

GROUPER EVALUATION & REMEDIATION REPORT

Reviewer: Howard Kramer, hkramer@colorado.edu

Technology Used: NVDA (ver. 2016.1), Firefox (ver. 48.0.2) on Windows 10 PC

Background

This report evaluates the Grouper group management software. This report is a follow-up to and reviews most of the accessibility issues mentioned in the CU-Boulder AUL report and recommends remedies and techniques for addressing these issues. In some cases this report addresses issues not mentioned in the AUL report. In many cases specific coding for remediation are provided.

Overview of Accessibility Problems

In general, the accessibility problems that occur in Grouper are due to the use of JavaScript instead of standard HTML coding. In many cases throughout the interface JavaScript with event handlers (e.g. `onmouseover()`) are used to present controls and form fields which can only be accessed with a mouse. This locks out anyone who is a keyboard-only or screenreader* user.

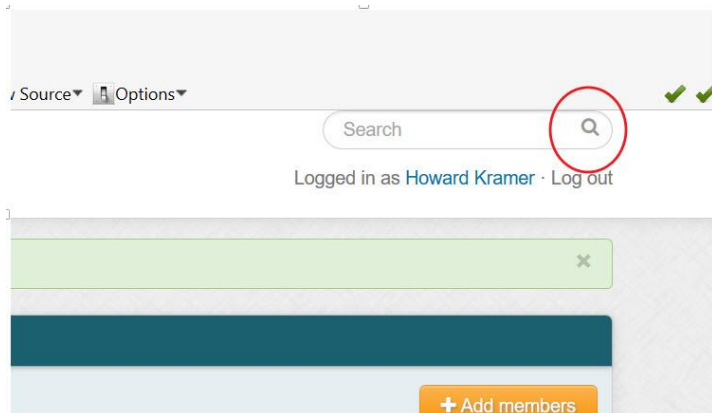
*(Blind individuals use screenreaders to convert onscreen objects and text to speech).

The other main group of accessibility issues are caused by the display of new content (sections that expand with previously hidden content when a button or link is selected). For example, the section that appears when the “Add Members” button is selected or the sub-menus displayed on selection of the “more” or “more actions” pull-down menus.

In the report below I recommend solutions that include the use of ARIA (sometimes referred to as WAI-ARIA) and JavaScript to make new content “visible” to screenreaders. In many cases, the implementation of ARIA attributes can also make the JavaScript designed interface controls accessible. Sometimes I recommend the use of simpler HTML interface objects instead of JavaScript.

The list of issues and recommended remedies follow.

Search Icon Not Labelled



(ver. 9/6/16 – 12:28 a.m.)

Problem:

The search icon is rendered through the following JavaScript and HTML:

```
<a onclick="return guiV2link('operation=UiV2Main.searchSubmit', {optionalFormElementNamesToSend: 'searchQuery'});" href="#" >
```

```
<i class="fa fa-search"></i>
```

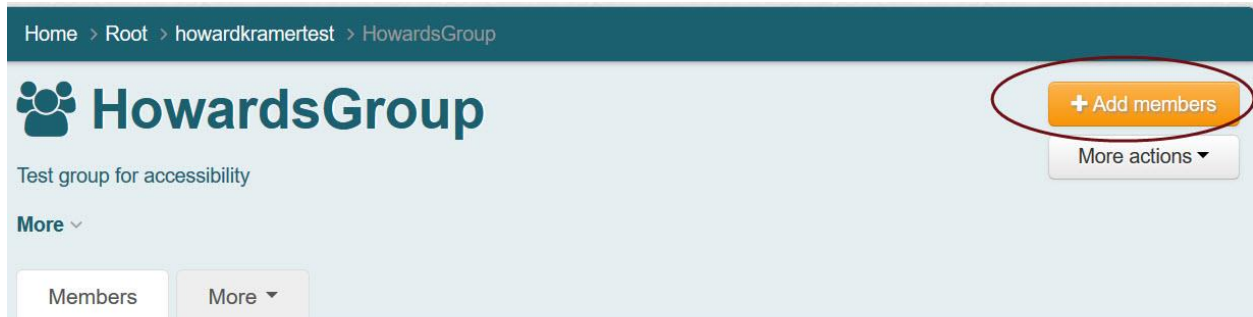
As a result, the NVDA screenreader announces this as "clickable" – not enough information to make it meaningful to a blind user.

Solution:

Adding the "aria-label" attribute to the "a" element above should provide a readable (and non-visible) text label to the icon:

```
<a onclick="return guiV2link('operation=UiV2Main.searchSubmit', {optionalFormElementNamesToSend: 'searchQuery'});" href="#" aria-label="search">
```

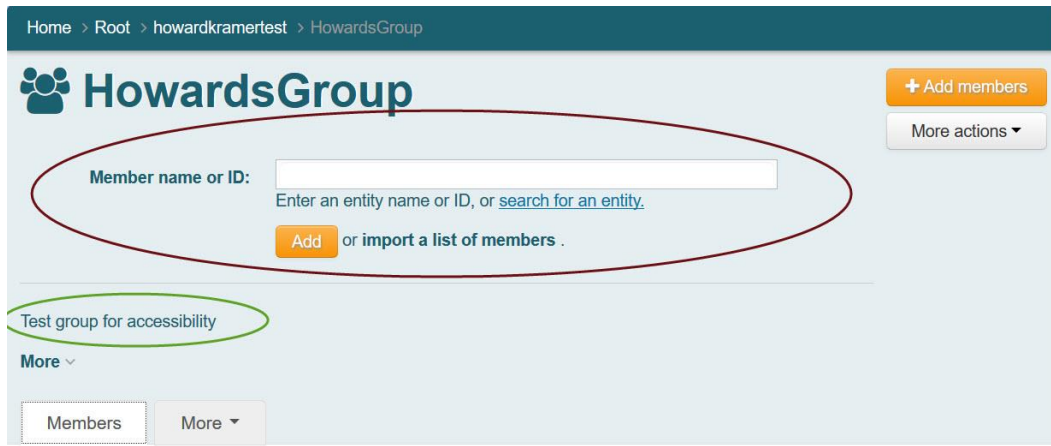
“Add members” displays new content not apparent to a screenreader



The screenshot shows a web page for 'HowardsGroup'. At the top, there is a breadcrumb trail: 'Home > Root > howardkramertest > HowardsGroup'. Below this is the group's logo and name, 'HowardsGroup', with the subtitle 'Test group for accessibility'. On the right side, there is a button labeled '+ Add members' which is circled in red. Below it is a 'More actions' dropdown menu. At the bottom left, there is a 'More' dropdown menu and a 'Members' button.

The following table lists all entities which are members of this group.

(ver. 9/6/16 – 12:28 a.m.)



Problem:

New Content Not “Visible” to ScreenReaders

Selecting “Add members” reveals new content visually (circled above in red) but not to the screenreader. In other words, there is no announcement to the screenreader that new content has appeared.

Focus Placed After New Content

In addition, after selecting "Add Members" focus takes the user past the new content and fields associated with adding a new member. The first field a keyboard user would encounter (using the down arrow) is the “more actions” button or (if using the Tab key) the "Test group for accessibility" (circled above in green)".

A keyboard user (including screenreader users) would be required to back tab back through a number of buttons and links such as "more," “Test group for accessibility,” "import a list of members," etc. before finding the "member name or ID" field. Of course, a blind user would have discover this on their own. They would not know they have been placed past the fields associated with adding a new member. It might take quite a while of exploring the page before (and if) they find the necessary content.

Solution:

Use jQuery or JavaScript with ARIA to announce the new content to screenreaders and to move the focus to the "member name or id field".

Specific Coding

The "control-group" div (shown below) should include the aria attributes: `aria-live="polite" aria-expanded="false"`.

(ver. 9/6/16 – 12:28 a.m.)

The screenshot shows a web application interface at the top and a browser's developer tools at the bottom. In the interface, a search form is visible with a label "Member name or ID:" circled in green. Below the label is a text input field with the placeholder text "Enter an entity name or ID, or [search for an entity](#)." To the right of the input field is an orange "Add" button followed by the text "or import a list of members .". The developer tools show the DOM tree for the form. The "control-group" div is highlighted in blue, and its HTML code is shown below it. The code includes a label with the text "Member name or ID:" which is also circled in green. The code is as follows:

```
<div class="control-group">
  <label class="control-label" for="add-block-input">Member name or ID:</label>
  <div class="controls">
  </div>
</div>
```

Adding the following jQuery (or equivalent JavaScript) should place the attribute `role="alert"` and change the attribute `aria-expanded` to `true` (from `false`) in the `"control-group"` div statement once the link `"show-add-block"` is selected (this is the ID of the `"Add members"` link). This will alert the screenreader that new content has appeared. The last statement below will place focus in the `"Member name or ID"` field. (Note: the current coding for this input field is quite complex with a number of hidden fields located at the same location. As with the other suggestions in this report, these solution should be tested with a screenreader once these changes are implemented).

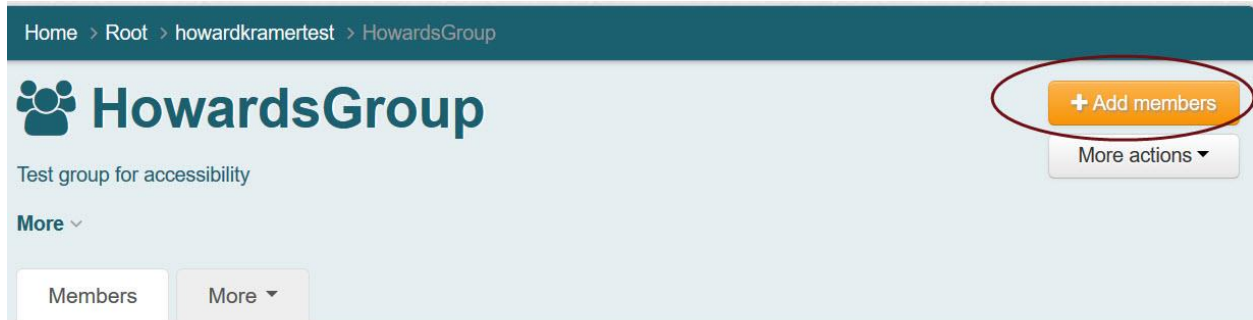
```
$( "a.show-add-block" ).click( function() {
  $( "#control-group" ).attr( "role", "alert" );
  $( "#control-group" ).attr( "aria-expanded", "true" );
  document.getElementById( "groupAddMemberCombold" ).focus();
} );
```

You can read and see more examples on making collapsible and expandable content accessible at: https://www.w3.org/WAI/GL/wiki/Using_aria-expanded_to_indicate_the_state_of_a_collapsible_element

(ver. 9/6/16 – 12:28 a.m.)

From AUL report, page 11:

“Add member” appears visually as a button but is read as a link.



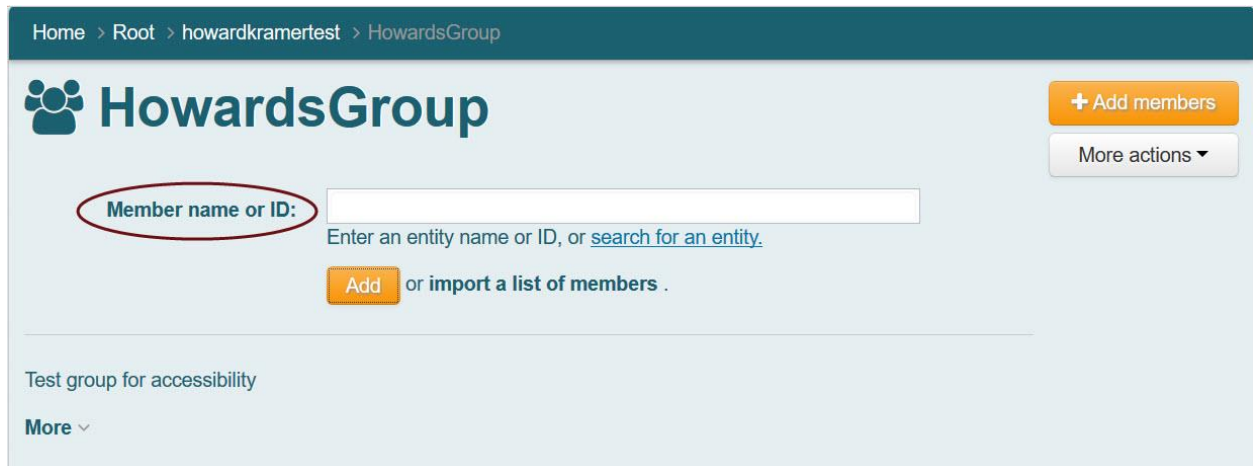
The following table lists all entities which are members of this group.

Solution:

Add *role = "button"* to “Add member” statement as shown below

```
<a id="show-add-block" class="btn btn-medium btn-primary btn-block" onclick="$('#add-block-container').toggle('slow'); return false;" href="#" role="button">
```

“Member name or ID” field appears to be mislabeled and identified as a combo-box through use of ARIA role



(ver. 9/6/16 – 12:28 a.m.)



```
S Script DOM Net Cookies Search by t
ol-group < form#add-...highlight < div#add-members < div#add-b...ell.hide < div.span10 < div.row-fluid < div
iv class="span10">
| <h1>
| <div id="member-search" class="modal hide fade" aria-hidden="true" aria-labelledby="member-
search-label" role="dialog" tabindex="-1">
| <div id="add-block-container" class="well hide" style="display: block;">
|   <div id="add-members">
|     <form id="add-members-form" class="form-horizontal form-highlight" target="#">
|       <div class="control-group">
|         <label class="control-label" for="add-block-input">Member name or ID:</label>
|         <div class="controls">
|           <div id="add-members-container">
```

Problem:

“Member name or ID” field appears to be mislabeled

As shown in the above screenshot, the label “Member name or ID” is associated through ID to an input field with an ID of “add-block-input.” The ID of the input field (shown below – last circled item) is “groupAddMemberComboID.” There is another input field (also shown below) but does not match the label ID either. As a result the screenreader user receives no helpful feedback when they tab or move to the input field.

Solution:

Change the associated ID (i.e. for = “ID”) in the label statement shown above to match the ID of the input field: “groupAddMemberComboID”

Problem:

“Member name or ID” field is announced as a combobox (see screenshot below)

The Member ID or name field is identified as a combo box through the containing div which has ARIA attribute *role = “combobox”* (code shown above circled in green). I assume this has been implemented because the interface uses a JavaScript routine to search for members who match the text entered. However, this interface element does not behave as a true combobox which would be populated with values before the input of any text. As a result identifying this as a combobox is confusing to the screenreader user.

(ver. 9/6/16 – 12:28 a.m.)

Enter an entity name or ID, or [search for an entity.](#)

Search by text or CSS s

```
tbody < table < div#add-m...ontainer < div.controls < div.control-group
  <div id="widget_groupAddMemberComboId" class="dijit dijitReset
  dijitInline dijitLeft dijitTextBox dijitComboBox
  dijitValidationTextBox" data-dojo-attach-
  point="_popupStateNode" aria-haspopup="true" role="combobox" s
  tyle="width: 30em;" widgetid="groupAddMemberComboId">
    <div class="dijitReset dijitRight dijitButtonNode
    dijitArrowButton dijitDownArrowButton
    dijitArrowButtonContainer" role="presentation" data-dojo-
    attach-point="_buttonNode" style="display: none;">
    <div class="dijitReset dijitValidationContainer">
      <input class="dijitReset dijitInputField
      dijitValidationIcon
      dijitValidationInner" type="text" role="presentation" r
      eadonly="readonly" tabindex="-1" value="X" >
    </div>
    <div class="dijitReset dijitInputField
    dijitInputContainer">
      <input id="groupAddMemberComboId" class="dijitReset
      dijitInputInner" type="text" role="textbox" data-dojo-
      attach-
      point="textbox,focusNode" autocomplete="off" tabindex="
      0" aria-required="false" value="" aria-invalid="false">
      <input type="hidden" name="groupAddMemberComboName" val
      ue="">
    </div>
  ...
```

Solution:

Remove the `role="combobox"` attribute and instead use `aria-describedby` to provide a description of how the field operates. This attribute would probably best be placed in the `"widget_groupAddMemberComoiD"` div.

Then create an element with the text describing how the lookup field operates.

For example, `aria-describedby = "explainmembersearch"` would point to an object (such as a paragraph) with an id of `"explainmembersearch"`. This element can be set to be visually hidden (but don't use `display: none`).

Here's an example of CSS code from Chase Bank used to hide the span class `chaseutil-hidevisual`:

```
span.chaseutil-hidevisual {
  height: 1px;
  left: -99999px;
  overflow: hidden;
  position: absolute;
  width: 10px;
}
```

(ver. 9/6/16 – 12:28 a.m.)

For the instruction pointed to by aria-describedby I suggest something such as: "enter at minimum the first two letters of the name or id of the member to search for. The up and down arrow can then be used to move through any matches. If you receive an error messages of 'the value entered is not valid,' there are no matches."

Problem:

Error message 'the value entered is not valid' is confusing (to all users)

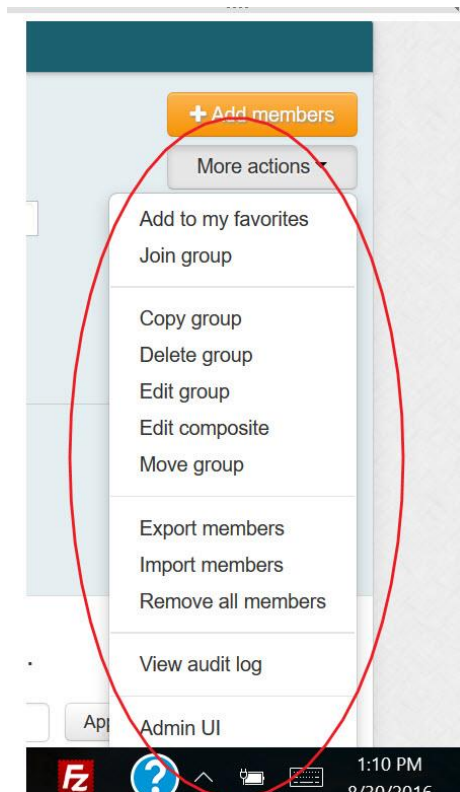
The error message displayed when there is no match in the "Member name or ID" field is accessible but confusing. Instead of 'the value entered is not valid' I suggest a message such as "no matches found."

Solution:

You can customize the error messages through the html5 constraint validation API (see <http://codetheory.in/custom-validation-messages-for-html5-form-constraints/> as an example)

Problem:

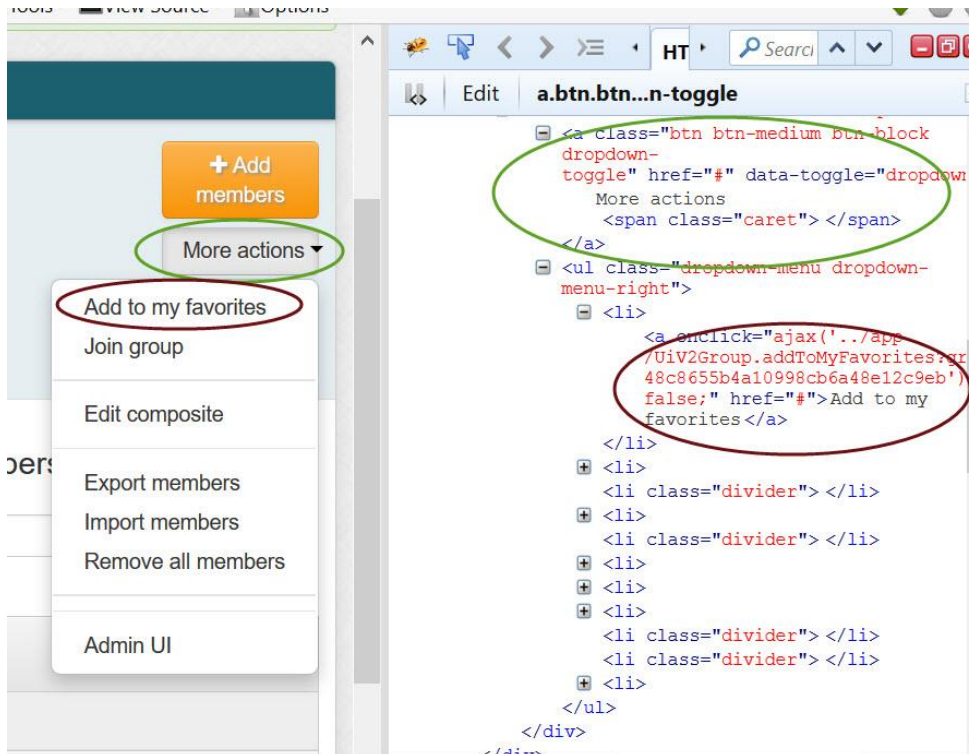
Pull down menus throughout site do not indicate their function as a pull down menu or announce changed state once selected



For example, in the "More Actions" pull-down menu (shown above), there is no cue that this is a pull down menu or an announcement that the menu has been opened.

Solutions:

Add ARIA to indicate menu role and state of expand or collapse



For example, in the above "more actions" pull down menu, we would start by adding `aria-haspopup="true"` and `aria-expanded="false"` and `role="menu"` to the anchor object for the "More actions" link, as shown below.

```
<a class="btn btn-medium btn-block dropdown-toggle" href="#" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" role="menu">
  More actions
  <span class="caret"></span>
</a>
```

Adding a JavaScript action (another onclick handler) to change `aria-expanded="true"` once the "More action" link is selected would provide feedback to the screenreader that the menu is in expanded mode.

Focus should also be moved to the first item in the list upon selection of the "more actions" button using the jQuery or equivalent JavaScript shown on the bottom of page 4.

(ver. 9/6/16 – 12:28 a.m.)

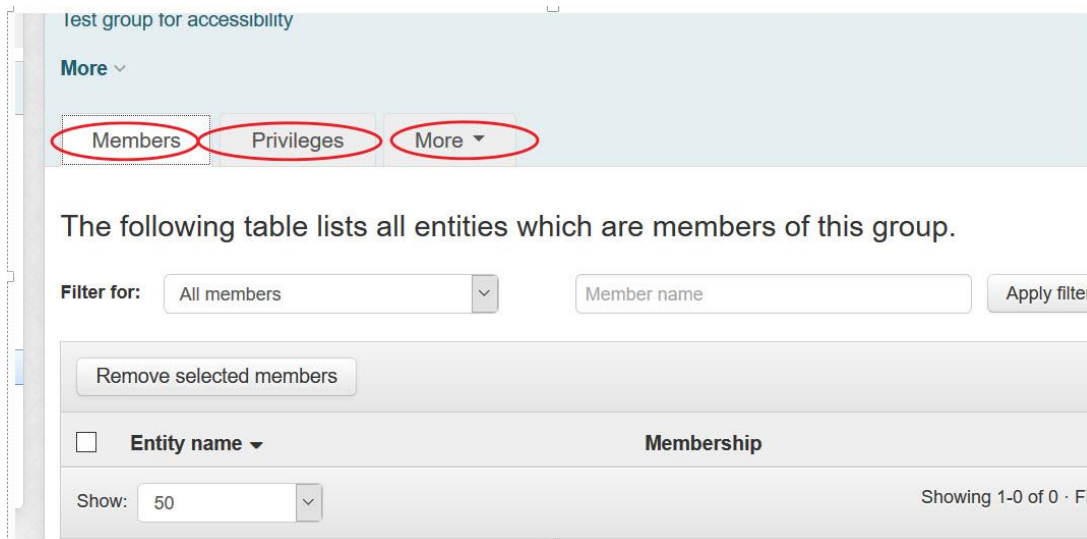
The techniques described above can and should be applied to all the pull-down menus on the site.

There is a tutorial and examples of accessible pull-down menus at <https://www.w3.org/WAI/tutorials/menus/flyout/#flyoutnavkbtn>

There are other examples of accessible pull-down menus at

- http://heydonworks.com/practical_aria_examples/#submenus
- <http://a11y.nicolas-hoffmann.net/subnav-dropdown/>

Section tabs read as links



Problem:

Although it is visually clear that "Members" and "Privileges," are section tabs, the screenreader user simply hears these as links.

Solution:

Use `aria-role="tab"` for each of the tabs (circled) and use JavaScript or jQuery to change the attribute `aria-selected="true"` as each tab is selected and change not-activated tabs to `aria-selected="false"`. This will inform the screenreader user that these are tabs and when each tab is selected or non-selected.

See <http://accessibility.athena-ict.com/aria/examples/tabpanel2.shtml> for more information and an example of how this would be implemented.

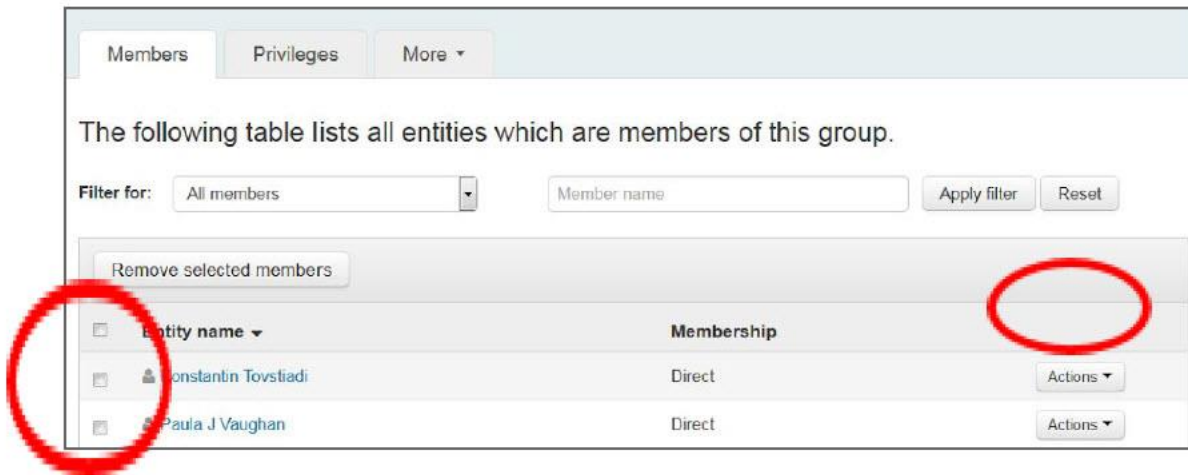
(ver. 9/6/16 – 12:28 a.m.)

From AUL report, page 11:

Group membership entity table lacks column header for fourth column

Problem:

The definition is read as four columns but the fourth column (containing the Actions dropdowns) has no column header to let the screen reader user know what is in that column.



Solution:

Add column heading text to the <th> statement as shown below

Note that this text can be hidden from the visual user using CSS (See page 7 for the proper way to hide text with CSS)

```
<th class="sorted">Entity name</th>
<th data-hide="phone">Membership</th>
<th style="width:100px;" class="hidden">Choose action</th>
```

Problem

The check boxes for each row have no meaningful labels

Solution:

Add a label using aria-label as shown below

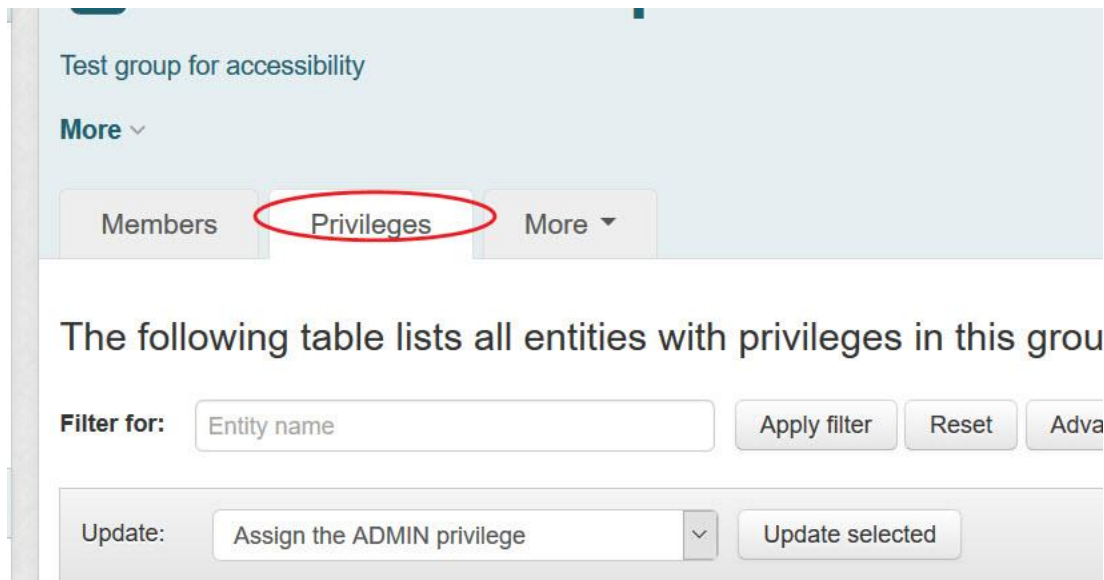
(Note: removal seems to be the only group action available using this checkbox, hence the label below)

```
<label class="checkbox checkbox-no-padding">
<input class="membershipCheckbox" type="checkbox"
value="b88c0a8caad040dca3996f92817086cb:e70e6dd43cfa40248f76602ba008588b"
name="membershipRow_0" aria-label="select this member for removal">
```

(ver. 9/6/16 – 12:28 a.m.)

</label>

Privileges tab takes focus back to top of page



Problem:

When the "privileges" tab is selected, focus is brought to the top of the page (focus should return to "privileges" tab) or to the first item under the "Privileges" section. This forces the screenreader user to move back through the whole page to return to the content under the "privileges" section.

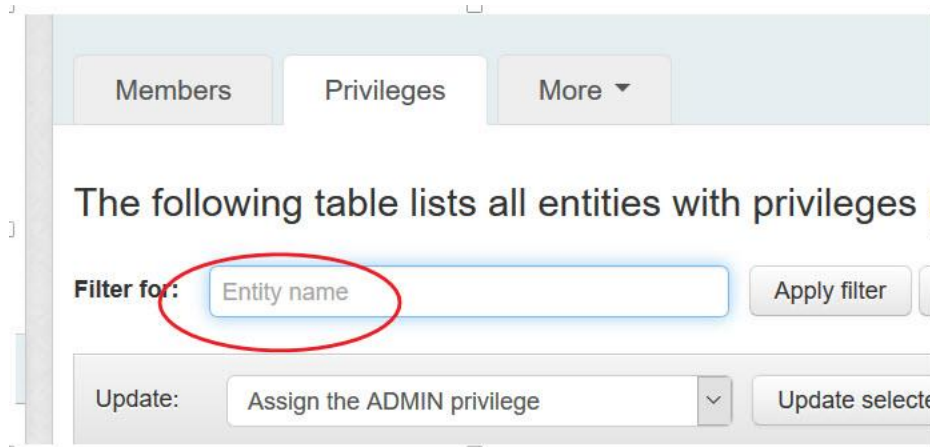
Solution:

The JavaScript and other techniques described previously for "Add Members" on page 4 can be used to control the placement of the focus when the "privileges" tab is selected. Focus should be placed at the first control or link after the "Privileges" tab – in this case it appears to be the "entity name" field.

Alternatively (and perhaps more helpful), make the paragraph "The following table lists all ..." focusable with `tabindex="0"` and set focus to that item after the "Privileges" tab is selected.

```
<p class="lead" id="starthere" tabindex="0">The following table lists all entities with privileges in this group.</p>
```

Placeholder text not accessible



Problem:

Placeholder text is not “visible” to a screenreader

Solution:

This can be corrected by using the “aria-label” attribute. For this input field, the code would appear as:

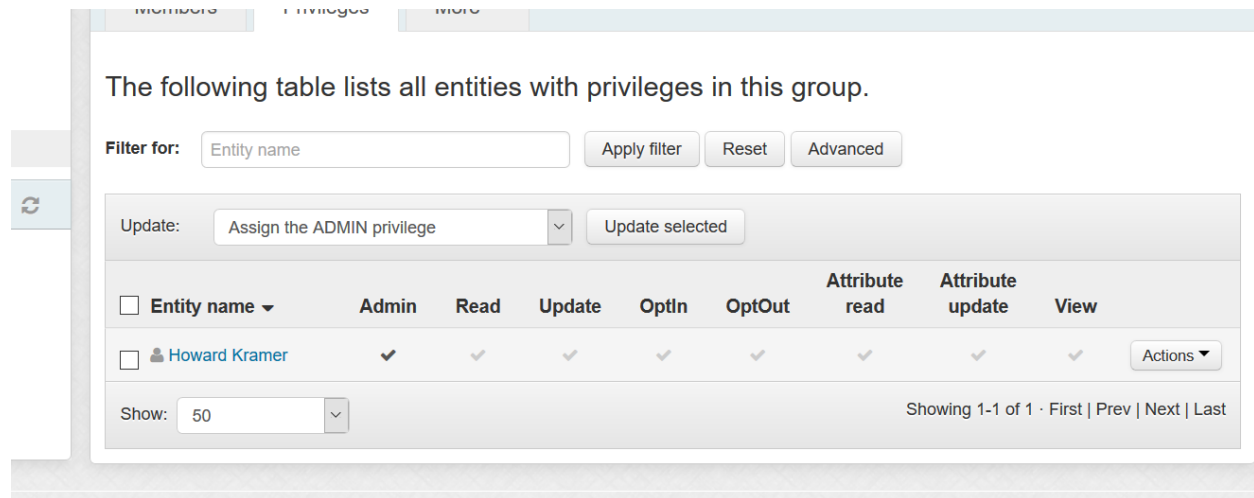
```
<input id="table-filter" class="span12" type="text" name="privilegeFilterText" placeholder="Entity name" aria-label="enter entity name">
```

For a longer description the ARIA attribute "aria-describedby" could be used. See the example on page 7.

Note: I found this area of the site confusing regarding the function of the filter and other options. You could also add text to an id associated with “aria-describedby” that explains how the options in this section operate.

(ver. 9/6/16 – 12:28 a.m.)

Privileges interface not accessible to keyboard-only or screenreader users



Problem:

The table section of the Privileges interface can only be accessed with a mouse

As mentioned in the AUL report, the privileges table interface is not accessible to keyboard-only or screenreader users. Specifically, none of the faux checkboxes for setting user privileges is accessible by the keyboard. Placing `tabindex="0"` inside the "a" element (initial segment displayed below) of the faux checkboxes could make it more accessible but it is recommended that the JavaScript-created checkboxes be replaced by standard checkboxes.

```
<a class="btn btn-inverse btn-super-mini remove" tabindex="0" ....
```

JavaScript could then be built up around the standard checkboxes to achieve some of the enhanced visual effects. In other words, after starting with standard checkboxes, JavaScript could be used to provide additional effects on mouse hover, etc.

There is a good article on JavaScript event handlers and accessibility at:

<http://webaim.org/techniques/javascript/eventhandlers> and on the use of `tabindex` at:

<http://webaim.org/techniques/keyboard/tabindex>.

(ver. 9/6/16 – 12:28 a.m.)

Focus is placed within table on Group Management page

Problem:

(Taken from page 7 of AUL report):

When the group management page is displayed for a specific group, screen reader focus lands within the Entity table portion of the display bypassing the page heading, page description, and table heading. Consequently, the screen reader user does not know what page they have landed on nor what the table they have landed in is for.

Solution:

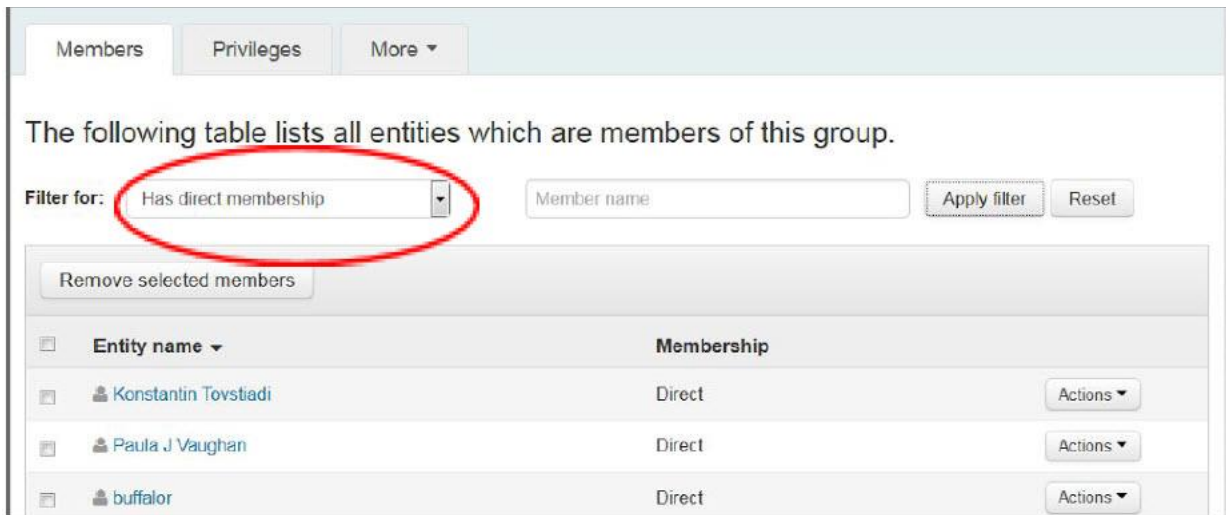
(Taken from page 7 of AUL report):

Upon arriving at the group management page, focus should be put at the group name heading and navigation should take the screen reader through the main content and table name before arriving at the table rows.

Use the focus technique shown on page 4 to accomplish this recommendation.

(Taken from page 5 of AUL report):

The revised display of entity names is not announced after applying a filter to the entity list.



The screenshot shows a web interface for group management. At the top, there are tabs for 'Members', 'Privileges', and 'More'. Below the tabs, a heading reads 'The following table lists all entities which are members of this group.' Underneath the heading is a 'Filter for:' dropdown menu with 'Has direct membership' selected, which is circled in red. To the right of the dropdown are input fields for 'Member name', 'Apply filter', and 'Reset'. Below the filter section is a 'Remove selected members' button. The main content is a table with two columns: 'Entity name' and 'Membership'. The table contains three rows of data:

Entity name	Membership
Konstantin Tovstadi	Direct
Paula J Vaughan	Direct
buffalor	Direct

(ver. 9/6/16 – 12:28 a.m.)

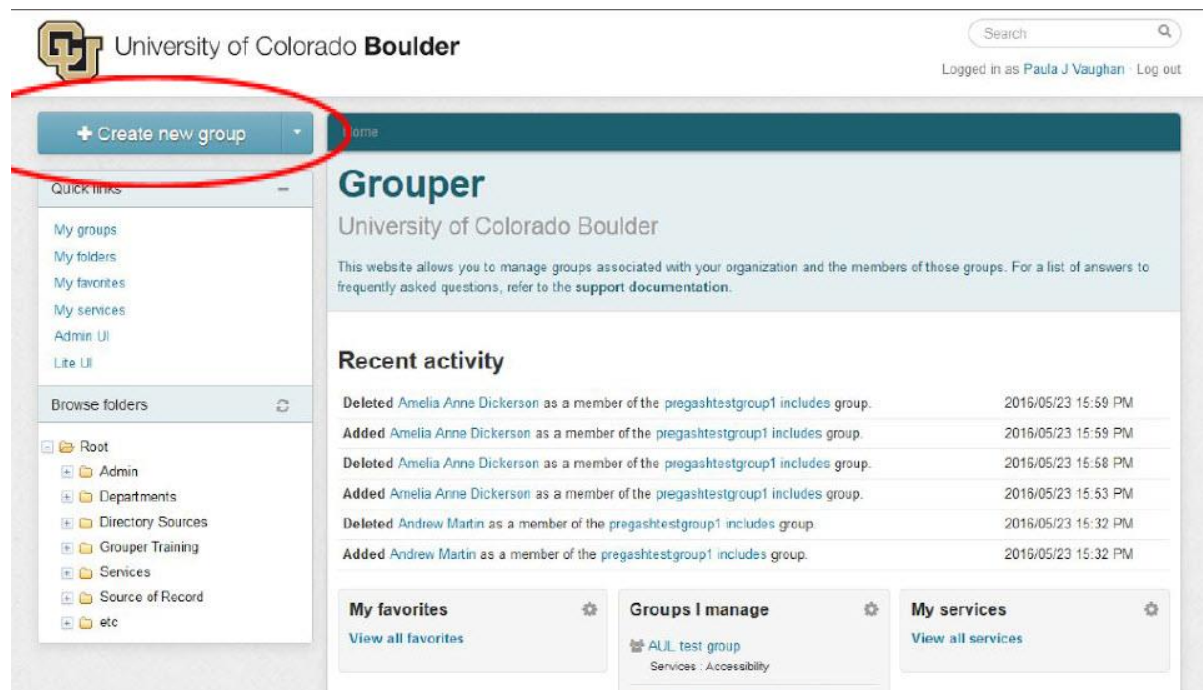
Solution:

Use the *aria role="alert"* and *aria-expanded* attributes shown on page 4 to make the new content apparent to screenreader users. In addition, sending focus to first name in the list would enhance accessibility. This technique is also shown on page 4.

(Taken from page 6 of AUL report):

Navigation Issues

On the Grouper home page, “Create new group” is not read by screen readers until after all of the main content is read, including everything within “My favorites”, “Groups I Manage”, and “My Services.” Given the importance of the function, it should perhaps come earlier in the flow.



Solution:

The HTML code could be reordered on this page or preferably, some hidden “skip to main content,” “skip to quicklinks and other options” links could be placed at the top of the page.

Sample code would look as follows:

```
<a href="#maincontent" class="">skip to main content</a>
```

See webaim.org for an example of how this technique is implemented. The skip option is the first one seen when tabbing through the page from the top.

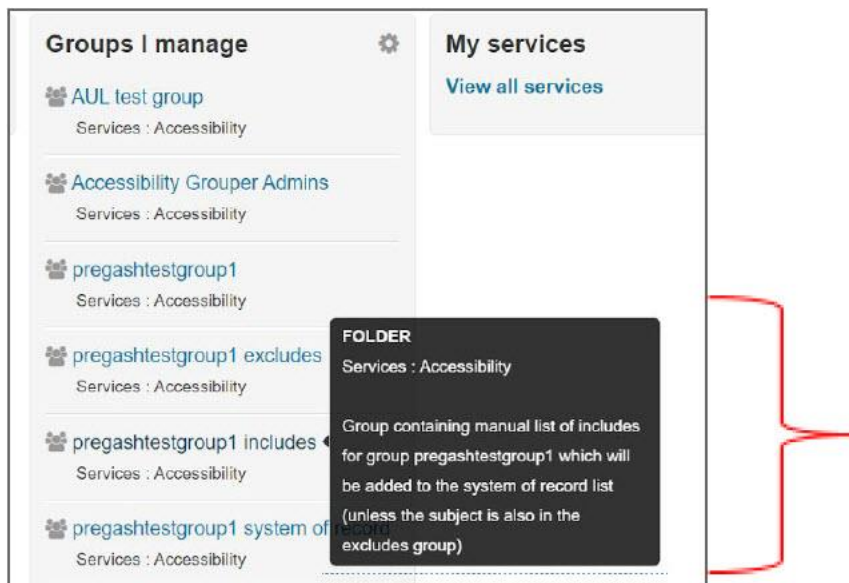
(ver. 9/6/16 – 12:28 a.m.)

(Taken from page 8 of AUL report):

Information displayed via mouse hover not readable by screen readers

Screen readers are completely dependent upon keyboard action; no “hover” information is available to screen reader users. Example (in addition to the Privileges table mousehover issues noted in the Severe Issues section):

- Descriptions of groups within the Groups I Manage list are not available to screen reader users.



Solution:

I suggest placing a small question mark icon or graphic next to each of these items which would open up a dialog box when selected with the enter key. Use the alt attribute (if a graphic is used for the icon) or aria-label if an anchor to provide a meaningful label for the icon. For example, “more information about this entity.”

For examples of accessible dialog boxes, see:

- <https://www.marcozehe.de/2015/02/05/advanced-aria-tip-2-accessible-modal-dialogs/>
- <https://www.nczonline.net/blog/2013/02/12/making-an-accessible-dialog-box/>
- http://heydonworks.com/practical_aria_examples/#warning-dialog

(ver. 9/6/16 – 12:28 a.m.)

(Taken from page 9 of AUL report):

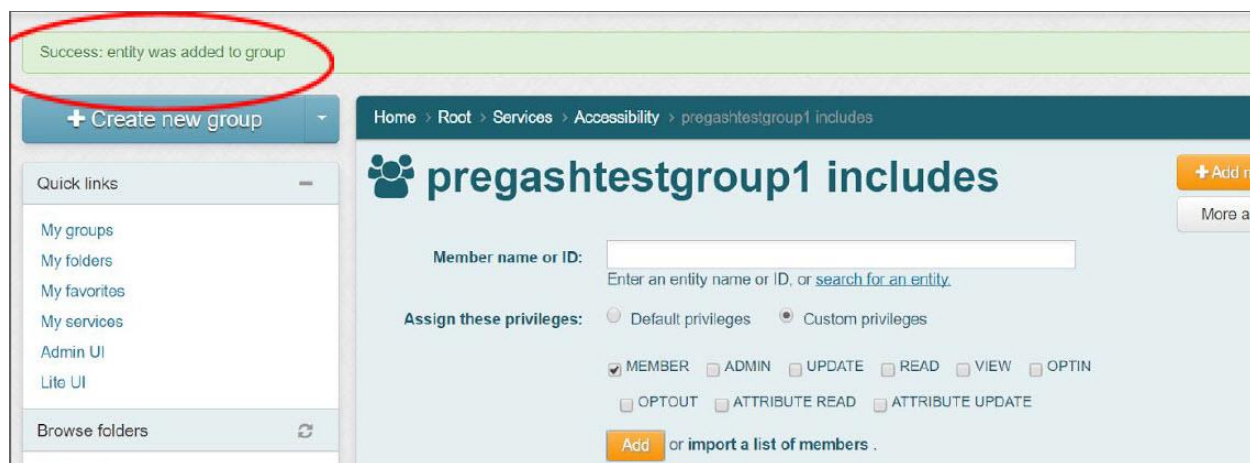
Error (and success) messages are displayed above the page heading and all content, unapparent to screenreaders

Problem:

Rather than sending focus to (and reading) the success message, focus stays at the action button (for example, at the “Add” button for the Add member process).

o The screen readers users did not even know to look for the messages without the prompting of a sighted observer. Instead, they had to assume the requested action was completed.

o Magnifier users did not see the messages because the colored bar appeared above the field of magnification and the text appeared too far to the left of the field of magnification.



This reviewer will add that this notification is barely noticeable by the visual user because of its placement on the top left and poor color contrast.

Solutions:

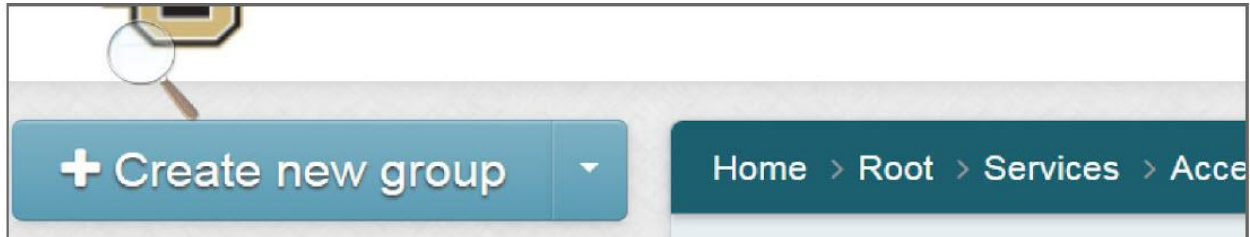
- **Place the notifications and errors closer to the main content area of the page and use a background/foreground color combination with at least a 4.5:1 contrast ratio. Use a tool such as the TPG Color Contrast Analyzer <https://www.paciellogroup.com/resources/contrastanalyser/> to evaluate and select sufficient color combinations.**
- **Use the ARIA role="alert" to make these notifications apparent to the screenreader and low vision user. See https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques/Using_the_alert_role for an example on how to implement this feature.**

From AUL report, page 10:

(ver. 9/6/16 – 12:28 a.m.)

Labeling - Magnifier testing revealed that the Create New Group button is a faded color.

Consequently, the magnifier tester assumed the button was inactive.



Solution:

Use the Color Contrast Analyzer listed on page 18 to select a higher contrast color combination.

From AUL report, page 11:

The bread crumb above the main content is not identified as a bread crumb to the screen reader, losing its value as a navigation tool for the screen reader user.



Solution:

Place a heading before the breadcrumb with text of "breadcrumb." Use CSS to hide this from the visual user.

From AUL report, page 12:

There are two links on the Group management page labeled "More" each with a very different purpose.

Problem:

The first leads to a display of "more" information about the group and the second offers "more" actions associated with the group. Screen reader users searching for links hear "More" without associated context and have no idea what each "More" refers to.

(ver. 9/6/16 – 12:28 a.m.)



Solution:

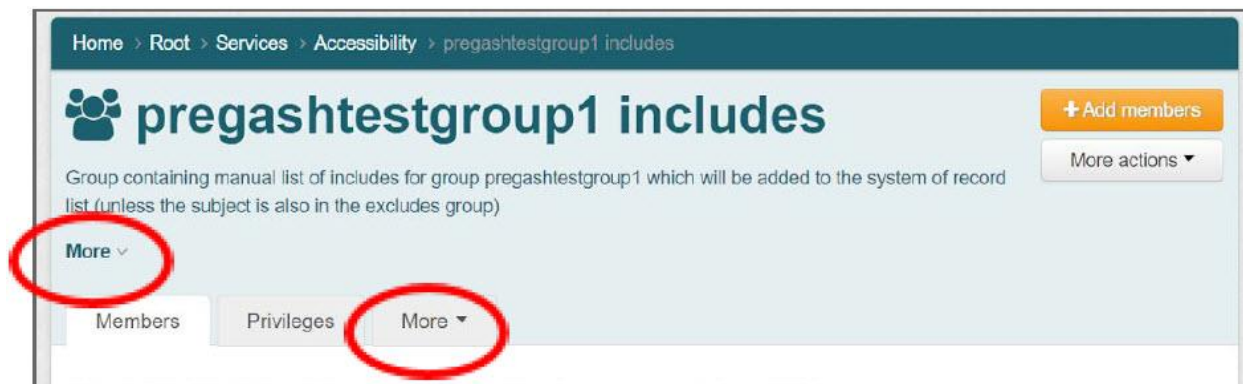
Use the aria-label attribute to rename these items for the screenreader user so that their function is clearer.

From AUL report, page 12:

Two links within screen reader link lists are labeled with “Actions” with very different purposes.

Problem:

The first pertains to group actions, the second pertains to member entity actions. Screen reader users searching for links hear “Actions” without associated context and have no idea what each “Actions” refers to.



Solution:

Same as that proposed for the previous item

(ver. 9/6/16 – 12:28 a.m.)

From AUL report, page 13:

The Add Members entity name search list is not alphabetical.

Problem:

Screen reader users have to read through the entire list name-by-name to find the desired person. If alphabetical, they could search by first letter(s).

(Note: this can also harm findability for the visual user.)

Search for an entity X

vaughan

Exact ID match

Choose a data source

Entity Name
Paula J Vaughan (vaughan)
Travis Taylor Vaughan (vaughatt)
Brenden Guy Vaughan (vaughanb)
Charles Mills Vaughan (vaughacm)
Katrina Eileen Vaughan (vaughank)
Terry Vaughan (vaughant)
Allison Taylor Vaughan (vaughaat)

Solution:

This is a programming issue. I assume Grouper programmers can sort these results alphabetically.

From AUL report, page 5:

Problem:

When “Custom privileges” is chosen during the Add Member/Assign privileges process, the new display of custom privilege choices are not announced. Screenshots below show the display before and after choosing the Custom privileges radio button (with new content highlighted).

(ver. 9/6/16 – 12:28 a.m.)



pregashtestgroup1 includes

+ Add members

More actions ▾

Member name or ID: Paula J Vaughan (vaughan)
Enter an entity name or ID, or [search for an entity](#)

Assign these privileges: Default privileges Custom privileges

Add or import a list of members



pregashtestgroup1 includes

+ Add members

More actions ▾

Member name or ID: Paula J Vaughan (vaughan)
Enter an entity name or ID, or [search for an entity](#)

Assign these privileges: Default privileges Custom privileges

MEMBER ADMIN UPDATE READ VIEW OPTIN
 OPTOUT ATTRIBUTE READ ATTRIBUTE UPDATE

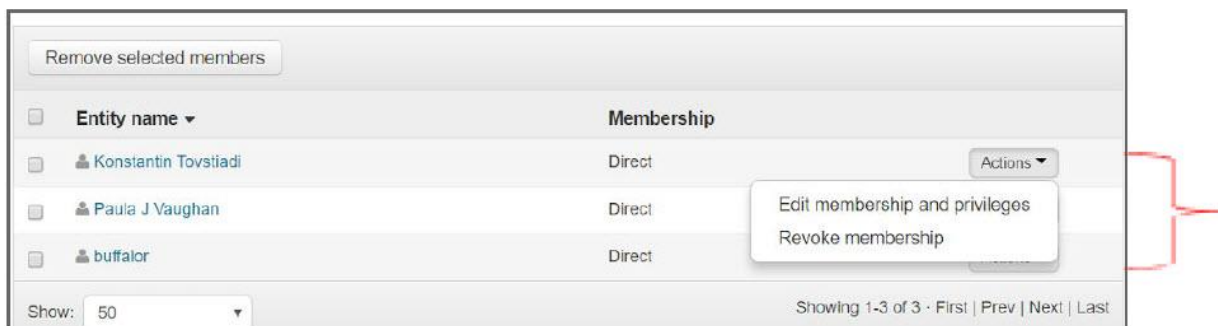
Add or import a list of members

Solution:

This problem is identical to the problem discussed on page 3: new content presented after selecting “New members” is not announced to screenreader. Follow the solutions provided there to address this similar issue.

From AUL report, page 5:

The drop down list from the entity actions button is not announced when Actions is selected. The screen reader user is left wondering what actions there might be.



Remove selected members

<input type="checkbox"/> Entity name ▾	Membership	Actions ▾
<input type="checkbox"/> Konstantin Tovstiadi	Direct	
<input type="checkbox"/> Paula J Vaughan	Direct	Edit membership and privileges Revoke membership
<input type="checkbox"/> buffalor	Direct	

Show: 50 ▾

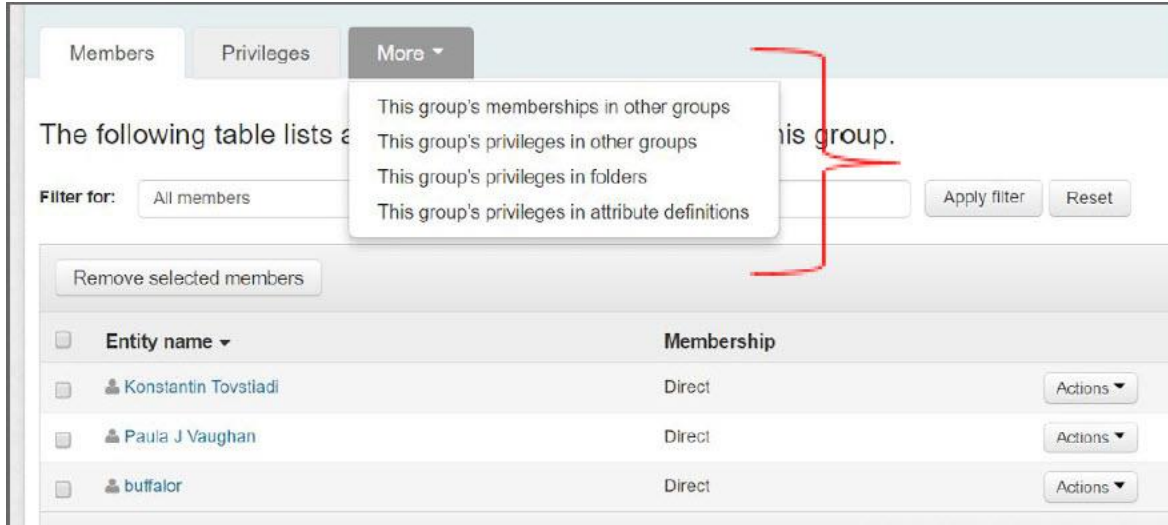
Showing 1-3 of 3 · First | Prev | Next | Last

The accessibility issue and solution for pull-down menus are discussed on pages 8 - 10.

(ver. 9/6/16 – 12:28 a.m.)

From AUL report, page 6:

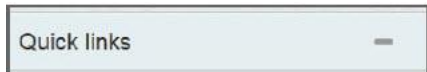
The drop down list from the More tab/link is not announced when More is selected.



The accessibility issue and solution for pull-down menus are discussed on pages 8 - 10.

From AUL report, page 10:

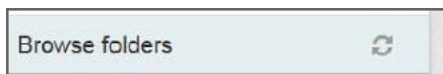
“Quick Links” is read as a button whereas it is actually an expandable list.



The accessibility issue and solution for pull-down menus are discussed on pages 8 – 10 except that in this case aria-expanded=“true” is the starting state.

From AUL report, page 10:

The graphic after “Browse Folders” is not labeled and is read only as “clickable.”



Similar to the issue of the unlabeled search icon described on page 2, the refresh icon should be labeled using the aria-label attribute.

(ver. 9/6/16 – 12:28 a.m.)

`<i class="fa fa-refresh" onclick="dojNitMenu()" style="cursor: pointer;" aria-label="refresh folder browse"></i>`

From AUL report, page 10:

The Recent Activity listing (defined as a table to the screen readers) on the Grouper home page has no meaningful labels.

The screenshot shows a web browser window with a 'Recent activity' table. The table contains six rows of activity logs. Below the table, the browser's developer tools are open, showing the HTML structure of the table. The table is defined as a table with a class of 'table table-bottom-borders'. The HTML code shows the table structure with a header row and a body row.

Recent activity	
Added Howard Kramer as a member of the HowardsGroup group.	2016/09/05 17:15 PM
Deleted Howard Kramer as a member of the HowardsGroup group.	2016/09/05 17:15 PM
Added Pregash J. Devasagayam as a member of the HowardsGroup group.	2016/09/05 17:14 PM
Added Howard Kramer as a member of the HowardsGroup group.	2016/09/04 14:28 PM
Deleted 'Admin' privilege to Howard Kramer for the group HowardsGroup.	2016/09/01 19:36 PM
Assigned 'Attribute read' privilege to Howard Kramer for the group HowardsGroup.	2016/08/30 14:02 PM

```
<div class="span12">
  <h3>Recent activity</h3>
  <table class="table table-bottom-borders">
    <thead></thead>
    <tbody>
      <tr>
        <td>
          <td>2016/09/05 17:15 PM</td>
        </tr>
    </tbody>
  </table>
</div>
```

Problem:

In other words there is no header row for this table. Note that the table begins after the heading level 3 “Recent activity.”

Solutions:

Create a header row for the table portion of the table. This header row can be hidden using CSS as shown on the bottom of page 7. In addition, CSS could be used to define the table header row has having a height of 1px (or less) to make it non-visible.

Example code:

```
<thead>
<tr>
<th>Recent activity</th>
<th>activity date</th>
</tr>
</thead>
```