MenuOptions Documentation

Release 1.8.3-8

Mike Etts

Feb 17, 2018

Contents

CHAPTER 1

MenuOptions was created for one reason:

To reduce - *to an absolute minimum* - the # of keystrokes and clicks required for data entry & navigation.

CHAPTER 2

What it looks like:

Features:

- Input masking
- multi-column autcomplete
- menu system based on JSON
- rocker control
- auto-configuration

Other benefits:

- uses color highlighting to show autocomplete matches
- mouseover filtering to reduce choices
- it can utilize data from a variety of JSON types (array, array of arrays, single object, array of objects)
- the value associated with with the label string is saved in the input element automatically (in the menu_opt_key - no need to manually update a hidden field)

CHAPTER 3

Prerequisites:

- jQuery version >=1.9
- jQuery ui version >= 1.10
- download MenuOptions from git
- download MenuOptions from npm

CHAPTER 4

See the live examples

at MenuOptions.org

Contents:

4.1 Quick start instructions

4.1.1 Installation

```
npm install menuoptions
        -- or --
git clone https://github.com/compsult/MenuOptions.git
```

The important part of the installation is under *dist*

Here are the 6 files needed for MenuOptions. Note: this is a subset of the directory tree

```
├── css
│   └── menuoptions.min.css
├── imgs
│   ├── greencheck.png
│   ├── red_x.png
│   ├── rocker.png
│   └── ui-bg_glass_75_dadada_1x400.png
└── js
    └── jquery.menuoptions.min.js
```

You can use any directory name for the javascript and css files but the images directory needs to be called *imgs* (it's referenced in the menuoptions.css file). You can get around this restriction by editing menuoptions.css to use your directory name.

4.1.2 Quick start to create an input mask

Below are examples of using the currently available input masks

```
$('input#MdYtest').menuoptions({
    "ClearBtn": true,
    "Mask": "Mon DD, YYYY"
});
$('input#YMDtest').menuoptions({
    "ClearBtn": true,
    "Mask": "YYYYMMDD"
});
$('input#Phonetest').menuoptions({
    "ClearBtn": true,
    "Mask": "(999) 999-9999"
});
$('input#Timetest').menuoptions({
    "ClearBtn": true,
    "Mask" : 'HH:MM AM'
});
```

4.1.3 Quick start to create menu

see Quick start menu demo

Pass in a array of objects in the format:

```
{ <textToDisplay> : <href link|javascript> }
```

to MenuOptions to create a simple drop down menu.

The example below has 2 URLs and a javascript command.

```
var Data =  [ {"javascript": function() { alert('Run some javascript'); }},
              {"Google": "http://www.google.com"},
              {"Yahoo": "http://www.yahoo.com"}];

$('button[id$="menutest"]').menuoptions({
    "Data": Data;
    "MenuOptionsType": "Navigate", // Navigate is for menus
});
```

4.1.4 Quick start to create select drop down

see Quick start select demo

You can create a select drop down with a simple array:

```
var Data = [ "January","February","March","April","May","June","July",
              "August","September","October","November","December" ];

$('input#selecttest').menuoptions({
    "Data": Data
});
```

4.2 Masks

4.2.1 How pre-defined masks work

Note : User defined masks work slightly differently than pre-defined masks masks

1. Each key stroke is evaluated.
2. If the key stroke is valid, the standard mask will be shown.
3. If the key stroke is invalid, allowable values will be shown.
4. For fixed length masks, a green check will appear when the input is both valid and complete.

See masks demo here

4.2.2 Mask key specifications

Help

You can specify one of three positions to show help (and error) messages

Notes:

1. the default is 'right' (the other options are 'top' and 'bottom')

```
$('input#YMDtest').menuoptions({
    "onSelect": function(mo, data) {
         console.log(mo, data.newVal, data.newCode, data.type );
     },
    "ClearBtn": true,
    "Help": 'bottom', // or 'top' or 'right'
    "Mask": "YYYYMMDD"
});
```

4.2.3 User defined Masks

Requirements for user defined masks

1. **"Mask" must be an object** (not a string, as in pre-defined masks)
2. **"Mask" object must contain the 'Whole' key** which specifies the use defined RegExp

Notes:

1. You do not get the character by character validation (and therefore, the character specific error messages) with user defined masks.
2. If you use FixedLen with a user defined masks, it is helpful to make the 'HelpMsg' massage be that exact length. This will make the progress highlighting behave as intended (i.e., showing the user how many valid characters were entered and how many need to be entered to fufill the 'FixedLen' requirement).

Example

```
$('input#DrName2').menuoptions({
    "ClearBtn": true,
    "Help": 'bottom',
    "Mask": {
            Whole : '^[ A-Za-z0-9\-.,]*$',
            HelpMsg : "Doctor Name"
            },
    "Justify": 'left'
});
```

Whole (required)

This is the RegExp that matches the input and signifies that the input is completed (the one exception is if the FixedLen is defined - in that case, the input is not complete until the FixedLen character count is reached)

HelpMsg (optional)

When using a User Defined masks, you can specify the Help message text to display while user is inputting data into the input element.

FixedLen (optional)

When using a User Defined masks, you can specify a FixedLen. To pass validation, the input string must be this exact length.

4.2.4 Pre-defined masks

YYYYMMDD

```
"Mask": "YYYYMMDD"
```

Mon DD, YYYY

```
"Mask": "Mon DD, YYYY"
```

USphone

Note: the "Phone" mask saves the phone number as numbers (formatting is stripped) in the menu_opt_key

```
"Mask": "USphone"
```

HH:MM AM

```
"Mask": "HH:MM AM"
```

Money

Note: the "Money" mask saves the amount as a float in the menu_opt_key

```
"Mask": "Money"
```

See masks demo

4.3 Parameters specifications for menus

Show me the menu demo

4.3.1 Parameter list for menus

Table 4.1: Paramters for Menus

Parameter	Type	Allowable Values	Default	Required
BootMenuOfs	integer	positive integer	125	false
ColumnCount	integer	positive integer	1	false
Data	JSON object	array, object or array of objects	none	true
Filters	array of objects	{'str':'str'} or {'str':'RegExp'}	none	false
MenuOptionsTypeM	string	'Select' or 'Navigate'	'Select'	false
ShowAt	string	'Bottom' or 'Right'	'Bottom'	false
ShowDownArrow	string	'None or <color>'	'black'	false
Sort	array of strings	['alpha'|'num', 'desc'|'asc']	['alpha','asc']	false
Width	integer	positive integer	width of parent	false
Window	string	'repl'or 'new'	repl	false

4.3.2 Parameters explained for menus

BootMenuOfs

options: **positive integer**

BootMenuOfs is useful to control where the menu appears when you have a Bootstrap navbar menu that has been expanded from a collapsed state. This allows control of how far from the left that the menu will appear

```
'BootMenuOfs': 150,
```

ColumnCount

options: **positive integer**

MenuOptions defaults to a single column. To show have more than one column, use the *ColumnCount* parameter.

Data

options: **{}, or [{}, {}, . . .]**

MenuOptions menus accept the following in *Data*

1. **a single multikey object** { 1:"Jan", 2:"Feb",. . . }
2. **an array of single key objects** single key [{1:"Jan"},{2:"Feb"}. . .]

Filters

options: **[{ 'text : 'text' }, . . .] or [{'text': 'RegExp'}, . . .]**

Filters enable mouseover filering of menu items You can filter by plain text or by regular expression Here is an example of using Filters with a RegExp (Filters demo)

```
'Filters': [{ 'Biz' : '^(CNBC|MarketWatch)'}, {'Search' :'^(Google|Yahoo)'} ],
```

MenuOptionsType

options: **'Select' or 'Navigate' or 'Rocker'**

MenuOptions defaults to "Select". To create a menu drop down, call menuoptions with MenuOptionsType = "Navigate"

```
'MenuOptionsType': 'Navigate'
```

ShowAt

options: **'bottom' or 'right'**

MenuOptions accepts a string to tell it where to display the menu

```
"Bottom" means that the menu will appear underneath
"Right" means that the menu will appear to the right
```

ShowDownArrow

options: **None or <color>**

ShowDownArrow defaults to "black", meaning a down arrow will automatically be appended to the end of a menu drop down in the color black. Set ShowDownArrow to "None" if you would rather not see this arrow. Set ShowDownArrow to "silver" if you would like the arrow color to be silver.

```
'ShowDownArrow': 'silver'
```

Sort

options: **['alpha' or 'num', 'desc' or 'asc']**

Setting the property to an empty array will cause a Data array (or array of objects) to be displayed in the original order. With no sort, a single object will be displayed in random order.

Width

options: **positive integer**

MenuOptions will try to match the width of the parent element (it may be wider if the contents cannot fit). The Width parameter allows the user to override the default width.

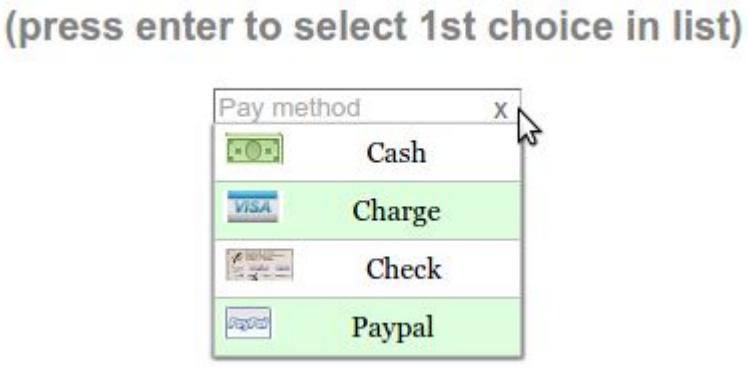
Window

options: **"repl" or "new"**

When a menu itme is clicked, you can opt to have a new browser window open by using the "new" option. The default will be to replace the current URL with the one that was just clicked.

```
'Window': 'new'
```

4.4 Parameters specifications for multi-column autocomplete



Show me the multi-column autocomplete demo

4.4.1 Parameter list for multi-column autocomplete

Table 4.2: Parameters for multi-column autocomplete

Parameter	Type	Allowable Values	Default	Required
ClearBtn	boolean	true or false	true	false
ColumnCount	integer	positive integer	1	false
Data	JSON object	(see Data section)	none	true
DataKeyNames	object	(see DataKeyNames section)	none	false
Disabled	boolean	true or false	false	false
DisableHiLiting	boolean	true or false	false	false
Filters	array of objects	{'str':'str'} or {'str':'RegExp'}	none	false
Height	integer	positive integer	height of dropdown	false
Help	strimg	'right'|'top'|'bottom'	'right'	false
InitialValue	object	{'ky'|'val': <value>}	{}	false
Justify	string	'right'|'left'|'center'	left	false
MenuOptionsType	string	'Select'|'Navigate'|'Rocker'	'Select'	false
onSelect	function	function()	none	false
PlaceHolder	<deleted>	<as of v1.6.1>	–	–
SelectOnly	boolean	true or false	false	false
ShowAt	string	'Bottom' or 'Right'	'Bottom'	false
Sort	array of strings	['alpha'|'num', 'desc'|'asc']	['alpha','asc']	false
TriggerEvent	<deleted>	<as of v1.5.1>	–	–
UseValueForKey	boolean	true or false	false	false
Width	integer	positive integer	width of dropdown	false

4.4.2 Parameters explained for MenuOption multi-column autocomplete

ClearBtn

options: **true or false**

ClearBtn instructs MenuOptions to place a clear button on the right side of the <input> element. It will clear that <input> element when clicked and cause the drop down list to appear.

ColumnCount

options: **positive integer**

MenuOptions defaults to a single column. To show have more than one column, use the ColumnCount parameter.

Data

options: **[] or [[], [], . . .], {} or [{}, {}, . . .]**

MenuOptions accepts the following in *Data*

1. **an array** ["Jan","Feb","Mar"...]

2. **an array of arrays** [[“Jan”,”Feb”,”Mar”],[“Apr”,”Jun”,”Jul”]. . .]
3. **a single multikey object** { 1:”Jan”, 2:”Feb”,. . . }
4. **an array of multikey or single key objects**
    (a) single key [{1:”Jan”},{2:”Feb”}. . .]
    (b) multikey [{1:”Jan”,2:”Feb”},{3:”Mar”, 4:”Apr”}. . .]
5. **an array of multikey objects where keys are specifed and extracted** (see DataKeyNames)

Notes:

1. Use arrays when you want the menu_opt_key to equal the displayed text, e.g., when using US State abbreviations. So the display would be ‘AL’ (for Alabama) and ‘AL’ would be stored in the menu_opt_key
2. Use objects when you want to save a code in menu_opt_key For example, if you want to display ‘January’ but save the code 1 in the menu_opt_key (and later save that code in a database or other persistent storage).

DataKeyNames

options: **{ “key”: <key name>, “value”: <value name> }**

`DataKeyNames` allows you to utilize Data that has extra, unneeded data, only picking out the key and value fields that you specify.

Below is the code used in Quick start multi-column autocomplete

```
var data = [{"mon_num":1, "mon_name":"January", "junk_key":"junk_val"},
            {"mon_num":2, "mon_name":"February", "junk_key":"junk_val"},
            {"mon_num":3, "mon_name":"March", "junk_key":"junk_val"},
            {"mon_num":4, "mon_name":"April", "junk_key":"junk_val"},
            {"mon_num":5, "mon_name":"May", "junk_key":"junk_val"},
            {"mon_num":6, "mon_name":"June", "junk_key":"junk_val"},
            {"mon_num":7, "mon_name":"July", "junk_key":"junk_val"},
            {"mon_num":8, "mon_name":"August", "junk_key":"junk_val"},
            {"mon_num":9, "mon_name":"September", "junk_key":"junk_val"},
            {"mon_num":10, "mon_name":"October", "junk_key":"junk_val"},
            {"mon_num":11, "mon_name":"November", "junk_key":"junk_val"},
            {"mon_num":12, "mon_name":"December", "junk_key":"junk_val"}];
$('input#selecttest').menuoptions({
    "Data": data,
    "onSelect": function(mo, data) {
        console.log(mo, data.newVal, data.newCode, data.type );
    },
    "DataKeyNames" : { "key": "mon_num", "value": "mon_name" },
    "ClearBtn": true,
    "InitialValue": { 'val': 'December'},
    "ShowAt": 'bottom',
    "Sort": []
});
$('input#scrolltest').menuoptions({
```

Notes:

1. `DataKeyNames` requires that Data be an array of objects **[{}, {}, . . .]**

Disabled

options: **true or false**

default: **false**

Setting Disabled to true will make the MenuOptions control disabled.

Setting Disabled to false will make the MenuOptions control enabled.

Note:

1. Setting Diasbled to true will disable the parent of the MenuOptions control. If you placed another element in that same , it will disabled as well To get around this, put the other element outside the containing the MenuOptions control.

DisableHiLiting

options: **true or false**

default: **false**

There are 3 conditions that will cause the background of an MenuOptions input element to change to pink (indicating incomplete or error).

1. the user type in an invalid character (i.e., a character not in the multi-column autocomplete).
2. when leaving the MenuOptions input element (blur event) and the value is incomplete.
3. when using setting the intial value (using InitialValue) and that initial value is invalid

Set to *true* to disable this feature.

Filters

options: **[{ 'text : 'text' }, . . .] or [{'text': 'RegExp'}, . . .]**

Filters enable mouseover filtering of multi-column autocomplete items You can filter by plain text or by regular expression (Example of using Filters with a RegExp)

```
'Filters': [{ 'Biz' : '^(CNBC|MarketWatch)'}, {'Search' :'^(Google|Yahoo)'} ],
```

Height

options: **positive integer**

MenuOptions will default to the actual height of the multi-column autocomplete list. Using the Height parameter is useful to get the exact height you want for your application. It is also useful for large list, since it will create a vertical scroll bar. (Example of using Height to create a vertical scroll bar)

```
'Height': 200
```

Help

You can specify one of three positions to show help (and error) messages

Notes:

1. the default is 'right' (the other options are 'top' and 'bottom')

```
$('input#YMDtest').menuoptions({
    "onSelect": function(mo, data) {
         console.log(mo, data.newVal, data.newCode, data.type );
     },
    "ClearBtn": true,
    "Help": 'bottom' // or 'top' or 'right'
});
```

InitialValue

options: **{ 'ky' or 'val' : <value> }**

You can use InitialValue to set (or reset) an initial value.

Note: you can use this to set the initial value (visible on the screen) or to set the key (the menu_opt_key) or both key and value. (see a demo that uses InitialValue)

These examples show using both forms of *InitialValue*

```
'InitialValue': { 'val':'Sicilian' },
'InitialValue': { 'ky': 1 },
```

Note: *InitialValue* can only be used at initialization time.

To reset the value after that time, use set_select_value

Justify

options: **left|right|center**

This will justify the text in the input element

```
"Justify": 'right',
```

MenuOptionsType

options: **'Select' or 'Navigate' or 'Rocker'**

MenuOptions defaults to "Select". To create a menu drop down, call menuoptions with MenuOptionsType = "Navigate". To create a Rocker control, call menuoptions with MenuOptionsType = "Rocker"

```
'MenuOptionsType': 'Navigate'
```

onSelect

options: **function(<MenuOptions instance>, data)**

When user selects an option, either by clicking or by pressing enter while in the text box, this function will be executed

data has 3 values

1. newVal (the new value that was selected)

2. newCode (the code that corresponds to new value that was selected)
3. type (this tells you if the selection was made by "Click", "EnterKey" or "RockerClick")

```
"onSelect": function(mo, data) {
    if ( data.type == "EnterKey" ) {
        $("form#tst").submit();
    }
    console.log(mo, data.newVal, data.newCode, data.type );
},
```

PlaceHolder

<deleted> as of version 1.6.1

Due to the whimsical nature of IE, (triggering an input event when placeholder is changed), this feature was removed

SelectOnly

options: **true or false**

This makes the <input> element read-only, i.e., data can only be entered by clicking a multi-column autocomplete item (note: this prevents the use of autocomplete).

ShowAt

options: **'bottom' or 'right'**

MenuOptions accepts a string to tell it where to display the multi-column autocomplete items

```
"Bottom" means that the multi-column autocomplete list will appear underneath
"Right" means that the multi-column autocomplete list will appear to the␣
↪right
```

Sort

options: **['alpha' or 'num', 'desc' or 'asc']**

Setting the property to an empty array will cause a Data array (or array of objects) to be displayed in the original order. With no sort, a single object will be displayed in random order.

UseValueForKey

options: **true or false**

UseValueForKey = true means that the visible option will be the same as the menu_opt_key. So if the visible option were "CA", the html built would look like:

```
<td menu_opt_key="CA">"CA"</td>.
```

Width

options: **positive integer**

MenuOptions will try to match the width of the parent element (it may be wider if the contents cannot fit). The Width parameter allows the user to override the default width.

4.5 User methods

(Click here to see demo that resets MenuOptions options & resets a MenuOptions input field)

4.5.1 add_menuoption_key *[deprecated]*

Alternative to add_menuoption_key

A call to MenuOptions with no parameters will auto-configure

Explanation:

if there is a **key** (a code repesenting the value) in the input field it will be replaced with the **value** (the text the user should see) and the menu_opt_key will be set to the **key** Alternatively, if there is a **value** in the input field it will be left as is and the menu_opt_key will be set to the key

```
$('input#delivery').menuoptions();
```

Useful for when a **value** is pasted into a select list field, *add_menuoption_key* will set the menu_opt_key, based on the text visible in the input field. So, for example, the user pasted "January" inot the month field, calling *add_menuoption_key* will cause the month code to be placed in the menu_opt_key field.

Usage:

```
$(<selector>).menuoptions('add_menuoption_key');
```

Thise example shows using *add_menuoption_key*

```
$('input#delivery').menuoptions('add_menuoption_key');
```

4.5.2 set_select_value *[deprecated]*

(Use this method instead)

allows the select list field to be set programmatically. Pass in an object with either 'ky' or 'val' as the key and the actual value.

Usage:

```
$(<selector>).menuoptions('set_select_value', { 'ky'|'val': <value>});
```

These examples show using both forms of *set_select_value*

```
$('input#delivery').menuoptions('set_select_value', {'val': 'Delivered'});
$('input#crust').menuoptions('set_select_value', {'ky': '3'}); // Thick
```

Note: to clear out a Rocker control (reset), set the 'val' to '' (empty string).

```
$('input#delivery').menuoptions('set_select_value', {'val': ''});
```

4.5.3 call MenuOptions with no parameters *(replaces set_select_value)*

```
// ----   Step #1 ----
// set the input contents using the value
$('input#delivery').val('pickup');
//        -- OR --
// set the input contents using the key
$('input#delivery').val('1');
// ----   Step #2 ----
// call MenuOptions with no parameters will auto-configure
$('input#delivery').menuoptions();
//        -- OR --
// call MenuOptions with no parameters to auto-configure ALL the MenuOptions widgets␣
↪on page
$('.ui-menuoptions').menuoptions();
```

4.5.4 refreshData *[deprecated]*

refreshData allows all parameters to be dynamically reset

Usage:

```
$(<selector>).menuoptions('refreshData', { 'option': 'option value', ...});
```

Using refreshData is no longer required to reset MenuOptions parameters.

4.5.5 [resetting MenuOptions data] *(replaces refreshData)*

```
$(<selector>).menuoptions({ 'option': 'option value', ...});
```

This example shows resetting a select list from input type to Rocker type and removing any previous Sort instructions

```
$('input#pizzatype').menuoptions({"MenuOptionsType":"Rocker", "Sort": []});
```

This example shows resetting a select list's Data

```
$('input#delivery').menuoptions({"Data": { 1: "Deliver", 2:"Pick up" } });
```

This example shows resetting a select list's Width

```
$('input#delivery').menuoptions({'Width' : 100 });
```

This example shows making a select list display to the right (instead of at bottom)

```
$('input#delivery').menuoptions({"ShowAt" : "right"});;
```

This example shows resetting a select list's ColumnCount

```
$('input#pizzatype').menuoptions({'ColumnCount' : 2 });
```

4.6 Using MenuOptions select with serialize

4.6.1 Pass output of serialize() to re_serialize()

See re_serialize demo with documention

in a call like the one below:

```
$('input#selecttest').menuoptions('re_serialize',$('form').serialize());
```

Note:

The selector must be an element that has been initialized with MenuOptions. So in the example above, $('input#selecttest') had to have been used initialized with a MenuOptions call (initialization example below).

```
 PayMethod   = { 1: "American Express", 2: "Visa", 3: "Mastercard", 4:"Discover", 5:
↪"Check",
                 6:"PayPal", 7:"Cash", 8:"Money Order"}

$('input#selecttest').menuoptions({
    "Data": PayMethod
});
```

4.7 Using dividers in a menu

4.7.1 Rules to use a divider

1. You can only use dividers in a single column menu. In other words, the ColumnCount must be set to 1
2. The Data must be an object or array of objects
3. The value must be set to 'divider'

```
{  'Search'  :'divider' }
             ^^^^^^^
```

in a call like the one below:

```
var Menu_w_Dividers =[ {  'Search'  : 'divider' },
            {  'Google'  :'http://www.google.com' },
            {  'Yahoo'  :'http://www.yahoo.com' },
            {  'Business'  : 'divider' },
            {  'CNBC'  :'http://www.cnbc.com' },
            {  'MarketWatch' :'http://www.MarketWatch.com' } ];

$('button[id="menu_divs_filts"]').menuoptions({
    'MenuOptionsType': 'Navigate', // this is a menu
    'Data': Menu_w_Dividers, // Data is array of objects
    // 2 mouseover filters using RegExps
    'Filters': [{ 'Biz' : '^(CNBC|MarketWatch)'}, {'Search' :'^(Google|Yahoo)'} ],
    'Sort': [], // don't sort, display in the original order
});
```

See this code in a demo

4.8 Using Filters in a menu or select list

4.8.1 How Filters work

Filters are specified using an array of objects. Each key show what the user will see, each value will be the actual filter that MenuOptions uses to filter out data.

```
{ 'Biz' : '^(CNBC|MarketWatch)'}
  ^^^^    ^^^^^^^^^^^^^^^^^^^^^
  user     MenuOptions searches
  sees     using this RegExp
  this
```

4.8.2 Using a plain text filter

If your Data is conforms to a pattern (e.g., times with AM PM), you can create a simple Filter like the one below

```
"Filters": [{'AM':'AM'},{'PM':'PM'}],
```

See simple filter demo

4.8.3 Using a regular expression (RegExp) filter

If your data is more varied, you can use RegExp Filters, like the one below:

```
'Filters': [{ 'Biz' : '^(CNBC|MarketWatch)'}, {'Search' :'^(Google|Yahoo)'} ],
```

See RegExp demo

4.9 FAQ

4.9.1 How do I reset the options in MenuOptions

see the instructions here

4.9.2 What do you mean auto-configuration?

Auto-configuration means that if you set the input field to either the key or the value, MenuOptions will automatically set the correct menu_opt_key and the correct value (what is shown to user). The command below will auto-configure all the MenuOptions widgets on a page:

```
$('input.ui-menuoptions').menuoptions();
```

For example:

Assume you are using month name and month code in your Data and the code 12 represents the month December. If you set the input field to "December", MenuOptions will automatically set menu_opt_key to the code 12. If you

set the input field to 12, MenuOptions will convert that and display December, while setting the menu_opt_key to the code 12.

4.9.3 What is the menu_opt_key?

menu_opt_key is an attribute of the input field that holds the code that corresponds to the visible text. So, if you input field shows "December" and the code for December is 12, the *menu_opt_key* would be set to 12.

```
<input type="text" name="month1" id="selecttest" menu_opt_key="12" class="ui-
↪menuoptions">
```

4.9.4 When I use jQuery.empty(), the widget does not get removed. How do I fix this?

The MenuOptions widget will detect the removal of the element it is applied to. However, calling jQuery.empty() does not appear to trigger the remove event *** so you will likely have to call the destroy () method, for example:

```
$(YourSelector + ' .ui-menuoptions').menuoptions('destroy');
```

4.9.5 I pasted data into a MenuOptions mulit-column autocomplete and now have errors when saving

MenuOptions expects that you have either clicked a selection or typed one in and pressed enter. When you paste data into a MenuOptions mulit-column autocomplete, just call MenuOptions again with no parameters

```
$(YourSelector + ' input.ui-menuoptions').menuoptions();
```

This will populate the attribute menu_opt_key that re_serialize() uses to get the value that corresponds with the text the user sees.

4.9.6 How would I reset (clear the data from) all MenuOptions mulit-column autocomplete and Rocker controls?

```
$(YourSelector + ' input.ui-menuoptions').val(''); // clear out the values
//  clear the menu_opt_key and clear Rocker to show no selection
$(YourSelector + ' input.ui-menuoptions').menuoptions();
```

4.9.7 The clear button (or 'X') is not aligned correctly

There are 2 situations where this can happen.

The first is when an input element is added dynamically (using javascript). The clear button is positioned using the jQuery UI position() function, which requires that the element be present in the DOM and visible.

The second is when the container that surrounds the input element is being resized, as when a browser draws a table and shrinks the <TD> that contains the input element.

There are 2 workarounds for this. The first is to call MenuOptions again (with no parameters) immediately after adding the element or after the layout change.

```
$(YourSelector + ' input.ui-menuoptions').menuoptions();
```

For dynamically added elements, you can wrap the menuoptions call with a setTimeout, like this:

```
setTimeout(function () {
    $('input#selecttest').menuoptions({
         "Data": { 1:"January",2:"February",3:"March",4:"April",5:"May", 6:"June",7:
↪"July",
                   8:"August",9:"September",10:"October",11:"November",12:"December" }
↪,
         "Sort": []
    });
}, 200 );
```

4.9.8 How do I display text and have a hidden value, like the HTML select control?

When creating your MenuOptions select control, pass it an object, like the code below:

```
PayMethod   = { 1: "American Express", 2: "Visa", 3: "Mastercard", 4:"Discover", 5:
↪"Check",
                6:"PayPal", 7:"Cash", 8:"Money Order"}

$('input[name="t"]').menuoptions({  "Data": PayMethod,
                                    "SelectOnly": true,
                                    "ClearBtn": true,
                                    "PlaceHolder": "Pay Method",
                                    "ColumnCount": 2,
                                    'Width': 225 });
```

For more details, see re_serialize()

4.9.9 When I hit enter in a MenuOptions select, it does not submit the form

That's correct. MenuOptions uses the Enter key to select the first dropdown element. If you want to submit the form when a user presses Enter, you can do so in the onSelect option, which returns the MenuOptions instance, newVal, newCode and type (EnterKey|Click|Rocker).

For more detals on onSelect see the docs

```
$('input#selecttest').menuoptions({
    "Data": { 1:"January",2:"February",3:"March",4:"April",5:"May", 6:"June",7:"July",
              8:"August",9:"September",10:"October",11:"November",12:"December" },
    "onSelect": function(mo, data) {
        if ( data.type == "EnterKey" ) {
            $("form#tst").submit();
        }
    },
    "Sort": [] // don't sort
});
```

This code is in quick start select demo

4.9.10 How can I create a vertical scroll bar for large lists?

Below is an example. Whenever you specify a Height that is less than the height of the mulit-column autocomplete dropdown, a vertical scroll bar will be created.

```
$('input#scrolltest').menuoptions({
    "Data": { 1:"January",2:"February",3:"March",4:"April",5:"May", 6:"June",7:"July",
              8:"August",9:"September",10:"October",11:"November",12:"December" },
    "onSelect": function(mo, data) {
        console.log(mo, data.newVal, data.newCode, data.type );
    },
    "InitialValue": { 'val': 'December'},
    "Height": 200,
    "Sort": []
});
```

This code is in quick start select demo

4.9.11 When I enter certain characters in a MenuOptions mulit-column autocomplete they disappear, why?

It only disappears when you enter a character that is not in any of the mulit-column autocomplete options

4.9.12 Can I use 'special' characters in a MenuOptions mulit-column autocomplete (parens, curly braces)?

Yes

4.9.13 Why do we need another input widget?

MenuOptions was created for one reason. To reduce - *to an absolute minimum* - the # of keystrokes and clicks required for data entry as well as navigation.

Features:

- **Input masking**
  - error messages that explain why the input key is invalid
  - hotkeys - a single key can fill a field (e.g., 't' fills in todays date in date fields)
- **Multi column autocomplete**
  - intelligent autocomplete (characters not in any mulit-column autocomplete item are automatically removed, saving keystrokes)
  - mouseover filtering lets user reduce choices by moving their mouse over a filter element
  - auto-configuration
- **Rocker control**
  - Binary options (true/false, yes/no, etc) that never hide a choice
- **Menus**

  - Built from JSON
  - mouseover filtering

Other benefits:

- offers the ability to combine multi column autocomplete and input mask functionality.
- uses color highlighting to show autocomplete matches
- the value associated with with the label string is saved in the input element automatically (in the menu_opt_key - no need to manually update a hidden field)
- it can utilize Data from a variety of of JSON types (array, array of arrays, single object, array of objects)

4.10 Change Log

4.10.1 1.7.1-3

Changed RockerControl from an option to a MenuOptionsType

The new format is demonstrated below:

```
$('input#on_off').menuoptions({"Sort": [],
     "Data": { 1: "On", 2:"Off" },
     "MenuOptionsType":"Rocker", // Rocker is now specified here
     "onSelect": function(mo, data) {  console.log(data); }
});
```

The old format is demonstrated below (will not work in versions > 1.7.1-2):

```
$('input#on_off').menuoptions({"Sort": [],
     "Data": { 1: "On", 2:"Off" },
     "RockerControl": True, // this won't work after 1.7.1-2
     "onSelect": function(mo, data) {  console.log(data); }
});
```

4.10.2 1.7.1-7

Path to static files has changed:

Table 4.3: New path for static files

Old path	New path
media/javascript	media/js
media/style	media/css
media/images	media/imgs

ShowDownArrow is no longer true or false

ShowDownArrow defaults to color black and allows that color to be overridden with any color you pass in. You can also pass in the "None" keyword, indicating that no arrow will be added to the menu header element.

The old format will now default to a black arrow being added to the menu header element.

```
$('button[id$="menutest"]').menuoptions({
    "Data": [ {"javascript": function() { alert('Some javascript was run'); } },
              {"Google": "http://www.google.com"},
              {"Yahoo": "http://www.yahoo.com"}],
    "MenuOptionsType": "Navigate",
    });
```

The new format (below), where arrow color is specified

```
$('button[id$="menutest"]').menuoptions({
    "Data": [ {"javascript": function() { alert('Some javascript was run'); } },
              {"Google": "http://www.google.com"},
              {"Yahoo": "http://www.yahoo.com"}],
    "MenuOptionsType": "Navigate",
    "ShowDownArrow": "silver" // color of arrow is now silver, not black
    });
```

4.10.3 1.7.3-15

deprecated refreshData

Instead, call MenuOptions the same way you would when initializing (code examples here)

4.10.4 1.7.4-7

deprecated add_menuoption_key

deprecated set_select_value

Added DataKeyNames

DataKeyNames allows you to utilize Data that has extra, unneeded data, only picking out the key and value fields that you specify.

Added data structure tests for menus

4.10.5 1.8.0

refactor source into several js file

add input masking

enable input masking and autocomplete together

CHAPTER 5

Indices and tables

- genindex
- modindex
- search