Understanding Focus Not Obscured

Status

This understanding document is part of the **draft** WCAG 2.2 content. It may change or be removed before the final WCAG 2.2 is published.

Focus Not Obscured Success Criterion text

> When a user interface component receives keyboard focus, the component is not entirely hidden due to author-created content.

NOTE:

If the interface is configurable so that the user can reposition content such as toolbars and non-modal dialogs, then only the initial positions of user-movable content would be considered for testing and conformance of this Success Criterion.

NOTE:

Where content disclosed by the user can obscure other content, if the user can reveal the item with focus without moving keyboard focus (for example, by dismissing with Esc), such disclosed content is not considered to fully obscure the item receiving focus.

Intent of Focus Not Obscured

The intent of this Success Criterion is to ensure that the item receiving keyboard focus is always visible in the user's viewport. For sighted people who rely on a keyboard (or on a device that operates through the keyboard interface, such as a switch or voice input), knowing the current point of focus is critical. The component with focus signals the interaction point on the page. Where users cannot see the item with focus, they may not know how to proceed, or may even think the system has become unresponsive.

In recognition of the complex responsive designs common today, this AA criterion allows for the component receiving focus to be *partially* obscured by other author-created content. A partly obscured component can still be very visible, although the more of it that is obscured, the less easy it is to see. For that reason, authors should attempt to design interactions to reduce the degree and frequency with which the item receiving focus is partly obscured. For best visiblity, *none* of the component receiving focus should be hidden. This preferred outcome is covered by the AAA criterion Focus Not Obscured (Enhanced).

Typical types of content that can overlap focused items are sticky footers, sticky headers, and non-modal dialogs. As a user tabs through the page, these layers of content can obscure the item receiving focus, along with its focus indicator.

A notification implemented as sticky content, such as a cookie banner, will fail this Success Criterion if it entirely obscures a component receiving focus. Ways of passing include making the banner modal so the user has to dismiss the banner before navigating through the page, or using scroll padding so the banner does not overlap other content. Notifications that do not require user action could also meet this criterion by closing on loss of focus.

Another form of obscuring can occur where light boxes or other semi-opaque effects overlap the item with focus. While less than 100 percent opacity is not causing the component to be "entirely hidden," such semi-opaque overlaps may cause a failure of 2.4.11 Focus Appearance. When a focus indicator can be covered by a semi-opaque component, the ability of the focus indicator to pass 2.4.11 should be evaluated (and pass) while the focus indicator is under the semi-opaque component. The intention in both situations is that the component receiving focus should never be obscured to the point a user cannot tell which item has focus.

Considerations

The following sub-sections provide clarity on how to assess Focus Not Obscured (Minimum) in certain scenarios.

User-movable

This SC contains a note regarding content that can be repositioned. If users can move content regions, then they can potentially position them such that they obscure other content that may receive focus. In such a case, the author is only responsible for ensuring that the movable content *in its initial position* does not obscure the item receiving focus.

This note is intended to accommodate a common interaction in complex applications such as authoring tools, where the main editing region (also called a canvas) can be enhanced by displaying toolbars or other panels, which can be repositioned around the canvas. It is possible to design such toolbars so they do *not* obscure focus. Authors are encouraged to do so, as well as pursue techniques which ensure equitable keyboard use of such toolbars. However, in recognition of the complexities involved in responsive design as well as in supporting the ability to transform the text size and spacing of content, only the starting position of such side panels is assessed.

Modal dialogs

A properly constructed modal dialog will always pass this SC. Even if it appears directly on top of an item with focus, the dialog takes focus on appearance, and thus the item receiving focus -- the dialog or one of its components -- is visible. A properly constructed modal maintains that focus and prevents interaction outside the modal until it is dismissed.

A dialog-like overlay that does not take focus on appearance and does not either constrain interaction to the overlay or dismiss itself on loss of focus (thus allowing focus to exit into the content behind it) will be at risk of failing this SC, where it is positioned such that it can obscure other focusable items.

User openable, persistent disclosures

Some disclosure patterns provide a mechanism for the user to open additional content that remains open until intentionally closed by the user. Accordions are a simple example of such a pattern. Chatbots and expandable side navigation are more complex examples. All of these patterns can be implemented so they are not at risk of failing this SC. Some possible approaches are:

- **When the additional content appears, it displaces existing content.** An accordion is an example of this. When an accordion is opened, the disclosed content shifts existing content further down the page. Since the new content does not obscure existing content, it cannot obscure the item with focus.
- **When the additional content appears, existing content reflows.** The popout sidebar on the WCAG standard is an example of this pattern. When the side menu is activated, it discloses a new section of information along the left side of the page. The main content area is reduced horizontally to accommodate this, and the existing main content reflows to fit in the thinner space.

As a result, there is no overlapping content between the two sections; the item receiving focus, whether in the left navigation or in the main content, will not be obscured by the other section.

- **When the additional content is disclosed, it takes focus and the tab ring is constrained to the new content until it is dismissed.** This modality is somewhat like a dialog, in that a user cannot navigate beyond the disclosed content by keyboard without dismissing it first (typically by pressing Esc). However, unlike in a modal dialog, in some implementations a pointer user may be able to interact with content outside the disclosed section without dismissing it. Since this pattern potentially creates an inequitable experience between keyboard and pointer users, it should be used cautiously. That said, it does prevent the disclosed content from obscuring the keyboard focus in the main content, and thus should pass this SC.
- **The disclosure expands into an area of the page containing no other content.** Many pages are designed with wide margins, providing significant white space into which new content can be disclosed. Many chatbots and toasters are designed to 'slide up' into the right unpopulated side of a page. Where authors are careful to ensure content is not obscured at each breakpoint in a responsive design, no obscuring of other operable content need occur.
- **When focus leaves the additional content, the additional content is hidden or collapsed.** This is very similar to patterns discussed next under Non-persistent disclosed information. A distinguishing factor can be that the user's last point of interaction in the disclosure is preserved (it persists) even though it may be hidden until a user returns. Some trees and left navigation pattterns behave this way.

Where disclosed content persists and causes existing content to be obscured, it will be at risk of failing this criterion if it prevents users from seeing the item receiving focus.

Non-persistent disclosed information

A number of components on the web disclose additional content (on activation or on focus) intended for immediate user interaction or information. These disclosures often open on top of other content, obscuring it. Examples of such disclosures are menu items, select element items, combobox lists (and other dropdown items), date picker calendars, and tooltips. The common trait of all these disclosures is that they are only expected to persist until either acted on or no longer the primary point of user interaction. Such non-persistent disclosures do not fail this SC so long as they don't obscure the item with focus. However, if an author allows such disclosures to persist after the user has 1) activated one of the disclosed items or 2) moved the focus away from the triggering item and the disclosure, the additional content is at risk of failing this SC by obscuring the item with focus.

Benefits of Focus Not Obscured

- Sighted users who rely on a keyboard interface to operate the page will be able to see the component which gets keyboard focus. Such users include those who rely on a keyboard or on devices which use the keyboard interface, including speech input, sip-and-puff software, onscreen keyboards, scanning software, and a variety of assistive technologies and alternate keyboards.
- People with limited or low vision, who may primarily user a pointer for screen orientation and repositioning, nonetheless benefit from a visible indication of the current point of keyboard interaction, especially where magnification reduces the overall viewing portion of the screen.
- People with attention limitations, short term memory limitations, or limitations in executive processes benefit by being able to discover where the focus is located.

Examples of Focus Not Obscured

- A page has a sticky footer (attached to the bottom of the viewport). When tabbing down the page the focused item is not hidden by the footer.

Resources

Techniques for Focus Not Obscured

Sufficient Techniques

1. CSS: Using scroll-padding to ensure a sticky header does not entirely obscure the focused item (Potential future technique).

Additional Techniques (Advisory)

Failures

1.