

Hidden controls working document subgroup

David MacDonald , John Kirkwood, Sarah Horton, Michael Gower

Proposal

Fallback 2: Drop "and understand "and "at the point of interaction"

Information needed to recognize how to operate user interface components is provided to users without requiring pointer hover or keyboard focus.

Exceptions:

- Hiding the information needed to identify the control is essential
- The control is provided specifically to enhance the experience for keyboard navigation. (e.g., skip links)
- The information needed to identify the user interface components is available through an equivalent control that is visible on the same page or on a different step in a multi-step process without requiring pointer hover or keyboard focus.
- The visibility of the control is determined by the user agent and not modified by the author
- A mechanism is available to make the information persistently visible.

NOTE: controls can be available via a visible entry point such as a submenu, etc.

Success Criterion 3.2.7 Visible User Interface Components (Level AA)

Information needed to identify user interface components is visible without requiring pointer hover or keyboard focus, <add>as triggers</add>.

Exceptions:

- Hiding the information needed to identify the control is essential
- The control is provided specifically to enhance the experience for keyboard navigation. (e.g., skip links)

- The information needed to identify the user interface components is available through an equivalent control that is visible on the same page or on a different step in a multi-step process without requiring pointer hover or keyboard focus.
- The visibility of the control is determined by the user agent and not modified by the author
- A mechanism is available to make the information persistently visible.

NOTE: controls can be available via a visible entry point such as a submenu, etc.

EDITOR'S NOTE: The working group is seeking input on the requirements for author supplied media controls that appear on hover. Should there be an explicit exception for media players? Or, should the current exception that allows an author to provide a mechanism to make them persistently visible enough of an exception?

Option

Pointer hover or keyboard focus are not the only ways to make visible Information that is needed to identify user interface components.

Possible replacement for bullet 3

Content can be operated by user interface components and does not require pointer hover or keyboard focus to make the user interface components visible+1

[There are a couple of fallbacks below which leave out "and understand" and "at the point of interaction"]

Rationale based on functional needs

This section shows the language of the proposed success criteria broken down, to illustrate how functional needs for different user groups are met by different requirements. For more detail, see the [User Needs](#) summary of the [Basecamp: Visual user favoring pointer/Edit interaction with text and checkbox that display control](#) brainstorming notes.

- Information needed to recognize and understand how to operate user interface components is (Common Needs)
- Provided without requiring pointer hover or keyboard focus (Common Needs)
- Provided at the point of interaction without requiring pointer hover (Vision & Visual: Use with limited vision)
- Provided without requiring pointer hover (Mobility: Use without mobility, Use with limited mobility, Use with limited reach or range)
- Provided without requiring pointer hover (Motor: Use without hands, Use with limited strength, Use without fine point control, Use with tremors)
- Provided at the point of interaction without requiring pointer hover (Attention: Use with limited ability to focus attention, Use with limited ability to direct attention, Use with limited ability to shift attention)
- Provided at the point of interaction without requiring pointer hover (Memory: Use with limited short-term or working memory, Use with limited medium or long-term memory)
- Provided at the point of interaction without requiring pointer hover (Executive: Use with limited planning, organization, sequencing, and execution ability)

Techniques

- Making a control persistently visible
- Providing an equivalent function offered through a persistently visible control
- Making the control is available via a visible entry point
- Providing a mechanism to make the control persistently visible
- Making a control visible as a result of a choice made by the user on a previous step of a multi step process.

NOTE: Controls that are outside of the viewport and are visible when scrolled into view with other visible content are persistently visible.

visible entry point: A dedicated control that opens to one or more sub groups of controls, which is visible on the page.

This could be in understanding:

NOTE: Text that serves another purpose(s) besides being a visible entry point, such as a person's name, that can be clicked to enter characteristics about that user, would not be a visible entry point control because:

- it is not on the page for the explicit purpose of opening more options. It has a primary purpose other than opening more options
- It is not visibly distinct as actionable

- It is not a label because it does not sufficiently describe the purpose of the control.

Possible fallbacks

Fallback 1: Drop the words "and understand"

Information needed to recognize how to operate user interface components is provided to users at the point of interaction without requiring pointer hover or keyboard focus.

Exceptions:

- [same as above]

Fallback 2: Drop "and understand "and "at the point of interaction"

Information needed to recognize how to operate user interface components is provided to users without requiring pointer hover or keyboard focus.

Exceptions:

- [same as above]

=====

[END OF READING FOR THE PROPOSAL - BELOW IS BACKGROUND AND EXISTING SC]

=====

Applicable design principles

From *Universal Principles of Design, Design of Everyday Things*

- **Visibility:** The usability of a system is improved when its status and methods of use are clearly visible. Design systems that clearly indicate the system status, possible actions, and consequences of actions. Without visibility: "Whenever the number of possible actions exceeds the number of controls, there is apt to be difficulty." (Donald Norman) The visible on pointer hover design pattern we are trying to address with this guidelines is one where actions provided through controls that are not persistently visible (i.e., that display on hover) and therefore, the possible actions are not visible.

- **Affordance/Conventions:** Affordance is a property in which the physical characteristics of an object or environment influence its function, like push plates on doors. With interfaces, affordances make possible actions visible by providing cues and hints, like buttons and underlined links. With the visible on pointer hover design pattern, possible actions are not made visible using affordances or conventions until they display on pointer hover.
- **Recognition Over Recall:** Memory for recognizing things is better than memory for recalling things. Minimize the need to recall information from memory whenever possible. With the visible on pointer hover design pattern, nothing is provided in the interface to help users recognize interactivity. To access functionality, users are relied on to recall that the controls display on hover.
- **Gulf of Execution:** “The gap between a user’s goal for action and the means to execute that goal.” https://en.wikipedia.org/wiki/Gulf_of_execution With our work to improve the accessibility of the visible on pointer hover design pattern, we aim to close the gap between the user’s goal to successfully access functionality and complete tasks and the need for clean designs (discussed below, under [Why this pattern?](#)).
- **Progressive Disclosure:** “A strategy for managing information complexity in which only necessary or requested information is displayed at any given time.” Progressive disclosure minimizes the complexity of an interface by providing access to primary controls and only revealing secondary controls when necessary or relevant (e.g., formatting controls that display when a text input area has focus, a dropdown menu control that displays secondary options when activated). With this SC, we aim to make sure the primary controls in this pattern are displayed without requiring pointer hover or keyboard focus.

User Needs

To ensure users can recognize that an element is a control and understand its purpose and how to operate it.

Common Needs

These user needs are based on a [walkthrough of the Basecamp edit to-do text task flow](#).

- User needs to know that the to-do text is interactive, so they know they can trigger the control to access the edit feature.
- User needs to know which controls aren’t relevant to the edit to-do text task, so they don’t activate the wrong feature.
- User needs to know that triggering either the to-do text or the show/hide text will activate the same edit feature, so they are not confused about how to access the feature.

- User needs to recognize controls and understand how they work, so they can activate the feature they are looking for.
- User needs to understand which controls are enabled, so they don't try to operate elements that are not actionable.

Functional Needs

Use with limited vision

1. User needs to be able to see/be aware of the affordances (cursor pointer, show/hide text), so they know that the to-do text is a control and understand the feature the to-do text control provides access to.

Use without mobility, Use with limited mobility, Use with limited reach or range, Use without hands

2. User needs information about what the to-do text control provides access to, so they can understand the feature the to-do control provides access to.
3. User needs to know that the to-do text is a control, so they can use speech to access the feature.
4. User needs to know that triggering the to-do text will not select the checkbox, so they have confidence they will not inadvertently check off the to-do.

Use with limited ability to focus attention, Use with limited ability to direct attention, Use with limited ability to shift attention

5. User needs to not be distracted by elements appearing and disappearing that are not at their point of focus and interaction, so they can focus on the task at hand.
6. User needs to not have to attend to elements that are not at the point of focus and interaction, so they can focus on the task at hand.
7. User needs to not have to hunt around an interface to locate a control, so they can focus on the point of interaction and follow the path toward task completion.

Use with limited short-term or working memory, Use with limited medium or long-term memory

8. User needs to recognize the to-do text as a control that provides access to the edit feature, so that when they want to edit a to-do, they know what to do.

Use with limited planning, organization, sequencing, and execution ability

9. User needs clear access the edit to-do feature, so the path to accessing the feature and updating the text is straightforward.

Examples Doc

https://docs.google.com/document/d/1Uh2ZsQXRIC_S2gtJE5mMSIHewB1K2gBBc0VL3hL4Yk/edit#

Issues list

<https://github.com/w3c/wcag/issues?q=is%3Aissue+is%3Aopen+label%3A%223.2.7+Hidden+Controls%22+-label%3A%22Survey+-+Added%22+-label%3A%22Duplicate%22+>

Current Success Criterion Wording

Success Criterion **3.2.7 Hidden Controls** (Level AA): Controls needed to progress or complete a **process** are visible at the time they are needed without requiring pointer hover or keyboard focus, or a mechanism is available to make them persistently visible.

Current Understanding

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.

Intent

The intent of this Success Criterion is to ensure that controls needed to progress or complete a process can be easily found by people with cognitive disabilities when they are needed.

People with low executive function, impaired memory, and other cognitive and learning disabilities may not be able to find controls needed to progress if they are hidden until focus is placed on them or a pointer hovers over them. They may also not remember where the control is the next time they interact with the site.

Some design approaches hide controls needed to complete tasks and require certain user interactions, such as mouse-over, to display these controls. These required interactions can leave users with cognitive disabilities without a path forward.

Controls should be visible without user interaction (i.e., persistently visible) when they are needed to progress. In multistep processes or multipart forms, the control may be hidden in an earlier step/part, however, at the time the user can move forward the control needs to be persistently visible without having interacted with the control. Similarly, this SC does not require inactive controls to be persistently visible until the user is able to progress. For example, when a form includes required fields be completed before the submit button becomes active, the submit button does not need to be persistently visible while it is inactive.

When multiple controls are included in a single location, some indication of those controls such as a drop down indicator should be visible without user interaction.

In some cases, controls are provided in multiple locations on a page or at multiple points within a process. In these cases, at least one of the instances of the controls needs to be visible without user interaction. For example, in an email process, some controls such as trash may be visible using pointer hover in the list of emails but are always visible on the view of the email itself. Because the controls are persistently visible when needed on the email view, they do not need to be persistently visible on the list of emails.

The intent of this SC is to help individuals with cognitive and learning disabilities locate controls they need to successfully complete their desired task. The controls addressed in this SC should also meet the SC for sufficient text (1.4.3/1.4.6) and non-text (1.4.11) contrast.

Controls such as video players, web chats, and carousels include controls that are only visible on hover since they overlay the contents being displayed. These controls are not considered a process in terms of this SC but occasionally completing a process requires interacting with one of these controls. For example, completing training may require the user watch a video before proceeding to the next step. In this case, a control needed to play the video needs to be visibly persistent. This may be

accomplished by making the video controls all visibly persist or by adding a redundant play control to the page.

Benefits

- Users with cognitive and learning disabilities can more easily locate controls that allow them to successfully progress through a process.
- Users with memory impairments do not need to remember where controls are located

Examples

- **Web Editor Success:** In a web editor, once a page is open for editing, all the possible locations for editing display a pencil icon.
- **Web Editor Success:** In a web editor, the possible locations for editing are hidden until a user hovers over them but a radio button in settings can be changed so that the icons are all shown by default.
- **Email Success:** When writing an email, controls needed to trash or forward the email are hidden by hover on the email list page but visually persist at the top of the email screen.

Techniques

Each numbered item in this section represents a technique or combination of techniques that the WCAG Working Group deems sufficient for meeting this Success Criterion. However, it is not necessary to use these particular techniques. For information on using other techniques, see [Understanding Techniques for WCAG Success Criteria](#), particularly the "Other Techniques" section.

Sufficient Techniques

1. [Provide persistently visible controls](#)
2. Use simplification="important" to indicate the control is important (future)

Key Terms

process

series of user actions where each action is required in order to complete an activity

Successful use of a series of Web pages on a shopping site requires users to view alternative products, prices and offers, select products, submit an order, provide shipping information and provide payment information.

An account registration page requires successful completion of a Turing test before the registration form can be accessed.

Previous workup understanding

<https://docs.google.com/document/d/1DPtCqWHjrj3QZ4afsqzmWDd-zMSf39RsMqSpR2QGCG/edit#heading=h.ej41yvbs80dk>

Hidden controls (was Essential Controls)	N e w to find the most important thing	C Rachael C Montgome Cry A	https://docs.google.com/document/d/1DPtCqWHjrj3QZ4afsqzmWDd-zMSf39RsMqSpR2QGCG/edit	In editors draft	https://www.w3.org/2002/09/wbs/35422/essential-controls/results	https://www.w3.org/2020/02/11-ag-minutes.html#item02
--	---	-------------------------------------	---	------------------	---	---

Brainstorming 30 Nov 2020

Basecamp: Visual user favoring pointer/Edit interaction with text and checkbox that display control

- User Action: User wants to assign a to-do, sees the to-do item on the screen, does not see any way to edit the to-do or take any action besides checking it off as done
- System Response: Nothing
- User Response:
 - Looks for menus or other controls
 - Activates other menus/controls that aren't relevant to task
 - Uses pointer to point at things
 - Notices a menu appears when pointing at the to-do item text

- Common User Needs
 - To know the to-do text is interactive, so they know they can trigger it
 - To know what controls aren't relevant, so they don't activate the wrong feature
 - To know triggering the to-do or the hidden revealed text will activate the same feature, so they know how to access the feature
 - To recognize what controls are and how they work, so they can activate the feature they are looking for
 - To understand which are enabled, so they don't try to work things that don't work
- Functional Needs
 - Functional Needs: Essential
 1. Use without physical harm or risk (to self or others within a physical environment)
- Functional Needs: Sensory
 - Vision & Visual
 2. Use without vision
 3. Use with limited vision
 - a. User needs to be able to see/be aware of the affordances (cursor pointer, show/hide text), so they can know the thing is interactive and what it does
 4. Use without color perception
 5. Use with limited color perception
 6. Use with limited depth perception
 7. Use with limited orientation or spatial tracking
 8. Use with photosensitivity (too much or too little)
 - Hearing & Auditory
 9. Use without hearing
 10. Use with limited hearing
 11. Use with limited auditory processing (speech)
 12. Use with sensorineural hearing loss (limited frequency range) related to age or Presbycusis (gradual loss over time)
 - Sensory Intersections
 13. Use without vision and hearing
 14. Use with vestibular issues
 15. Use without spatial auditory awareness or perception (needs diegetic sound)
- Functional Needs: Physical
 - Mobility
 16. Use without mobility
 - a. User needs to know what the control does, so they activate it intentionally and with knowledge of what's they're going to get
 - b. User needs to know that the text is a control to access the edit feature, so they can access the edit feature
 - c. User needs to be confident that activating the text, so they don't inadvertently check off the to-do checkbox

- 17. Use with limited mobility
- 18. Use with limited reach or range
- Motor
 - 19. Use without hands
 - 20. Use without multiple touchpoint gesture
 - 21. Use with limited strength
 - 22. Use without fine point control
 - 23. Use without physical tracking speed
 - 24. Use with tremors
- Physical & Sensory Intersections
 - 25. Use with limited kinesthetic perception (orientation, position, weight distribution, movement)
 - 26. Use with limited tactile perception, sensory processing, or touch pressure sensitivity
 - 27. Use with chronic pain impacting input or interaction modality
- Speech
 - 28. Use without vocalization
 - 29. Use with limited vocalization or volume
- Functional Needs: Cognitive
 - Attention
 - 30. Use with limited ability to focus attention
 - a. User needs to not be distracted by elements that are not at the point of interaction, so they can focus on the task
 - b. User needs to not have to attend to things that are not at the point of interaction, so they can focus on the task
 - c. User needs to not have to hunt for where the control is, so they can follow a path, follow their point of interaction
 - 31. Use with limited ability to direct attention
 - a. User needs to not be distracted by elements that are not at the point of interaction, so they can focus on the task
 - b. User needs to not have to attend to things that are not at the point of interaction, so they can focus on the task
 - c. User needs to not have to hunt for where the control is, so they can follow a path, follow their point of interaction
 - 32. Use with limited ability to shift attention
 - a. User needs to not be distracted by elements that are not at the point of interaction, so they can focus on the task
 - b. User needs to not have to attend to things that are not at the point of interaction, so they can focus on the task
 - c. User needs to not have to hunt for where the control is, so they can follow a path, follow their point of interaction
- Language & Communication
 - 33. Use with limited ability to comprehend spoken language
 - 34. Use without ability to read

- 35. Use with limited ability to recognize written language
- 36. Use with limited ability to comprehend written language
- 37. Use without ability to write
- 38. Use with limited ability to correctly write (or type) words and use punctuation
- 39. Use without understanding symbols
- 40. Use without understanding metaphors, idioms, euphemisms, or specific dialect of culture or location
- Learning
 - 41. Use with limited ability of math and numeric concepts
 - 42. Use with limited compositional skill (simultaneous thinking and input)
 - 43. Use with limited coordinational skill (motoric skills, visual-spatial organizational memory, and social)
- Memory
 - 44. Use with limited short-term or working memory
 - a. User needs to know how the text item works, so they can recognize it as the interactive feature they are looking for
 - 45. Use with limited medium or long-term memory
 - a. User needs to know how the text item works, so they can recognize it as the interactive feature they are looking for
 - 46. Use with limited sensory memory
- Executive
 - 47. Use with limited planning, organization, sequencing, and execution ability
 - a. An alternative path to the function may be challenging to someone with limited sequencing ability
 - b. Making the path clear, expectation of how the path will unfold
 - c. Cognitive load, put information up-front so they can self-pace
 - 48. Use with limited emotional control and self monitoring
 - 49. Use with limited judgement
- Mental Health
 - 50. Use with debilitating fear or anxiety
- Cognitive & Sensory Intersections
 - 51. Use with interocular transfer of visual memory (retrieval based on limited acuity in a single eye)
 - 52. Use with limited phonological or phonemic awareness
- Functional Needs: Independence
 - Independence
 - 53. Use without autonomy or agency
 - 54. Use without privacy

Why this pattern?

- Minimal visual noise
- Leverage repetitiveness of interface, e.g., people learn the interface and apply it

- Trying to do more with less, instead of one thing doing one thing, they want to do many things with one thing
 - Designers want to be innovative
 - Designers like the economy of it
 - Want people to go wow, that's slick!
- Spacing, limited amount of space to work, try to leverage, make affordances
- Top tasks are visual, secondary things bounce off of top tasks
- Clean design, mobile-friendly
- Less stuff
- Focus on task working on, make them work for features that aren't as common
- Desktop, anything possible is in menus, persist
- Don't have that with web

1st iteration of rewrite

Information needed to recognize and understand how to operate user interface components is provided to users at the point of interaction without requiring pointer hover or keyboard focus.

Jpn's description of other users it helps

user uses 800x600 resolution at the default browser zoom. The user looks closely at the monitor – say 4 inches from the display. Due to the distance from the monitor, the acuity of the person and Scotomas (blind spots) the user is only able to clearly read 4 or 5 words on the screen in a small circle – but can see that there are objects, text, colors, and other items around on the page. In fact there are hundreds of words or controls that appear – but are not legible. When a user can't find what they are looking for the user starts moving their head around the screen looking for controls. If that is not successful then using the mouse requires the user to jiggle the mouse and move his/her head around looking for the mouse pointer. Then hovering over objects one by one to find out what they do. Using the keyboard means likely starting from the top of the page and tabbing through many times to find the content on the page – likely trying to visually track where the keyboard focus is on the page at each tab stop looking to see if something appears on focus. Tracking the visual keyboard focus is very difficult as it jumps around, can be in different forms and go through many controls even when the page is WCAG conformant. All of the while all of this moving around the screen with their head is causing terrible neck pain. Keep in mind that users with low vision are likely moving their physical head side to side up and down continually as they mouse over each and every control. It's not just moving your eyes and if it were trying to control your eye gaze when you have nystagmus is exhausting as well.

Thus, one of the main challenges is simply locating related controls on the screen, moving the mouse from control to control or getting the keyboard to the point where you were looking and tabbing. It's not uncommon for a low vision user to be looking where the mouse or keyboard is not located – for some users most of the time the mouse or keyboard is not where they are reading! One user I am describing is not using any magnification software but the text is large enough on the screen to read – but they can only read a few words clearly – they scroll through the screen with the mouse wheel.

Finding content by searching visually is very exhausting. When a user doesn't see something on the screen they assume it must be around but they are just missing it, it's in their blind spot or it must be hidden in a menu, etc. Then they have to go searching until they likely give up.

On a separate note most screen readers today do not focus elements when the virtual/browser cursor reaches content. NVDA and VoiceOver on iOS use to but I don't believe they do anymore – JAWS has never to my knowledge. Many screen reader users are not tabbing around the page. This means for screen reader users unless there are clear roles or hidden instructions screen reader users may not be aware that they need to tab to focus an element. Most advanced users are navigating by heading, paragraph, control, etc. which only moves the screen reader's point of regard and not the system focus.

For speech recognition users hovering the mouse would require using a mouse grid and use of the keyboard to focus content would require saying press tab to move from control to control. This is simply not how users of speech recognition use their technology. Most users are activating controls by speaking the name of the control or by roles such as click link, etc. Trying to focus or hover just to find out if the control had something on focus/hover would be verbally exhausting.