Method: Character Key Shortcuts Method Template 10/2021

Instructions document:
Writing Testable WCAG 3.0 Outcomes
For
2.14 Character key shortcuts - 3.0 Style 2021-05-06 Kim

Introduction

Outcome

This method supports the outcome: Single key shortcuts for mobility issues.

Platform

- Content implemented in any technology

Technology

- HTML and JavaScript

Input aspects for testing

- Things you need to identify before running the test. In ACT, HTML and CSS are aspects for test
- Understanding the language is an input aspect
- SIght may be an input aspect
- [technical sources for input, such as DOM tree, Accessibility tree and CSS styling]

Summary

(from WCAG 2.1 Technique G217)

Some authors provide shortcuts in their applications to allow for faster user interaction. However, where such shortcuts involve only character keys (letters, numbers, punctuation or symbol characters) without modifiers, they create challenges for speech input users as well as some keyboard users who lack the ability to type accurately.

To prevent accidental activation, authors must allow users to turn off or reconfigure shortcuts that are made up of only character keys. Reconfiguring the shortcut may involve the ability to add a modifier key such as Ctrl, or authors may elect to allow users to alter the character keys assigned in addition to adding a modifer. Providing a mechanism to turn off or reassign the shortcut ensures more users can successfully interact with the application.

Note

If the keyboard shortcut is only active when a particular user interface component has focus, then the author does not need to provide an override mechanism. For example, in a select element, it is expected behaviour that pressing a letter key will reposition to the next item within the select element that begins with that character. This shortcut is only triggered when the select element has focus, and so it does not interfere with a user's ability to interact with the rest of the page.

An initial challenge for testers is identifying if author-created shortcuts exist, and then determining if existing shortcuts are triggered by character keys without modifiers. Where testers have access to the develop team, the existence and nature of shortcut keys can often be determined simply by asking involved designers and developers. Where reliable information from the authors is not available, the presence of shortcuts can possibly be identified by checking code (for example, in javascript, the presence of keydown, keyup and keypress listeners). Another means of identifying shortcuts is to review documentation. Where none of these strategies provide information on the presence of keyboard shortcuts, manual tests will need to be completed to search for the existence of character key shortcuts. Review the test procedure in F99: Failure of Success Criterion 2.1.4 due to implementing character key shortcuts that cannot be turned off or remapped for guidance.

How it solves user need

(from Understanding/character-key-shortcuts Benefits section)

- Speech users will be able to turn off single-key shortcuts so they can avoid accidentally firing batches of them at once. This will allow speech users to make full use of programs that offer single-key shortcuts to keyboard users.

- Keyboard-only users who have dexterity challenges can also be prone to accidentally hitting keys. Those users would be able to avoid problematic single character shortcuts by turning them off or modifying them to include at least one non-character key.
- Allowing *all* shortcut keys to be remapped can help users with some cognitive disabilities, since the same shortcuts can be assigned to perform the same actions across different applications.

When to use

This method should be used anywhere there are singlekey shortcuts implemented for users.

Background

W3C Resources

- {related resources}

Non-W3C Resources

- {related resources}

Accessibility Support

Accessibility support from ACT

Single key shortcuts should not override assistive technology.

Assumptions

- [assumptions]

Examples

- {list of examples, unless using sub-sections, then delete this list}

Passed

(from WCAG 2.1 Technique G217)

Example 1:

Author provides a simple mechanism to disable shortcuts in a web-based email client

An email client contains single-character shortcuts so that keyboard users can more rapidly interact with the application (e.g., the R key replies to the current item in the inbox, the D key deletes it, the S key stars it). These keys do not apply while users are composing emails; however, users may become confused about whether the keyboard focus is in a compose pane or the inbox and, while in the inbox, inadvertently delete a message by trying to compose a word containing the letter D. To overcome this problem, the author provides a toggle button for the shortcuts, allowing users to easily turn them off or on.

Example 2:

Author provides a menu option where users can remap shortcuts

A web-based application has single-character shortcuts for functions which are different than those a user regularly uses with a software application installed locally. Since the author has provided a menu where the user can reassign both the shortcut keys and the modifiers, the user is able to reassign the shortcut to a familiar combination.

Failed

{Example Name}

{Use a copy of this section for each example if not using a bullet list of examples}

{explanation}

<

Inapplicable

{Example Name}

{Use a copy of this section for each example if not using a bullet list of examples}

{explanation}

Tests

Get Started

{Tips or link to information for beginners in testing.}

Summary

[overview of the test]

Applicability

This outcome applies to any [element names] element that is [condition] and for which one of the following is true:

{excluded, ignored, exception}

[element]

Expectations

[detail of the expectation]

Glossary

[term]

definition

[term]

definition