Mobile Site Accessibility Testing Methodology

Step 1: Identify devices

Step 2: Identify site type and variations

Step 3: Test critical issues

Step 4: Test mobile issues

Step 5: Test mobile assistive technology and feature support

# Table of Contents

[Introduction 3](#_Toc46835673)

[A note on WCAG2.1 3](#_Toc46835674)

[Mobile sites versus native apps 3](#_Toc46835675)

[Relationship with other Mobile Site documents 3](#_Toc46835676)

[Mobile Site Testing Methodology 4](#_Toc46835677)

[Overview 4](#_Toc46835678)

[Step 1: Identify devices 5](#_Toc46835679)

[Identify devices 5](#_Toc46835680)

[Step 2: Identify the site type and variations of the page 6](#_Toc46835681)

[Step 3: Test critical issues 8](#_Toc46835682)

[Exit traps 8](#_Toc46835683)

[Swipe / scroll traps 8](#_Toc46835684)

[Layer trap 8](#_Toc46835685)

[Step 4: Test mobile issues 9](#_Toc46835686)

[Alternatives 9](#_Toc46835687)

[Display 9](#_Toc46835688)

[Actionable items 10](#_Toc46835689)

[Navigational aids 10](#_Toc46835690)

[Audio and video 10](#_Toc46835691)

[Forms 10](#_Toc46835692)

[Mobile / desktop interaction 11](#_Toc46835693)

[Step 5: Test mobile assistive technology and feature support 12](#_Toc46835694)

[Assistive technologies and mobile features 12](#_Toc46835695)

[Acknowledgements 14](#_Toc46835696)

[Relationship to existing Accessibility testing standards 14](#_Toc46835697)

[ICT Accessibility Testing Symposium Mobile Site Sub-Committee 14](#_Toc46835698)

[Contacts 14](#_Toc46835699)

[Gian Wild 14](#_Toc46835700)

[Peter McNally 14](#_Toc46835701)

# Introduction

The ICT Accessibility Testing Symposium has developed a methodology for evaluating the accessibility of mobile websites. This document is an amalgamation of accepted mobile site accessibility testing standards from around the world, including additional developments from the ICT Accessibility Testing Symposium’s Mobile Site Sub-Committee (for more information, see Acknowledgements on page 14).

WCAG2 success criteria are applicable to mobile sites, however, not all aspects of mobile site accessibility are specifically covered by WCAG2. It is the opinion of this committee that merely conforming WCAG2 (or WCAG 2.1) does not provide for a fully accessible experience for users with disabilities.

Please note that this methodology does not include those issues already included in WCAG2, however does include issues identified in WCAG2.1. This guide was written with the intent to clarify the unique needs of users with disabilities who use mobile websites and to raise the bar for the web development community. This is a work-in-progress, and, as such, we do not make a claim that conforming to these requirements will ensure that your mobile site is fully accessible to all users.

## A note on WCAG2.1

The Committee decided that it was important to also include issues added in WCAG2.1 so that testers who were testing against WCAG2 would also benefit from the mobile-related errors published in WCAG2.1. This methodology is more stringent than WCAG2 and WCAG2.1 in some areas, and these are detailed in the Test Cases documents.

## Mobile sites versus native apps

There is a great difference between mobile sites and native apps – native apps utilize a completely different codebase. Therefore, the ICT Accessibility Testing Symposium has decided to separate the native app methodology from the mobile site testing methodology.

## Relationship with other Mobile Site documents

There are two overview documents:

* Mobile Site Methodology (**this current document**)
* About Mobile Site Testing – Devices, assistive technologies, site types, variations of a page and capturing errors.

There are three sets of test cases documents, which detail how to test a particular requirement in the methodology, why it is important and example failures and passes of the requirement.

* Mobile Site Critical Test Cases
* Mobile Site Test Cases
* Mobile Site Assistive Technology Test Cases

# Mobile Site Testing Methodology

## Overview

Please note that this methodology does not include those errors already included in WCAG2. In order to ensure your mobile site is fully accessible, you need to meet WCAG2 and this mobile site and native app testing methodology.

The following steps should be followed to ensure that a mobile site is accessible to users

* Step 1: Identify devices (page 5)
* Step 2: Identify site type and variations (page 6)
* Step 3: Test critical issues (page 8)
* Step 4: Test mobile issues (page 9)
* Step 5: Test mobile assistive technology and feature support (page 12)

## Step 1: Identify devices

What needs to be tested is dependent on the site type, variations of the page and chosen devices.

### Identify devices

Recommended devices and browser combinations:

* iPhone, Safari
* iPad, Safari
* Android phone, Chrome

#### Other devices to consider

* Android tablet (for example, Samsung Tab A or Chromebook), Chrome
* Alternative devices such as a Kindle device

#### Recommendations

Test on the latest version of iOS.

Test on latest two versions of Android.

Where a site is directly aimed at people with a particular kind of disability, it is worth considering including assistive devices and/or other assistive technologies used by potential users. Including these devices will help to ensure that a tested page is not only accessible, but also user-friendly. A good example of such a site is [Bookshare](https://www.bookshare.org/cms/), an online library for people with reading disabilities. Among the users of this service you can find users of braille notetakers and other mobile devices, and their user experiences are likely to be different from the experience of standard screen reader users.

## Step 2: Identify the site type and variations of the page

There are three different site types:

* **Desktop websites:** non-responsive sites that have only one site view or display, whether viewed on desktop or mobile or tablet device – a viewer would see the exact same thing;
* **Responsive websites:** user views or experiences that change depending on the screen size or other feature as determined in break points by the developer – usually the mobile site is simpler than the desktop website; and
* **m.dot sites:** sites that have a particular display for mobile and tablet. The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.

The testing methods will be dependent on what site type is being tested, and, in the case of responsive sites, how variations in content are controlled.

#### Desktop website testing

The desktop website must be tested against the entirety of WCAG2 on the mobile.

##### Recommended testing:

* WCAG2 testing of the desktop site on the desktop
* WCAG2 testing of the desktop site on mobile
* This mobile methodology of the desktop site on the mobile and desktop

#### m.dot site testing

The m.dot site must also be tested against the entirety of WCAG2, **in addition** to the standard www version of the site.

##### Recommended testing:

* WCAG2 testing of the www site on the desktop
* WCAG2 testing of the www site on mobile
* This mobile methodology on the www site on mobile and desktop
* WCAG2 testing of the m.dot site on the desktop
* WCAG2 testing of the m.dot site on mobile
* This mobile methodology on the m.dot site on mobile and desktop

#### Responsive website testing

It is important that **each variation of the page is tested**, and that **all functionality is available on all variations of the page**. The testing methods for responsive website testing are dependent on whether there are variations of the page. All versions of the site on the various devices will be required for comprehensive testing. User experience and accessibility can vary greatly among these differentiators; never apply results from one situation to another.

Determine which of the following triggers the variation of the page:

* The **device** (e.g. iPhone, desktop, Android, etc.);
* The **operating system** (e.g. Windows, iOS, Android, etc.);
* The **browser** (e.g. Safari, IE 11, Chrome, etc.); and
* The **screen size** (e.g. 280 by 720, 1920 by 1080, 320 by 480, etc.).

##### Recommended testing:

* WCAG2 testing of each variation on the desktop
* WCAG2 testing of each variation that can be accessed on mobile
* This mobile methodology of each variation on mobile and desktop

## Step 3: Test critical issues

### Exit traps

#### Applies to: All users

Ensure there is always an accessible actionable item (e.g. a close button that meets color contrast requirements and has an accessible name) that closes any feature or page that overlays the current page (such as a full-page ad) or an important mobile feature.

### Swipe / scroll traps

#### Applies to: Touch users

Ensure you do not override standard mobile touch functions (swiping, scrolling, etc.) on the majority of the page unless the entire screen is a map.

### Layer trap

#### Applies to: All users (but mostly encountered by screen reader users)

The user should not be trapped on a non-visible layer.

## Step 4: Test mobile issues

### Alternatives

Functionality that can be operated by device motion or user motion, interaction and/or gesture can also be operated by user interface components, and responding to the motion, interaction and/or gesture can be disabled to prevent accidental actuation, except for certain situations (for more information see [SC 2.5.4: Motion Actuation](https://www.w3.org/WAI/WCAG21/Understanding/motion-actuation.html)).

Any touch gesture must have an alternate gesture, such as a link (for more information see [SC 2.5.1: Pointer Gestures](https://www.w3.org/WAI/WCAG21/Understanding/pointer-gestures.html) and SC [1.4.13: Content on Hover or Focus](https://www.w3.org/WAI/WCAG21/Understanding/content-on-hover-or-focus.html)).

Alternatives are provided for geolocation functionality that is mandatory (for example, requiring a specific geolocation before functionality appears), unless the geolocation is essential for legal reasons, or doing so would invalidate the activity. This applies to geolocation via GPS, user statement, IP address or other methods.

Changes of state of non-standard controls (e.g. hamburger menu, star ratings) are clearly indicated.

Audio cues have an equivalent, accessible, visual cue.

Status messages are available to all users without receiving focus (for more information see [SC 4.1.3: Status Messages](https://www.w3.org/WAI/WCAG21/Understanding/status-messages.html)).

All abbreviations are expanded the first time they are used on the page or a glossary of abbreviations and their expansions is provided (for more information see [SC 3.1.4: Abbreviations](https://www.w3.org/TR/UNDERSTANDING-WCAG20/meaning-located.html)).

Where the text requires reading ability more advanced than Flesch Kincaid level 8, a summary or description of content is provided (for more information see [SC 3.1.5: Reading Level](https://www.w3.org/TR/UNDERSTANDING-WCAG20/meaning-supplements.html)).

Controls, primary headings, links, field labels and page titles are not ambiguous when read aloud.

### Display

Web pages do not contain more than three flashes in a one second period (for more information see [SC 2.3.2: Three Flashes](https://www.w3.org/TR/UNDERSTANDING-WCAG20/seizure-three-times.html)).

Changes of context must always be user-initiated unless it is time-sensitive or an emergency (for more information see [SC 3.2.5: Change on Request](https://www.w3.org/TR/UNDERSTANDING-WCAG20/consistent-behavior-no-extreme-changes-context.html)).

Size of touch targets is at least 44 by 44 CSS pixels (approximately 7 to 10 millimeters) (for more information see [SC 2.5.5: Target Size](https://www.w3.org/TR/WCAG21/#target-size)).

Touch targets have sufficient inactive space between them (inactive space of at least 22 by 22 CSS pixels, or 4 to 5 mm, should be provided around active elements).

Do not use fixed size containers for blocks of text, unless the display is essential.

Justified text has not been used.

Text and actionable items (including text and non-actionable items that will become actionable) or items that convey meaning should have a minimum color contrast ratio of 4.5:1 when compared with the surrounding background.

The site can be used in portrait mode (for more information see [SC 1.3.4: Orientation](https://www.w3.org/WAI/WCAG21/Understanding/orientation.html)).

The site can be used in landscape mode (for more information see [SC 1.3.4: Orientation](https://www.w3.org/WAI/WCAG21/Understanding/orientation.html)).

The site does not swap orientation unexpectedly.

Animation triggered by interaction can be disabled (for more information see [SC 2.3.3: Animation from Interactions](https://www.w3.org/WAI/WCAG21/Understanding/animation-from-interactions.html)).

Pinch zoom is operable, unless an accessible font resizing feature has been included in the website that allows the user to increase the size of content at least two times the size of the standard font size.

Horizontal scrolling is not required at all when the page is resized.

### Actionable items

When additional content appears on hover, focus or input, it is dismissable, hoverable and persistent (for more information see [SC 1.4.13: Content on Hover or Focus](https://www.w3.org/WAI/WCAG21/Understanding/content-on-hover-or-focus.html)).

Native UI controls, objects, alerts and elements have been used.

Where text links are used, the text is visually descriptive (for more information see [SC 2.4.9: Link Purpose (Link Only)](https://www.w3.org/TR/UNDERSTANDING-WCAG20/navigation-mechanisms-link.html)).

When direct input via the keyboard is not required, provide options for the user to achieve the same result (i.e. use dropdown, radio buttons and checkboxes, etc.).

Infinite scrolling has not been used.

Color alone should not be used to indicate actionable items (if not underlined) within inline text. A secondary method, such as underlines, should be used in addition to color.

Actionable elements are triggered only on removal of touch (for more information see [SC 2.5.2: Pointer Cancellation](https://www.w3.org/WAI/WCAG21/Understanding/pointer-cancellation)).

### Navigational aids

Visual indicators (such as arrows, next and previous buttons) have been used to indicate swipe or scroll areas or additional functionality.

Single character key shortcuts can be turned off, modified by the user or are active only on focus (for more information see [SC 2.1.4: Character Key Shortcuts](https://www.w3.org/WAI/WCAG21/Understanding/character-key-shortcuts.html)).

Blocks of content have descriptive headings (for more information see [SC 2.4.10: Section Headings](https://www.w3.org/TR/UNDERSTANDING-WCAG20/navigation-mechanisms-headings.html)).

Prior to starting a process, users are warned if there is a timeout and the length of time of inactivity that will trigger the timeout. Please note you will also need to comply with WCAG2 SC 2.2.1 Timing Adjustable (for more information see [SC 2.2.6: Timeouts](https://www.w3.org/WAI/WCAG21/Understanding/timeouts.html)).

Navigation features such as breadcrumbs, next and previous buttons are provided (for more information see [SC 2.4.8: Location](https://www.w3.org/TR/UNDERSTANDING-WCAG20/navigation-mechanisms-location.html)).

ARIA document landmarks have been used to appropriately describe document structure.

### Audio and video

All audio and video have an accessible transcript (for more information see [SC 1.2.8: Media Alternative (Prerecorded)](https://www.w3.org/TR/UNDERSTANDING-WCAG20/media-equiv-text-doc.html)).

Captions must be Closed Captions.

Live captions and audio descriptions are provided for any live audio or video.

### Forms

Visual and audio CAPTCHAs are not used.

All complex forms contain context-sensitive help as instructions at the beginning of the form and/or specific instructions at each field (for more information see [SC 3.3.5: Help](https://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-context-help.html)).

All submitted forms are reversible by the user, checked for errors by the application or confirmed prior to submission (for more information see [SC 3.3.6: Error Prevention (All)](https://www.w3.org/TR/UNDERSTANDING-WCAG20/minimize-error-reversible-all.html)).

Field labels are positioned adjacent to their input field and appear closest to their respective input field in relation to other field labels and other input fields.

Fields must have an associated visible label which is also programmatically associated with the field (please note that placeholding characters do not meet this requirement) (for more information see [SC 2.5.3: Label in Name](https://www.w3.org/WAI/WCAG21/Understanding/label-in-name.html)).

If there is a visible text label for an input field or component, the accessible name matches the visual name (for more information see [SC 2.5.3: Label in Name](https://www.w3.org/WAI/WCAG21/Understanding/label-in-name.html)).

Forms interact appropriately with the keyboard – for example, providing submission on Enter and moving between fields.

All HTML5 INPUT TYPES and AUTOCOMPLETE values are used according to the specification and where they are applicable (for more information see [SC 1.3.5: Identify Input Purpose](https://www.w3.org/WAI/WCAG21/Understanding/identify-input-purpose.html) and particularly the list of [input types that should be used](https://www.w3.org/TR/WCAG21/#input-purposes)).

### Mobile / desktop interaction

Item labelling between different types of a site (desktop, m.dot and/or responsive) and different variations of a responsive site is consistent.

Links between different types of a site (desktop, m.dot and/or responsive) have been provided where the site is not solely a responsive site.

## Step 5: Test mobile assistive technology and feature support

All actionable items and important content can be accessed and activated by assistive technologies or mobile features.

### Assistive technologies and mobile features

#### iOS

##### iPhone

* VoiceOver
* Keyboard
* Switch
* Zoom
* Reduce Motion
* Invert colors
* Grayscale
* Reader View and increase text

##### iPad

* VoiceOver
* Keyboard
* Switch
* Zoom
* Reduce Motion
* Invert colors
* Grayscale
* Reader View and increase text

#### Android

* TalkBack
* Keyboard
* Switch
* Magnification
* Remove Animations
* Color Inversion
* Grayscale
* Simplified view
* Increase text size with Android Chrome

##### Samsung (optional)

* Voice Assistant

##### Kindle / Amazon Firestick (optional)

* Voice View

# Acknowledgements

## Relationship to existing Accessibility testing standards

This document is based on:

* W3C Web Content Accessibility Guidelines, Version 2.0
* W3C Web Content Accessibility Guidelines, Version 2.1
* BBC Mobile Accessibility Guidelines
* AccessibilityOz Mobile Testing Methodology
* TPG Mobile Testing Guide

## ICT Accessibility Testing Symposium Mobile Site Sub-Committee

This document was developed by the ICT Accessibility Testing Symposium Mobile Sub-Committee. Members include: Gian Wild (Co-Chair), Peter McNally (Co-Chair), Brent Davis, Corbb O'Connor, Karen Herr, Kathryn Weber-Hottleman, Kathy Eng, Laura Renfro, Megha Rajopadhye, Mona Rekhi, Morgan Lee Kestner, Rafal Charlampowicz, Ryan Pugh, Steve Sawczyn, Sunish Gupta, Tom Lawton and Chris Law.

# Contacts

## Gian Wild

**Web:** [www.accessibilityoz.com](http://www.accessibilityoz.com)

**Phone:** 415 621 9366

**Email:** [gian@accessibilityoz.com](mailto:gian@accessibilityoz.com)

## Peter McNally

**Web:** [www.bentley.edu](http://www.bentley.edu)

**Phone:** 781 891 2893

**Email:** [pmcnally@bentley.edu](mailto:pmcnally@bentley.edu)