanation of conformance claims.]

2. Guidelines

Guideline 1. Ensure that the Authoring Tool is Accessible to Authors with Disabilities

The authoring tool is a software program with standard user interface elements and as such should follow relevant user interface accessibility guidelines.

The author may need a different presentation to edit the Web content than the one they wish ultimately to be displayed. This implies display preferences that do not manifest themselves in the ultimate markup or style declarations.

Authoring Web content requires editing a potentially large and complex document. In order to edit a document the author must be able to locate and select specific blocks of text, efficiently traverse the document, and quickly find and mark insertion points. Authors who use screen readers, refreshable braille displays, or screen

magnifiers can make limited use (if at all) of visual artifacts that communicate the structure of the document and act as sign posts when traversing the document. There are strategies that make it easier to navigate and manipulate a marked up document. A compressed view of the document allows the author to both get a good sense of the overall structure and to navigate that structure more easily.

Checkpoints:

1.1 Use all applicable operating system and accessibility standards and conventions (Priority 1 for standards and conventions which are essential to accessibility, Priority 2 for those that are important to accessibility, Priority 3 for those that are beneficial to accessibility). [Priority 1]
1.2 Allow the author to change the editing view [p. 13] without affecting the document markup. [Priority 1]

> This allows the author to edit the document according to their personal requirements, without changing the way the document looks or is rendered when published.

1.3 Allow the author to display and edit all properties of each element and object in an accessible fashion. [Priority 1]
1.4 Ensure the editing view allows navigation via the structure of the document. [Priority 1]
1.5 Enable editing of the structure of the document. [Priority 2]

Guideline 2. Generate standard markup

Conformance with standards promotes interoperability and accessibility.

Checkpoints:

2.1 Use applicable W3C Recommendations. [Priority 2]
2.2 Ensure that content is created in accordance with a published standard [Priority 1]

> This is necessary ion order to produce user agents which can transform the site to produce a presentation appropriate to a particular user's needs.

2.3 Ensure that document language used enables accessibility of content. [Priority 1]

> This is relevant both to the use of an existing document language, and to one which is created or extended for a specific purpose..

Guideline 3. Support accessible authoring practices

Methods for ensuring accessible markup vary with different markup languages. If markup is automatically generated, many authors will be unaware of the accessibility status of the final product unless they expend extra effort to make appropriate corrections by hand. Since many authors are unfamiliar with accessibility, these problems are likely to remain.

Many applications feature the ability to convert documents from other formats (e.g., Rich Text Format) into a markup format, such as HTML. Markup changes may also be made to facilitate efficient editing and manipulation. These processes are usually hidden from the user's view and may create inaccessible content or cause inaccessible content to be produced.

Checkpoints:

3.1 Implement all accessible authoring practices that have been defined for the markup language(s) supported by the tool. [Priority 1]
3.2 Produce content that conforms to the W3C's Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 15] . [Priority 1 for level-A conformance, Priority 2 for double-A conformance, Priority 3 for triple-A conformance]
3.3 Ensure that templates provided by the tool conform to W3C Web Content Accessibility Guidelines [WAI-WEBCONTENT] [p. 15] . [Priority 1 for level-A conformance, Priority 2 for double-A conformance, Priority 3 for triple-A conformance]
3.4 Preserve all accessibility content during transformations and conversions. [Priority 1]

Guideline 4. Ensure that no accessibility content is missing

Textual equivalents, including "alt-text", long descriptions, video captions, and transcripts are absolutely necessary for the accessibility of all images, applets, video, and audio files. However, the task of producing these equivalents is probably the most time-consuming accessibility recommendation made to the author.

The authoring tool can provide various mechanisms to assist the author in generating textual equivalents while ensuring that the author can determine whether the textual equivalent accurately reflects the information conveyed by the multimedia object.

Including professionally written descriptions for all multimedia files (e.g., clip-art) packaged with the tool will:

- Save users time and effort;
- Cause a significant number of professionally written descriptions to circulate on the Web;
- Provide users with convenient models to emulate when they write their own descriptions;
- Show authors the importance of description writing.

This will lead to an increase in the average quality of descriptions used.

Checkpoints:

4.1 Prompt the author to provide alternative content (e.g., captions, descriptive video). (Priority 1 for alternative content that is [Web-Content-Priority-1] [p. 15] , Priority 2 for alternative content that is [Web-Content-Priority-2] [p. 15] , Priority 3 for alternative content that is [Web-Content-Priority-3] [p. 15])
4.2 Prompt the author for all missing structural information (e.g., language changes, table headers). (Priority 1 for structural information that is [Web-Content-Priority-1] [p. 15] , Priority 2 for structural information that is [Web-Content-Priority-2] [p. 15] , Priority 3 for structural information that is [Web-Content-Priority-3] [p. 15])
4.3 Provide pre-written alternative content for all multimedia files packaged with the authoring tool. [Priority 2]

> Note: This text should be used for input field default, but not for automatic insertion.

4.4 Provide a mechanism to manage alternative content for multimedia objects, that retains and offers for editing pre-written or previously linked alternative content. [Priority 3]
4.5 Do not insert automatically generated (e.g., the filename) or place-holder (e.g., "image") equivalent text, except in cases where human-authored text has been written for an object whose function is known with certainty. [Priority 1]

Guideline 5. Integrate accessibility solutions into the overall "look and feel"

When a new feature is added to an existing software tool without proper integration, the result is often an obvious discontinuity. Differing color schemes, fonts, interaction styles and even application stability can be factors affecting user acceptance of the new feature.

Checkpoints:

5.1 Ensure that the highest-priority accessible authoring practices are among the most obvious and easily initiated by the author. [Priority 2]
5.2 Make generation of accessible content a naturally integrated part of the authoring process. [Priority 1]

Guideline 6. Provide methods of checking and correcting inaccessible content

Many authoring tools allow authors to create documents with little or no knowledge about the underlying markup. To ensure accessibility, authoring tools must be designed so that they may automatically identify inaccessible content, and enable its correction even when the markup itself is hidden from the author.

In supporting the creation of accessible Web content, authoring tools must take into account the differing authoring styles of their users. Some users may prefer to be alerted to problems when they occur, whereas others may prefer to perform a check after the document is completed. This is analogous to programming environments that allow users to decide whether to check for correct code during editing or at compile time.

Note that validity is an accessibility requirement, particularly for assistive technologies.

Checkpoints:

6.1 Check for and alert the author to accessibility problems. (Priority 1 for accessibility problems that are [Web-Content-Priority-1] [p. 15] , Priority 2 for accessibility problems that are [Web-Content-Priority-2] [p. 15] , Priority 3 for accessibility problems that are [Web-Content-Priority-3] [p. 15])
6.2 Allow users to control both the nature and timing of accessibility alerts. [Priority 2]
6.3 Assist authors in correcting accessibility problems. (Priority 1 for accessibility problems that are [Web-Content-Priority-1] [p. 15] , Priority 2 for accessibility problems that are [Web-Content-Priority-2] [p. 15] , Priority 3 for accessibility problems that are [Web-Content-Priority-3] [p. 15])
6.4 When removing unrecognized markup, alert the author (according to a configurable schedule). [Priority 2]
6.5 Provide the author with a summary of the document accessibility status on a configurable schedule. [Priority 3]
6.6 Allow the author to perform tag transformations. [Priority 3]
For example, to transform visually formatted elements to structure elements, or tables to lists.

Guideline 7. Promote accessibility in help and documentation

The issues surrounding Web accessibility are often unknown to Web authors. Help and documentation should explain accessibility problems and solutions, with examples.

Checkpoints:

7.1 Integrate accessible authoring practices in all applicable help topics. [Priority 1]
7.2 Explain the accessible authoring practices supported by the authoring tool. [Priority 1]
7.3 Do not use inaccessible markup in examples. [Priority 1]
7.4 Emphasize the universal benefit of accessible design. [Priority 3]

3. Terms and Definitions

[Editors' note: This section will be reviewed by the group, and is expected to be updated in future drafts]

Prompts and Alerts

User Configurable Schedule
: A user configurable schedule allows the user to determine the type of prompts and alerts that are used, including when they are presented.

Prompts
: Prompts are requests for user input, either information or a decision. Prompts require author response.

Alerts
: Alerts notify the author of something, or mark something for the author's attention. They may or may not require author response.

Authoring Tool
: As used in this document, an *Authoring Tool* is any software that is used to generate content for publishing on the Web. See also section 1.3 Scope of these guidelines.

Conversion Tool

Generation Tool

Site Management Tool

Publishing Tool

Image Editor

Video Editor

Multi-media Authoring Tool

Automated Markup Insertion Function

Transformation
: A process whereby one object is changed, according to a discrete set of rules, into another, equivalent, object. This includes any application or application feature that allows content that is marked up in a particular markup language to be transformed into another markup language, such as software that allows the author to change the DTD defined for the original document to another DTD. It also describes the substitution of textual equivalents for graphical or visually defined elements and objects, and the conversion from one element type to another within a document.

Document
: A *document* is a series of elements that are defined by a language (e.g., HTML 4.0 or an XML application).

Element
: An element is any identifiable object within a document, for example a character, word, image, paragraph or spreadsheet cell. In HTML and XML an element refers to a pair of tags and their content, or an "empty" tag - one that has no closing tag or content.

Property
: A property is a piece of information about an element, for example structural information (e.g., it is item number 7 in a list, or plain text) or presentation information (e.g., that it is marked as bold, its font size is 14). In XML and HTML properties of an element include the name of the element (e.g., IMG or DL), the values of its attributes, and information associated by means of a stylesheet. In a database, properties of a particular element may include values of the entry, and acceptable data types for that element.

Attributes
: in XML and HTML, an element may have any number of attributes. In the following example, the attributes of the beverage element are flavor, which has the value "lots", and colour, which has the value "red": <beverage flavor="lots" colour="red">my favorite</beverage> Some attributes are integral to document accessibility (e.g., the "alt", "title", and "longdesc" attributes in HTML

Rendered Content
: The *rendered content* is that which an element actually causes to be rendered by the user agent. This may differ from the element's structural content. For example, some elements cause external data to be rendered (e.g., the IMG element in HTML), and in some cases, browsers may render the value of an attribute (e.g., "alt", "title") in place of the element's content.

Accessible, Accessibility
: Within these guidelines, Accessible and Accessibility are used in the sense of being accessible to people regardless of disability.

Accessibility Solution, Accessible Authoring Practice
: These terms refer to Authoring practices that improve the accessibility of content generated by the tool..

Alternative Representations
: Certain types of content may not be accessible to all users (e.g., images or audio presentations), so alternative representations are used, such as transcripts for audio, or short functionally equivalent text (e.g., "site map link") and/or descriptive text equivalents (e.g., "Graph 2.5 shows that the population has doubled approximately every ten years for the last fifty years, increasing from about 10 million to 330 million in that time"). An object may have several alternative representations, for example a video, captions of the audio, audio description of the video, a series of still images, and textual representations of each of these.

Inserting an element

Editing an element

Views
: An authoring tool may offer several *views* of the same document. For instance, one view may show raw markup, a second may show a structured tree view, a third may show markup with rendered objects while a final view shows an example of how the document may appear if it were to be rendered by a

particular browser.

Editing view

What is displayed by the authoring tool to the author during the editing process.

Selection

Current User Selection

Focus

4. Acknowledgments

Many thanks to the following people who have contributed through review and comment: Jim Allan, Denis Anson, Kynn Bartlett, Harvey Bingham, Judy Brewer, Carl Brown, Dick Brown, Kelly Ford, Wendy Chisholm, Rob Cumming, Daniel Dardailler, Mark Day, BK Delong, Jamie Fox, Sylvain Galineau, Al Gilman, Eric Hansen, Phill Jenkins, Len Kasday, Brian Kelly, Jaap van Lelieveld, William Loughborough, Karen McCall, Charles Oppermann, Dave Pawson, Dave Poehlman, Bruce Roberts, Chris Ridpath, Gregory Rosmaita, Jim Thatcher, Irène Vatton, Gregg Vanderheiden, Pawan Vora, Jason White, and Lauren Wood.

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

5. 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/

[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-html40/

[W3C-RECS]
"W3C Technical Reports and Publications" The latest versions of W3C Recomendations are available at:
http://www.w3.org/TR

[WAI-AUTOOLS-TECH]
"Authoring Tool Accessibility Techniques (Working Draft)", J. Treviranus, J. Richards, I. Jacobs, and C. McCathieNevile eds. The latest working draft of these techniques is available at:
http://www.w3.org/WAI/AU/WAI-AUTOOLS/wai-autools-tech

[WAI-USERAGENT]
"User Agent Accessibility Guidelines", J. Gunderson and I. Jacobs, eds. These guidelines for designing accessible user agents are available at:
http://www.w3.org/TR/WAI-USERAGENT

[WAI-WEBCONTENT]
"Web Content Accessibility Guidelines 1.0", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds. These guidelines for designing accessible documents are available at:
http://www.w3.org/TR/WAI-WEBCONTENT

[WAI-WEBCONTENT-TECHS]
"Techniques for Web Content Accessibility Guidelines", W. Chisholm, G. Vanderheiden, and I. Jacobs, eds. These techniques for designing accessible documents are available at:
http://www.w3.org/TR/WAI-WEBCONTENT-TECHS/

[Web-Content-Priority]
Priorities defined by [WAI-WEBCONTENT] [p. 15] .