

[\[Table of Contents\]](#) | [\[Implementing ATAG 2.0\]](#)



Authoring Tool Accessibility Guidelines (ATAG) 2.0

W3C Candidate Recommendation 7 November 2013

This version:

<http://www.w3.org/TR/2013/CR-ATAG20-20131107/>

Latest version:

<http://www.w3.org/TR/ATAG20/>

Previous version:

<http://www.w3.org/TR/2013/WD-ATAG20-20130910/>

Latest Editor's Draft

<http://www.w3.org/WAI/AU/ATAG20/>

Editors:

Jan Richards, Inclusive Design Institute, OCAD University

Jeanne Spellman, W3C

Jutta Treviranus, Inclusive Design Institute, OCAD University

[Copyright](#) © 2013 [W3C](#)[®] ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)), All Rights Reserved. [W3C liability](#), [trademark](#) and [document use](#) rules apply.

Abstract

This standard provides guidelines for designing web content authoring tools that are both more accessible to authors with disabilities (Part A) and designed to enable, support, and promote the production of more accessible web content by all authors (Part B). See [Authoring Tool Accessibility Guidelines \(ATAG\) Overview](#) for an introduction and links to ATAG technical and educational material.

Status of This Document

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

W3C Candidate Recommendation of ATAG 2.0

This is the Candidate Recommendation of Authoring Tool Accessibility Guidelines (ATAG) 2.0 from the [Authoring Tool Accessibility Guidelines Working Group](#). This version integrates minor editorial changes identified since the publication of the 10 September 2013 Last Call Working Draft. The Working Group received [1 comment](#) on this draft which AUWG addressed by a change to Implementing ATAG 2.0. See the [automated diff document](#) for all the editorial changes.

A Candidate Recommendation is a document that has been widely reviewed and is ready for implementation. Publication as a Candidate Recommendation does not imply endorsement by the W3C Membership. Before the specification can progress to [Proposed Recommendation](#), the [CR exit criteria](#) must be met.

Besides these implementations, feedback on implementation and use of this specification is welcome, including from implementations not selected as part of the formal implementation report for exiting Candidate Recommendation. Comments should be sent to public-atag2-comments@w3.org ([Public Archive](#)).

Working closely with authoring tool developers, we have already documented [initial implementations](#) of all exit criteria with the exception of those items that are indicated as "at risk" in this status section; however, these implementation have not yet been formally tested, which will occur during the CR period. AUWG expects to show evidence of meeting the exit criteria no earlier than 7 March 2014.

Exit Criteria

The Authoring Tool Accessibility Guidelines Working Group intends to submit this document for consideration as a W3C Proposed Recommendation as soon as the following conditions are met. We expect to complete testing and show evidence of meeting all exit criteria, and change or remove the at-risk items as needed, no earlier than 7 March 2014.

1. [Tool by tool] Two independent [\[1\]](#) authoring tools must conform to ATAG 2.0 level AA (which includes level A).
2. [Tool by category] At least one authoring tool from each of the following authoring tool categories must conform to ATAG 2.0 Level A (i.e. will conform to all applicable Level A success criteria.):
 - WYSIWYG web page editing tools
 - Content management tools
 - Development tool for applet, scripts, or applications
 - Non-text media (e.g. video, audio, images) editing tools
 - Social media content authoring tools (e.g. blogs, wikis, social networks)
3. [Success criterion by success criterion] Each ATAG 2.0 success

criterion must be implemented [2] by two independent authoring tools. For the thirteen ATAG 2.0 success criteria that are dependent on WCAG 2.0 [3] for their levels, each ATAG 2.0 success criterion must be implemented for two WCAG 2.0 success criteria at each level: A, AA, and AAA. These six WCAG 2.0 success criteria are a sampling of the requirements of WCAG (e.g. text alternatives for non-text content, keyboard accessibility, sufficient contrast).

Note 1: "Independent authoring tools" are tools by different developers that do not share (or derive from) the same source code for the relevant feature(s). Sections of code that have no bearing on the implementation of this standard are exempt from this requirement. The authoring tools must be a shipping product or other publicly available version. Experimental implementations, specifically designed to pass the test suite and not intended for normal usage, are not permitted.

Note 2: "Implemented" refers to situations in which a success criterion is applicable to a given authoring tool and the authoring tool meets the success criterion. This is in contrast to situations in which a success criterion is not applicable.

Note 3: For example, if the WCAG success criteria at level A are satisfied, then the ATAG success criteria is satisfied at level A. If the WCAG success criteria at level A and AA are satisfied, then the ATAG success criteria is satisfied at level AA. If the WCAG success criteria at level A, AA, and AAA are satisfied, then the ATAG success criteria is satisfied at level AAA.

Features At Risk

As a part of the Candidate Recommendation process, any items that might change or where there may not be implementations are marked as "at risk." "At risk" in no way implies that these success criteria are less important to accessibility. It is a W3C requirement to identify any provision for which the Working Group believes it may not be able to document the required implementations by the end of the Candidate Recommendation period.

Items At Risk for Change

If at least two implementations of each of the following success criteria do not exist at the end of the Candidate Recommendation period, the success criteria may be modified as stated.

A.3.6.3 will be changed from "platform display and control settings" to "platform high contrast settings" as follows:

- A.3.6.3 Apply Platform Settings: The authoring tool respects changes in platform high contrast settings, unless authors select more

specific display and control settings using the authoring tool. (Level AA).

A.1.1.1 may be level A, AA or AAA depending on the level of the WCAG success criteria that are met by the authoring tool. If AUWG cannot identify two implementations at WCAG AAA level, then the success criterion will be changed to remove the AAA requirement as follows:

- A.1.1.1 Web-Based Accessible (WCAG): If the authoring tool contains web-based user interfaces, then those web-based user interfaces meet the WCAG 2.0 success criteria. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria).

Items At Risk for Removal

If at least two implementations of the following success criterion do not exist at the end of the Candidate Recommendation period, the following success criterion may be removed.

- B.2.4.4 Identify Template Accessibility (Enhanced) : If the authoring tool provides templates, then all of the templates are accessible template (to WCAG Level AA). (Level AAA)

The [Authoring Tool Accessibility Guidelines Working Group \(AUWG\)](#) intends to publish ATAG 2.0 as a W3C Recommendation. Until that time [Authoring Tool Accessibility Guidelines \(ATAG\) 1.0 \[ATAG10\]](#) is the stable, referenceable version. This Working Draft does not supersede ATAG 1.0.

Web Accessibility Initiative

This document has been produced as part of the W3C [Web Accessibility Initiative](#) (WAI). The goals of the AUWG are discussed in the [Working Group charter](#). The AUWG is part of the [WAI Technical Activity](#).

No Endorsement

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

Patents

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

[Essential Claim\(s\)](#) must disclose the information in accordance with [section 6 of the W3C Patent Policy](#).

Table of Contents

[Abstract](#)

[Status of This Document](#)

[Introduction](#)

[ATAG 2.0 Layers of Guidance](#)

[Levels of Conformance](#)

[Integration of Accessibility Features](#)

[ATAG 2.0 Guidelines](#)

[A. Make the authoring tool user interface accessible](#)

[A.1. Authoring tool user interfaces follow applicable accessibility guidelines](#)

- [A.1.1. \(For the authoring tool user interface\) Ensure that web-based functionality is accessible](#)
- [A.1.2. \(For the authoring tool user interface\) Ensure that non-web-based functionality is accessible](#)

[A.2. Editing-views are perceivable](#)

- [A.2.1. \(For the authoring tool user interface\) Make alternative content available to authors](#)
- [A.2.2. \(For the authoring tool user interface\) Ensure that editing-view presentation can be programmatically determined](#)

[A.3. Editing-views are operable](#)

- [A.3.1. \(For the authoring tool user interface\) Provide keyboard access to authoring features](#)
- [A.3.2. \(For the authoring tool user interface\) Provide authors with enough time](#)
- [A.3.3. \(For the authoring tool user interface\) Help authors avoid flashing that could cause seizures](#)
- [A.3.4. \(For the authoring tool user interface\) Enhance navigation and editing via content structure](#)
- [A.3.5. \(For the authoring tool user interface\) Provide text search of the content](#)
- [A.3.6. \(For the authoring tool user interface\) Manage preference settings](#)
- [A.3.7. \(For the authoring tool user interface\) Ensure that previews are at least as accessible as in-market user agents](#)

[A.4. Editing-views are understandable](#)

- [A.4.1. \(For the authoring tool user interface\) Help authors avoid and correct mistakes](#)
- [A.4.2. \(For the authoring tool user interface\) Document the user interface, including all accessibility features](#)

B. Support the production of accessible content

B.1. Fully automatic processes produce accessible content

- B.1.1. Ensure that automatically-specified content is accessible
- B.1.2. Ensure that accessibility information is preserved

B.2. Authors are supported in producing accessible content

- B.2.1. Ensure that accessible content production is possible
- B.2.2. Guide authors to produce accessible content
- B.2.3. Assist authors with managing alternative content for non-text content
- B.2.4. Assist authors with accessible templates
- B.2.5. Assist authors with accessible pre-authored content

B.3. Authors are supported in improving the accessibility of existing content

- B.3.1. Assist authors in checking for accessibility problems
- B.3.2. Assist authors in repairing accessibility problems

B.4. Authoring tools promote and integrate their accessibility features

- B.4.1. Ensure the availability of features that support the production of accessible content
- B.4.2. Ensure that documentation promotes the production of accessible content

Implementing ATAG 2.0 Conformance

Conformance Requirements

- Success Criteria Satisfaction
- Relationship to the Web Content Accessibility Guidelines (WCAG) 2.0
- Conformance Options and Levels
- Web Content Technologies Produced
- Live Publishing Authoring Tools

Conformance Claims (Optional)

- Required Components of a Conformance Claim
- Optional Components of a Conformance Claim

Disclaimer

Appendix A: Glossary

Appendix B: References

Appendix C: Acknowledgments

Introduction

This section is informative.

This is a Working Draft of the Authoring Tool Accessibility Guidelines

(ATAG) version 2.0. This document includes recommendations for assisting authoring tool developers to make their authoring tools more accessible to people with disabilities, including auditory, cognitive, neurological, physical, speech, and visual disabilities.

Authoring tool accessibility addresses the needs of two overlapping user groups with disabilities:

- authors of web content, whose needs are met by ensuring that authoring tool user interfaces are more accessible (addressed by [Part A of the Guidelines](#)), and
- end users of web content, whose needs are met by ensuring that all authors are enabled, supported, and guided by the authoring tools that they use toward producing accessible web content (WCAG) (addressed by [Part B of the Guidelines](#)).

It is important to note that while the requirements for meeting these two sets of user needs are separated for clarity within the guidelines, the accelerating trend toward user-produced content means that, in reality, they are deeply inter-connected. For example, when a user participates in an online forum, he (or she) frequently author content that is then incorporated with other content authored by other users. Accessibility problems in either the authoring user interface or the content produced by the other forum users would reduce the overall accessibility of the forum.

Notes:

1. The term "authoring tools" has a specific definition in ATAG 2.0. The definition, which includes several normative notes, appears in the [Glossary](#).
2. The term "accessible content (WCAG)" and related terms, such as "accessible template (WCAG)" is used by ATAG 2.0 to refer to "content that would conform to [WCAG 2.0](#)", at either Level A, AA, or AAA, assuming that any web content technologies relied upon to satisfy the WCAG 2.0 success criteria are accessibility supported. The definition of the term reflects the WCAG 2.0 note that even content that conforms to the highest level of WCAG 2.0 (Level AAA) may not be "accessible to individuals with all types, degrees, or combinations of disability". For more information, see "[Relationship to the Web Content Accessibility Guidelines \(WCAG\) 2.0](#)".
3. ATAG 2.0 does not include standard usability recommendations, except where they have a significantly greater impact on people with disabilities than on other people.
4. Authoring tools are just one aspect of web accessibility. For an overview of the different components of web accessibility and how they work together see:
 - [Essential Components of Web Accessibility](#)
 - [Web Content Accessibility Guidelines \(WCAG\) Overview](#)

- [User Agent Accessibility Guidelines \(UAAG\) Overview](#) (This will be of special interest to developers of “Combined User Agent/Authoring Tools” and “User Agents with Authoring Tool Modes”)

ATAG 2.0 Layers of Guidance

The individuals and organizations that may use ATAG 2.0 vary widely and include authoring tool developers, authoring tool users (authors), authoring tool purchasers, and policy makers. In order to meet the varying needs of these audiences, several layers of guidance are provided:

- **Parts:** ATAG 2.0 is divided into two parts, each reflecting a key aspect of accessibility with respect to authoring tools. [Part A](#) relates to the accessibility of authoring tool user interfaces to authors with disabilities. [Part B](#) relates to support by authoring tools for the creation, by any author (not just those with disabilities), of web content that is more accessible to end users with disabilities. Both parts include normative “Conformance Applicability Notes” that apply to all of the success criteria within that part: [Part A Conformance Applicability Notes](#) and [Part B Conformance Applicability Notes](#).
- **Principles:** Under each part are several high-level principles that organize the guidelines.
- **Guidelines:** Under the principles are guidelines. The guidelines provide the basic goals that authoring tool developers should work toward in order to make authoring tools more accessible to both authors and end users of web content with different disabilities. The guidelines are not testable, but provide the framework and overall objectives to help authoring tool developers understand the success criteria. Each guideline includes a brief rationale for why the guideline was included.
- **Success Criteria:** For each guideline, testable success criteria are provided to allow ATAG 2.0 to be used where requirements and conformance testing are necessary, such as in design specification, purchasing, regulation, and contractual agreements. In order to meet the needs of different groups and different situations, multiple levels of full and partial conformance are defined (see [Levels of Conformance](#)).
- **Implementing ATAG 2.0 document:** The [Implementing ATAG 2.0](#) document provides additional non-normative information for each success criterion, including a description of the intent of the success criterion, examples, and links to related resources.

See [Authoring Tool Accessibility Guidelines \(ATAG\) Overview](#) for links to additional ATAG technical and educational material.

Levels of Conformance

In order to ensure that the process of using ATAG 2.0 and [WCAG 2.0](#) together in the development of [authoring tools](#) is as simple as possible, ATAG 2.0 shares [WCAG 2.0](#)'s three level conformance model: Level A (lowest), AA (middle), AAA (highest). For more information, see [Understanding Levels of Conformance](#).

Integration of Accessibility Features

When implementing ATAG 2.0, [authoring tool developers](#) should carefully integrate features that support more accessible authoring into the same "look-and-feel" as other features of the [authoring tool](#). Close integration has the potential to:

- produce a more seamless product;
- leverage the existing knowledge and skills of [authors](#);
- make authors more receptive to new accessibility-related authoring requirements; and
- reduce the likelihood of author confusion.

Guidelines

The success criteria and the conformance applicability notes in this section are [normative](#).

Part A: Make the authoring tool user interface accessible

Part A Conformance Applicability Notes:

1. Scope of "authoring tool user interface": The Part A success criteria apply to all aspects of the [authoring tool user interface](#) that are concerned with producing the ["included" web content technologies](#). This includes [views of the web content being edited](#) and features that are independent of the content being edited (e.g. menus, button bars, status bars, user preferences, [documentation](#)).
2. Reflected content accessibility problems: The [authoring tool](#) is responsible for ensuring that [editing-views display the web content being edited](#) in a way that is more accessible to [authors with disabilities](#) (e.g. ensuring that [text alternatives in the content](#) can be [programmatically determined](#)). However, where an [authoring tool user interface accessibility problem](#) is caused directly by the content being edited (e.g. if an image in the content lacks a text alternative), then this would not be considered a deficiency in the accessibility of the [authoring tool user interface](#).
3. Developer control: The Part A success criteria only apply to the [authoring tool user interface](#) as it is provided by the [developer](#). They do not apply to any subsequent modifications by parties other

- than the authoring tool developer (e.g. user modifications of default settings, third-party plug-ins).
4. User agent features: Web-based authoring tools may rely on user agent features (e.g. keyboard navigation, find functions, display preferences, undo features) to satisfy success criteria. Conformance claims are optional, but any claim that is made must record the user agent(s).
 5. Accessibility of features provided to meet Part A: The Part A success criteria apply to the entire authoring tool user interface, including any features added to meet the success criteria in Part A (e.g. documentation, search functions). The only exemption is for preview features, as long as they meet the relevant success criteria in Guideline A.3.7. Previews are treated differently than editing-views because all authors, including those with disabilities, benefit when preview features accurately reflect the functionality of user agents that are actually in use by end users.
 6. Unrecognizable content: When success criteria require authoring tools to treat web content according to semantic criteria, the success criteria only apply when these semantics are encoded programmatically (e.g. text describing an image can only be considered a text alternatives for non-text content when this role is encoded within markup).

Principle A.1: Authoring tool user interfaces follow applicable accessibility guidelines

Guideline A.1.1: (For the authoring tool user interface) Ensure that web-based functionality is accessible.
[[Implementing A.1.1](#)]

Rationale: When authoring tools (or parts of authoring tools) are web-based, conforming to WCAG 2.0 will facilitate access by all authors, including those using assistive technologies.

A.1.1.1 Web-Based Accessible (WCAG): If the authoring tool contains web-based user interfaces, then those web-based user interfaces meet the WCAG 2.0 success criteria. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

[Implementing A.1.1.1](#)

Guideline A.1.2: (For the authoring tool user interface) Ensure that non-web-based functionality is accessible.
[[Implementing A.1.2](#)]

Rationale: When authoring tools (or parts of authoring tools) are non-web-based, following existing platform accessibility guidelines and implementing communication with platform accessibility services facilitates access by all authors, including those using assistive technologies.

A.1.2.1 Accessibility Guidelines: If the authoring tool contains non-web-based user interfaces, then those non-web-based user interfaces follow user interface accessibility guidelines for the platform. (Level A)

Note: The (optional) [explanation of conformance claim results](#) should record the user interface accessibility guidelines that were followed.

[Implementing A.1.2.1](#)

A.1.2.2 Platform Accessibility Services: If the authoring tool contains non-web-based user interfaces, then those non-web-based user interfaces expose accessibility information through platform accessibility services. (Level A)

Note: The (optional) [explanation of conformance claim results](#) should record the platform accessibility service(s) that were implemented.

[Implementing A.1.2.2](#)

Principle A.2: Editing-views are perceivable

Guideline A.2.1: (For the authoring tool user interface) Make alternative content available to authors.

[[Implementing A.2.1](#)]

Rationale: Some authors require access to alternative content in order to interact with the web content that they are editing.

A.2.1.1 Text Alternatives for Rendered Non-Text Content: If an editing-view renders non-text content, then any programmatically associated text alternatives for the non-text content can be programmatically determined. (Level A)

[Implementing A.2.1.1](#)

A.2.1.2 Alternatives for Rendered Time-Based Media: If an editing-view renders time-based media, then at least one of the following is true: (Level A)

(a) Option to Render: The authoring tool provides the option to render alternatives for the time-based media; or

(b) User Agent Option: Authors have the option to preview the time-based media in a user agent that is able to render the alternatives.

[Implementing A.2.1.2](#)

Guideline A.2.2: (For the authoring tool user interface)

Ensure that editing-view presentation can be programmatically determined. [[Implementing A.2.2](#)]

Rationale: Some authors need access to details about the editing-view presentation, via their assistive technology, when that presentation is used to convey status messages (e.g. underlining misspelled words) or provide information about how the end user will experience the web content being edited.

A.2.2.1 Editing-View Status Indicators: If an editing-view adds status indicators to the content being edited, then the information being conveyed by the status indicators can be programmatically determined. (Level A)

Note: Status indicators may indicate errors (e.g. spelling errors), tracked changes, hidden elements, or other information.

[Implementing A.2.2.1](#)

A.2.2.2 Access to Rendered Text Properties: If an editing-view renders any text formatting properties that authors can also edit using the editing-view, then the properties can be programmatically determined. (Level AA)

[Implementing A.2.2.2](#)

Principle A.3: Editing-views are operable

Guideline A.3.1: (For the authoring tool user interface)
Provide keyboard access to authoring features.
[[Implementing A.3.1](#)]

Rationale: Some authors with limited mobility or visual disabilities do not use a mouse and instead require keyboard interface access to all of the functionality of the authoring tool.

A.3.1.1 Keyboard Access (Minimum): All functionality of the authoring tool is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints. (Level A)

Note 1: Keyboard interfaces are programmatic services provided by many platforms that allow operation in a device independent manner. This success criterion does not imply the presence of a hardware keyboard.

Note 2: The path exception relates to the underlying function, not the input technique. For example, if using handwriting to enter text, the input technique (handwriting) requires path-dependent input, but the underlying function (text input) does not. The path exception encompasses other input variables that are continuously sampled from

pointing devices, including pressure, speed, and angle.

Note 3: This success criterion does not forbid and should not discourage other input methods (e.g. mouse, touch) in addition to keyboard operation.

[Implementing A.3.1.1](#)

A.3.1.2 No Keyboard Traps: If keyboard focus can be moved to a component using a keyboard interface, then focus can be moved away from that component using only a keyboard interface. If it requires more than unmodified arrow or tab keys or other standard exit methods, authors are advised of the method for moving focus away. (Level A)

[Implementing A.3.1.2](#)

A.3.1.3 Efficient Keyboard Access: The authoring tool user interface includes mechanisms to make keyboard access more efficient than sequential keyboard access. (Level AA)

[Implementing A.3.1.3](#)

A.3.1.4 Keyboard Access (Enhanced): All functionality of the authoring tool is operable through a keyboard interface without requiring specific timings for individual keystrokes. (Level AAA)

[Implementing A.3.1.4](#)

A.3.1.5 Customize Keyboard Access: If the authoring tool includes keyboard commands, then those keyboard commands can be customized. (Level AAA)

[Implementing A.3.1.5](#)

A.3.1.6 Present Keyboard Commands: If the authoring tool includes keyboard commands, then the authoring tool provides a way for authors to determine the keyboard commands associated with authoring tool user interface components. (Level AAA)

[Implementing A.3.1.6](#)

Guideline A.3.2: (For the authoring tool user interface)
Provide authors with enough time. [[Implementing A.3.2](#)]

Rationale: Some authors who have difficulty typing, operating the mouse, or processing information can be prevented from using systems with short time limits or that require fast reaction speeds, such as clicking on a moving target.

A.3.2.1 Auto-Save (Minimum): The authoring tool does not include session time limits or the authoring tool can automatically save edits made before the session time limits are reached. (Level A)

[Implementing A.3.2.1](#)

A.3.2.2 Timing Adjustable: The authoring tool does not include time

limits or at least one of the following is true: (Level A)

- (a) Turn Off: Authors are allowed to turn off the time limit before encountering it; or
- (b) Adjust: Authors are allowed to adjust the time limit before encountering it over a wide range that is at least ten times the length of the default setting; or
- (c) Extend: Authors are warned before time expires and given at least 20 seconds to extend the time limit with a simple action (e.g. "press the space bar"), and authors are allowed to extend the time limit at least ten times; or
- (d) Real-time Exception: The time limit is a required part of a real-time event (e.g. a collaborative authoring system), and no alternative to the time limit is possible; or
- (e) Essential Exception: The time limit is essential and extending it would invalidate the activity; or
- (f) 20 Hour Exception: The time limit is longer than 20 hours.

[Implementing A.3.2.2](#)

A.3.2.3 Static Input Components: The authoring tool does not include moving user interface components that accept input where the movement of these components cannot be paused by authors. (Level A)

[Implementing A.3.2.3](#)

A.3.2.4 Content Edits Saved (Extended): The authoring tool can be set to automatically save web content edits made using the authoring tool. (Level AAA)

[Implementing A.3.2.4](#)

Guideline A.3.3: (For the authoring tool user interface) Help authors avoid flashing that could cause seizures.

[[Implementing A.3.3](#)]

Rationale: Flashing can cause seizures in authors with photosensitive seizure disorder.

A.3.3.1 Static View Option: If an editing-view can play visual time-based content, then playing is not necessarily automatic upon loading the content and playing can be paused. (Level A)

[Implementing A.3.3.1](#)

Guideline A.3.4: (For the authoring tool user interface) Enhance navigation and editing via content structure.

[[Implementing A.3.4](#)]

Rationale: Some authors who have difficulty typing or operating the mouse benefit when authoring tools make use of the structure present in

web content to simplify navigating and editing the content.

A.3.4.1 Navigate By Structure: If editing-views expose the markup elements in the web content being edited, then the markup elements (e.g. source code, content renderings) are selectable and navigation mechanisms are provided to move the selection focus between elements. (Level AA)

[Implementing A.3.4.1](#)

A.3.4.2 Navigate by Programmatic Relationships: If editing-views allow editing of programmatic relationships within web content, then mechanisms are provided that support navigation between the related content. (Level AAA)

Note: Depending on the web content technology and the nature of the authoring tool, relationships may include, but are not limited to, element nesting, headings, labeling, programmatic definitions, and ID relationships.

[Implementing A.3.4.2](#)

Guideline A.3.5: (For the authoring tool user interface)
Provide text search of the content. [[Implementing A.3.5](#)]

Rationale: Some authors who have difficulty typing or operating the mouse benefit from the ability to use text search to navigate to arbitrary points within the web content being edited.

A.3.5.1 Text Search: If the authoring tool provides an editing-view of text-based content, then the editing-view enables text search, such that all of the following are true: (Level AA)

- (a) All Editable Text: Any text content that is editable by the editing-view is searchable (including alternative content); and
- (b) Match: Matching results can be presented to authors and given focus; and
- (c) No Match: Authors are informed when no results are found; and
- (d) Two-way: The search can be made forwards or backwards.

[Implementing A.3.5.1](#)

Guideline A.3.6: (For the authoring tool user interface)
Manage preference settings. [[Implementing A.3.6](#)]

Rationale: Some authors need to set their own display settings in a way that differs from the presentation that they want to define for the published web content. Providing the ability to save and reload sets of keyboard and display preference settings benefits authors who have needs that differ over time (e.g. due to fatigue).

A.3.6.1 Independence of Display: If the authoring tool includes display settings for editing-views, then the authoring tool allows authors to

adjust these settings without modifying the web content being edited.
(Level A)

[Implementing A.3.6.1](#)

A.3.6.2 Save Settings: If the authoring tool includes display and/or control settings, then these settings can be saved between authoring sessions. (Level AA)

[Implementing A.3.6.2](#)

A.3.6.3 Apply Platform Settings: The authoring tool respects changes in platform display and control settings, unless authors select more specific display and control settings using the authoring tool. (Level AA)

[Implementing A.3.6.3](#)

Guideline A.3.7: (For the authoring tool user interface) Ensure that previews are at least as accessible as in-market user agents. [[Implementing A.3.7](#)]

Rationale: Preview features are provided by many authoring tools because the workflow of authors often includes periodically checking how user agents will display the web content to end users. Authors with disabilities need the same opportunity to check their work.

A.3.7.1 Preview (Minimum): If a preview is provided, then at least one of the following is true: (Level A)

- (a) In-Market User Agent: The preview renders content using a user agent that is in-market; or
- (b) UAAG (Level A): The preview conforms to the User Agent Accessibility Guidelines 1.0 Level A [[UAAG](#)].

[Implementing A.3.7.1](#)

A.3.7.2 Preview (Enhanced): If a preview is provided, then authors can specify which user agent performs the preview. (Level AAA)

[Implementing A.3.7.2](#)

Principle A.4: Editing-views are understandable

Guideline A.4.1: (For the authoring tool user interface) Help authors avoid and correct mistakes. [[Implementing A.4.1](#)]

Rationale: Some authors with disabilities may be more susceptible to input errors due to factors such as difficulty making fine movements

and speech recognition system errors.

A.4.1.1 Content Changes Reversible (Minimum): All authoring actions are either reversible or the authoring tool requires author confirmation to proceed. (Level A)

[Implementing A.4.1.1](#)

A.4.1.2 Settings Change Confirmation: If the authoring tool provides mechanisms for changing authoring tool user interface settings, then those mechanisms can reverse the setting changes, or the authoring tool requires author confirmation to proceed. (Level A)

[Implementing A.4.1.2](#)

A.4.1.3 Content Changes Reversible (Enhanced): Authors can sequentially reverse a series of reversible authoring actions. (Level AAA)

Note: It is acceptable to clear the authoring action history at the end of authoring sessions.

[Implementing A.4.1.3](#)

Guideline A.4.2: (For the authoring tool user interface) Document the user interface, including all accessibility features. [[Implementing A.4.2](#)]

Rationale: Some authors may not be able to understand or operate the authoring tool user interface without documentation.

A.4.2.1 Describe Accessibility Features: For each authoring tool feature that is used to meet [Part A](#) of ATAG 2.0, at least one of the following is true: (Level A)

- (a) Described in the Documentation: Use of the feature is explained in the authoring tool's documentation; or
- (b) Described in the Interface: Use of the feature is explained in the authoring tool user interface; or
- (c) Platform Service: The feature is a service provided by an underlying platform; or
- (d) Not Used by Authors: The feature is not used directly by authors (e.g. passing information to a platform accessibility service).

Note: The accessibility of the documentation is covered by [Guideline A.1.1](#) and [Guideline A.1.2](#).

[Implementing A.4.2.1](#)

A.4.2.2 Document All Features: For each authoring tool feature, at least one of the following is true: (Level AA)

- (a) Described in the Documentation: Use of the feature is explained in the authoring tool's documentation; or
- (b) Described in the Interface: Use of the feature is explained in the authoring tool user interface; or
- (c) Platform Service: The feature is a service provided by an

underlying platform; or

(d) Not Used by Authors: The feature is not used directly by authors (e.g. passing information to a platform accessibility service).

Note: The accessibility of the documentation is covered by [Guideline A.1.1](#) and [Guideline A.1.2](#).

[Implementing A.4.2.2](#)

Part B: Support the production of accessible content

Part B Conformance Applicability Notes:

1. Author availability: Any Part B success criteria that refer to authors only apply during authoring sessions.
2. Developer control: The Part B success criteria only apply to the authoring tool as it is provided by the developer. This does not include subsequent modifications by parties other than the authoring tool developer (e.g. third-party plug-ins, user-defined templates, user modifications of default settings).
3. Applicability after the end of an authoring session: Authoring tools are responsible for the web content accessibility (WCAG) of web content that they automatically generate after the end of an author's authoring session (see [Success Criterion B.1.1.1](#)). For example, if the developer changes the site-wide templates of a content management system, these would be required to meet the accessibility requirements for automatically-generated content. Authoring tools are not responsible for changes to the accessibility of content that the author causes, whether it is author-generated or automatically-generated by another system that the author has specified (e.g. a third-party feed).
4. Authoring systems: As per the ATAG 2.0 definition of authoring tool, several software tools (identified in any [conformance claim](#)) can be used in conjunction to meet the requirements of Part B (e.g. an authoring tool could make use of a third-party software accessibility checking tool).
5. Accessibility of features provided to meet Part B: The [Part A](#) success criteria apply to the entire authoring tool user interface, including any features that must be present to meet the success criteria in Part B (e.g. checking tools, repair tools, tutorials, documentation).
6. Multiple authoring roles: Some authoring tools include multiple author roles, each with different views and content editing permissions (e.g. a content management system may separate the roles of designers, content authors, and quality assurers). In these cases, the Part B success criteria apply to the authoring tool as a whole, not to the view provided to any particular authoring role. Accessible content support features should be made available to any authoring role where it would be useful.
7. Unrecognizable content: When success criteria require authoring

tools to treat web content according to semantic criteria, the success criteria only apply when these semantics are encoded programmatically (e.g. text describing an image can only be considered a text alternatives for non-text content when this role is encoded within markup).

Principle B.1: Fully automatic processes produce accessible content

Guideline B.1.1: Ensure that automatically-specified content is accessible. [[Implementing B.1.1](#)]

Rationale: If authoring tools automatically produce web content that includes accessibility problems (WCAG), then this will impose additional repair tasks on authors.

B.1.1.1 Content Auto-Generation After Authoring Sessions (WCAG): The authoring tool does not automatically generate web content after the end of an authoring session or authors can specify that the content be accessible web content (WCAG). (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

Note: This success criterion applies only to automatic processes specified by the authoring tool developer. It does not apply when author actions prevent generation of accessible web content (WCAG).

[Implementing B.1.1.1](#)

B.1.1.2 Content Auto-Generation During Authoring Sessions (WCAG): If the authoring tool provides the functionality for automatically generating web content during an authoring session, then at least one of the following is true: (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

- (a) Accessible: The content is accessible web content (WCAG) without author input; or
- (b) Prompting: During the automatic generation process, authors are prompted for any required accessibility information (WCAG); or
- (c) Automatic Checking: After the automatic generation process, accessibility checking is automatically performed; or
- (d) Checking Suggested: After the automatic generation process, the authoring tool prompts authors to perform accessibility checking.

Note 1: Automatic generation includes automatically selecting templates for authors.

Note 2: This success criterion applies only to automatic processes specified by the authoring tool developer. It does not apply when author actions prevent generation of accessible web content (WCAG).

[Implementing B.1.1.2](#)

Guideline B.1.2: Ensure that accessibility information is preserved. [[Implementing B.1.2](#)]

Rationale: Accessibility information (WCAG) is critical to maintaining comparable levels of web content accessibility (WCAG) between the input and output of web content transformations.

B.1.2.1 Restructuring and Recoding Transformations (WCAG): If the authoring tool provides restructuring transformations or re-coding transformations, and if equivalent mechanisms exist in the web content technology of the output, then at least one of the following is true: (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

- (a) Preserve: Accessibility information (WCAG) is preserved in the output; or
- (b) Warning: Authors have the default option to be warned that accessibility information (WCAG) may be lost (e.g. when saving a vector graphic into a raster image format); or
- (c) Automatic Checking: After the transformation, accessibility checking is automatically performed; or
- (d) Checking Suggested: After the transformation, the authoring tool prompts authors to perform accessibility checking.

Note 1: For text alternatives for non-text content, see [Success Criterion B.1.2.4](#).

Note 2: This success criteria only applies when the output technology is "included" for conformance.

[Implementing B.1.2.1](#)

B.1.2.2 Copy-Paste Inside Authoring Tool (WCAG): If the authoring tool supports copy and paste of structured content, then any accessibility information (WCAG) in the copied content is preserved when the authoring tool is both the source and destination of the copy-paste and the source and destination use the same web content technology. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

[Implementing B.1.2.2](#)

B.1.2.3 Optimizations Preserve Accessibility: If the authoring tool provides optimizing web content transformations, then any accessibility information (WCAG) in the input is preserved in the output. (Level A).

[Implementing B.1.2.3](#)

B.1.2.4 Text Alternatives for Non-Text Content are Preserved: If the authoring tool provides web content transformations that preserve non-text content in the output, then any text alternatives for that non-text content are also preserved, if equivalent mechanisms exist in the web

content technology of the output. (Level A).

Note: This success criterion only applies when the output technology is ["included"](#) for conformance.

[Implementing B.1.2.4](#)

Principle B.2: Authors are supported in producing accessible content

Guideline B.2.1: Ensure that accessible content production is possible. [[Implementing B.2.1](#)]

Rationale: To support accessible web content (WCAG) production, at minimum, it is possible to produce web content that conforms with [WCAG 2.0](#) using the authoring tool.

B.2.1.1 Accessible Content Possible (WCAG): The authoring tool does not place restrictions on the web content that authors can specify or those restrictions do not prevent [WCAG 2.0](#) success criteria from being met. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

[Implementing B.2.1.1](#)

Guideline B.2.2: Guide authors to produce accessible content. [[Implementing B.2.2](#)]

Rationale: By guiding authors from the outset toward the creation and maintenance of accessible web content (WCAG), web content accessibility problems (WCAG) are mitigated and less repair effort is required.

B.2.2.1 Accessible Option Prominence (WCAG): If authors are provided with a choice of authoring actions for achieving the same authoring outcome (e.g. styling text), then options that will result in accessible web content (WCAG) are at least as prominent as options that will not. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

[Implementing B.2.2.1](#)

B.2.2.2 Setting Accessibility Properties (WCAG): If the authoring tool provides mechanisms to set web content properties (e.g. attribute values), then mechanisms are also provided to set web content properties related to accessibility information (WCAG). (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

Note: For the prominence of the mechanisms, see [Success Criterion B.4.1.4](#).

[Implementing B.2.2.2](#)

Guideline B.2.3: Assist authors with managing alternative content for non-text content. [[Implementing B.2.3](#)]

Rationale: Improperly generated alternative content can create web content accessibility problems (WCAG) and interfere with accessibility checking.

Note: This guideline only applies when non-text content is specified by authors (e.g. inserting an image). When non-text content is automatically added by the authoring tool, see [Guideline B.1.1](#).

B.2.3.1 Alternative Content is Editable (WCAG): If the authoring tool provides functionality for adding non-text content, then authors are able to modify programmatically associated text alternatives for non-text content. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

Note: An exception can be made when the non-text content is known to be decoration, formatting, invisible or a CAPTCHA.

[Implementing B.2.3.1](#)

B.2.3.2 Automating Repair of Text Alternatives: The authoring tool does not attempt to repair text alternatives for non-text content or the following are all true: (Level A)

- (a) No Generic or Irrelevant Strings: Generic strings (e.g. "image") and irrelevant strings (e.g. the file name, file format) are not used as text alternatives; and
- (b) In-Session Repairs: If the repair attempt occurs during an authoring session, authors have the opportunity to accept, modify, or reject the repair attempt prior to insertion of the text alternative into the content; and
- (c) Out-of-Session Repairs: If the repair attempt occurs after an authoring session has ended, the repaired text alternatives are indicated during subsequent authoring sessions (if any) and authors have the opportunity to accept, modify, or reject the repair strings prior to insertion in the content.

[Implementing B.2.3.2](#)

B.2.3.3 Save for Reuse: If the authoring tool provides the functionality for adding non-text content, when authors enter programmatically associated text alternatives for non-text content, then both of the following are true: (Level AAA)

- (a) Save and Suggest: The text alternatives are automatically saved and suggested by the authoring tool, if the same non-text content is reused; and
- (b) Edit Option: The author has the option to edit or delete the

saved text alternatives.

[Implementing B.2.3.3](#)

Guideline B.2.4: Assist authors with accessible templates. [[Implementing B.2.4](#)]

Rationale: Providing accessible templates (WCAG) can have several benefits, including: immediately improving the accessibility of the web content (WCAG) of being edited, reducing the effort required of authors, and demonstrating the importance of accessible web content (WCAG).

B.2.4.1 Accessible Template Options (WCAG): If the authoring tool provides templates, then there are accessible template (WCAG) options for a range of template uses. (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

[Implementing B.2.4.1](#)

B.2.4.2 Identify Template Accessibility: If the authoring tool includes a template selection mechanism and provides any non-accessible template (WCAG) options, then the template selection mechanism can display distinctions between the accessible and non-accessible options. (Level AA)

Note: The distinction can involve providing information for the accessible templates, the non-accessible templates or both.

[Implementing B.2.4.2](#)

B.2.4.3 Author-Created Templates: If the authoring tool includes a template selection mechanism and allows authors to create new non-accessible templates (WCAG), then authors can enable the template selection mechanism to display distinctions between accessible and non-accessible templates that they create. (Level AA)

Note: The distinction can involve providing information for the accessible templates (WCAG), the non-accessible templates or both.

[Implementing B.2.4.3](#)

B.2.4.4 Accessible Template Options (Enhanced): If the authoring tool provides templates, then all of the templates are accessible template (to WCAG Level AA). (Level AAA)

[Implementing B.2.4.4](#)

Guideline B.2.5: Assist authors with accessible pre-authored content. [[Implementing B.2.5](#)]

Rationale: Providing accessible pre-authored content (WCAG) (e.g. clip art, synchronized media, widgets) can have several benefits, including:

immediately improving the accessibility of web content (WCAG) being edited, reducing the effort required of authors, and demonstrating the importance of accessibility.

B.2.5.1 Accessible Pre-Authored Content Options: If the authoring tool provides pre-authored content, then a range of accessible pre-authored content (to WCAG Level AA) options are provided. (Level AA)

[Implementing B.2.5.1](#)

B.2.5.2 Identify Pre-Authored Content Accessibility: If the authoring tool includes a pre-authored content selection mechanism and provides any non-accessible pre-authored content (WCAG Level AA) options, then the selection mechanism can display distinctions between the accessible and non-accessible options. (Level AA)

Note: The distinction can involve providing information for the accessible pre-authored content, the non-accessible pre-authored content or both.

[Implementing B.2.5.2](#)

Principle B.3: Authors are supported in improving the accessibility of existing content

Guideline B.3.1: Assist authors in checking for accessibility problems. [[Implementing B.3.1](#)]

Rationale: When accessibility checking is an integrated function of the authoring tool, it helps make authors aware of web content accessibility problems (WCAG) during the authoring process, so they can be immediately addressed.

B.3.1.1 Checking Assistance (WCAG): If the authoring tool provides authors with the ability to add or modify web content in such a way that a WCAG 2.0 success criterion can be violated, then accessibility checking for that success criterion is provided (e.g. an HTML authoring tool that inserts images should check for alternative text; a video authoring tool with the ability to edit text tracks should check for captions). (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

Note: Automated and semi-automated checking is possible (and encouraged) for many types of web content accessibility problems (WCAG). However, manual checking is the minimum requirement to meet this success criterion. In manual checking, the authoring tool provides authors with instructions for detecting problems, which authors carry out by themselves. For more information on checking, see [Implementing ATAG 2.0 – Appendix B: Levels of Checking Automation](#).

[Implementing B.3.1.1](#)

B.3.1.2 Help Authors Decide: If the authoring tool provides accessibility checking that relies on authors to decide whether potential web content accessibility problems (WCAG) are correctly identified (i.e. manual checking and semi-automated checking), then the accessibility checking process provides instructions that describe how to decide. (Level A)

[Implementing B.3.1.2](#)

B.3.1.3 Help Authors Locate: If the authoring tool provides checks that require authors to decide whether a potential web content accessibility problem (WCAG) is correctly identified (i.e. manual checking and semi-automated checking), then the relevant content is identified to the authors. (Level A)

Note: Depending on the nature of the editing-view and the scope of the potential web content accessibility problem (WCAG), identification might involve highlighting elements or renderings of elements, displaying line numbers, or providing instructions.

[Implementing B.3.1.3](#)

B.3.1.4 Status Report: If the authoring tool provides checks, then authors can receive an accessibility status report based on the results of the accessibility checks. (Level AA)

Note: The format of the accessibility status report is not specified and they might include a listing of problems detected or a [WCAG 2.0](#) conformance level, etc.

[Implementing B.3.1.4](#)

B.3.1.5 Programmatic Association of Results: If the authoring tool provides checks, then the authoring tool can programmatically associate accessibility checking results with the web content that was checked. (Level AA)

[Implementing B.3.1.5](#)

Guideline B.3.2: Assist authors in repairing accessibility problems. [[Implementing B.2.3](#)]

Rationale: When repair is an integral part of the authoring process, it greatly enhances the utility of checking and increases the likelihood that web content accessibility problems (WCAG) will be properly addressed.

B.3.2.1 Repair Assistance (WCAG): If checking (see [Success Criterion B.3.1.1](#)) can detect that a [WCAG 2.0](#) success criterion is not met, then repair suggestion(s) are provided: (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

Note: Automated and semi-automated repair is possible (and encouraged)

for many types of web content accessibility problems (WCAG). However, manual repair is the minimum requirement to meet this success criterion. In manual repair, the authoring tool provides authors with instructions for repairing problems, which authors carry out by themselves. For more information on repair, see [Implementing ATAG 2.0 – Appendix C: Levels of Repair Automation](#).

[Implementing B.3.2.1](#)

Principle B.4: Authoring tools promote and integrate their accessibility features

Guideline B.4.1: Ensure the availability of features that support the production of accessible content. [[Implementing B.4.1](#)]

Rationale: The accessible content support features will be more likely to be used, if they are turned on and are afforded reasonable prominence within the authoring tool user interface.

B.4.1.1 Features Active by Default: All accessible content support features are turned on by default. (Level A)

[Implementing B.4.1.1](#)

B.4.1.2 Option to Reactivate Features: The authoring tool does not include the option to turn off its accessible content support features or features which have been turned off can be turned back on. (Level A)

[Implementing B.4.1.2](#)

B.4.1.3 Feature Deactivation Warning: The authoring tool does not include the option to turn off its accessible content support features or, if these features can be turned off, authors are informed that this may increase the risk of content accessibility problems (WCAG). (Level AA)

[Implementing B.4.1.3](#)

B.4.1.4 Feature Prominence: All accessible content support features are at least as prominent as features related to either invalid markup, syntax errors, spelling errors or grammar errors. (Level AA)

[Implementing B.4.1.4](#)

Guideline B.4.2: Ensure that documentation promotes the production of accessible content. [[Implementing B.4.2](#)]

Rationale: Some authors need support in determining how to use accessible content production features (e.g. how to respond to prompts

for text alternatives, how to use accessibility checking tools). Demonstrating accessible authoring as routine practice, or at least not demonstrating inaccessible practices, will help to encourage acceptance of accessibility by some authors.

B.4.2.1 Model Practice (WCAG): A range of examples in the documentation (e.g. markup, screen shots of WYSIWYG editing-views) demonstrate accessible authoring practices (WCAG). (Level A to meet WCAG 2.0 Level A success criteria; Level AA to meet WCAG 2.0 Level A and AA success criteria; Level AAA to meet all WCAG 2.0 success criteria)

[Implementing B.4.2.1](#)

B.4.2.2 Feature Instructions: Instructions for using any accessible content support features appear in the documentation. (Level A)

[Implementing B.4.2.2](#)

B.4.2.3 Tutorial: The authoring tool provides a tutorial for an accessible authoring process that is specific to that authoring tool. (Level AAA)

[Implementing B.4.2.3](#)

B.4.2.4 Instruction Index: The authoring tool documentation contains an index to the instructions for using any accessible content support features. (Level AAA)

[Implementing B.4.2.4](#)

Conformance

This section is normative.

Conformance means that the authoring tool satisfies the applicable success criteria defined in the guidelines section. This conformance section describes conformance and lists the conformance requirements.

Conformance Requirements

Success Criteria Satisfaction

The first step in determining ATAG 2.0 conformance is to assess whether the Success Criteria have been satisfied. The potential answers are:

- Not Applicable: The ATAG 2.0 [definition of authoring tool](#) is inclusive and, as such, it covers software with a wide range of capabilities and contexts of operation. In order to take into account authoring tools with limited feature sets (e.g. a photo editor, a CSS editor, a status update field in a social networking application), many of the ATAG 2.0 success criteria are conditional, applying only to authoring tools with the given features(s). If a

conformance claim is made, an explanation of why the success criterion is not applicable is required.

- Yes: In the case of some success criteria, this will include a [Level](#) (A, AA, or AAA) with reference to WCAG 2.0. If a [conformance claim](#) is made, an explanation is optional, yet strongly recommended.
- No: If a conformance claim is made, an explanation is optional, yet strongly recommended.

Relationship to the Web Content Accessibility Guidelines (WCAG) 2.0

At the time of publishing, WCAG 2.0 [[WCAG20](#)] is the current W3C Recommendation regarding web content accessibility. For this reason, ATAG 2.0 refers to WCAG 2.0 when setting requirements for (1) the accessibility of web-based authoring tool user interfaces (in [Part A](#)) and (2) how authors should be enabled, supported, and guided toward producing web content that is more accessible to end users with disabilities (in [Part B](#)).

In particular, ATAG 2.0 refers to WCAG 2.0 within its definition of the term "accessible content" (and related terms, such as "accessible template"). The definition of "accessible content" is content that would conform to WCAG 2.0, at either Level A, AA, or AAA, under the assumption that any web content technologies that are relied upon to satisfy the WCAG 2.0 success criteria are accessibility supported. The phrase "at either Level A, AA, or AAA" takes into account that the definition of "accessible content" can differ depending on the context of use (e.g. in a Level A success criterion of ATAG 2.0 versus in a Level AAA success criterion). The definition also includes two notes:

- The first is "If accessibility support for the relied upon technologies is lacking, then the content will not conform to WCAG 2.0 and one or more groups of end-users with disabilities will likely experience difficulty accessing the content."
- The second is "Conformance to WCAG 2.0, even at the highest level (i.e. Level AAA), still may not make content 'accessible to individuals with all types, degrees, or combinations of disability'." In order to highlight success criteria or defined terms in ATAG 2.0 that depend on WCAG 2.0, they are marked with the parenthetical: "(WCAG)".

Note on "accessibility-supported ways of using technologies":

Part of conformance to WCAG 2.0 is the requirement that "only accessibility-supported ways of using technologies are relied upon to satisfy the WCAG 2.0 success criteria. Any information or functionality that is provided in a way that is not accessibility supported is also

available in a way that is accessibility supported.” In broad terms, WCAG 2.0 considers a web content technology to be “accessibility supported” when (1) the way that the web content technology is used is supported by users’ assistive technology and (2) the web content technology has accessibility-supported user agents that are available to end users.

This concept is not easily extended to authoring tools because many authoring tools can be installed and used in a variety of environments with differing availabilities for assistive technologies and user agents (e.g. private intranets versus public websites, monolingual sites versus multilingual sites). Therefore:

ATAG 2.0 does not include the accessibility-supported requirement. As a result, ATAG 2.0 success criteria do not refer to WCAG 2.0 “conformance”, and instead refer to “meeting WCAG 2.0 success criteria”.

Once an authoring tool has been installed and put into use, it would be possible to assess the WCAG 2.0 conformance of the web content that the authoring tool produces, including whether the WCAG 2.0 accessibility-supported requirement is met. However, this WCAG 2.0 conformance assessment would be completely independent of the authoring tool’s conformance with ATAG 2.0.

Conformance Options and Levels

There are two types of conformance, each with three levels:

ATAG 2.0 Conformance (Level A, AA, or AAA)

This conformance option may be selected when an authoring tool can be used to produce accessible web content (WCAG) without additional tools or components. The level of conformance is determined as follows:

- Level A: The authoring tool satisfies all of the [applicable](#) Level A success criteria.
- Level AA: The authoring tool satisfies all of the applicable Level A and Level AA success criteria.
- Level AAA: The authoring tool satisfies all of the applicable success criteria.

Note 1: The [Part A Conformance Applicability Notes](#) and [Part B Conformance Applicability Notes](#) must be applied.

Note 2: If the minimum conformance level (Level A) has not been achieved (i.e. at least one applicable Level A success criterion has not been met), it is still beneficial to publish a statement specifying which success criteria were met.

Partial ATAG 2.0 Conformance – Process Component (Level A, AA, or AAA)

This conformance option may be selected when an authoring tool would require additional tools or components in order to conform as a complete authoring system. This option may be used for components with very limited functionality (e.g. a plug-in) up to nearly complete systems (e.g. a markup editor that only lacks accessibility checking functionality).

The level of conformance (A, AA, or AAA) is determined as above except that, for any "no" answers, the tool must not prevent the success criteria from being met by another authoring process component as part of a complete authoring system.

Note 1: Authoring tools would not be able to meet partial conformance if they prevent additional authoring process components from meeting the failed success criteria (e.g. for security reasons).

Note 2: The [Part A Conformance Applicability Notes](#) and [Part B Conformance Applicability Notes](#) must be applied.

Partial ATAG 2.0 Conformance – Platform Limitations (Level A, AA, or AAA)

This conformance option may be selected when an authoring tool is unable to meet one or more success criteria because of intrinsic limitations of the platform (e.g. lacking a platform accessibility service). The (optional) explanation of conformance claim results should explain what platform features are missing.

Web Content Technologies Produced:

Authoring tools conform to ATAG 2.0 with respect to the production of specific web content technologies (e.g. Level A Conformance with respect to the production of XHTML 1.0).

If an authoring tool is capable of producing multiple web content technologies, then the conformance may include only a subset of these technologies as long as the subset includes both any technologies that the developer sets for automatically-generated content or that the developer sets as the default for author-generated content. The subset may include "interim" formats that are not intended for publishing to end users, though this is not required.

Live publishing authoring tools:

ATAG 2.0 may be applied to authoring tools with workflows that involve live authoring of web content (e.g. some collaborative tools). Due to the challenges inherent in real-time publishing, conformance to [Part B](#) of ATAG 2.0 for these authoring tools may involve some combination of support before (e.g. support in preparing accessible slides), during (e.g. live captioning as WCAG 2.0 requires at Level AA) and after the

live authoring session (e.g. the ability to add a transcript to the archive of a presentation that was initially published in real-time). For more information, see [Implementing ATAG 2.0 – Appendix E: Authoring Tools for Live Web Content](#).

Conformance Claims (Optional)

Note: As with any software application, authoring tools can be collections of components. A conformance claim can only be made by a responsible entity. Any other attempted “claims” are, in fact, reviews.

Required Components of a Conformance Claim

1. Date of the claim.
2. ATAG 2.0 version and URI
3. [Conformance level](#) satisfied.
4. Authoring tool information: The name of the authoring tool and sufficient additional information to specify the version (e.g. vendor name, version number (or version range), required patches or updates, human language of the user interface or documentation).
 - Note: If the authoring tool is a collection of software applications (e.g. a markup editor, an image editor, and a validation tool), then information must be provided separately for each application, although the conformance claim will treat them as a whole.
5. Platform(s): The platform(s) upon which the authoring tool operates:
 - For user agent platform(s) (used to evaluate web-based authoring tool user interfaces): provide the name and version information of the user agent(s).
 - For platforms that are not user agents (used to evaluate non-web-based authoring tool user interfaces): provide the name and version information of the platform(s) (e.g. desktop operating system, mobile operating system, cross-OS environment) and the name and version of the platform accessibility service(s) employed.
6. A list of the web content technologies produced by the authoring tool that are included in the claim. If there are any web content technologies produced by the authoring tool that are not included in the conformance claim, these must be listed separately. If the authoring tool produces any web content technologies by default, then these must be included.
7. Results for each of the success criteria: Yes, No, [Not Applicable](#)

Optional Components of a Conformance Claim

In addition to the required components of a conformance claim above,

consider providing additional information to assist authors. Recommended additional information includes:

1. An explanation of the success criteria results (Yes, No). (strongly recommended)
2. Information about how the web content technologies produced can be used to create accessible web content (e.g. links to technology-specific WCAG 2.0 techniques).
3. Information about any additional steps taken that go beyond the success criteria to enhance accessibility.
4. A machine-readable metadata version of the conformance claim.
5. A description of the authoring tool that identifies the types of editing-views that it includes.

Disclaimer

Neither W3C, WAI, nor AUWG take any responsibility for any aspect or result of any ATAG 2.0 [conformance claim](#) that has not been published under the authority of the W3C, WAI, or AUWG.

Appendix A: Glossary

This section is [normative](#).

This appendix contains definitions for all of the significant/important/unfamiliar terms used in the normative parts of this standard, including terms used in the [Conformance](#) section. Please consult <http://www.w3.org/TR/qaframe-spec/> for more information on the role of definitions in standards quality.

accessibility problems

ATAG 2.0 recognizes two types of accessibility problems:

- authoring tool user interface accessibility problems: Aspects of an [authoring tool user interface](#) that does not meet a success criterion in [Part A](#) of ATAG 2.0.
- web content accessibility problems (WCAG): Aspects of [web content](#) that does not meet a [WCAG 2.0](#) success criterion (Level A, AA or AAA).

accessibility information (WCAG)

Information that [web content](#) must contain in order to meet a [WCAG 2.0](#) success criterion (Level A, AA or AAA). Examples include: [programmatically associated alternative content](#) (e.g. text alternatives for images), [role](#), and state information for widgets, [relationships within complex tables](#)).

Note: For the purposes of ATAG 2.0, only [programmatically determinable](#) accessibility information qualifies. For additional

examples, see [Appendix A of the Implementing ATAG 2.0 document](#).
 accessible content support features

Any features of an authoring tool that directly support authors in increasing the accessibility of the web content being edited. These are features that must be present to meet the success criteria in [Part B](#) of ATAG 2.0.

alternative content

Web content that is used in place of other content that some people are not able to access. Alternative content fulfills essentially the same function or purpose as the original content. [WCAG 2.0](#) recognizes several general types of alternative content:

- text alternatives for non-text content: Text that is programmatically associated with non-text content or referred to from text that is programmatically associated with non-text content. For example, an image of a chart might have two text alternatives: a description in the paragraph after the chart and a short text alternative for the chart indicating in words that a description follows.
- alternatives for time-based media: Web content that serves the same function or purpose as one or more tracks in a time-based media presentation. This includes: captions, audio descriptions, extended audio descriptions, sign language interpretation as well as correctly sequenced text descriptions of time-based visual and auditory information that also is capable of achieving the outcomes of any interactivity in the time-based presentation.
- media alternative for text: Media that presents no more information than is already presented in text (directly or via text alternatives). A media alternative for text is provided for people who benefit from alternate representations of text. Media alternatives for text may be audio-only, video-only (including sign-language video), or audio-video.

Importantly, from the perspective of authoring tools, alternative content may or may not be:

- programmatically associated alternative content: Alternative content whose location and purpose can be programmatically determined from the original content for which it is serving as an alternative. For example, a paragraph might serve as a text alternative for an image, but it is only programmatically associated if this relationship is properly encoded (e.g. by "aria-labeledby").
 Note: ATAG 2.0 typically refers to programmatically associated alternative content.

assistive technology

Software (or hardware), separate from the authoring tool, that

provides functionality to meet the requirements of people with disabilities (authors and end users). Some authoring tools may also provide direct accessibility features. Examples include:

- screen magnifiers, and other visual reading assistants, which are used by people with visual, perceptual, and physical print disabilities to change text font, size, spacing, color, synchronization with speech, etc. in order improve the visual readability of rendered text and images;
- screen readers, which are used by people who are blind to read textual information through synthesized speech or Braille;
- text-to-speech software, which is used by some people with cognitive, language, and learning disabilities to convert text into synthetic speech;
- speech recognition software, which are used by some people who have some physical disabilities;
- alternative keyboards, which are used by some people with physical disabilities to simulate the keyboard (including alternate keyboards that use head pointers, single switches, sip/puff, and other special input devices);
- alternative pointing devices, which are used by some people with physical disabilities to simulate mouse pointing and button activations.

audio

The technology of sound reproduction. Audio can be created synthetically (including speech synthesis), recorded from real-world sounds, or both.

author actions preventing generation of accessible web content

When the actions of authors prevent authoring tools from generating accessible web content (WCAG). Examples include: turning off accessible content support features, ignoring prompts for accessibility information (WCAG), providing faulty accessibility information (WCAG) at prompts, modifying the authoring tool (e.g. via scripting, macros), and installing plug-ins.

authors

People who use authoring tools to create or modify web content. The term may cover roles such as content authors, designers, programmers, publishers, testers, etc. (see [Part B Conformance Applicability Note 6: Multiple authoring roles](#)). Some authoring tools control who may be an author by managing author permissions.

author permission

Authorization that allows modification of given web content.

authoring action

Any action that authors can take using the authoring tool user interface that results in editing web content (e.g. typing text, deleting, inserting an element, applying a template). In contrast, most authoring tool user interfaces also enable actions that do not edit content (e.g. saving, publishing, setting preferences, viewing

documentation).

- reversible authoring action: An authoring action that can be immediately and completely undone by the authoring tool upon a cancel request by an author. Examples of cancel requests include: "cancel", "undo", "redo" (when it used to reverse "undo"), "revert", and "roll-back"

Note: It is acceptable for an authoring tool to collect a series of text entry actions (e.g. typed words, a series of backspaces) into a single reversible authoring action.

authoring outcome

The content or content modifications that result from authoring actions. Authoring outcomes are cumulative (e.g. text is entered, then styled, then made into a link, then given a title).

authoring practice

An approach that authors follow to achieve a given authoring outcome (e.g. controlling presentation with style sheets). Depending on the design of an authoring tool, authoring practices may be chosen by authors or by the authoring tool. Authoring practices may or may not be:

- accessible authoring practices (WCAG): An authoring practice in which the authoring outcome conforms to [WCAG 2.0](#) at Level A, AA, or AAA. Some accessible authoring practices require accessibility information (WCAG).

authoring session

A state of the authoring tool in which web content can be edited by an author.

- end of an authoring session: The point at which the author has no further opportunity to make authoring actions without starting another session. The end of an authoring session may be determined by authors (e.g. closing a document, publishing) or by the authoring tool (e.g. when the authoring tool transfers editing permission to another author on a collaborative system).

Note: The end of the authoring session is distinct from publishing. Automatic content generation may continue after the end of both the authoring session and initial publishing (e.g. content management system updates).

authoring tool

Any web-based or non-web-based application(s) that can be used by authors (alone or collaboratively) to create or modify web content for use by other people (other authors or end users).

Note 1: "application(s)": ATAG 2.0 may be conformed to by stand-alone applications or by collections of applications. If a conformance claim is made, then the claim must provide identifying

information for each application and also for any required extensions, plug-ins, etc.

Note 2: "alone or collaboratively": Multiple authors may contribute to the creation of web content and, depending on the authoring tool, each author may work with different views of the content and different author permissions.

Note 3: "to create or modify web content": This clause rules out software that collects data from a person for other purposes (e.g. online grocery order form) and then creates web content from that data (e.g. a web-based warehouse order) without informing the person (however, [WCAG 2.0](#) would still apply). This clause also rules out software used to create content exclusively in non-web content technologies.

Note 4: "for use by other people": This clause rules out the many web applications that allow people to modify web content that only they themselves experience (e.g. web-based email display settings) or that only provide input to automated processes (e.g. library catalog search page).

Examples of software that are generally considered authoring tools under ATAG 2.0:

- web page authoring tools (e.g. WYSIWYG HTML editors)
- software for directly editing source code
- software for converting to web content technologies (e.g. "Save as HTML" features in office document applications)
- integrated development environments (e.g. for web application development)
- software that generates web content on the basis of templates, scripts, command-line input or "wizard"-type processes
- software for rapidly updating portions of web pages (e.g. blogging, wikis, online forums)
- software for generating/managing entire websites (e.g. content management systems, courseware tools, content aggregators)
- email clients that send messages using web content technologies
- multimedia authoring tools
- software for creating mobile web applications

Examples of software that are not considered authoring tools under ATAG 2.0 (in all cases, WCAG 2.0 still applies if the software is web-based):

- customizable personal portals: ATAG 2.0 does not apply because the web content being edited is only available to the owner of the portal
- e-commerce order forms: ATAG 2.0 does not apply because the purpose of an e-commerce order form is to order a product, not communicate with other people via web content, even if the data collected by the form actually does result in web content (e.g. online tracking pages)

- stand-alone accessibility checkers: ATAG 2.0 does not apply because a stand-alone accessibility checker with no automated or semi-automated repair functionality does not actually modify web content. An accessibility checker with repair functionality or that is considered as part of a larger authoring process would be considered an authoring tool.

authoring tool user interface

The display and control mechanism that authors use to operate the authoring tool software. User interfaces may be non-web-based or web-based or a combination (e.g. a non-web-based authoring tool might have web-based help pages):

- authoring tool user interface (non-web-based): Any parts of an authoring tool user interface that are not implemented as web content and instead run directly on a platform that is not a user agent (e.g. Windows, Mac OS, Java Virtual Machine, iOS, Android).
- authoring tool user interface (web-based): Any parts of an authoring tool user interface that are implemented using web content technologies and are accessed by authors via a user agent.

Authoring tool user interfaces may or may not be:

- accessible authoring tool user interfaces: Authoring tool user interfaces that meet the success criteria of a level in [Part A](#) of ATAG 2.0.

checking, accessibility

The process by which web content is evaluated for web content accessibility problems (WCAG). ATAG 2.0 recognizes three types of checking, based on increasing levels of automation of the tests:

- manual checking: Checking in which the tests are carried out by authors. This includes the case where authors are aided by instructions or guidance provided by the authoring tool, but where authors must carry out the actual test procedure.
- semi-automated checking: Checking in which the tests are partially carried out by the authoring tool, but where authors' input or judgment is still required to decide or help decide the outcome of the tests.
- automated checking: Checking in which the tests are carried out automatically by the authoring tool without any intervention by authors.

An authoring tool may support any combination of checking types.
content (web content)

Information and sensory experience to be communicated to the end user by means of a user agent, including code or markup that defines

the content's structure, presentation, and interactions. In ATAG 2.0, the term is primarily used to refer to the output that is produced by the authoring tool. Content produced by authoring tools may include web applications, including those that act as web-based authoring tools. Content may or may not be:

- accessible content (WCAG): Content that would conform to [WCAG 2.0](#), at either Level A, AA, or AAA, assuming that any web content technologies relied upon to satisfy the WCAG 2.0 success criteria are accessibility supported.
 - Note 1: If accessibility support for the relied upon technologies is lacking, then the content will not conform to WCAG 2.0 and one or more groups of end users with disabilities will likely experience difficulty accessing the content.
 - Note 2: Conformance to WCAG 2.0, even at the highest level (i.e. Level AAA), still may not make content "accessible to individuals with all types, degrees, or combinations of disability".
- content being edited: The web content that an author can modify during an authoring session. The content being edited may be a complete piece of content (e.g. image, style sheet) or only part of a larger piece of content (e.g. a status update). The content being edited only includes content in web content technologies that the authoring tool supports (e.g. a WYSIWYG HTML editor allows editing of the HTML content of a web page editable, but not the images).

content properties

The individual pieces of information that make up the web content (e.g. the attributes and contents of elements, style sheet information).

content (structured)

Web content that includes machine-readable internal structure (e.g. markup elements), as opposed to unstructured content, such as raster image formats or plain human language text.

content generation (content authoring, content editing)

The act of specifying the actual web content that will be rendered, played or executed by the end user's user agent. While the precise details of how content is created in any given system may vary widely, responsibility for the generation of content can be any combination of the following:

- author generated content: Web content for which authors are fully responsible. The author may only be responsible down to a particular level (e.g. when asked to type a text label, the author is responsible for the text, but not for how the label is marked up; when typing markup in a source editing-view, the author is not responsible for the fact that UNICODE is used to

encode the text).

- automatically-generated content: Web content for which developer-programmed functionality is fully responsible (e.g. what markup to output when an author requests to start a new document, automatically correcting markup errors).
- third-party content generation: Web content for which a third-party author is responsible (e.g. community shared templates).

content rendering

User interface functionality that authoring tools present if they render, play or execute the web content being edited. ATAG 2.0 recognizes several types of content renderings:

- conventional renderings (or "WYSIWYG"): When content is rendered in a way that is similar to the default rendering a user agent would create from the same content. While "WYSIWYG", standing for "What-you-see-is-what-you-get" is the common term, differences between user agents and end user settings mean that in reality there is no single typical end user experience; or
- unconventional renderings: When content is rendered differently than it would be in a typical user agent (e.g. rendering an audio file as a graphical waveform); or
- partial renderings: When some aspects of the content are rendered, played, or executed, but not others (e.g. a frame-by-frame video editor renders the graphical, but not the timing aspects, of a video).

content transformations

Processes that take content in one web content technology or non-web content technology (e.g. a word processing format) as input and produce content that has been optimized, restructured or recoded:

- Optimizing Content Transformations: Transformations in which the content technology is not changed and the structural features of the content technology that are employed also stay the same. Changes would not be expected to result in information loss (e.g. removing whitespace, replacing in-line styles with an external style sheet).
- Restructuring Content Transformations: Transformations in which the content technology stays the same, but the structural features of the technology used to markup the content are changed (e.g. linearizing tables, splitting a document into pages).
- Recoding Content Transformations: Transformations in which the content technology used to encode the content is changed (e.g. HTML to XHTML, a word processing format to HTML).

Note: Clipboard operations, in which content is copied to or pasted from the platform clipboard, are not considered content

transformations.

control settings

Settings that relate to how authors operate the authoring tool, for example using the keyboard or mouse.

developer

Any entities or individuals responsible for programming the authoring tool. This includes the programmers of any additional software components included by the Claimant in the [conformance claim](#). In some cases, development of the authoring tool is complete before authors can use it to publish web content. However, in other cases (e.g. some web-based authoring tools), the developer may continue to modify the authoring tool even after content has been published, such that the content experienced by the end user is modified.

direct accessibility features

Features of an authoring tool that provide functionality to meet the requirements of authors with disabilities (e.g. keyboard navigation, zoom features, text-to-speech). Additional or specialized functionality may still be provided by external assistive technology.

display settings

Settings that relate to how authors perceive the authoring tool. These include:

- audio display settings: the characteristics of audio output of music, sounds, and speech. Examples include volume, speech voices, voice speed, and voice emphasis.
- visual display settings: the characteristics of the on-screen rendering of text and graphics. Examples include fonts, sizes, colors, spacing, positioning, and contrast.
- tactile display settings: the characteristics of haptic output. Examples include the magnitude of the haptic forces and the types of vibration.

documentation

Any information that supports the use of an authoring tool. This information may be provided electronically or otherwise and includes help, manuals, installation instructions, sample work flows, tutorials, etc.

document object

The internal representation of data in the source by a non-web based authoring tool or user agent. The document object may form part of a platform accessibility service that enables communication with assistive technologies. Web-based authoring tools are considered to make use of the document object that is maintained by the user agent.

element

A pair of markup tags and its content, or an "empty tag" (one that requires no closing tag or content).

end user

A person who interacts with web content once it has been authored. This includes people using assistive technologies.

human language

Language that is spoken, written or signed (through visual or tactile means) to communicate with humans.

informative

For information purposes and not required for conformance.

keyboard interface

Keyboard interfaces are programmatic services provided by many platforms that allow operation in a device independent manner. A keyboard interface can allow keystroke input even if particular devices do not contain a hardware keyboard (e.g. a touchscreen-controlled device can have a keyboard interface built into its operating system to support onscreen keyboards as well as external keyboards that may be connected).

Note: Keyboard-operated mouse emulators, such as MouseKeys, do not qualify as operation through a keyboard interface because these emulators use pointing device interfaces, not keyboard interfaces.

keyboard trap

A user interface situation in which a keyboard interface may be used to move focus to, but not from, a user interface component or group of components.

label

Text or other component with a text alternative that is presented to users to identify a component. A label is presented to all users whereas the name may be hidden and only exposed by assistive technology. In many (but not all) cases the name and the label are the same.

live

Information captured from a real-world event that is published with no more than a broadcast delay.

Note: A broadcast delay is a short (usually automated) delay, for example used in order to give the broadcaster time to queue or censor the audio (or video) feed, but not sufficient to allow significant editing.

markup language

A system of text annotations (e.g. elements in HTML) and processing rules that may be used to specify the structure, presentation or semantics of content. Examples of markup languages include HTML and SVG.

- markup of some content is the set of annotations that appear in the content.

name

Text by which software can identify a user interface component to the author or end user. The name may be hidden and only exposed by assistive technology, whereas a label is presented to all users. In

many (but not all) cases, the label and the name are the same.
 non-text content

Any content that is not a sequence of characters that can be programmatically determined or where the sequence is not expressing something in human language. This includes ASCII Art (which is a pattern of characters), emoticons, and images representing text.

normative

Required for conformance. One may conform in a variety of well-defined ways to ATAG 2.0. Content identified as "informative" or "non-normative" is never required for conformance.

option

When an author is presented with choices.

- default option: A setting or value for an option that is assigned automatically by the authoring tool and remains in effect unless canceled or changed by the author.

platform

The software environment within which the authoring tool operates. Platforms provide a consistent operational environment on top of lower level software platforms or hardware. For web-based authoring user interfaces, the most relevant platform will be a user agent (e.g. browser). For non-web-based user interfaces, the range of platforms includes, but may not be limited to, desktop operating systems (e.g. GNOME desktop on Linux, Mac OS, Windows), mobile operating systems (e.g. Android, BlackBerry, iOS, Windows Phone), or cross-OS environments (e.g. Java), etc.

Note 1: Many platforms mediate communication between applications operating on the platform and assistive technology via a platform accessibility service.

Note 2: Accessibility guidelines for developers exist for many platforms.

platform accessibility service

A programmatic interface that is specifically engineered to provide communication between applications and assistive technologies (e.g. MSAA, IAccessible2 and UI Automation for Windows applications, AXAPI for Mac OS X applications, GNOME Accessibility Toolkit API for GNOME applications, Java Access for Java applications). On some platforms, it may be conventional to enhance communication further by implementing a document object.

plug-in

A program that runs as part of the authoring tool (e.g. a third-party checking and repair tool) and that is not part of web content being edited. Authors generally choose to include or exclude plug-ins from their authoring tool.

pre-authored content

Pieces of web content, created prior to an authoring session, that the authoring tool developer makes available to authors to use in the content being edited. Examples include clip art, sample videos,

user interface widgets.

Note 1: For templates, an incomplete form of pre-authored content, see [Guideline B.2.4](#).

Note 2: If the authoring tool uses pre-authored content automatically, see [Guideline B.1.1](#).

- accessible pre-authored content (WCAG): Pre-authored content that is either already accessible web content (WCAG) or would be accessible, if it was appropriately inserted into an empty document.

Note: If extensive author input is required to make use of pre-authored content, then the content may in fact be a template.

pre-authored content selection mechanism

A function beyond standard file selection that allows authors to select pre-authored content to use in an authoring session (e.g. clip art gallery, widget palette).

presentation

Rendering of the content in a form to be perceived by authors or end users.

programmatically determined (programmatically determinable)

Information that is encoded in a way that allows different software, including assistive technologies, to extract and present the information in different modalities. ATAG 2.0 uses this term in two contexts:

- Processing content: Whether the authoring tool is able to extract information from the web content (e.g. to extract the language of content from the markup).
- Communication between the authoring tool and assistive technology: For non-web-based user interfaces, this means making use of platform accessibility services, APIs, and, in some cases, document object models. For web-based user interfaces, this means ensuring that the user agent can pass on the information (e.g. through the use of WAI-ARIA).

prominence

A heuristic measure of how likely authors are to notice a user interface component in a user interface that they are operating. Prominence is affected by numerous factors, including: the number of navigation steps required, the reading order position, visual properties (e.g. size, spacing, color), and even the modality of use (e.g. mouse vs. keyboard use).

- at least as prominent: For ATAG 2.0, a user interface component A is considered to be "at least as prominent" as another component B when, from a default state, component A becomes displayed (and enabled) with the same number or less "opening" actions than are required for component B to become displayed (and enabled).

Note 1: When a container is open, all of the enabled components in the container (e.g. items in a list, items in a menu, buttons in a toolbar, all components in a dialog box) are considered to be displayed (and therefore are at least as prominent as each other), even if the container must be scrolled for them to become visible. This takes into account that different screen sizes and author settings will affect which components are visible at a given time.

Note 2: "Opening actions" are actions made by authors on components within the user interface that result in new components becoming displayed or enabled. For example: (a) keyboard shortcut to a top-level menu item to display a sub-menu, (b) keyboard selection on a button to display a dialog box, (c) mouse click on a checkbox to enable previously disabled sub-items, etc. Actions that do not cause new components to become actionable (e.g. moving focus, scrolling a list), are not counted as "opening actions".

Note 3: Keyboard shortcuts to components in closed containers are not counted as "opening actions" because the components have no prominence when they are not displayed. The same is true when authors must use "search" to reveal components in closed containers.

Note 4: The "default state" is the state of the authoring tool at the beginning of an authoring session as set by the developer. The default state of many document authoring tools is an editing-view.

prompt

Any authoring tool initiated request for a decision or piece of information from authors. The term covers both requests that must be responded to immediately (e.g. modal dialog boxes) as well as less urgent requests (e.g. underlining a misspelled word).

publishing

Any point at which the authors or authoring tool make web content available to end users (e.g. uploading a web page, committing a change in a wiki, live streaming).

range

More than one item within a multi-item set.

Informative Note: ATAG 2.0 uses the term "range" where absolute measurements may not be practical (e.g. the set of all help documentation examples, the set of all templates). While the strict testable requirement is the definition "More than one item within a multi-item set", implementers are strongly encouraged to implement the success criteria more broadly.

relationships

Meaningful associations between distinct pieces of content.

repair (accessibility)

The process by which web content accessibility problems that have been identified within web content are resolved. ATAG 2.0 recognizes

three types of repair, based on increasing levels of automation:

- manual repair: Where the repairs are carried out by authors. This includes the case where authors are aided by instructions or guidance provided by the authoring tool, but where authors carry out the actual repair procedure;
- semi-automated repair: Where the repairs are partially carried out by the authoring tool, but where authors' input or judgment is still required to complete the repair; and
- automated repair: Where the repairs are carried out automatically by the authoring tool without any intervention by authors.

restrictions, restricted web content authoring

When the web content that authors can specify with an authoring tool either must include or must not include certain content (e.g. elements, attributes, widgets). Many authoring tools restrict authoring in some way, which can either benefit accessibility (e.g. if text alternatives for non-text content are required) or detract from accessibility (e.g. if attributes for defining text alternatives are not available). In contrast, authoring tools that allow unrestricted web content authoring do not require any particular content to be included or not included (e.g. many source editing-views).

role

Text or a number by which software can identify the function of a component within web content (e.g. a string that indicates whether an image functions as a hyperlink, command button, or check box).

sequential keyboard access

Using a keyboard interface to navigate the focus one-by-one through all of the items in an ordered set (e.g. menu items, form fields) until the desired item is reached and activated. This is in contrast to direct keyboard access methods such as keyboard shortcuts and the use of bypass links.

technology (web content)

A mechanism for encoding instructions to be rendered, played or executed by user agents. Web content technologies may include markup languages, data formats, or programming languages that authors may use alone or in combination to create end user experiences that range from static web pages to multimedia presentations to dynamic web applications. Some common examples of web content technologies include HTML, CSS, SVG, PNG, PDF, Flash, Silverlight, Flex, and JavaScript.

templates

Content patterns that are filled in by authors or the authoring tool to produce web content for end users (e.g. document templates, content management templates, presentation themes). Often templates will pre-specify at least some authoring decisions.

- accessible templates (WCAG): Templates that can be filled in to create web content that meets the [WCAG 2.0](#) success criteria (Level A, AA or AAA), when both of the following are true:
 - a. The author correctly follows any instructions provided (e.g. correctly responding to prompts, correctly replacing highlighted placeholders); and
 - b. No further authoring occurs

Note: Under these conditions, some templates will result in completely empty documents, which are considered accessible by default.

template selection mechanism

A function beyond standard file selection that allows authors to select templates to use as the basis for new content or to apply to existing content.

time limit

The amount of time that an authoring tool provides to authors to perform a task (e.g. read a message, select an item, save a change). Examples include: authoring session timeouts, time-based presentations (e.g. tutorial video).

tutorial

A type of documentation that provides step-by-step instructions for performing multi-part tasks.

user agent

Any software that retrieves, renders and facilitates end user interaction with web content (e.g. web browsers, browser plug-ins, media players)

- In-Market User Agent: A user agent that can be procured by members of the public (free or otherwise). Usually, an in-market user agent will be a separate software from the authoring tool; however, sometimes a software may combine user agent and authoring tool functionality. These cases include:
 - Preview-Only: If the user agent can only render web content that it receives from the associated authoring functionality, then the software is an authoring tool with a preview feature. Such preview-only features are not considered in-market user agents.
 - User Agent with Authoring Tool Mode: If the user agent functionality must retrieve and open web content before it can be sent to the authoring tool functionality, then the software is a user agent with an authoring tool mode. If the user agent is used to preview content produced by the authoring tool mode, then it is to be considered an in-market user agent.
 - Combined User Agent/Authoring Tool: A user agent in which the default mode of user interaction enables editing the web content. These tools do not need previews because

the author is already experiencing the content in the same way as end users.

user interface component

A part of the user interface or content display (including content renderings) that is perceived by authors as a single control for a distinct function.

video

The technology of moving pictures or images. Video can be made up of animated or photographic images, or both.

view

A user interface function that authors use to interact with the web content being edited. ATAG 2.0 categorizes views according to whether they support editing:

- editing-views: Views in which some or all of the content is editable; or
- previews: Views in which no authoring actions are provided (i.e. the view is not editable). Previews are provided to present the web content being edited by the authoring tool as it would appear to end users of user agents. Previews may be implemented using actual in-market user agents, but this is not necessary.

ATAG 2.0 also recognizes several approaches to presenting the content in a view:

- source views: The content is presented in unrendered form (e.g. plain text editors); or
- rendered views: Content renderings (conventional, unconventional or partial) are presented; or
- property views: Only properties of the content are presented. The authoring tool then uses these properties to automatically generate the content to be published (e.g. CMS calendar widget that generates a calendar from the numeric month and year).

workflow

A customary sequence of steps or tasks that authors follow to produce a content deliverable. If an authoring tool is composed of a collection of applications (e.g. markup editor, image editor, and validation tool), then its workflows may include use of one or more of the applications.

Appendix B: References

This section is informative.

For the latest version of any W3C standards please consult the list of [W3C Technical Reports](http://www.w3.org/TR/) at <http://www.w3.org/TR/>. Some documents listed

below may have been superseded since the publication of this document.

[ATAG10]

["Authoring Tool Accessibility Guidelines 1.0"](#), J. Treviranus, C. McCathieNeville, I. Jacobs, and J. Richards, eds., 3 February 2000.

[UAAG]

["User Agent Accessibility Guidelines 1.0"](#), I. Jacobs, J. Gunderson, and E. Hansen, eds. 17 December 2002.

[WCAG20]

["Web Content Accessibility Guidelines 2.0"](#), B. Caldwell, M. Cooper, L. Guarino Reid, and G. Vanderheiden, eds. 11 December 2008.

Appendix C: Acknowledgments

Appendix Editors:

- Jan Richards (Adaptive Technology Resource Centre, University of Toronto)
- Jeanne Spellman (W3C)
- Roberto Scano (IWA/HWG)

Participants active in the AUWG at the time of publication:

- Tom Babinszki (IBM)
- Tim Boland (National Institute for Standards and Technology)
- Alastair Campbell (Nomensa)
- Cherie Ekholm (Microsoft)
- Alex Li (Microsoft)
- Alessandro Miele (Invited Expert)
- Greg Pisocky (Adobe)
- Jan Richards (Inclusive Design Institute, OCAD University)
- Roberto Scano (IWA/HWG)
- Jeanne Spellman (W3C)
- Jutta Treviranus (WG Chair; Inclusive Design Institute, OCAD University)

Other previously active AUWG participants and other contributors to ATAG 2.0:

Previous Editors:

Tim Boland, NIST

Matt May (until June 2005 while at W3C)

Kynn Bartlett, Giorgio Brajnik, Judy Brewer, Wendy Chisholm, Daniel Dardailler, Geoff Deering, Barry A. Feigenbaum, Katie Haritos-Shea, Kip Harris, Phill Jenkins, Len Kasday, Marjolein Katsma, William Loughborough, Karen Mardahl, Matt May, Charles McCathieNeville, Ann

McMeekin, Matthias Müller-Prove, Liddy Nevile, Sueann Nichols, Graham Oliver, Wendy Porch, Sarah Pulis, Bob Regan, Chris Ridpath, Andrew Ronksley, Gregory Rosmaita, Dana Simberkoff, Reed Shaffner, Michael Squillace, Heather Swayne, Gregg Vanderheiden, Carlos Velasco, and Jason White.

This document would not have been possible without the work of [those who contributed to ATAG 1.0](#).

This publication has been funded in part with Federal funds from the U.S. Department of Education, National Institute on Disability and Rehabilitation Research (NIDRR) under contract number ED-0SE-10-C-0067. The content of this publication does not necessarily reflect the views or policies of the U.S. Department of Education, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

[\[Contents\]](#)