

```

1  (function ($) {
2      Drupal.behaviors.improved_multi_select = {
3          attach: function(context, settings) {
4              if (settings.improved_multi_select && settings.improved_multi_select.
5                  selectors) {
6                  var options = settings.improved_multi_select;
7
8                  for (var key in options.selectors) {
9                      var selector = options.selectors[key];
10                     $(selector, context).once('improvedselect', function() {
11                         var $select = $(this),
12                             moveButtons = '',
13                             improvedselect_id = $select.attr('id'),
14                             $cloned_select = null,
15                             cloned_select_id = '';
16                         if (options.orderable) {
17                             // If the select is orderable then we clone the original select
18                             // so that we have the original ordering to use later.
19                             $cloned_select = $select.clone();
20                             cloned_select_id = $cloned_select.attr('id');
21                             cloned_select_id += '-cloned';
22                             $cloned_select.attr('id', cloned_select_id);
23                             $cloned_select.appendTo($select.parent()).hide();
24                             // Move button markup to add to the widget.
25                             moveButtons = '<span class="move_up" sid="' + $select.attr('id') +
26                                 '">' + Drupal.checkPlain(options.buttontext_moveup) + '</span>' +
27                                 '<span class="move_down" sid="' + $select.attr('id')
28                                 + '">' + Drupal.checkPlain(options.
29                                     buttontext_movedown) + '</span>';
30                         }
31                         $select.parent().append(
32                             '<div class="improvedselect" sid="' + $select.attr('id') + '"
33                             id="improvedselect-' + $select.attr('id') + '">' +
34                             '<div class="improvedselect-text-wrapper">' +
35                             '<input type="text" class="improvedselect_filter" sid="' +
36                             $select.attr('id') + '" prev="" />' +
37                             '</div>' +
38                             '<ul class="improvedselect_sel"></ul>' +
39                             '<ul class="improvedselect_all"></ul>' +
40                             '<div class="improvedselect_control">' +
41                             '<span class="add" sid="' + $select.attr('id') + '">' + Drupal
42                             .checkPlain(options.buttontext_add) + '</span>' +
43                             '<span class="del" sid="' + $select.attr('id') + '">' + Drupal
44                             .checkPlain(options.buttontext_del) + '</span>' +
45                             '<span class="add_all" sid="' + $select.attr('id') + '">' +
46                             Drupal.checkPlain(options.buttontext_addall) + '</span>' +
47                             '<span class="del_all" sid="' + $select.attr('id') + '">' +
48                             Drupal.checkPlain(options.buttontext_delall) + '</span>' +
49                             moveButtons +
50                             '</div>' +
51                             '<div class="clear"></div>' +
52                             '</div>');
53                         if ($select.find('optgroup').has('option').length > 0) {
54                             $select.parent().find('.improvedselect').addClass('has_group');
55                             // Build groups.
56                             $('#improvedselect-' + improvedselect_id + '
57                                 .improvedselect-text-wrapper', context)

```

```

47     .after('<div class="improvedselect_tabs-wrapper" sid="" +
        $select.attr('id') + '"><ul
        class="improvedselect_tabs"></ul></div>');
48     $select.find('optgroup').has('option').each(function() {
49         $('#improvedselect-' + improvedselect_id + '
        .improvedselect_tabs', context)
50         .append('<li><a href="">' + $(this).attr('label') +
        '</a></li>');
51     });
52     // Show all groups option.
53     $('#improvedselect-' + improvedselect_id + ' .improvedselect_tabs'
        , context)
54     .prepend('<li class="all"><a href="">' + Drupal.t('All') +
        '</a></li>');
55     // Select group.
56     $('#improvedselect-' + improvedselect_id + ' .improvedselect_tabs
        li a', context).click(function(e) {
57         var $group = $(this),
58             sid = $group.parent().parent().parent().attr('sid');
59         $('#improvedselect-' + improvedselect_id + '
        .improvedselect_tabs li.selected', context).removeClass(
        'selected').find('a').unwrap();
60         $group.wrap('<div>').parents('li').first().addClass('selected');
61
62         // Any existing selections in the all list need to be unselected
63         // if they aren't part of the newly selected group.
64         if (!$group.hasClass('all')) {
65             $('#improvedselect-' + improvedselect_id + '
        .improvedselect_all li.selected[group!=' + $group.text() + ']'
        , context).removeClass('selected');
66         }
67
68         // Clear the filter if we have to.
69         if (options.groupresetfilter) {
70             // Clear filter box.
71             $('#improvedselect-' + improvedselect_id + '
        .improvedselect_filter', context).val('');
72         }
73         // Force re-filtering.
74         $('#improvedselect-' + improvedselect_id + '
        .improvedselect_filter', context).attr('prev', '');
75         // Re-filtering will handle the rest.
76         improvedselectFilter(sid, options, context);
77         e.preventDefault();
78     });
79     // Select all to begin.
80     $('#improvedselect-' + improvedselect_id + ' .improvedselect_tabs
        li.all a', context).click();
81 }
82
83 $select.find('option, optgroup').each(function() {
84     var $opt = $(this),
85         group = '';
86     if ($opt.attr('tagName') == 'OPTGROUP') {
87         if ($opt.has('option').length) {
88             $('#improvedselect-' + improvedselect_id + '
        .improvedselect_all', context)

```

```

89         .append('<li isgroup="isgroup" so="---" + $opt.attr('label')
+ '---">--- ' + $opt.attr('label') + ' ---</li>');
90     }
91 }
92 else {
93     group = $opt.parent("optgroup").attr("label");
94     if (group) {
95         group = ' group="' + group + '"';
96     }
97     if ($opt.attr('selected')) {
98         $('#improvedselect-' + improvedselect_id + '
.improvedselect_sel', context)
99         .append('<li so="' + $opt.attr('value') + '"' + group + '>'
+ $opt.text() + '</li>');
100     }
101     else {
102         $('#improvedselect-' + improvedselect_id + '
.improvedselect_all', context)
103         .append('<li so="' + $opt.attr('value') + '"' + group + '>'
+ $opt.text() + '</li>');
104     }
105 }
106 });
107 $('#improvedselect-' + improvedselect_id + ' .improvedselect_sel li,
#improvedselect-' + improvedselect_id + ' .improvedselect_all
li[isgroup!="isgroup"]', context).click(function() {
108     $(this).toggleClass('selected');
109 });
110 $select.hide();
111 // Double click feature request.
112 $('#improvedselect-' + improvedselect_id + ' .improvedselect_sel li,
#improvedselect-' + improvedselect_id + ' .improvedselect_all
li[isgroup!="isgroup"]', context).dblclick(function() {
113     // Store selected items.
114     var selected = $(this).parent().find('li.selected'),
115         current_class = $(this).parent().attr('class');
116     // Add item.
117     if (current_class == 'improvedselect_all') {
118         $(this).parent().find('li.selected').removeClass('selected');
119         $(this).addClass('selected');
120         $(this).parent().parent().find('.add').click();
121     }
122     // Remove item.
123     else {
124         $(this).parent().find('li.selected').removeClass('selected');
125         $(this).addClass('selected');
126         $(this).parent().parent().find('.del').click();
127     }
128     // Restore selected items.
129     if (selected.length) {
130         for (var k = 0; k < selected.length; k++) {
131             if ($(selected[k]).parent().attr('class') == current_class) {
132                 $(selected[k]).addClass('selected');
133             }
134         }
135     }
136 });

```

```

137
138     // Set the height of the select fields based on the height of the
139     // parent, otherwise it can end up with a lot of wasted space.
140     $('.improvedselect_sel, .improvedselect_all').each(function() {
141         $(this).height($(this).parent().height() - 35);
142     });
143 });
144 }
145
146 $('.improvedselect_filter', context).keyup(function() {
147     improvedselectFilter($(this).attr('sid'), options, context);
148 });
149
150 // Add selected items.
151 $('.improvedselect .add', context).click(function() {
152     var sid = $(this).attr('sid');
153     $('#improvedselect-' + sid + ' .improvedselect_all .selected', context
154     ).each(function() {
155         $opt = $(this);
156         $opt.removeClass('selected');
157         improvedselectUpdateGroupVisibility($opt, 1);
158         $('#improvedselect-' + sid + ' .improvedselect_sel', context).append
159         ($opt);
160     });
161     improvedselectUpdate(sid, context);
162 });
163
164 // Remove selected items.
165 $('.improvedselect .del', context).click(function() {
166     var sid = $(this).attr('sid');
167     $('#improvedselect-' + sid + ' .improvedselect_sel .selected', context
168     ).each(function() {
169         $opt = $(this);
170         $opt.removeClass('selected');
171         $('#improvedselect-' + sid + ' .improvedselect_all', context).append
172         ($opt);
173         improvedselectUpdateGroupVisibility($opt, 0);
174     });
175     // Force re-filtering.
176     $('#improvedselect-' + sid + ' .improvedselect_filter', context).attr(
177     'prev', '');
178     // Re-filtering will handle the rest.
179     improvedselectFilter(sid, options, context);
180     improvedselectUpdate(sid, context);
181 });
182
183 // Add all items.
184 $('.improvedselect .add_all', context).click(function() {
185     var sid = $(this).attr('sid');
186     $('#improvedselect-' + sid + ' .improvedselect_all
187     li[isgroup!=isgroup]', context).each(function() {
188         $opt = $(this);
189         if ($opt.css('display') != 'none') {
190             $opt.removeClass('selected');
191             improvedselectUpdateGroupVisibility($opt, 1);
192             $('#improvedselect-' + sid + ' .improvedselect_sel', context).
193             append($opt);

```

```

187     }
188   });
189   improvedselectUpdate(sid, context);
190 });
191
192 // Remove all items.
193 $('improvedselect .del_all', context).click(function() {
194   var sid = $(this).attr('sid');
195   $('#improvedselect-' + sid + ' .improvedselect_sel li', context).each(
196     function() {
197       $opt = $(this);
198       $opt.removeClass('selected');
199       $('#improvedselect-' + sid + ' .improvedselect_all', context).append
200         ($opt);
201       improvedselectUpdateGroupVisibility($opt, 0);
202     });
203   // Force re-filtering.
204   $('#improvedselect-' + sid + ' .improvedselect_filter', context).attr(
205     'prev', '');
206   // Re-filtering will handle the rest.
207   improvedselectFilter(sid, options, context);
208   improvedselectUpdate(sid, context);
209 });
210
211 // Move selected items up.
212 $('improvedselect .move_up', context).click(function() {
213   var sid = $(this).attr('sid');
214   $('#improvedselect-' + sid + ' .improvedselect_sel .selected', context
215     ).each(function() {
216       var $selected = $(this);
217       // Don't move selected items past other selected items or there will
218       // be problems when moving multiple items at once.
219       $selected.prev(':not(.selected)').before($selected);
220     });
221   improvedselectUpdate(sid, context);
222 });
223
224 // Move selected items down.
225 $('improvedselect .move_down', context).click(function() {
226   var sid = $(this).attr('sid');
227   // Run through the selections in reverse, otherwise problems occur
228   // when moving multiple items at once.
229   $('#improvedselect-' + sid + ' .improvedselect_sel .selected',
230     context).get().reverse().each(function() {
231     var $selected = $(this);
232     // Don't move selected items past other selected items or there will
233     // be problems when moving multiple items at once.
234     $selected.next(':not(.selected)').after($selected);
235   });
236   improvedselectUpdate(sid, context);
237 });
238 }
239 }
240 };
241
242 /**
243  * Filter the all options list.

```

```

239  */
240  function improvedselectFilter(sid, options, context) {
241    $filter = $('improvedselect_filter', context);
242    // Get current selected group.
243    var $selectedGroup = $('#improvedselect-' + sid + 'improvedselect_tabs
li.selected:not(.all) a', context),
244    text = $filter.val(),
245    pattern,
246    regex,
247    words;
248
249    if (text.length && text !== $filter.attr('prev')) {
250      $filter.attr('prev', text);
251      switch (options.filtertype) {
252        case 'partial':
253        default:
254          pattern = text;
255          break;
256        case 'exact':
257          pattern = '^' + text + '$';
258          break;
259        case 'anywords':
260          words = text.split(' ');
261          pattern = '';
262          for (var i = 0; i < words.length; i++) {
263            if (words[i]) {
264              pattern += (pattern) ? '\\b' + words[i] + '\\b' : '\\b' + words[i
] + '\\b';
265            }
266          }
267          break;
268        case 'anywords_partial':
269          words = text.split(' ');
270          pattern = '';
271          for (var i = 0; i < words.length; i++) {
272            if (words[i]) {
273              pattern += (pattern) ? '|' + words[i] + ' : words[i];
274            }
275          }
276          break;
277        case 'allwords':
278          words = text.split(' ');
279          pattern = '^';
280          // Add a lookahead for each individual word.
281          // Lookahead is used because the words can match in any order
282          // so this makes it simpler and faster.
283          for (var i = 0; i < words.length; i++) {
284            if (words[i]) {
285              pattern += '(?=.*?\\b' + words[i] + '\\b)';
286            }
287          }
288          break;
289        case 'allwords_partial':
290          words = text.split(' ');
291          pattern = '^';
292          // Add a lookahead for each individual word.
293          // Lookahead is used because the words can match in any order

```

```

294         // so this makes it simpler and faster.
295         for (var i = 0; i < words.length; i++) {
296             if (words[i]) {
297                 pattern += '(?=.*?' + words[i] + ')*';
298             }
299         }
300         break;
301     }
302
303     regex = new RegExp(pattern, 'i');
304     $('#improvedselect-' + sid + ' .improvedselect_all li', context).each(
305         function() {
306             $opt = $(this);
307             if ($opt.attr('isgroup') != 'isgroup') {
308                 var str = $opt.text();
309                 if (str.match(regex) && (!$selectedGroup.length || $selectedGroup.text
310                     () == $opt.attr('group'))) {
311                     $opt.show();
312                     if ($opt.attr('group')) {
313                         // If a group is selected we don't need to show groups.
314                         if (!$selectedGroup.length) {
315                             $opt.siblings('li[isgroup="isgroup"][so="---" + $opt.attr(
316                                 'group') + '---"]').show();
317                         }
318                     }
319                 }
320             }
321             else {
322                 $opt.siblings('li[isgroup="isgroup"][so="---" + $opt.attr(
323                     'group') + '---"]').hide();
324             }
325         }
326     }
327 }
328 }
329 });
330 }
331 else {
332     if (!text.length) {
333         $filter.attr('prev', '');
334     }
335     $('#improvedselect-' + sid + ' .improvedselect_all li', context).each(
336         function() {
337             var $opt = $(this);
338             if ($opt.attr('isgroup') != 'isgroup') {
339                 if (!$selectedGroup.length || $selectedGroup.text() == $opt.attr(
340                     'group')) {
341                     $opt.show();
342                 }
343             }
344             else {
345                 $opt.hide();

```

```

343     }
344     improvedselectUpdateGroupVisibility($opt, 0);
345     }
346   });
347 }
348 }
349
350 /**
351  * Update the visibility of an option's group.
352  *
353  * @param $opt
354  *   A jQuery object of a select option.
355  * @param numItems
356  *   How many items should be considered an empty group. Generally zero or one
357  *   depending on if an item has been or is going to be removed or added.
358  */
359 function improvedselectUpdateGroupVisibility($opt, numItems) {
360   var $selectedGroup = $opt.parents('.improvedselect').first().find(
361     '.improvedselect_tabs li.selected:not(.all) a');
362
363   // Don't show groups if a group is selected.
364   if ($opt.parent().children('li[isgroup!="isgroup"][group="" + $opt.attr(
365     'group') + "]:visible').length <= numItems || $selectedGroup.length) {
366     $opt.siblings('li[isgroup="isgroup"][so="---" + $opt.attr('group') +
367       '---"]').hide();
368   }
369   else {
370     $opt.siblings('li[isgroup="isgroup"][so="---" + $opt.attr('group') +
371       '---"]').show();
372   }
373 }
374
375 function improvedselectUpdate(sid, context) {
376   // If we have sorting enabled, make sure we have the results sorted.
377   var $select = $('#' + sid),
378       $cloned_select = $('#' + sid + '-cloned');
379
380   if ($cloned_select.length) {
381     $select.find('option, optgroup').remove();
382     $('#improvedselect-' + sid + ' .improvedselect_sel li', context).each(
383       function() {
384         var $li = $(this);
385         $select.append($('</option>').attr('value', $li.attr('so')).attr(
386           'selected', 'selected').text($li.text()));
387       });
388     // Now that the select has the options in the correct order, use the
389     // cloned select for resetting the ul values.
390     $select = $cloned_select;
391   }
392   else {
393     $select.find('option:selected').attr("selected", "");
394     $('#improvedselect-' + sid + ' .improvedselect_sel li', context).each(
395       function() {
396         $('#' + sid + ' [value="" + $(this).attr('so') + "]', context).attr(
397           "selected", "selected");
398       });
399   }
400 }

```



```
392
393     $select.find('option, optgroup').each(function() {
394         $opt = $(this);
395         if ($opt.attr('tagName') == 'OPTGROUP') {
396             if ($opt.has('option').length) {
397                 $('#improvedselect-' + sid + ' .improvedselect_all', context).append($
                    ('#improvedselect-' + sid + ' .improvedselect_all [so="---' + $opt.
                        attr('label') + '---"]', context));
398             }
399         }
400         else {
401             // When using reordering, the options will be from the cloned select,
402             // meaning that there will be none selected, which means that items
403             // in the selected list will not be reordered, which is what we want.
404             if ($opt.attr("selected")) {
405                 $('#improvedselect-' + sid + ' .improvedselect_sel', context).append($
                    ('#improvedselect-' + sid + ' .improvedselect_sel [so="' + $opt.attr(
                        'value') + '"', context));
406             }
407             else {
408                 $('#improvedselect-' + sid + ' .improvedselect_all', context).append($
                    ('#improvedselect-' + sid + ' .improvedselect_all [so="' + $opt.attr(
                        'value') + '"', context));
409             }
410         }
411     });
412     // Don't use the $select variable here as it might be the clone.
413     // Tell the ajax system the select has changed.
414     $('#'+ sid, context).trigger('change');
415 }
416
417 })(jQuery, Drupal);
418
```