

UI Tags

Unlike [generic tags](#), UI tags do not provide much control structure or logic. Rather, they are focussed on using data, either from your action/value stack or from the Data Tags, and displaying data in rich and reusable HTML. All UI tags are driven by *templates* and *themes*. While generic tags simply output some content directly from the tag (if there is any content to output), the UI tags defer to a template, often grouped together as a theme, to do the actual rendering.


Template support allows UI tags to build a rich set of reusable HTML components that can be customized to fit exact requirements. For details, see [Themes and Templates](#).

Themes and Templates	A must-read explanation of how themes and templates are used when rendering UI tags.
Form Tags	provide all form-related UI output, such as <i>form</i> , <i>textfield</i> , and <i>select</i> .
Non Form Tags	provide all non-form-related UI output, such as <i>a</i> , <i>div</i> , and <i>tabbedPanel</i> .

Language Specific Tag Support

The framework strives to support multiple development environments. The framework does not impose a single template language. Almost any common language can be used, and there are hooks for new languages. By default, almost every single tag is supported in JSP, Velocity, and FreeMarker. In each of these sections, you'll find examples and techniques for applying the generic tag reference toward your specific language or template choice.

- [JSP Tags](#)
- [Velocity Tags](#)
- [FreeMarker Tags](#)

 Please make sure you have read the [Tag Syntax](#) document and understand how tag attribute syntax works.

Within the form tags, there are two classes of tags: the form tag itself, and all other tags, which make up the individual form elements. The behavior of the form tag is different than the elements enclosed within it.

Next: [Themes and Templates](#)