

# Undo API

## Overview

The Undo subcomponent provides undo support for any component that bears a model. For more information about subcomponents, see [Subcomponents](#).

**Subcomponent Name:** `fluid.undoDecorator`

## Model

To support Undo, a component must bear a *model*, a collection of Javascript objects which constitute the data which it is operating on. A model:

- consists of pure data, i.e. Javascript objects containing only other objects and primitives, without any functions;
- is public, i.e. accessible as a member of the component's top-level `that`.

The actual contents of a model is, of course, entirely dependent on the component.

To support Undo, a component's model must:

- be called `model`

### On This Page

- [Overview](#)
  - [Model](#)
  - [Implementing that.updateModel](#)
  - [Support](#)
- [Construction](#)
  - [Parameters](#)
    - [component](#)
    - [options](#)
- [Options](#)
- [Dependencies](#)

### See Also

- [Subcomponents](#)
- [Inline Edit API](#)
- [Fluid Component API](#)

### Still need help?

Join the [infusion-users mailing list](#) and ask your questions there.

## Implementing that.updateModel

To support the Undo subcomponent, the component using it must implement a public function on its `that` object called `updateModel`:

```
fluid.myComponent = function(container, options) {
  var that = fluid.initView("fluid.myComponent", container, options);
  ...
  that.updateModel = function (newValue, source) {
  };
};
```

The `updateModel()` function must update the locally stored model using the `newValue` provided. The `source` parameter is the subcomponent that triggered the model change.

## Support

Currently, the Fluid components that support Undo are:

- [Inline Edit](#)

## Construction

```
fluid.undoDecorator(component, options);
```

## Parameters

### component

The `component` parameter is the parent component object, i.e. the `that` object returned by the parent component's creator function.

### options

The `options` parameter is an optional collection of name-value pairs that configure the Pager and its subcomponents, as described below in the [#Options](#) section.

## Options

Name	Description	Values	Default
<code>selectors</code>	Javascript object containing selectors for various fragments of the Undo decorator	The object can contain any subset of the following keys: undoContainer undoControl redoContainer redoControl Any values not provided will revert to the default.	<p>In v1.1:</p> <pre>selectors: {   undoContainer: ".flc-undo-undoContainer",   undoControl: ".flc-undo-undoControl",   redoContainer: ".flc-undo-redoContainer",   redoControl: ".flc-undo-redoControl" }</pre> <p>In v1.2:</p> <pre>selectors: {   undoContainer: ".flc-undo-undoControl",   undoControl: ".flc-undo-undoControl",   redoContainer: ".flc-undo-redoControl",   redoControl: ".flc-undo-redoControl" }</pre>

renderer	A function that renders the markup for the undo controls	function	<p><b>In v1.1:</b> A function that generates the following:</p> <pre>&lt;span class='flc-undo' aria-live='polite' aria-relevant='all'&gt;   &lt;span class='flc-undo-undoContainer'&gt;[&lt;a href='#' class='flc-undo-undoControl'&gt;undo&lt;/a&gt;]&lt;/span&gt;   &lt;span class='flc-undo-redoContainer'&gt;[&lt;a href='#' class='flc-undo-redoControl'&gt;redo&lt;/a&gt;]&lt;/span&gt; &lt;/span&gt;</pre> <p><b>In v1.2:</b> A function that generates the following:</p> <pre>&lt;span class='flc-undo' role='region' aria-live='polite' aria-relevant='all'&gt;   &lt;a href='#' class='flc-undo-undoControl'&gt;...string specified in strings option...&lt;/a&gt;   &lt;a href='#' class='flc-undo-redoControl'&gt;...string specified in strings option...&lt;/a&gt; &lt;/span&gt;</pre>
<b>New in v1.2:</b> strings	Javascript object containing named strings for use in the interface.	The object can contain any subset of the following keys: undo redo Any values not provided will revert to the default.	<pre>strings: {   undo: "undo edit",   redo: "redo edit" }</pre>

## Dependencies

The Undo functionality's dependencies can be met by including the minified `InfusionAll.js` file in the header of the HTML file:

```
<script type="text/javascript" src="InfusionAll.js"></script>
```

Alternatively, if you are including individual files, you must include `Undo.js`:

```
<... other dependencies ...>
<script type="text/javascript" src="components/undo/js/Undo.js"></script>
```