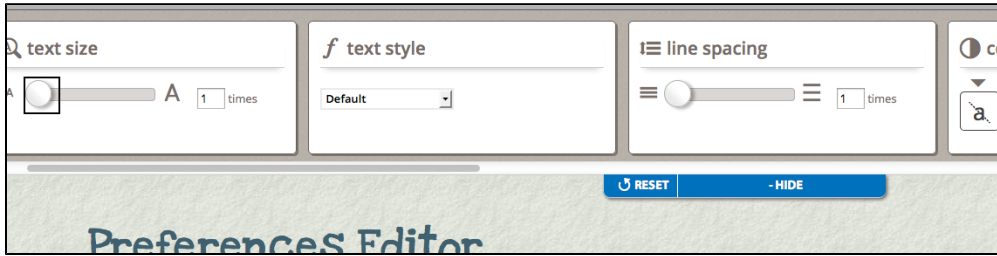


Tutorial - User Interface Options



The **User Interface Options component (UI Options)** allows users to transform the presentation of the user interface and content resources so that they are personalized to the individual user's needs.

UI Options does three things:

- places a preferences editor dialog with a set of six panels in a collapsible panel at the top of the page, accessible through a button in the upper right corner of the page;
- instantiates a cookie-based Settings Store for storing the user's preferences; and
- acts upon the user's preferences.

This tutorial assumes that:

- you are already familiar with HTML, Javascript and CSS,
- you are familiar with what the UI Options preferences editor is and does, and
- now you just want to know how to add it to your website.

See Also

[UI Options showcase demo](#)
[User Interface Options API](#)
[Working With A Preferences Editor On Your Site](#)
[Preferences Framework](#)

Scenario

You're putting together a website that you know will have a diverse audience. You'd like to allow your visitors to customize the presentation of the site to their individual needs, by enlarging the text or increasing the visual contrast, for example. This tutorial will show you how to add the Infusion [UI Options](#) component to your site.

These are the basic steps to add UI Options to your website:

- [Setup: Download and install the Infusion library](#)
- [Step 1: Prepare your page](#)
 - [1a: The Sliding Panel](#)
 - [1b: The Table of Contents](#)
- [Step 2: Add dependencies to the page](#)
- [Step 3: Add the UI Options component](#)

The rest of this tutorial will explain each of these steps in detail.

Setup: Download and install the Infusion library

1. Get the current source code from github as a ZIP file: <https://github.com/fluid-project/infusion/archive/master.zip>
2. Unpack the zip file you just downloaded and cd into the "infusion-master" folder that results.
3. If necessary install Node.js (<http://nodejs.org/download/>) and Grunt ("npm install -g grunt-cli").
4. Make your own custom build by running the "grunt" command in the Terminal. See the [README.md file](#) for instructions on how to make a custom build of Infusion.
5. The grunt command will create a zip file in the products folder. Unzip that file and move the resulting `infusion` folder somewhere convenient for your development purposes, likely in a `lib` folder in your site hierarchy.
6. This `infusion` will include a single file containing all of the JavaScript you need: `infusion-custom.js`. You will link to this file in the headers of your HTML files.

Step 1: Prepare your page

1a: The Sliding Panel

The UI Options component includes HTML templates for all the controls, so you don't need to create any HTML for them. You only need to add a small amount of markup to the top of your page to tell UI Options where to render itself.

Add the following markup as the first thing within your `<body>` tag:

```
<div class="flc-prefsEditor-separatedPanel fl-prefsEditor-separatedPanel">
  <!-- This is the div that will contain the Preference Editor component -->
  <div class="flc-slidingPanel-panel flc-prefsEditor-iframe"></div>
  <!-- This div is for the sliding panel that shows and hides the Preference Editor controls -->
  <div class="fl-panelBar">
    <span class="fl-prefsEditor-buttons">
      <button id="reset" class="flc-prefsEditor-reset fl-prefsEditor-reset"><span class="fl-icon-undo"><
/      span> Reset</button>
      <button id="show-hide" class="flc-slidingPanel-toggleButton fl-prefsEditor-showHide"> Show/Hide<
/      button>
    </span>
  </div>
</div>
```

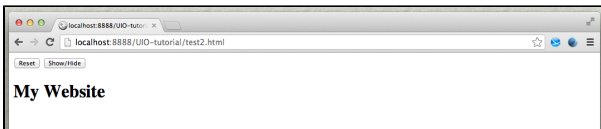
The main `<div>` in this snippet contains two things:

1. a `<div>` where an `iframe` will be inserted, containing the Fat Panel UI Options controls (line 3), and
2. a `<div>` where the sliding panel and button will be created (lines 5-10).

The elements in this snippet all have particular class names attached to them, and it's important to keep them:

- the class names starting with `flc-` are used to identify the elements to UI Options;
- the class names starting with `fl-` are used for visual styling.

If you open this page in your browser now, you'll only see the button in the upper left corner, since we haven't set up the CSS, and UI Options isn't present on the page yet:



Note that it doesn't matter what text you put in the button: The UI Options component will add a label and update it to reflect whether or not the panel is currently open. You can configure the text that the component uses by setting its configuration parameters. (After you've finished the tutorial, check out the instructional demo [Fat Panel UI Options - Custom Show-Hide Button](#) for an example of configuring the button text.)

1b: The Table of Contents

One of the UI Options controls allows users to add a Table Of Contents to the top of the page. You need to add a placeholder `<div>` to your page for the Table Of Contents. It should have a class of `"flc-toc-tocContainer"` like this:

```
<div class="flc-toc-tocContainer"> </div>
```

Where exactly on your page you put this `<div>` is up to you, but it will depend on the exact layout of your page. It should be pretty close to the top, so that it's easily visible and accessible quickly for keyboard-only users. You can, of course, add additional classes of your own to style the Table Of Contents to fit in with the look of your site.

Step 2: Add dependencies to the page

UI Options depends upon [Fluid Skinning System \(FSS\)](#) and the Infusion Framework, so you will need to add to your pages dependencies for

- the FSS CSS files,
- the UI Options own CSS files, and
- the main Infusion JavaScript file, `infusion_custom.js`.

In the header of your file, link to the FSS CSS files with `<link>` tags (you may have to adjust the paths to reflect where you've saved the Infusion package).

```

<!-- Required CSS files -->
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/fss/css/fss-layout.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/fss/css/fss-text.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-theme-bw-
prefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-theme-wb-
prefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-theme-by-
prefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-theme-yb-
prefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-theme-lgdg-
prefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-theme-dglg-
prefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/fss/fss-text-prefsEditor.
css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/lib/jquery/ui/css/fl-theme-by/by.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/lib/jquery/ui/css/fl-theme-yb/yb.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/lib/jquery/ui/css/fl-theme-bw/bw.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/lib/jquery/ui/css/fl-theme-wb/wb.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/lib/jquery/ui/css/fl-theme-lgdg/lgdg.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/lib/jquery/ui/css/fl-theme-dglg/dglg.css" />

```

- Note: The theme "dark-grey on light-grey" is planned for a future release. Therefore `css/fl-theme-dglg/dglg.css` is currently optional.

We'll also need the UI Options CSS files:

```

<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/PrefsEditor.css" />
<link rel="stylesheet" type="text/css" href="lib/infusion/framework/preferences/css/SeparatedPanelPrefsEditor.
css" />

```

We'll use the `<script>` tag to link to the Infusion library:

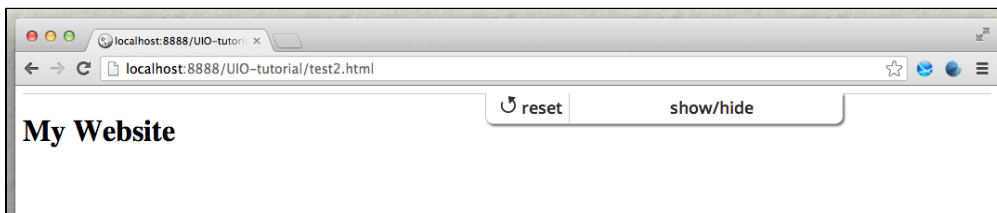
```

<!-- The Infusion Library -->
<script type="text/javascript" src="lib/infusion/infusion-custom.js"></script>

```

Note that the `infusion-custom.js` file is a concatenation of all of the required JavaScript files and will be minified (i.e. all of the whitespace removed) if you've built the minified version. If so, it might be difficult to debug with. If you want to be able to debug the code, you might want to choose the "source" version when you create your Infusion bundle.

If you open this page in your browser now, you'll only see that the button has been styled differently: it is in the upper right corner and the font has been changed. You can also see the bar of the sliding panel. The button still doesn't do anything, since we still haven't added the UI Options component to the page.



Step 3: Add the UI Options component

The simplest way to add the UI Options component to your page is using a `<script>` tag near the top of the page. We suggest placing it right before the UI Options markup created in [Step 1](#).

Add the script block as shown below:

```

<body>
  <script type="text/javascript">
    $(document).ready(function () {
      fluid.uiOptions.prefsEditor(".flc-prefsEditor-separatedPanel", {
        tocTemplate: "lib/infusion/components/tableOfContents/html/TableOfContents.html",
        templatePrefix: "lib/infusion/framework/preferences/html/",
        messagePrefix: "lib/infusion/framework/preferences/messages/"
      });
    })
  </script>

  <div class="flc-prefsEditor-separatedPanel fl-prefsEditor-separatedPanel">
    ...
  </div>

  <!-- the rest of your page here -->
</body>

```

This script calls the `fluid.uiOptions.prefsEditor()` function to create the component. The function takes two arguments:

1. the selector of the container for the component, and
2. an options object for configuring the component.

The selector for our UI Options will be the classname of the `<div>` we created in [Step 1](#). In this markup, the selector is `".flc-prefsEditor-separatedPanel"`, on line 11 in the code shown above.

The options tell the component about two things:

- where to find the UI Options HTML templates included in Infusion: the `templatePrefix` option, and
- where to find the message bundles, the strings that will be used in the interface: the `messagePrefix` option.

In the code above, the `messagePrefix` option is referencing the default strings provided by the component. If you need to translate the interface into a different language, see [Tutorial - Localizing UI Options](#).

Congratulations!

UI Options is now fully functional on your page. Now, when you load your page in your browser and click on the "Show Display Preferences" button, you will see the UI Options controls, as shown in the image below. If you adjust the controls, you will see your changes being applied to the page.

