Version 2.0;

Last revised: 04/07/2018

MARKER TYPES:

A marker type determines the nature of the location and the pin color on the NM map, each having its own display and expectations:

* marker = ‘Visitor Ctr’ implies a yellow pin positioned at a Visitor Center (National or State Park); The info window popup that is displayed when the yellow pin is clicked shows all hikes associated with that Visitor Center.
* marker = ‘At VC’ is a hike associated with ‘Visitor Ctr’. This designation allows the javascript to associated those hikes with the corresponding yellow pin and not place them separately on the map.
* marker = ‘Cluster’ implies a blue pin positioned at a location representing the area associated with hikes that have a common trail head or are in close proximity to other hikes. The info window popup when that pin is clicked will display all the hikes associated with that cluster. These are not otherwise displayed on the map.
* marker = ‘Normal’ implies a hike that is not associated with a Visitor Center, and does not have a trail head in common with or in close proximity to another hike. A normal hike has a red pin positioned at its trailhead, and will display the hike when in the info window popup when the red pin is clicked.

*The rest of this document deals with Cluster groups during Edit:*

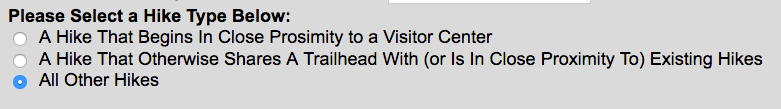
FOR marker = ‘Cluster’, two database items required definition:

* cgroup: this item is a letter (or compound letters) which identifies hikes in the same group;
* cname: this is the name of the selected group seen in the drop-down boxes (when displayed: see following section).

Cluster assignments can be applied in two different places:

1. When creating a new page: The default condition of ‘Hike Type’ radio buttons is:

Fig. 1



If the radio button is selected indicating “A Hike That Otherwise Shares a Trailhead With (or Is In Close Proximity To) Existing Hikes”, a drop-down box showing all existing hike groups (clusters) appears. If the user wishes to add a new group, a checkbox is provided, and the new group can be added on the subsequent page (Editor).

Fig. 2



IF the checkbox is checked, the data item in the drop-down will be ignored.

The page displayed is based on ‘startNewPg.php’. The data is passed to the server in the form’s action=”submitNewPg.php’. The radio button for the cluster type marker is <input id=”cluster” />, which contains the ‘Select’ and ‘Checkbox’ elements in the <em id=”cls”> block. The block is initially not displayed (see ‘startNewPg.css’), but when the radio button is selected, the javascript in ‘startNewPg.js’ displays the block (and makes sure the Visitor Center block is turned off in case it was previously displayed).

The data passed to the server is based on the option showing in the <select id=”clsel” name=”clus”> drop-down. The checkbox is <input name=”mknewgrp” />

The ‘submitNewPg.php’ script examines the passed-in marker type to see if the ‘Cluster’ type was the submitted radio button choice. The state of the checkbox is rendered as $newclus = true (or false). If the marker is ‘Cluster’ AND $newclus is false, then the cgroup and cname fields are determined by the passed-in data. The data assigned to the new page in EHIKES is based on this determination. Note that if $newclus is true, the only item set is ‘marker’, and not cgroup and cname.

After inserting the new page data into EHIKES, the page is redirected to the editor. See the following detailed description of cluster editing.

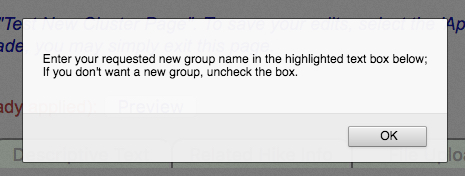
1. When editing a page:

The cluster section of the editor (tabldisplay.php) will appear slightly differently depending on whether or not the editor was invoked from the new page redirection, or directly from the user edit selection on the main page.

*State of tab1display.php cluster section when having been redirected from the submitNewPg.php script, AND having checked the ‘select a new group’ checkbox on startNewPg.php:*

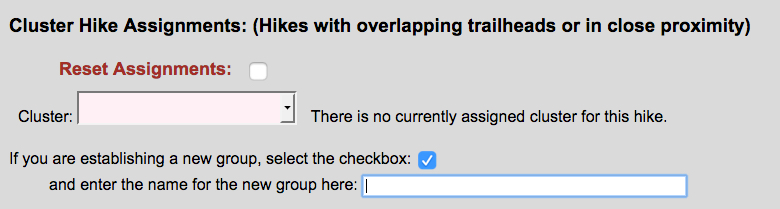
First, a popup alerts the user that a new cluster group was requested and that he/she is required to provide a name in the appropriate box for the new cluster. Also, if the user changes his/her mind, he/she may revert to ‘normal’ editing by deselecting the page’s ‘new group’ checkbox.

Fig. 3



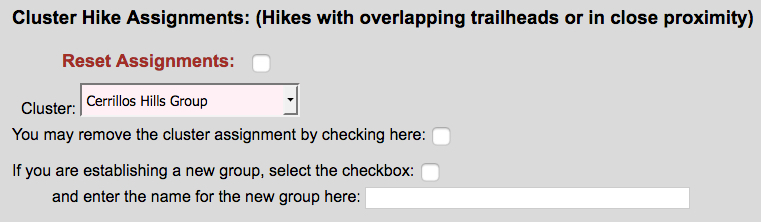
Once the window is dismissed by clicking ‘OK’, the focus moves to the new group input box with the ‘new group’ checkbox checked:

Fig. 4



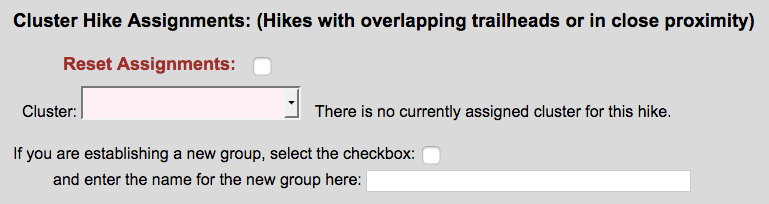
*State of tab1display.php cluster section when having been redirected from submitNewPg.php with a group selected in the drop-down on startNewPg.php and the ‘new group’ checkbox NOT checked:*

Fig. 5



*State of tab1.display.php cluster section when invoked from the main page using ‘New/Active Edits’ or edit from ‘Published’ (and hike-in-edit has no current cluster assignment):*

Fig. 6

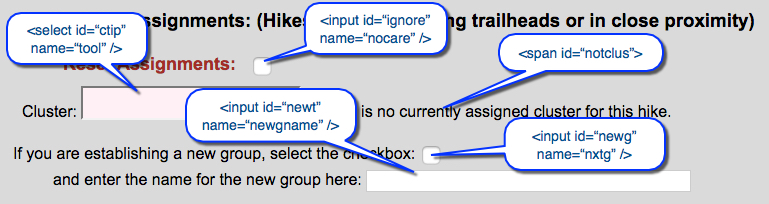


*State of tab1.display.php cluster section when invoked from the main page using ‘New/Active Edits’ or edit from ‘Published’ (and hike-in-edit has a current cluster assignment): see Fig. 5.*

The above states highlight the tab1.display data items used to prepare data for saving, via the form’s action=”saveTab1.php” script. The manipulated pieces of *visible* data are:

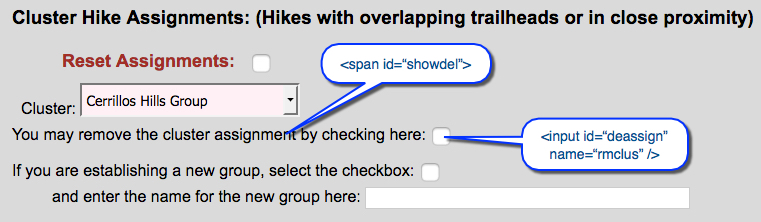
For empty cluster assignment:

Fig. 7



For existing cluster assignment:

Fig. 8



There is additional ‘hidden’ data that is required to be passed to the server in order to make correct decisions about cluster assignments, and appears in <input type=”hidden” /> elements. If the item is only used by javascript and not required by the server script, it is assigned its data value in a <p> tag. In some cases, the same data is required by both. *Both displayed and hidden data is detailed in ‘OPERATION’ below.*

**OPERATION**:

*Overall (general) operation*:

* 1. If a cluster group is not yet assigned, one may be selected from the drop-down box (<select id=”ctip”>)
  2. If a cluster group is assigned, the group may be changed by selecting a new group from the ‘Cluster’ drop-down box (<select id=”ctip”>)
  3. If the user wishes to remove the cluster group and return the marker to a “Normal” type hike, a checkbox is provided (<input id=”deassign” />)
  4. If the user wishes to specify a new group, the corresponding checkbox can be checked (<input id=”newg” />) AND a new name provided in the text box (<input id=”newt” />): NOTE: the javascript keys off of the word “Group” at the end of the text name, so please add that to the new group name.
  5. If any editing mistakes are made, checking the ‘Reset Assignments’ box (<input id=”ignore” />) will clear the edits and restore the states to their current, incoming database settings.

HTML:

The html for tab1 is contained in the module ‘tab1display.php’. It consists of *displayed items (tags)*, items (tags) that are *not displayed*, but accessible to javascript, and hidden inputs *passed* via the form submit to ‘saveTab1.php’. Tags with styling=”display:none” contain information that the javascript (js) will need in order to establish the initial values for the cluster assignments, and to modify the page when changes are made, if needed. Tags that are hidden form elements contain information needed by the php save module. These values are obtained from the database and provided by php upon loading the tab1 page.

[Key: element id; form name; value on page load]

*displayed:*

In this section of tab1 (Cluster Hike Assignments), the following tags are (or can be) displayed:

1. checkbox: label – “Reset Assignments”; id = ‘ignore’; name = “nocare”; on load, unchecked;
2. drop-down box (with available clusters): label – “Cluster”; id = “ctip”; name = “htool”; on load values = (php $cnames);
3. text: “You may remove the cluster assignment…” id=”showdel”; on load not displayed;
4. checkbox: id =“deassign”; name = “rmclus”; on load, unchecked & not displayed;
5. text: “There is currently no assigned cluster…” id=”notclus”; on load not displayed;
6. checkbox: label – “If you are establishing a new group…”; id = “newg”; name = “nxtg”; on load value = “NO”, unchecked;
7. text input: label – “… enter the name ... here”; id = “newt”; name = “newgname”; on load, no value;

*display: none* (for js)

* <p>: id = “mrkr”; onload = $hikeMarker; (marker type)
* <p>: id = “greq”; onload = $grpReq; (YES if marker type is ‘Cluster’ and cgroup is empty – this happens when the user selected ‘new group’ when creating a new page; NO otherwise;)
* <p>: id = “group”; onload = $hikeGrpTip; (cname from database)

*hidden inputs*: (for php)

* <input> name = “pmrkr”; onload = $hikeMarker; (marker type)
* <input> name = “pclus”; onload = $hikeClusGrp; (cgroup from database)
* <input> name = “pcnme”; onload = $hikeGrpTip; (cname from database)
* <input> id = “grpchg”; name = “chgd”; on load, “NO”; (“YES” when grp is changed)
* <input> name = “grpcnt”; onload = $dbCount; (count of existing cluster names in database)

VALUES LOADED BY php FROM DATABASE:

The script ‘dataForEditor.php’ collects the current database values and initializes the tab1display.php elements with those values.

JAVASCRIPT:

The javascript is currently contained in editDB.js, which serves all the tabs. For tab1, the section dealing with cluster assignments is found from line 18 – 120. As the possibilities can be complex, an accompanying state diagram may be helpful. The state diagram (below) provides a basic outline of what the display items do during transitions. The state outputs correspond to the sections identified above in Figures 7 and 8 and are arbitrarily assigned output state id’s as identified in the ‘displayed’ list above (o1 – o7).

Line 18: read the marker type from <p id=”mrkr”>;

Lines 19 – 26: text message output variables are established;

Line 27: The php-loaded incoming cluster assignment is read from the hidden text in <p id=”group”>;

Line 28: The value above is displayed in the drop-down (<select id=”ctip”>)

Lines 29 – 40:

* If a ‘new page creation’ invoked the editor, and the user chose the “…new group…” checkbox (id=”greq” will be ‘YES’), the following actions are performed:
  + Check the new group checkbox (<input id=”newg” />);
  + Change the value of the checkbox to “YES”;
  + Place the user cursor in the new group text box (<input id=”newt” />);
  + Turn on the page text that says: “There is no currently assigned cluster for this hike” (id=”notclus”);
  + Inform the user via a popup that he/she should enter the new group name highlighted, or uncheck the box to proceed without a new group name.
* Otherwise, the id=”greq” will be ‘NO’:
  + If <p id=”group”> is empty, turn on id=”notclus”;
  + Otherwise, turn on id=”showdel”;

Lines 41-50: Code comments;

Line 51: a var ‘fieldflag’ is set false: if true, the code will check to make sure an entry occurred in the new group text box when the checkbox is checked;

Lines 53 – 69: If the ‘Reset Assignments’ checkbox (id=”ignore”) is checked, restore all of the default cluster settings, alert the user that the action was performed and uncheck the checkbox;

Lines 71 – 77: If the id=”deassign” checkbox is clicked (“You may remove the cluster assignment by checking here:”), the code sets the value of that checkbox accordingly. That value gets passed to php when the form is submitted;

Lines 79 – 98: When the id=”newg” checkbox is either checked or unchecked, the appropriate actions take place: in either case ensure that the drop-down (id=”ctip”) is restored to its incoming value

* Box is checked:
  + Change the value of the checkbox to “YES”;
  + If the incoming value of the group drop-down was empty, turn off the text id=”notclus”;
  + Set the fieldflag = true (forces checking that text has been entered before applying changes);
  + Set the hidden input id=”grpchg” to “NO”;
  + Alert the user about adding a new group;
* Box is unchecked:
  + Change the value of the checkbox to “NO”;
  + If the incoming value of the group drop-down was empty, set the drop-down (id=”ctip”) to empty and turn on the appropriate text comment (id=”notclus”);
  + Clear the new group name text box (id=”newt”);
  + Set the fieldflag to false.

Lines 100 – 119: When the selection in the drop-down box (id=”ctip”) changes:

* If the id=”newg” (new group) checkbox is NOT checked (value = ‘NO’):
  + If and only the new selection is the same as the original:
* If the marker type was NOT ‘Cluster’:
  + Alert the user that the type will be changed to ‘Cluster’
  + Turn on the appropriate text
* If the marker type was already ‘Cluster’:
  + Alert the user about the old and new settings
* Set the id=”grpchg” hidden input to “YES”
* Otherwise, the id=”newg” (new group) checkbox IS checked:
  + Set the id=”grpchg” hidden input to “NO”
  + Alert user that the id=”newg” box is checked and therefore these and other actions are ignored

PHP (When form is submitted, ie “Apply” is clicked):

*The form action when submitted is ‘saveTab1.php’*

With respect to clusters, the following items are collected from the submit:

* $marker (from hidden input, name ‘pmarker’);
* $clusGrp (from hidden input, name ‘pclus’);
* $cgName (from hidden input, name ‘pcnme’); (NOTE may be reassigned below);
* $delClus (from checkbox, name ‘rmclus’);
* $nextGrp (from checkbox, name ‘nxtg’);
* $grpChg (from hidden input, name ‘chgd’);
* $cgName (from text input box, name ‘newgname’); (conditionally acquired)

The list of all known clusters (including any new ones in the editor) is created and assigned:

* $cnames -> names of the clusters
* $groups -> group letters corresponding to the names

The section of the code dealing with clusters resides in lines 93-136 with descriptive comments preceding in lines 85-92.

* Lines 97-100: Deleting a cluster if $delClus is checked: return the marker type to ‘Normal’ and set the cluster name and group to empty;
* Lines 101-117: If a new group is to be added: select the next available group letter(s); set marker type to ‘Cluster’; set group to newly selected letter(s); overwrite the value of $cgName with the text entered in ‘newgname’;
* Lines 118-131: The group drop-down value was changed: marker type is ‘Cluster’; $cgName is the value in the select box (‘htool’); the cluster letter(s) is assigned to $clusGrp by scanning through the list of name-letter associations and finding a match;
* Lines 132-135: Nothing changed in the assignment;
* Line 136: Escaping the group name for entry into the database;

BASIC STATE DIAGRAM:

