# Verifying Accessibility: An Integral Step of Content Development and Maintenance

By: Donna Lewis, CPACC, ADS

New content—webpages, forms, documents, etc.—should be created to be accessible from the first pixel applied to the digital page. But how can you be sure it meets accessibility guidelines? For that matter, what if you have webpages, documents, or forms that were created before you were aware of accessibility guidelines; how can you verify they are accessible?

Evaluating and verifying your materials are accessible is a key step in the iterative, recursive cycle of developing and maintaining content. Including evaluation passes into any new content development plan helps reduce time and money to remediate inaccessible material.

## Developing New Content Is A Cyclical Process

### Plan

To make certain new content you develop is accessible, think of the development process as a process with looping steps. Begin with a clear identification of what the document or webpage needs to communicate. Try to identify key aspects of the audience that is most likely to be viewing your content. Can you think of ways to make the content perceivable and accessible to audience members with diverse abilities? Can you imagine ways that you might create material that would pose a problem for people to perceive or interact with? Might you be building in any barriers to clear communication?

### Develop/Code

After identifying what you need to communicate and to whom, began the actual development process. Of course, the steps and approaches in this stage will vary depending on what you need to communicate, what means—document, email, webpage, etc.—of distribution, and the expected audience. At the most basic level, this is the stage when you begin to actually create the content, and here is where it is essential to begin to evaluate the accessibility.

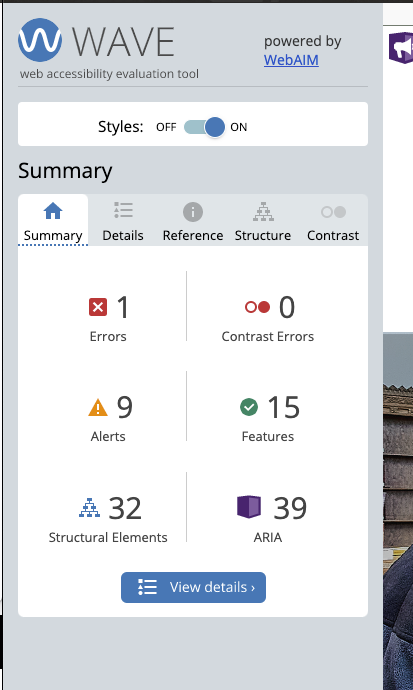
### Evaluate Accessibility

#### Automated Accessibility Checking Tools

Once some content is created, evaluate its accessibility. The Web Accessibility Initiative has published [Easy Checks – A First Review of Web Accessibility](https://www.w3.org/WAI/test-evaluate/preliminary/), a very useful process for evaluating the accessibility of webpages. Automated accessibility checking tools are also available for both webpages and documents created in specific applications.

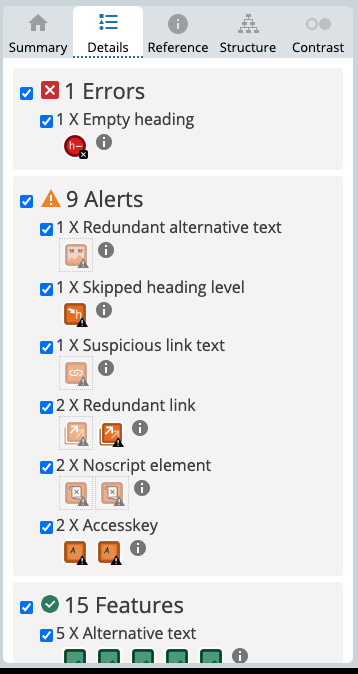
Automated checking tools do not catch all accessibility problems in webpages or documents, but they can identify potential problem areas for thorough manual evaluation. These tools cannot confirm accessibility; instead, they are useful for finding concerns such as images with missing alternative text, suspect or duplicate link names, missing or out of order headings, and other areas of concern. Once these items of concern are identified, they can be evaluated and resolved before continuing through the development process. If necessary, you can loop back to earlier stages of the development process to address more far-reaching or complex accessibility issues.

Most automated tools evaluate the webpage or document and return a list of error, alerts of warnings, and other features such as ARIA and various types of structural features. WAVE, a tool for evaluating websites, returns the following results after reviewing a webpage:

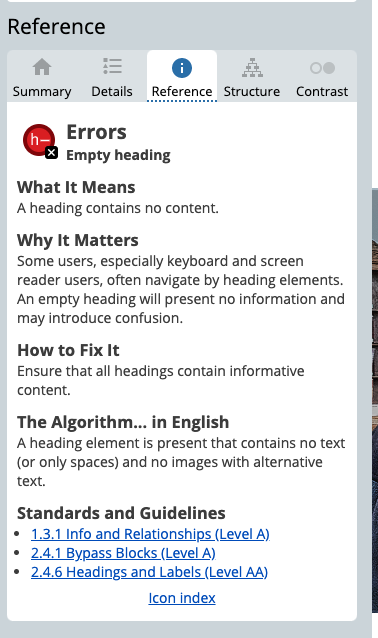


The designer can then access further details regarding these findings from other tabs. More information on the Error identified can be located on the Details and References tabs.

The Details tab provides a list of findings with symbols that can be used to locate specific ares of the webpage needing review and/or revision.



The Reference tab provides additional information about a given issue, including a definition of the element, an explanation of why the element matters for the webpage’s accessibility, what can be done to address the item, and links to specific Web Content Accessibility Guidelines (WCAG) standards and other laws or guidelines related to the identified issue.



Some useful automated checking tools for webpages include:

* + WAVE -- <https://wave.webaim.org/>
  + WebDev toolbars associated with most browsers
    - [Web Developer Extension for Chrome](https://chrome.google.com/webstore/detail/web-developer/bfbameneiokkgbdmiekhjnmfkcnldhhm)
    - [Web Developer Extension for Firefox](https://addons.mozilla.org/en-US/firefox/addon/web-developer/)
    - [Other tools are also helpful](https://www.w3.org/WAI/test-evaluate/tools/selecting/)

Color contrast checkers can also be useful for both evaluating webpages and documents. Some useful checkers include:

* + [Paciiello Group Color Checker](https://www.paciellogroup.com/color-contrast-checker/)
  + [WebAIM.org Color Checker](https://webaim.org/resources/linkcontrastchecker/?fcolor=FFFFFF&bcolor=09AAAA)
  + [Colorzilla (for Firefox)](https://www.colorzilla.com/firefox/)  
    (Colorzilla works through Firefox browser. Selects colors that can then be compared to determine contrast ratios)

#### Application-Specific Accessibility Checking Tools

Automated accessibility checking tools are also available in many source applications, in which content is initially built, including Microsoft Office 365 applications, as well as Adobe Acrobat.

To access these checkers:

* Microsoft Word:
  + File > Info > Check (Word for PC);
  + Tools > Check Accessibility (Word for Mac)

The checker organizes the results into three categories:

* + Errors—These issues render the content very difficult or impossible to access.
  + Warnings—These problems may make the content difficult to use.
  + Tips—This information provides hints on better organizing information in the document.

The checker also provides information on why the identified issue should be fixed and how to fix it.

* Adobe Acrobat
  + Open Document 🡪 Open Accessibility Tool (Click More Tools if you do not see the Accessibility Tool) 🡪 Run an Accessibility Check

#### Manual Checks

##### Keyboard Navigation

Some aspects of webpages or documents must be evaluated for accessibility through manual checks. These include keyboard navigation, screen reader navigation, and logical reading/tab order.

To be accessible, content in webpages and documents must be navigable and interactive using only a keyboard. To verify your content does not require the use of a mouse or other pointing device, check that the content can be navigated using only the tab, space bar, and/or arrow keys. As you use the keyboard to move through the page and interact with the content, also make sure that focus indicators, boxes or underlines indicating your location, are clear and easily perceived.

##### Screen Reader Navigation

Screen readers, assistive technology used to convert text-based content into speech, must be able to navigate and interact with your content. Screen reader applications convert webpage or document content into electronic speech or Braille that can be read using a Braille keyboard.

Screen reader applications are specific to given operating systems, for instance

* JAWS/ZoomText operates in Windows. These two programs are often bundled together as [Fusion](https://www.freedomscientific.com/products/software/fusion/) and are very powerful, but very expensive.
* [NVDA](https://www.nvaccess.org/download/) also operates in Windows and is also very powerful and feature rich. However, it is an open-source software that is free to download.
* VoiceOver is the built-in screen reader for Apple devices, including computer systems, phones, and tablets. It is also powerful and feature rich.
* Application-Specific Screen Readers
  + These screen readers are available in Office 365 and Adobe Acrobat. They are less powerful that the previously mention screen readers and have fewer available commands for interacting with content. They are particularly useful to use as a “second opinion,” to complement findings from another screen reader.

To evaluate your content using a screen reader:

* Tab to all controls and through the materials, verifying that the reading order and tab order are logical. The order of material you see on your screen may not reflect the order in which the content is read aloud by the screen reader.
* Activate all buttons and links using the screen reader’s keyboard commands.
* Verify that the webpage or document has no hidden features requiring a mouse over/mouse hover to activate.
* Make sure that no keyboard traps that prevent navigation away from an area of content exist.
* When/if pop-ups open, the focus must relocate to the pop-up dialog box, and the screen reader must read the box’s content.

To be considered accessible, your content must be able to be navigated regardless of the operating system, browser, or screen reader used to access it.

## Evaluating Existing Or Legacy Content

Verify that your existing webpages or document are accessible using these steps as well. Content that was developed before you knew about accessibility guidelines, such as WCAG or Section 508, may pose significant challenges to revise to comply with these guidelines. However, although “born accessible” content is more likely to be accessible or be more easily corrected to meet guidelines, most existing or legacy content can be remediated. When possible, begin the evaluation process as soon as possible. For example, if you will be teaching a course with digital content created before you understood the requirement for accessibility, give yourself plenty of lead time to evaluate the material and identify needed revisions to make it accessible before the next course offering begins. Create a remediation plan. Move through the material making revisions, periodically evaluating the changes you’ve made to ensure that those changes are accessible.

The goal of creating new webpages, documents, course materials, etc., is to communicate information. To achieve that goal, the content must be created in such a way that it is accessible to users with varying abilities, using differing technology, and having different preferences for interacting with the materials. Verifying that your content is robustly accessible increases the likelihood your content will be communicated directly and unambiguously.