BlueCross BlueShield
of North Carolina

Accessibility Guide

Contents

Introduction

At Blue Cross and Blue Shield of North Carolina, we want to ensure users of all abilities can access our site and applications. To meet this goal, our consumer-facing web pages should, at a minimum, comply with the standards for web accessibility as set forth in Web Content Accessibility Guidelines (WCAG) 2.0 (minimum of Level AA conformance) established by the World Wide Web Consortium (W3C). Any web pages or documents related to selling or servicing health plans for our Affordable Care Act (ACA) products must meet this standard.

We encourage, but do not require, compliance with Criteria 1.2.3: Audio Description or Media Alternative (Prerecorded) and Criteria 1.2.5: Audio Description (Prerecorded).

In addition, any web pages or documents related to selling or servicing health plans for our Medicare members must comply with Section 508 of the Rehabilitation Act (29 U.S.C. 794d), which directly references the WCAG 2.0 Level AA standards, so the same guidelines apply.

This guide covers Blue Cross NC guidelines for web accessibility.

Testing Tools

Windows

We recommend using Chrome or Firefox for testing, as these browsers have built-in features that can speed the testing process. In addition, install the aXe extension for Chrome or Firefox. This tool performs basic automated testing for accessibility issues. For screen reader testing, we recommend NVDA, a stand-alone screen-reader application. If you are unable to install a stand-alone reader, you can perform some basic testing with the ChromeVox extension for Chrome.

Mac

By default, keyboard navigation is disabled in many Mac browsers. You will need to enable keyboard navigation. We recommend using Chrome or Firefox for testing, as these browsers have built-in features that can speed the testing process. In addition, install the aXe extension for Chrome or Firefox. For screen reader testing, you can use VoiceOver, the built-in Mac screen reader.

Web-based Tools

If you are unable to install browser tools due to security restrictions, you can use the Wave Online checker for automated testing and Web Anywhere for in-browser screen reader testing. However, these tools do not work as well as native tools. They also will only work on non-authenticated, production sites.

Mobile

Android and iOS smartphones have a variety of built-in accessibility tools.

Accessibility Standards

These standards and best practices ensure legal compliance and a great user experience.

Alternate Text

Alternate text is a brief, text-based description of a visual element. It is generally a hidden property of an image.

Why does it matter?

Visually impaired users cannot see pictures and images on a page, so they rely on alternate text to understand visual elements. In addition, robots that scan the site for search engines use alternate text to index images.

Standards

Non-text Content (1.1.1)
Images of Text (1.4.5)

Checklist

- □ Every image on the page has clear, appropriate alternate text.
- □ Decorative images and graphics are loaded by CSS or have null/empty alt values (alt="").
- □ Complex graphics (graphs, charts, etc.) either have a descriptive caption or a link to a description.
- □ Alternate text on image links is descriptive enough to convey appropriate information and direction to members.
- □ Any image-based form buttons have clear, accurate alternate text.
- □ If the same visual presentation can be made using text alone, an image is not used to present that text.

Testing Techniques

Use aXe to quickly check a page. AXe will flag any images without alternate text. You should also manually check a sample of the images to make sure the alternate text is appropriate.

References

- WebAIM – Alternative text
- WebAIM – Alt text and linked images
- W3C – H37: Using alt attributes on img elements
- W3C – H67: Using null alt text and no title attribute on img elements for images that AT should ignore

Code Standards

These standards address good coding practices

Why does it matter?

Valid, well-formed code will work better across a variety of assistive technologies.

Standards

Info and Relationships (1.3.1)
Resize text (1.4.4)
Page Titled (2.4.2)
Language of Page (3.1.1)
Language of Parts (3.1.2)
Parsing (4.1.1)
Name, Role, Value (4.1.2)

Checklist

- □ The page is readable and functional when the text size is doubled.
- □ The web page has a descriptive and informative page title.
- □ The language of the page is identified using the HTML lang attribute
- □ The language of page content that is in a different language is identified using the lang attribute
- □ The HTML of a page of validates without significant errors.
- □ Semantic markup is used to designate headings (<h1>), lists (<ul>, <ol>, and <dl>), emphasized or special text (<strong>, <code>, <abbr>, <blockquote>, for example), etc. Semantic markup is used appropriately.
- □ Markup is used in a way that facilitates accessibility. This includes following the HTML/XHTML specifications and using forms, form labels, frame titles, etc. appropriately.

Testing Techniques

Use aXe to quickly check for an identified language attribute. It will also identify some common HTML validation errors. To check zoom, zoom in to 200% and ensure the page is still readable and functional. Manually check the page title (generally located on the tab or above the address bar in your browser).

References

- W3C – H25: Providing a title using the title element
- W3C – H57: Using language attributes on the html element
- W3C – H58: Using language attributes to identify changes in the human language
- W3C – H75: Ensuring that Web pages are well-formed

Content

A page's content needs to be generally usable and navigable. This includes providing a consistent and sensible navigation structure, clear and understandable instructions. In addition, link text needs to be descriptive enough to convey appropriate information and direction, without relying on surrounding text.

Why does it matter?

All users will benefit from a consistent and sensible navigation structure. In addition, many screen reader users rapidly jump through the links on a page. When they do, the reader only reads the linked text, not the text around it. Sighted users also often scan a page rapidly, looking for just the links. For this reason, links and linked elements should make sense.

Standards

Sensory Characteristics (1.3.3)
Headings and Labels (2.4.6)
Consistent Navigation (3.2.3)
Consistent Identification (3.2.4)
Link Purpose (In Context) (2.4.4)
Multiple Ways (2.4.5)

Checklist

- □ Page headings are informative.
- □ Large blocks of text are broken up with useful headings.
- □ Navigation links that are repeated on web pages do not change order when navigating through the site.
- □ Elements that have the same functionality across multiple web pages are consistently identified.
- □ Instructions do not rely upon shape, size, or visual location (e.g., "Click the square icon to continue" or "Instructions are in the right-hand column").
- □ Instructions do not rely upon sound (e.g., "A beeping sound indicates you may continue.").
- □ The purpose of each link (or form image button or image map hotspot) can be determined from the link text alone, or from the link text and its context (e.g., surrounding paragraph, list item, table cell, or table headers).
- □ Links (or form image buttons) with the same text that go to different locations are readily distinguishable.
- □ Multiple ways are available to find other web pages on the site.

Testing Techniques

Manually check pages for sensible headings, navigation structure, and link text. In particular, look for vague or unhelpful links (e.g., "click here") or links that have the same text but different destinations (e.g., four "learn more" links that go to the same place). AXe can help identify out-of-order or skipped heading levels

References

- W3C – G91 : Providing link text that describes the purpose of a link
- WebAIM – Screen Readers and Links

Design

The visual design of a page is an important component of accessibility.

Why does it matter?

Many users, especially older users, have low-vision. While they can read text, it needs to be large enough to see and have high enough contrast.

Color blindness is also very common. In the United States, about 10% of men are color blind[1].

Standards

Use of Color (1.4.1)
Contrast (Minimum) (1.4.3)
Images of Text (1.4.5)

Checklist

- □ Color is not the only way to distinguish important information.
- □ Links are either underlined or have a 3:1 contrast ratio with surrounding text and change visually on focus and hover.
- □ Text has enough contrast. The background and foreground colors have a contrast ratio of 3:1 for large text (18px and above) and 4.5:1 for smaller text.
- □ If the same visual presentation can be made using text alone, an image is not used to present that text.
- □ The font is readable and not too small (a least 9pt, 12pt preferred). (**Optional**)

Testing Techniques

Use aXe to quickly check text contrast on the page. You can also use the WAVE Color Contrast Checker to check the contrast ratio between two colors.

Manually check for any page content where color is the only way to distinguish important information (e.g., incorrect form fields highlighted in red with no additional markers.)

References

- WAVE Color Contrast Checker
- WebAIM – Designing for Color Blindness
- WebAIM – Fonts
- W3C: G183: Using a contrast ratio of 3:1 with surrounding text and providing additional visual cues on focus for links or controls where color alone is used to identify them

[1] From the National Library of Medicine

Documentation and Support

In addition to the site itself, your technical support and documentation should also be accessible.

Why does it matter?

Proper documentation will help your users find and use the accessible features of your site or application. In addition, users with accessibility issues should have the same access to user support.

Standards

Any standards relevant to documentation and support pages

Checklist

- □ The documentation and support pages are accessible.
- □ The documentation includes any accessibility features, including any special keyboard shortcuts. **(Optional)**
- □ If there is audio technical support, like a phone line, there is a non-audio alternative, like a support email address, for users with hearing or speech difficulties. **(Optional, required for Medicare)**

Testing Techniques

Documentation and support pages should meet the other standards in this guide.

References

- Web Accessibility – Information, documentation, and support

Files and Plug-Ins

In addition to the HTML content of a page, linked files and plug-ins must also be accessible

Why does it matter?

We often provide critical content, like benefits statements, in PDF or another file format. Users with disabilities must be able to read them as well.

Standards

PDF accessibility is measured with the PDF/UA standard.

Checklist

- □ Any linked files are also accessible.

Testing Techniques

Manual verify a link to the plug-in page is available. To test documents, use their internal testing tools; see the references for more information.

References

- Adobe – PDF Accessibility Overview (pdf)
- Adobe – Using the Acrobat XI Pro Accessibility Checker (pdf)
- Microsoft – Accessibility in Microsoft Office 2010
- Microsoft – Accessibility in Microsoft Office 2013
- WebAIM – PDF Accessibility: Acrobat and Accessibility

Forms

Forms need appropriate tags and labels to work with assistive technology, particularly form labels.

Why does it matter?

Users with disabilities need to be able to complete forms.

Standards

Non-text Content (1.1.1)
Info and Relationships (1.3.1)
Meaningful Sequence (1.3.2)
Keyboard (2.1.1)
Timing Adjustable (2.2.1)
Focus Order (2.4.3)
Headings and Labels (2.4.6)
Focus Visible (2.4.7)
On Focus (3.2.1)
On Input (3.2.2)
Error Identification (3.3.1)
Error Suggestion (3.3.3)
Labels or Instructions (3.3.2)
Error Prevention (Legal, Financial, Data) (3.3.4)
Name, Role, Value (4.1.2)

Checklist

- □ Each form element has a corresponding label element (preferred) or a title attribute.
- □ Any image-based form buttons have clear, accurate alternate text.
- □ Related form fields are grouped.
- □ The tab order of the fields (the order a user travels through them with the tab key) matches the visual order.
- □ Users can navigate and submit a form using only a keyboard.
- □ If the form has a time limit, there is a way to extend that limit.
- □ It is visually apparent which form field has the current keyboard focus (i.e., as you tab through the field, you can see where you are).
- □ Form fields are identified as required or optional. Any special formatting instructions are obvious and linked to the form field
- □ Form validation errors are presented in an efficient, intuitive, and accessible manner.
- □ If an input error is detected, provide suggestions for fixing the input in a timely and accessible manner
- □ Sufficient labels, cues, and instructions for required interactive elements are provided via instructions, examples, properly positioned form labels, and/or fieldsets/legends.
- □ If the user can change or delete legal or financial data, the changes/deletions can be reversed, verified, or confirmed.

- □ When a page element receives focus, it does not result in a substantial change to the page that could confuse or disorient the user.
- □ When a user inputs information or interacts with a control, it does not result in a substantial change to the page without warning.
- □ On a form validation, focus automatically moves to the error message or first invalid field **(optional)**

Testing Techniques

Use aXe to quickly check form labels. Manually verify that a user can submit a form using only keyboards, including navigating any form validation messages.

References

- WebAIM – Creating Accessible Forms
- WebAIM - Usable and Accessible Form Validation and Error Recovery
- W3C – G149: Using user interface components that are highlighted by the user agent when they receive focus
- W3C – H44: Using label elements to associate text labels with form controls
- W3C – H65: Using the title attribute to identify form controls when the label element cannot be used
- W3C – H71: Providing a description for groups of form controls using fieldset and legend elements
- W3C – ARIA16: Using aria-labelledby to provide a name for user interface controls

Keyboard Accessibility

Users should be able to use the site with only a keyboard.

Why does it matter?

Many users have a mobility issue that prevents them from using a traditional keyboard and mouse. Some will navigate using only the keyboard. Others might use alternative input devices that mimic keyboard input.

Standards

Keyboard (2.1.1)
No Keyboard Trap (2.1.2)
Focus Order (2.4.3)
Focus Visible (2.4.7)

Checklist

- □ The entire page is keyboard-navigable. Users can reach every interactive element of the page using a keyboard, including all user interface controls, using the default navigation keys (tab for forward, shift + tab for back, enter or spacebar to trigger).
- □ There are no keyboard traps that prevent keyboard-only users from navigating.
- □ The navigation order of links, form elements, etc. is logical and intuitive.
- □ It is visually apparent which page element has the current keyboard focus (i.e., as you tab through the page, you can see where you are).

Testing Techniques

Navigate the page using only a keyboard. Verify you can access all the links and interactive elements on the page and the tab order is sensible.

References

- WebAIM – Keyboard Accessibility
- W3C – H4: Creating a logical tab order through links, form controls, and objects
- W3C – H91: Using HTML form controls and links
- W3C – SCR35: Making actions keyboard accessible by using the onclick event of anchors and buttons
- MDN – Keyboard-navigable JavaScript widgets

Multimedia

Audio or video content should also have captions or transcripts, and multimedia players should have keyboard controls. Videos or animated graphics should avoid rapid flickering effects.

Why does it matter?

Users who are deaf or hard of hearing need captions to understand audio or video content. Captions can also help users who are in noisy environments or don't have headphones or speakers.

In addition, videos that flicker rapidly can trigger migraines or seizures in sensitive users.

Standards

Audio-only and Video-only (Prerecorded) (1.2.1)
Captions (Prerecorded) (1.2.2)
Audio Control (1.4.2)
Keyboard (2.1.1)
Three Flashes or Below Threshold (2.3.1)

Checklist

- □ No autoplay (preferred) OR a mechanism is provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.
- □ Audio-only files have transcripts.
- □ Videos without an audio track have a text-based description.
- □ Videos have synchronized captions.
- □ No page content flashes more than 3 times per second unless that flashing content is sufficiently small and the flashes are of low contrast and do not contain too much red.
- □ Audio or video players have keyboard controls.

Testing Techniques

Manually check audio and visual elements on the page. If you are concerned about a particular video, you can use Photosensitive Epilepsy Analysis Tool (PEAT) to check it for epilepsy triggers.

References

- WebAIM – Captions , Transcripts, and Audio Descriptions
- WebAIM – Seizure Disorders
- W3C – G19: Ensuring that no component of the content flashes more than three times in any 1-second period
- W3C – G87: Providing closed captions
- W3C – G176: Keeping the flashing area small enough

Skip Navigation Link

Skip navigation (skip nav) links are links at the top of a page that skip navigation menus and other repetitive content that comes before the page content.

Why does it matter?

Without a skip nav link, keyboard and screen reader users will be forced to navigate the entire navigation menu before each page.

Standards

Bypass Blocks (2.4.1)

Checklist

- □ There is a way to skip over repetitive navigation menus, if they exist.
- □ The skip nav link is the first focusable object on a page.
- □ The skip nav link is either visible or appears on focus.

Testing Techniques

Manually verify the skip navigation link exists and takes users to the appropriate place.

Note that a skip nav link is not required if there is no repetitive header content, or if the header content is contained entirely in an expandable and collapsible menu (e.g., a “hamburger” menu on a mobile page.)

References

- WebAIM – "Skip Navigation" Links
- W3C – SCR28: Using an expandable and collapsible menu to bypass block of content
- W3C – G1: Adding a link at the top of each page that goes directly to the main content area

Tables

HTML tables used to provide data need to be tagged correctly.

Why does it matter?

Complex or confusing tables can be very difficult to navigate with a screen reader or keyboard.

Standards

Info and Relationships (1.3.1)
Name, Role, Value (4.1.2)

Checklist

- □ Data tables have marked header cells.
- □ Tables are not used for layout (preferred), or layout tables do not have header cells.
- □ If a large or complex table has multiple levels of headers, the headers have an appropriate scope **(Optional)**.

Testing Techniques

Use aXe to check your page. Make sure that none of your data tables are marked as layout tables. Manually check any tables with multiple levels of headers.

References

- WebAIM – Creating Accessible Tables
- W3C – H63: Using the scope attribute to associate header cells and data cells in data tables

Timed elements

Timed elements are any page element that has a time limit or that changes or moves without direct user control.

Why does it matter?

Users with disabilities may require a longer time to read information or complete a form. In addition, user may find moving, blinking, scrolling, and auto-updating content distracting

Standards

Timing Adjustable (2.2.1)
Pause, Stop, Hide (2.2.2)

Checklist

- □ Every timed element (e.g., sliders, timed forms) has a way to either pause the content or extend the time.
- □ Automatically moving, blinking, or scrolling content that lasts longer than 5 seconds can be paused, stopped, or hidden by the user.
- □ Automatically updating content can be paused, stopped, or hidden.

Testing Techniques

Manually verify that the time extension feature is available and works.

References

- W3C – G4: Allowing the content to be paused and restarted from where it was paused
- W3C – SCR16: Providing a script that warns the user a time limit is about to expire
- W3C – SCR1: Allowing the user to extend the default time limit

Appendix A: WCAG 2.0 Level AA Quick Checklist

Number	Success Criteria	Level	Pass/Fail
1.1.1	Non-text Content	Level A	
1.2.1	Audio-only and Video-only (Prerecorded)	Level A	
1.2.2	Captions (Prerecorded)	Level A	
1.2.3	Audio Description or Media Alternative (Prerecorded)	Level A	Not required
1.2.4	Captions (Live)	Level AA	
1.2.5	Audio Description (Prerecorded)	Level AA	Not required
1.3.1	Info and Relationships	Level A	
1.3.2	Meaningful Sequence	Level A	
1.3.3	Sensory Characteristics	Level A	
1.4.1	Use of Color	Level A	
1.4.2	Audio Control	Level A	
1.4.3	Contrast (Minimum)	Level AA	
1.4.4	Resize text	Level AA	
1.4.5	Images of Text	Level AA	
2.1.1	Keyboard	Level A	
2.1.2	No Keyboard Trap	Level A	
2.2.1	Timing Adjustable	Level A	
2.2.2	Pause, Stop, Hide	Level A	
2.3.1	Three Flashes or Below Threshold	Level A	
2.4.1	Bypass Blocks	Level A	
2.4.2	Page Titled	Level A	
2.4.3	Focus Order	Level A	
2.4.4	Link Purpose (In Context)	Level A	
2.4.5	Multiple Ways	Level AA	
2.4.6	Headings and Labels	Level AA	
2.4.7	Focus Visible	Level AA	
3.1.1	Language of Page	Level A	
3.1.2	Language of Parts	Level AA	
3.2.1	On Focus	Level A	
3.2.2	On Input	Level A	
3.2.3	Consistent Navigation	Level AA	
3.2.4	Consistent Identification	Level AA	
3.3.1	Error Identification	Level A	
3.3.2	Labels or Instructions	Level A	
3.3.3	Error Suggestion	Level AA	
3.3.4	Error Prevention (Legal, Financial, Data)	Level AA	
4.1.1	Parsing	Level A	
4.1.2	Name, Role, Value	Level A	

For the complete text of each success criteria, see WCAG 2.0 Guidelines.