Four Things You Should Know Before Testing with VoiceOver on Mac

Jonathan Whiting

WebAIM.org
6807 Old Main Hill
jonathan@webaim.org

Abstract

The VoiceOver screen reader is a very capable screen reader that is used by many developers to test for screen reader accessibility. However, there are four important things that web developers and accessibility evaluators should understand before testing with VoiceOver:

1. The majority of desktop screen reader users are on Windows.
2. VoiceOver does not have a forms or application mode.
3. Default keyboard accessibility settings must be changed.
4. ARIA descriptions are read after a very long pause in older versions of VoiceOver.

Introduction

Web professionals are increasingly using Macs. As they become more aware of the importance of web accessibility, more and more of them are using VoiceOver to test for screen reader accessibility. This is excellent news, but there are at least four important differences between VoiceOver and most Windows screen readers that accessibility testers should understand.

1. The majority of desktop screen reader users are on Windows.

WebAIM regularly conducts surveys of screen reader users. The most recent WebAIM screen reader user survey (webaim.org/projects/screenreadersurvey6/) was conducted in July 2015, with over 2500 respondents. Of the respondents, 93% use a screen reader due to a disability. While this sample was not controlled and may not represent all screen reader users, it is still a valuable resource since more rigorous large-scale studies are not available at this time.

When respondents were asked which screen readers they use on their desktop or laptop, 8% of respondents indicated that VoiceOver was their *primary* screen reader and 31% reported that they *commonly* use VoiceOver (rounded to the nearest whole number). While these numbers suggest VoiceOver is a fairly common screen reader, the fact is that over two-thirds of respondents do *not* commonly use the Mac Operating System. Although you can gain a great

deal of insight when testing with VoiceOver, relying on it alone will not give you a complete picture of screen reader accessibility.

Although this paper deals with VoiceOver for Mac, it is worth noting that VoiceOver is by far the most common *mobile* screen reader. In this same survey, 57% of respondents indicated they commonly used VoiceOver for their mobile needs. TalkBack for Android was a distant second at 18%.

2. VoiceOver does not have a forms or application mode.

Windows screen readers have different interaction modes when they are reading through content vs. interacting with forms and widgets (Watson, 2014). These modes have different names in each screen reader, but for the sake of consistency, they will be called ‘read mode’, ‘forms mode’, and ‘application mode’.

Read mode

In read mode, the arrow keys are used for reading. For example, the down arrow reads the next item or section while the right arrow reads the next letter. There are also a number of shortcut keys that can be used to jump to common page elements. For example, the H key moves to the next heading and T moves through tables. On most websites, screen reader users spend the majority of their time in this mode.

Forms mode

When navigating into a form control, the screen reader typically uses an audio tone to indicate that it has entered forms mode. This mode allows the user to interact with form controls using the keyboard. Pressing the H key in forms mode, for example, will type this letter into a text box instead of moving to the next heading. Pressing the down arrow will select the next radio button in a group instead of reading the next item. A different audio tone announces when a screen reader leaves forms mode and returns to read mode. Most screen reader users are familiar with this mode.

Application mode

In order to interact with many online “widgets,” the screen reader must stop using keystrokes for reading and instead pass them on to the browser. For example, in an accessible tree menu, users should be able to move between sections with the up and down arrow keys and expand or collapse these sections with the right and left arrow keys (keystrokes normally reserved for reading).

Functionally speaking, forms and application mode are the same. In fact, they are the same in some screen readers. However, for developers and testers, there is still an important difference: HTML form controls trigger this mode automatically, and this process is understood by most screen reader users. For application widgets, this mode must be triggered with the correct ARIA roles. This is not as well understood by many screen reader users, and this confusion is at the heart of many accessibility issues.

ARIA, screen reader modes, and VoiceOver

Certain ARIA roles will tell a screen reader to change from reading mode to application mode. These roles include `menu`, `menubar`, `tab`, `tablist`, `grid`, `toolbar`, and of course `application`. ARIA roles that trigger application mode are often misused by well-meaning developers. The misuse of ARIA and lack of understanding of screen reader modes currently results in significant accessibility issues on the web. The prevalence of these issues is on the rise.

For example, if a group of links at the top of a webpage have incorrectly been assigned an ARIA `role="menu"`, this triggers application mode and the screen reader user cannot read through this area. Alternatively, if a developer creates a menu that is controlled with the arrow keys and it does not have the appropriate ARIA roles, a screen reader user may move right past it without knowing what it is or how to interact with it.

VoiceOver does not have the single-key controls that are common in Windows screen readers. For instance, to move to the next heading in a Windows screen reader, you press the H key. To move to the next heading in VoiceOver, you use command + control + option + H. To read the next item in VoiceOver, the user presses control + option + right arrow. Because of these required key combinations, there is no need for VoiceOver to toggle between modes. This is good news for VoiceOver users, who may not experience issues when the above roles are misused. However, this is not ideal for testing because an incorrect ARIA role that can make a webpage highly inaccessible for the majority of screen reader users may go undetected when testing with VoiceOver.

3. Default keyboard accessibility settings must be changed.

In modern Windows browsers, pressing the tab key will move through all links and form controls (text fields, checkboxes, buttons, etc.). But by default, not all of these elements receive keyboard focus on a Mac. Mac keyboard accessibility can also vary when VoiceOver is running and when it is disabled. To ensure consistent accessibility across all major Mac browsers, do the following two things:

Enable full keyboard accessibility in System Preferences.

To enable full keyboard access, select **System Preferences** > **Keyboard** > **Shortcuts**. At the bottom of the page, there is a setting that reads “Full Keyboard Access: In windows and dialogs, press Tab to move keyboard focus between:”. By default, the “Text boxes and lists only” option is selected. Change this to “**All controls**”.

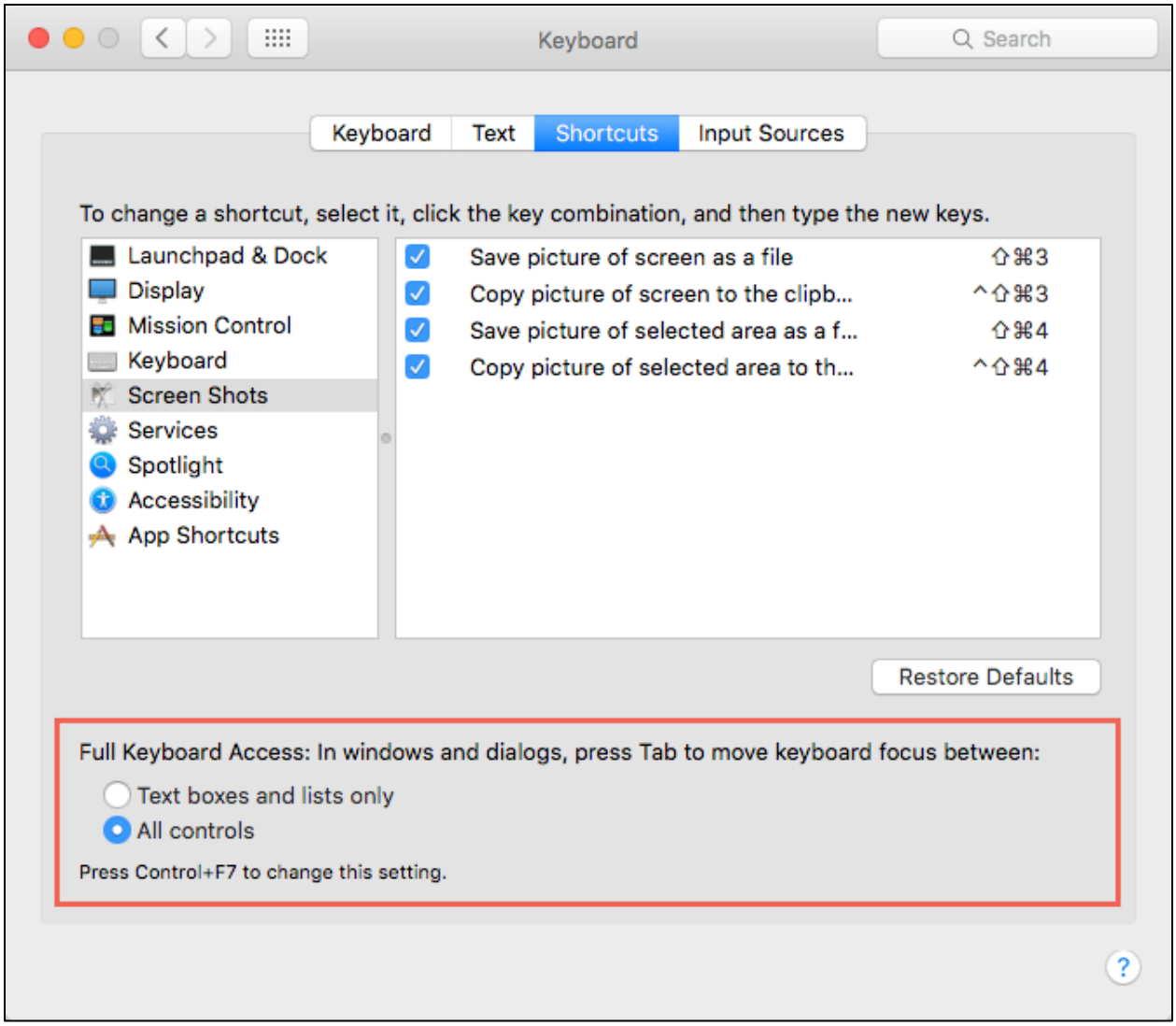

Figure 1: Screenshot of keyboard access settings.

Note: If this setting is not changed manually, it will automatically change to “All controls” every time VoiceOver opens and then revert back to “Text boxes and lists only” when VoiceOver closes. That change in behavior can be very confusing when testing keyboard accessibility without a screen reader.

Enable full keyboard accessibility in Safari.

Historically, accessibility support has been best in the Safari browser. However, links cannot receive keyboard focus by default in Safari, even with the above system preference enabled. To address this, in the **Safari** menu, select **Preferences** > **Advanced** > **Accessibility** and check “**Press Tab to highlight each item on a webpage**”.

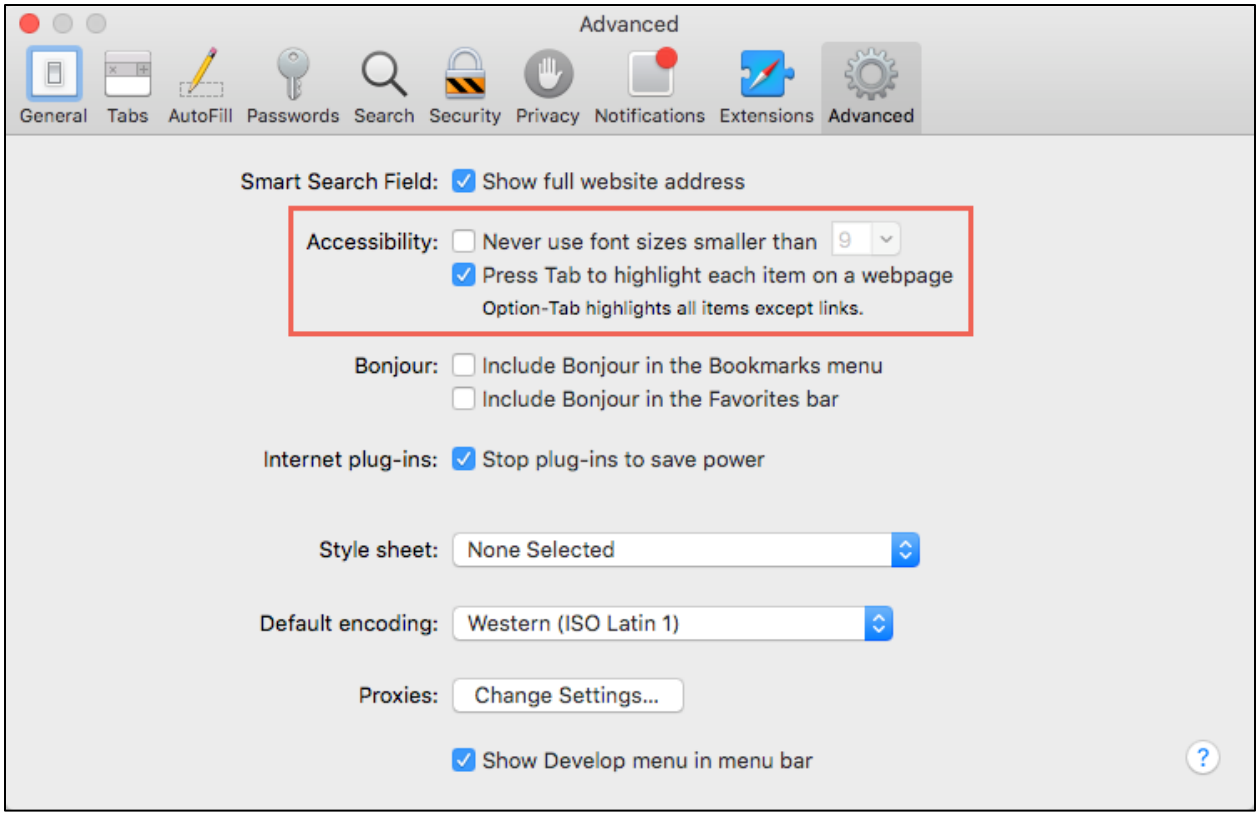

Figure 2: Screenshot of Safari keyboard accessibility settings.

(Becka11y, 2014)

4. ARIA descriptions are read after a very long pause in older versions of VoiceOver.

Often there are times when a text description should be associated to an element (usually a form control). For example, below in Figure 3 there is a form field with a message of "Must be 8-15 characters and have letters and numbers":

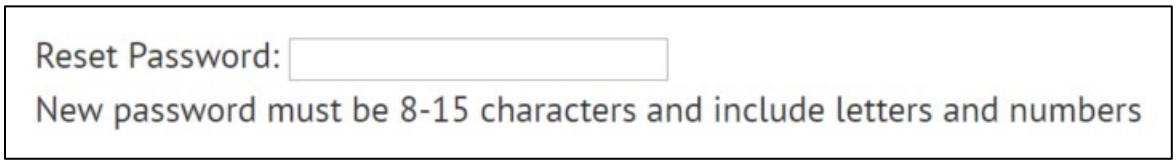

Figure 3: An example of a form field that has a label and description.

This additional information should be read by a screen reader when the password field is encountered, which is accomplished with the `aria-describedby` attribute.

In older versions of the Mac operating system, there is an issue when reading ARIA descriptions—there is a very long seven second delay after a control is focused before reading

ARIA descriptions. This can cause users and testers to miss this important information. It also leads to a common but incorrect practice of avoiding `aria-describedby` completely and using `aria-label` or `aria-labelledby` to present labels and descriptions. Fortunately, this issue was addressed in a September 2015 update (OS X 10.11, El Capitan).

If you encounter this issue during testing and you cannot update to a newer version of macOS, this delay can be shortened to less than half a second with VoiceOver settings. To shorten this delay, open your **System Preferences** and choose **Accessibility > VoiceOver > Open VoiceOver Utility > Verbosity > Hints**, and then drag the slider down on "**Speak hints after delay**".

Conclusion

VoiceOver is a very powerful screen reader and its usage is on the rise. In order to most effectively use VoiceOver to test web accessibility and to best ensure an accessible experience for all screen reader users, VoiceOver users should understand these differences and consider additional testing in Windows screen readers.

References

Beca11y (2014, July 7). Keyboard Navigation in Mac Browsers. Retrieved from http://www.weba11y.com/blog/2014/07/07/keyboard-navigation-in-mac-browsers/

King, Matt (2016, September 11) Re: Three things you should know before using VoiceOver for testing [Blog comment]. Retrieved from http://webaim.org/blog/three-things-voiceover/

Watson, L. (2014, September 21). Understanding screen reader interaction modes. Retrieved from http://tink.uk/understanding-screen-reader-interaction-modes/

WebAIM (2015, August 28). Screen Reader User Survey #6 Results. Retrieved from http://webaim.org/projects/screenreadersurvey6/

Copyright notice