

Integrating Plugin UI with Clearspace

You can integrate your plugins into the Clearspace user interface by overriding and adding to the UI components that Clearspace defines by default. For instance, you can add a new tab to a space's main page, add a new "Action" link to user profiles, and add a new menu to the user bar (that navigation bar at the top of Clearspace pages).

To add your UI to Clearspace, you add `<component>` elements to your plugin's `plugin.xml` file. Your `<component>` elements tells Clearspace which part of its UI you're "hooking into" and what should happen when the user interacts with your UI. All of the elements you can use are described in the schema at <http://www.jivesoftware.com/schemas/clearspace/1.4/ui-components.xsd>. Not all UI `<component>` elements are applicable in every component. Structure your `plugin.xml` component as described in the examples below.

The XML shape defined by the schema allows for complex URLs so that you can take advantage of the WebWork and FreeMarker components that Clearspace is made up of. For more information, see [Using FreeMarker Markup in Configuration](#).

Note: Clearspace uses the `ui-components.xml` file to define its own UI components. If you want to see this file you'll find it inside `clearspace.jar`, which is itself inside the `clearspace-<version_number>.war` file in the `/WEB-INF/lib` directory.

Integration Points

Through your `<component>` elements, you can integrate with Clearspace by adding:

- an [Admin Console](#) tab and sub-tabs.
- menus to the [user bar](#) (the navigation bar at the top of Clearspace pages).
- [user profile tabs](#).
- [community/space page tabs](#).
- [community page Action box links](#).
- [user profile Action box links](#).
- [document page Action box links](#).
- [discussion page Action box links](#).
- [blog page Action box links](#).

You tell Clearspace which of these you're integrating with through `id` attributes in your `<component>` element stanza. Here's an example:

The component id attribute identifies the part of Clearspace UI you're integrating with. This value must be one of those defined in the schema in order to integrate with existing Clearspace user interface.

Label text.

Tooltip text.

Your id value uniquely identifies your addition.

```
<component id="community-tabs">
  <tab id="community-tab" name="Example Plugin Tab"
    description="Example of a new community tab."
    cssClass="link-hello-world">
    <url><![CDATA[<@ww.url action="example"/>]]></url>
  </tab>
</component>
```

CSS class to style the label text. This could be defined in your CSS file or in Clearspace's.

Destination URL to use when the user clicks the item. Here, the URL points to a WebWork action defined in the plugin's xwork-plugin.xml file.

Admin Console

The admin console is made up of first- and second-level tabs. Each second-level tab displays links along one side (by default, the left side) that navigate to new administration pages. This UI is visible on admin console pages.

Adding a `<component>` element:

- Required `<component>` id attribute value: `admin-console`
- `<tab>` element to add a new first-level tab.
- `<section>` element to add a new second-level tab.
- `<item>` element to add a new link on the second level tab.
- Each of the strings in the XML must be represented by an `i18n` key you've included in the `plugin_i18n.properties` file.
- The FreeMarker template from which the page is rendered must include a reference to a `pageTitle` property. Here's an example: `<content tag="pageTitle">${pageTitle}</content>`

Example

Adds a new tab to the admin console with one section and one item.

```
<!-- Adds an Example tab to the admin console. -->
<component id="admin-console">
  <tab id="tab-helloworld" name="plugin.example.admin.example" url="hello.jspa"
    description="plugin.example.tab.tooltip">
    <section id="section-helloworld" name="plugin.example.admin.hello"
      description="plugin.example.section.tooltip">
      <item id="page-helloworld" name="plugin.example.admin.helloworld" url="hello.jspa"
        description="plugin.example.item.tooltip"/>
    </section>
  </tab>
</component>
```



User Bar

The user bar is the navigation bar at the top of each Clearspace page. By default, it includes menus labeled "New," "Your Stuff," "Browse," and so on. You can add new menu tabs and commands, as well as items under existing menu tabs. This UI is visible on all pages.

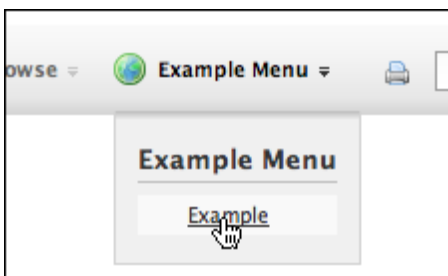
Adding a `<component>` element:

- Required `<component>` id attribute value: user-bar
- `<tab>` element to add a new menu tab.
 - The id attribute should be your unique value for a new menu. Use a value of the form `jiveUserMenu<position>`. The position is the menu's position in the horizontal order of menus.
 - To add to a Clearspace menu, use the existing menu's value.
- `<item>` elements to add a new link items to the menu. The id attribute is a unique value for the UI you're adding.

Example

Adds a new "Example Menu" menu tab that contains an Example link.

```
<components>
  <component id="user-bar">
    <tab id="jiveUserMenu6" accesskey="M" name="Example Menu"
      cssClass="jive-userbar-example">
      <item id="myexample" name="Example" url="example"/>
    </tab>
  </component>
</components>
```



User Profile Page Tabs

User profiles have tabs with pages that collect that user's content and other information about the user.

You can add a new tab to profiles with this component. This UI is visible on user profiles to all users who can see profiles.

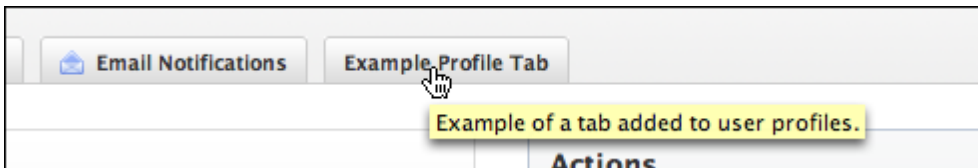
Adding a <component> element:

- Required <component> id attribute value: user-profile
- <tab> element to add a new first-level tab. The url element or attribute must point to a valid WebWork action.

Example

Adds a new tab to the user profile called "Example Profile Tab". The WebWork action example.jspa is imported into the page.

```
<components>
  <component id="user-profile">
    <tab id="tab-community-calendar"
      name="Example Profile Tab" description="Example of a tab added to user profiles.">
      <url><![CDATA[<@ww.url action="example" />]]></url>
    </component>
  </components>
```



If the url in the components section points to a webwork action, the following things must happen for your webwork page to function correctly.

- The webwork action
 - Must extend `com.jivesoftware.community.action.ViewProfile`
 - Must call `super.execute()` in the `execute()` method

```
public class ExampleProfileAction extends ViewProfile {
  public String execute() {
    String result = super.execute();
    // your code below
  }
}
```

- The freemarker page should
 - define a variable `view` with the value of the `tab.id`.
 - include `/template/global/include/view-profile-header.ftl`
 - include the css style sheet `/styles/jive-profile.css`
 - add the breadcrumb using the `sitemesh content` tag.

```
<html>
<head>
```

```

<title>My Profile Page</title>
<style type="text/css" media="screen">
  @import "<@ww.url value='/styles/jive-profile.css' includeParams='none' />";
</style>
<content tag="breadcrumb">
  <@ww.action name="community-breadcrumb" executeResult="true"
ignoreContextParams="true" />
  <a href="<@ww.url value="/people" includeParams="none" />"><@ww.text
name='settings.people.brdrmb.link' /></a>
</content>
</head>
<body>
  <#assign view = 'profile-example-tab' />
  <#include '/template/global/include/view-profile-header.ftl' />
  <h1>Hello World</h1>
</body>
</html>

```

Community Page Tabs

Each community and sub-community has a "home" page with tabs that collect content in the community. You can add a tabs to these pages.

Adding a <component> element:

- Required <component> id attribute value: community-tabs
- <tab> element to add a new tab. tionSupport.
- The url must point to a valid url.

Example

```

<!-- Adds a New Tab tab to a community page. -->
<components>
  <component id="community-tabs">
    <tab id="community-tab" name="Example Community Tab"
      description="Example of a new community tab." cssClass="jive-link-community">
      <url><![<@ww.url action='example' />]></url>
    </tab>
  </component>
</components>

```



If the url in the components section points to a webwork action, the following things must happen for your webwork page to work properly.

- The webwork action
 - Must extend com.jivesoftware.community.action.CommunityActionSupport
 - Must call super.execute() in the execute() method

```

public class ExampleCommunityAction extends CommunityAction {
    public String execute() {
        String result = super.execute();
        // your code below
    }
}

```

- The freemarker page should
 - define a variable view with the value of the tab id.
 - include /template/global/include/community-header.ftl
 - include the css style sheet /styles/jive-community.css
 - add the breadcrumb using the sitemesh content tag.

```

<html>
  <head>
    <#assign view = 'community-example-tab' />
    <title>My Page Title</title>
    <style type="text/css" media="screen">
      @import "<@ww.url value='/styles/jive-community.css' includeParams='none' />";
    </style>
    <content tag="breadcrumb">
      <@ww.action name="community-breadcrumb" executeResult="true"
ignoreContextParams="true">
        <@ww.param name="communityID" value="{community.ID?c}" />
        <@ww.param name="view" value="{view}" />
      </@ww.action>
    </content>
  </head>
  <body>
    <#include '/template/global/include/community-header.ftl' />
    <h1>Hello World</h1>
  </body>
</html>

```

Community Page Actions

You can add action links to the Actions box on community pages.

Adding a <component> element:

- Required <component> id attribute value: community-actions
- <tab> element to specify that the link goes in the Actions box.
- <item> element to give details about the link itself. The url element or attribute is the link's destination, such as a WebWork action defined in your plugin's xwork-plugin.xml file.

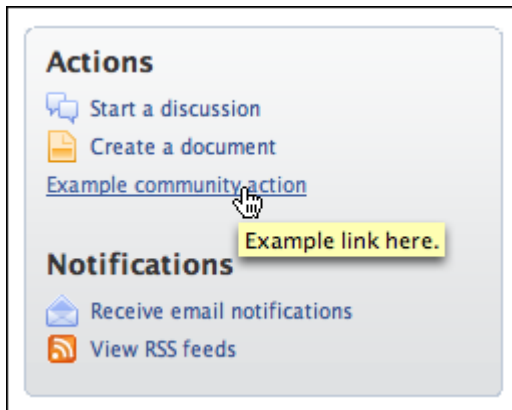
Example

```

<components>
  <component id="community-actions">
    <tab id="community-actions-tab">
      <item id="community-actions-link" name="Example community action"
description="Example link here.">
        <url><![CDATA[<@ww.url action="example" />]]></url>
      </item>
    </tab>
  </component>

```

```
</components>
```



Document Page Actions

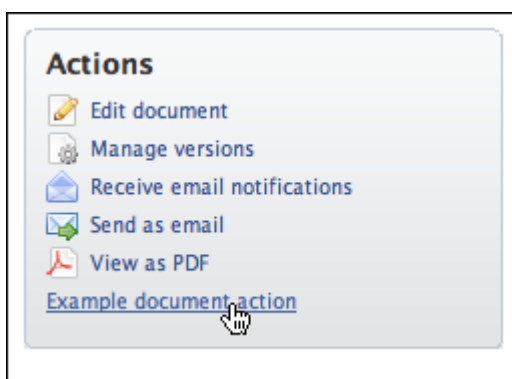
You can add action links to the Actions box on document pages.

Adding a `<component>` element:

- Required `<component>` id attribute value: `document-actions`
- `<tab>` element to specify that the link goes in the Actions box.
- `<item>` element to give details about the link itself. The `url` element or attribute is the link's destination, such as a WebWork action defined in your plugin's `xwork-plugin.xml` file.

Example

```
<components>
  <component id="document-actions">
    <tab id="document-actions-tab">
      <item id="document-actions-link" cssClass="jive-link-edit"
        name="Example document action">
        <url><![CDATA[<@ww.url action="example" />]]></url>
      </item>
    </tab>
  </component>
</components>
```



Thread Page Actions

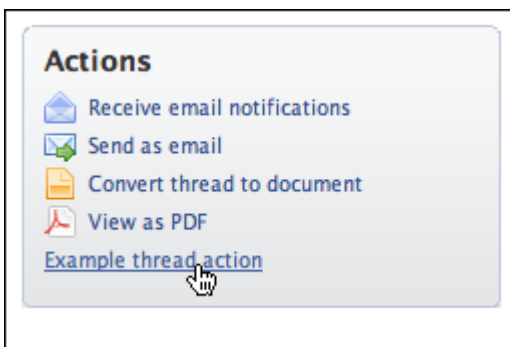
You can add action links to the Actions box on discussion thread pages.

Adding a `<component>` element:

- Required `<component>` id attribute value: `thread-actions`
- `<tab>` element to specify that the link goes in the Actions box.
- `<item>` element to give details about the link itself. The `url` element or attribute is the link's destination, such as a WebWork action defined in your plugin's `xwork-plugin.xml` file.

Example

```
<components>
  <component id="thread-actions">
    <tab id="thread-actions-tab">
      <item id="thread-actions-link" name="Example thread action"
        cssClass="link-hello-world">
        <url><![CDATA[<@ww.url action="example" />]]></url>
      </item>
    </tab>
  </component>
</components>
```



User Profile Page Actions

User profiles have two faces: one for the person whose profile it is, and one for other users. You can add a new Action box link to either (or both, with two `<component>` stanzas), depending on the value you specify for the `<component>` element's id attribute.

Adding a `<component>` element:

- Required `<component>` id attribute value: `profile-actions-self` for links that are visible only to the user who owns the profile; `profile-actions` for other visiting users.
- `<tab>` element to specify that the link goes in the Actions box.
- `<item>` element to give details about the link itself. The `url` element or attribute is the link's destination, such as a WebWork action defined in your plugin's `xwork-plugin.xml` file.

Example

```

<components>
  <component id="profile-actions-self">
    <tab id="profile-actions">
      <item id="profile-actions-link-self" name="Example profile action (self)"
        description="This is visible only to the user whose profile this
is."><url><![CDATA[<@ww.url
        action="example" />]]></url>
      </item>
    </tab>
  </component>
</components>

```



Blog Page Actions

You can add links to the Actions box on blog post pages.

Adding a `<component>` element:

- Required `<component>` id attribute value: `blog-actions`.
- `<tab>` element to specify that the link goes in the Actions box.
- `<item>` element to give details about the link itself. The `url` element or attribute is the link's destination, such as a WebWork action defined in your plugin's `xwork-plugin.xml` file.

Example

```

<components>
  <component id="blog-actions">
    <tab id="blog-actions-tab">
      <item id="blog-actions-link" name="Example blog action">
        <url><![CDATA[<@ww.url action="example" />]]></url>
      </item>
    </tab>
  </component>
</components>

```



Using FreeMarker Markup in Configuration

The shape for your `<component>` elements (as defined by the schema) provides flexibility so that you can include complex URLs as the destination for user clicks. You can give the URL as a value of either a `url` attribute or a `<url>` element. A `<url>` element is especially useful for complex URLs, such as a URL to WebWork action that will include parameters used by FreeMarker (which might include characters that are invalid in an attribute and outside a CDATA).

For example, you can specify a URL with a `url` attribute:

```
<item id="myitem" name="My Item" url="myitem.jspa" />
```

You can also specify it with a `<url>` child element:

```
<item id="myitem" name="My Item">
  <url>myitem.jspa</url>
</item>
```

The `<url>` element above would also work as simply `<url>myitem</url>`, where `myitem` is assumed to be a WebWork action.

When the URL is rendered it has full access to everything in the FreeMarker context on the given page. So if the page has a `Community` object in the context under the key `community` you could do the following. Here, the URL passes to the `myitem` action a `communityID` property value set to the `community`'s ID:

```
<item id="myitem" name="My Item">
  <url><![CDATA[myitem.jspa?communityID=${community.ID?c}]</url>
</item>
```