

## Intoduction

ClicColorPicker is a picker that can be used to select a color, or a gradient (with optional opacity on either).

Using the ClicColorPicker requires jquery and the jquery-ui draggable plugin.

## Creation

The syntax for creating a control will be familiar to anyone who has used jquery.

```
$('#picker').ClicColorPicker();
```

Similar to most jquery controls, you can pass options in.

```
$('#red').ClicColorPicker({'startColor':'rgb(255,0,0)'});
```

Here's a full list of the values that can be passed in, and their default values.

```
{
  startColor:'#FFFFFF', // can be hex or rgb(..), or rgba(...) if opacity is enabled
  type:'simple', // can be full, simple, or gradient, more on these modes later...
  enableOpacity:false, //determines whether opacity selection is enabled
  defaultPalette:'websafe', // which palette is selected by default, only applies to simple mode
  openerCssClass:null, // css class for the element that opens the panel
  mainPanelCssClass:null, // css for the panel which lets you choose your color
  onChanged:null // function which is executed when the user selects a color
}
```

So a more typical example of the way it would be used is like this:

```
$('#picker').ClicColorPicker({
  'type':'gradient',
  'enableOpacity':true,
  'onChanged': function (e) {
    alert("the color was changed");
  }
});
```

## Layout

Creating a ClicColorPicker will get you a control that looks like this:



And when the user click on it to open the picker, the will get the following:

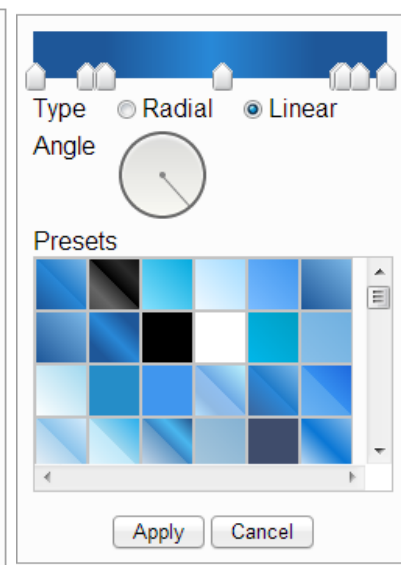
Simple picker



Full picker



Gradient Picker



When you click on one of the gradient stops in the gradient picker, it uses the full picker to let you choose your color for that stop.

In addition, note that the color picker text boxes are editable, and will update the selected color when enter is pressed, or it loses focus.

## Code

To facilitate reuse, various bits of the picker are made from various components.

- **ClicColorPicker** is the “parent” control. It draws the box users click to open a panel and draw a panel which it populates differently depending on settings.
- **ClicSimplePicker** is a panel which will let the user select from pre-defined palletes or enter their own value into the ColorTextBox.
- **ColorTextBox** is the box which shows the current color and lets users enter in a new color.
- **ClicFullColorPicker** draws the area where people can select any color. Uses the nouslider and ColorTextBox.
- **ClicGradientPicker** draws the ClicGradientSlider, and a ClicPresetPicker

- **ClicGradientSlider** is the control users click to add gradient stops, or change the color of existing ones.
- **ClicPresetPicker** draws an area full of pre-defined gradients to quickly create a gradient.

All of these controls have a similar structure. There's an init where the settings are defined and an application "state" is created. There's some drawing of html and setting up internal events which alter the application state. And finally there is updating the ui based on application state. Communication between the controls happens the same way most jquery objects talk to each other, through event handling functions and methods.

In addition to the "controls", there are some code files. I used the "nested object namespacing" namespace pattern found [here](#).

- "Helper" files, **ColorMethods** which contains methods to parse hex string into colors, write css, etc, and **ClicUiLib** which contains methods for standard jquery control creation and rendering of html
- "Data": **GradientPresets** which contains json data that populates the preset picker.

In addition to code I wrote, there are third party dependencies.

- **jquery**
- **jquery-ui** (only requires draggable)
- [nouislider](#)
- [toggles](#)

Since making the browser request multiple files is slow, **it's strongly suggested that the code gets minified** before going into production.

## Roadmap

This is stuff I'd like to see in no particular order...

- Works on **mobile devices** (doesn't have to be awesome, but right now it's broken)
- **Multi-lingual** (there's some logic in place in most of the child controls but it's not all wired up to ClicColorPicker yet)
- **Design**, I'm not an artist I'm sure the buttons and stuff can be nicer. I should match your fonts?
- **Double click**, to select and "apply" in one step.