Preferences Editor

On This Page

See Also

- Primary Schema for Preferences Framework
- Auxiliary Schema for Preferences Framework
- Builder
- Tutorial - Creating a Preferences Editor Using the Preferences Framework

Overview

One of the primary functions of the Infusion Preferences Framework is to allow you to create a Preferences Editor: a collection of adjusters that users can use to set their interface preferences.

The Preferences Framework provides a utility that creates and instantiates a preferences editor in a single step, given primary and auxiliary schemas.

```
var prefsEditor = fluid.prefs.create(container[, options]);
```

Parameters

`container`	(required) (String) A CSS-style selector that will contain the preferences editor markup.
`options`	(optional) (Object) Configuration options. See Options below for more information.

Return Value

Object	(Object) The preferences editor instance.

Options

Name	Description	Values	Default
`build`	(Optional) Configuration options for the builder; see Builder Options below for more information.	Object	{}
`prefsEditor`	(Optional) Configuration options for the preferences editor itself. See PrefsEditor Options below for more information.	Object	{}

Builder Options

Name	Description	Values	Default

`gradeNames`	(Optional) A list of grade names to be used for the builder. This option can be used to specify the names of grades that define schemas, as an alternative to specifying the schemas through the direct options. If you do not provide the `auxiliarySchema` option, you must include the grade name of a grade that includes an auxiliary schema.	Array of strings	none
`primarySchema`	(Optional) A JavaScript object providing primary schema details. See Processing the Schemas below for more details.	Object	`{}`
`auxiliarySchema`	(Optional) A JavaScript object providing auxiliary schema details. See Processing the Schemas below for more details. If you do not specify the grade name of a grade that includes an auxiliary schema, you must include this option.	Object	`{}`

NOTE: You *must* provide at least one of

- the `auxiliarySchema` option, or
- a `gradeName` indicating an auxiliary schema.

If you provide both, they will be merged (with the `auxiliarySchema` overriding anything in the grade schema), but you must provide at least one.

PrefsEditor Options

Name	Description	Values	Default

`prefsEditorType`	(Optional) The string name of a grade of preference editor.	The Preferences Framework provides three built-in types of editor:
		
		`"fluid.prefs.separatedPanelPrefsEditor"`
		`"fluid.prefs.fullNoPreview"`
		`"fluid.prefs.fullPreview"`
		
		Integrators can use one of these grades, or define their own grade, using one of these grades as a base grade.
			`"fluid.prefs.separatedPanelPrefsEditor"`

`storeType`	(Optional) The string name of a grade of a Settings Store.	Integrators can define their own store grade by using the built-in default grade `"fluid.globalSettingsStore"` as a base grade.	`"fluid.globalSettingsStore"`
`enhancerType`	(Optional) The string name of a grade of a UI Enhancer.	Integrators can define their own enhancer grade by using the built-in default grade `"fluid.pageEnhancer"` as a base grade.	`"fluid.pageEnhancer"`
`templatePrefix`	(Optional) A string value representing the relative path to the directory containing the templates. This value will overwrite the `templatePrefix` value supplied by auxiliary schemas.		
`messagePrefix`	(Optional) A string value representing the relative path to the directory containing the message files. This value will overwrite the `messagePrefix` value supplied by auxiliary schemas.		
`prefsEditor`	(Optional) The data structure that configures the `prefsEditor` component. See Preferences Editor for what is accepted in the data structure.		
`enhancer`	(Optional) The data structure that configures the `uiEnhancer` component. See UI Enhancer for what is accepted in the data structure.		
`store`	(Optional) The data structure that configures the `store` component. See Settings Store for what is accepted in the data structure.		
`listeners`	(Optional) A data structure defining listener functions for supported events. See Infusion Event System for more information about registering event listeners.	The Preferences Framework supports one event: `onReady`: Fires after the preferences editor is rendered and ready to use.	

Usage

The simplest way to create a separated panel preferences editor is to provide the primary and auxiliary schema using the options:

```
var prefsEditor = fluid.prefs.create("#myPrefsEditor", {
    build: {
        primarySchema: {...},
        auxiliarySchema: {...}
    }
});
```

The preferences editor will be instantiated and rendered into the container specified as the first argument to `fluid.prefs.create()`.

Examples

```
 fluid.prefs.create("#myPrefsEditor", {
    build: {
        gradeNames: ["fluid.prefs.auxSchema.starter"],
        auxiliarySchema: {
            "template": "prefsEditorPreview.html",
            "tableOfContents": {
                "enactor": {
                    "tocTemplate": "../../../components/tableOfContents/html/TableOfContents.html"
                }
            }
        }
    },
    prefsEditor: {
        prefsEditorType: "fluid.prefs.fullPreview"
    }
})
```