

Table of Contents API

Production Status: SNEAK PEEK

The Table of Contents component examines a page for HTML heading elements, and generates and formats a list of links to headings that can be used to navigate the page. It is used by the [User Interface Options](#) component.

See Also

[Table of Contents Model Builder](#)

[Table of Contents Levels](#)

[User Interface Options API](#)

Creator	<code>fluid.tableOfContents(container, options)</code>
Supported Events	<code>onReady</code> Fires when the component has finished instantiating and ready to use. <code>onRefresh</code> Fires when the table of contents needs to be refreshed. This firing of this event also triggers the refresh. <code>afterRender</code> Fires after the table of contents has been rendered.
Methods	<code>headingTextToAnchorInfo(headings, GUID)</code> Returns the information required for generating the anchor for one of the headings. <code>generateGUID()</code> Generates a unique ID. This ID is used in the process of creating anchors for the headings. <code>locateHeadings()</code> Returns a jQuery object containing all of the headings to be included in the Table of Contents. <code>hide()</code> Hides the table of contents. <code>insertAnchor(name, element, anchorClass)</code> Inserts anchors before each heading. <code>show()</code> Shows the table of contents.
Options	listeners See #Supported Events for information. selectors See below.
Selectors	<code>headings</code> Identifies the headings in the page <code>tocContainer</code> Container for the generated table of contents <code>tocAnchors</code> The selector to identify the generated table of contents anchors.
Subcomponents	<code>levels</code> Renders the generated table of contents. <code>modelBuilder</code> Builds an internal model of the structure of the headings in the document.

Creator

[back to top](#) Use the following function to create a Table of Contents component:

fluid.tableOfContents(container, options)

Method	<code>fluid.tableOfContents(container, options);</code>
---------------	---

Description	Instantiates a Table of Contents component applied to the specified container.
Parameters	<p><i>container</i> A CSS-based selectors, single-element jQuery object, or DOM element that identifies the root DOM node where the Table of Contents component should search for headings.</p> <p><i>options</i> An optional data structure that configures the Table of Contents component, as described below.</p>
Returns	The Table of Contents component
Examples	<pre>var toc = fluid.tableOfContents("#main");</pre>
Notes	The Table of Contents only processes headings found inside the element provided by the <code>container</code> argument. This allows the component to be applied to only a section of a page, if desired. Keep in mind that the generated list of links will be placed inside the <code>container</code> , as specified by the <code>tocContainer</code> selector.

Supported Events

[back to top](#) Listeners can be attached to any supported events through a component's `listeners` option. Values can be a function reference (not a string function name) or an anonymous function definition, as illustrated below:

```
var myComponent = component.name("#myContainerID", {
  listeners: {
    eventName1: functionName,
    eventName2: function (params) {
      ...
    }
  }
});
```

For information on the different types of events, see [Infusion Event System](#).

onReady

Description	This event fires when the DOM is fully instantiated and ready to use.
Type	default
Parameters	<i>The instantiated table of contents component</i>
Availability	Infusion 1.5 and later

onRefresh

Description	This event fires when the Table of Contents Levels need to be refreshed. The firing triggers the refresh of the Table of Contents Levels subcomponent.
Type	default
Parameters	<i>none</i>
Availability	Infusion 1.5 and later

afterRender

Description	This event fires Fires when the table of contents is rendered.
Type	default
Parameters	<i>The Table of Contents Levels subcomponent</i>
Availability	Infusion 1.5 and later

Methods

[back to top](#) This component has public methods that can be invoked by integrators as necessary. These methods are typically implemented as [Invokers](#), which resolve their arguments from the environment at invocation time. These methods can be configured by the integrator if necessary: arguments can be changed, and in fact the entire implementation function can be replaced by a custom function (though it is likely rare that this would be necessary or desirable).

Configuration of invokers should be carried by creating a grade, using the following pattern:

```
fluid.defaults("<new name of overriding grade>", {
  gradeNames: ["<name of grade being overridden>", "autoInit"],
  invokers: {
    <invokerName>: {
      funcName: <implementation function name>,
      args: [<array of argument specifications>]
    }
  }
});
```

Example of overriding an invoker

The Infusion [Pager](#) component has a method called `acquireDefaultRange` which, by default, simply returns the total range. An integrator may wish to override this using their own custom function:

```
fluid.defaults("my.pager", {
  gradeNames: ["fluid.pager", "autoInit"],
  invokers: {
    acquireDefaultRange: {
      funcName: "my.pager.acquireDefaultRange"
    }
  }
});
```

[headingTextToAnchorInfo\(headings, GUID\)](#)

Description	Returns the information required for generating the anchor for the heading.
Parameters	<i>heading</i> The heading, usually one of headings generated by <code>that.locate("headings")</code> <i>GUID</i> The unique ID used to identify the generated anchor.
Configurable	<i>yes</i> Default implementation: "fluid.tableOfContents.headingTextToAnchorInfo"
Availability	Infusion v1.5
See Also	generateGUID()

[generateGUID\(\)](#)

Description	Used by the component in the creation of unique anchors for the headings. The default implementation uses fluid.allocateSimpleId .
Parameters	<i>none</i>
Configurable	<i>yes</i> Default implementation: "fluid.allocateSimpleId"
Availability	Infusion v1.4
See also	headingTextToAnchorInfo(headings, GUID)

[locateHeadings\(\)](#)

Description	Used by the component to locate the headings to be used to generate the Table of Contents from. The default implementation finds all of the headings specified by the <code>headings</code> selector, minus the headings matching selectors specified in the <code>ignoreForToC</code> option.
--------------------	--

Parameters	<i>none</i>
Configurable	<i>yes</i> Default implementation: "fluid.tableOfContents.locateHeadings"
Availability	Infusion v1.9

insertAnchor(name, element, anchorClass)

Description	Used by the component to create and insert anchors before each element linked to by the table of contents. The default implementation creates an HTML <a> element using the <code>name</code> parameter as both the name and id of the element and applies <i>anchorClass</i> to the element.
Parameters	<i>name</i> The name used in the creation of the anchor. <i>element</i> The element to insert the anchor before. <i>anchorClass</i> The css class to be applied to the anchor.
Configurable	<i>yes</i> Default implementation: "fluid.tableOfContents.insertAnchor"
Availability	Infusion v1.4 (note that the component signature changed in 1.5)

hide()

Description	Hides the generated table of contents.
Parameters	<i>none</i>
Configurable	<i>no</i>
Availability	Infusion v1.0
See also	show()

show()

Description	Shows the generated table of contents.
Parameters	<i>none</i>
Configurable	<i>no</i>
Availability	Infusion v1.0
See also	hide()

Options

[back to top](#) The second argument to the creator function is the options argument. This is a JavaScript object containing name/value pairs: The name is the name of the option and the value is the desired setting. Components define their own default values for options, but integrators can override these defaults by providing new values using the options argument. For technical information about how options are merged with defaults, see [Options Merging](#).

selectors

Description	See below for details.
--------------------	------------------------

ignoreForToc

Description	A key/value pairing where the keys are selectors representing headings to NOT include in the Table of Contents.
--------------------	---

Example	<pre>"ignoreForToC": { "overviewPanel": ".flc-overviewPanel" }</pre>
----------------	--

Selectors

[back to top](#) One of the options that can be provided to the component is a set of CSS-based selectors identifying where in the DOM different elements can be found. The value for the option is itself a Javascript object containing name/value pairs:

```
selectors: {
  selector1Name: "selector 1 string",
  selector2Name: "selector 2 string",
  ...
}
```

The component defines defaults for these selectors: If you use those defaults in your markup, you do not need to specify the selectors option. If you do choose to override any of the selectors, you can specify your custom selector using this option.

The selectors supported by Table of Contents are described below.

headings

Description	This selector will be used to identify all of the headings in the page. These headings will be used in the table of contents (after filtering).
Default	":header:visible"
Example	<pre>fluid.tableOfContents(".myContainer", { selectors: { headings: "h1,h2,h3" // limit to only three levels deep } });</pre>

tocContainer

Description	This selector identifies the container into which the generated table of contents will be inserted. This will typically be an empty <div> element.
Default	".flc-toc-tocContainer"
Example	<pre>fluid.tableOfContents(".myContainer", { selectors: { tocContainer: "#tableOfContentsDiv" } });</pre>

tocAnchors

Description	This selector identifies the generated table of contents anchors. These anchors are removed at every refresh before new anchors are added.
Default	".flc-toc-anchors"

Example

```
fluid.tableOfContents(".myContainer", {
  selectors: {
    tocAnchors: "#tableOfContentsAnchors"
  }
});
```

Subcomponents

Subcomponents can be configured through the parent component's `components` option using the following pattern:

```
parent.component.name(".myContainer", {
  components: {
    <subcomponentName>: {
      options: {
        <subcomponent options>
      }
    }
  }
});
```

levels

Description	The <code>levels</code> subcomponent renders the generated table of contents based on a template.
Default implementation	fluid.tableOfContents.levels
See also	Table of Contents Levels

modelBuilder

Description	The <code>modelBuilder</code> subcomponent builds an internal model of the structure of the headings in the document. This model is used by the Table of Contents component to generate a hierarchical table of contents.
Default implementation	fluid.tableOfContents.modelBuilder
See also	Table of Contents Model Builder