

Configuration File Differences



EditLive!'s [Advanced API](#) and [plugin](#) functionality is only supported with an EditLive! [Enterprise Edition](#) license.

Instances of EditLive! using either the Advanced APIs or the default editor both need to reference an EditLive! Configuration File. For a complete list of all the configuration file elements available to EditLive!, please see the [Configuration File Elements](#) article.

Using the Advanced APIs, developers can customize in Java the actions to occur on clicking custom toolbar and menu items. The following documentation extends on the EditLive! configuration file elements depicted in the [Configuration File Elements](#) article:

- `<customMenuItem>`
- `<customToolBarButton>`
- `<customComboBoxItem>`

`<customMenuItem>` Configuration File Element

This element specifies the properties for a developer-defined custom menu item for use within EditLive!.

Configuration Element Tree Structure

```
<editLive>
<menuBar>
<menu>
<customMenuItem.../>
```

```
<editLive>
...
<menuBar>
...
<menu>
  <customMenuItem... />
</menu>
...
</menuBar>
...
</editLive>
```

Required Attributes

name

The name which uniquely defines the custom menu item.

text

The text to place on the menu for this item.

action

The action which this menu item performs when clicked on. This attribute has the following possible values:

- *insertHTMLAtCursor* - Insert the given HTML at the cursor
- *insertHyperlinkAtCursor* - Insert the given hyperlink at the cursor
- *raiseEvent* - Call a JavaScript function with the name specified in the value attribute. An event is also fired to the [ELJBean](#) using this configuration file. The following values will be assigned to the Java TextEvent sent with the event:
 - *Action Command* - `TextEvent.CUSTOM_ACTION`
 - *Extra String* - The string specified in the **value** attribute
 - *Extra Object* - A Java Map containing the attributes of the element selected when the custom menu item was selected.

The event sent to the bean will also have the value `TextEvent.CustomAction.RAISE_EVENT` added to the extra *int* property of the event.

When the event has been handled (by using the event method `setHandled(boolean b)`, where `b = true`), the javascript function defined in value will be called.

- *customPropertiesDialog* - Fires an event for the [ELJBean](#) using this configuration file. The following values will be assigned to the TextEvent sent with the event:
 - *Action Command* - `TextEvent.SHOW_CUSTOM_PROPERTIES`
 - *Extra String* - The string specified in the **value** attribute
 - *Extra Object* - A Java Map containing the attributes of the element selected when the custom menu item was selected.

value

The value of this attribute depends on the value specified in the action attribute:

- *insertHTMLAtCursor* - value will be a string of HTML
- *InsertHyperlinkAtCursor* - value will be a URL
- *raiseEvent* - value will be the name of a JavaScript function
- *customPropertiesDialog* - value will be the Extra String field of the TextEvent sent with the [ELJBean](#) event.



When using the *insertHTMLAtCursor* **action** the HTML to be inserted must be URL encoded in the configuration file. For example, `<p>HTML to insert<p>` becomes `%3Cp%3EHTML%20to%20insert%3C/p%3E`.

Optional Attributes

imageURL

The URL of the image to be placed on the menu with the menu item text. The image should be of a .gif format and be a size of sixteen (16) pixels high and sixteen (16) pixels wide.



This URL can be relative or absolute. Relative URLs are relative to the location of the page in which EditLive! is embedded.

enableintag

This attribute defines in which tags the function should be enabled. For example, when set to *td* the function will be enabled when the cursor is within a `<td>` tag (i.e. a table cell).

Examples

The following example demonstrates how to define a custom menu item for use within EditLive!. The menu item defined in this example will insert HTML at the cursor; note that the value in the example below is URL encoded.

```
<editLive>
...
<menuBar>
...
<menu name="Example">
  <customMenuItem
    name="customItem1"
    text="Custom Item"
    imageURL="http://www.someserver.com/image16x16.gif"
    action="insertHTMLAtCursor"
    value="%3Cp%3EHTML%20to%20insert%3C/p%3E" />
  </menu>
...
</menuBar>
...
</editLive>
```

The following example demonstrates how to define a custom properties dialog which is launched from a custom menu item for use within EditLive!. The custom properties dialog will be available for use when the cursor is inside any `<td>` tag.

```
<editLive>
...
<menuBar>
...
<menu name="Example">
  <customMenuItem
    name="customProperties1"
    text="Custom td Properties"
    action="customPropertiesDialog"
    value="customTDFunction"
    enableintag="td" />
  </menu>
...
</menuBar>
...
</editLive>
```

Remarks

The `<customMenuItem>` element can appear multiple times within the `<menu>` element.

The `<customMenuItem>` element must be a complete tag; it cannot contain a tag body. Therefore the tag must be closed in the same line. See the example below:

```
<customMenuItem name=... />
```

Text assigned to the value attribute must be [URL encoded](#) as it is in the example above.

`<customToolBarButton>` Configuration File Element

This element will cause a particular button to be present on the toolbar within Tiny EditLive!

Configuration Element Tree Structure

```
<editLive>
<toolbars>
<toolbar>
<customToolBarButton.../>
```

```
<editLive>
...
<toolbars>
...
<toolbar>
  <customToolBarButton ... />
</toolbar>
...
</toolbars>
...
</editLive>
```

Required Attributes

name

The name which uniquely defines the custom toolbar button.

text

The tooltip text for this custom toolbar button.

action

The action which this toolbar button performs when clicked on.

Note: This attribute has the following possible values:

- *insertHTMLAtCursor* - Insert the given HTML at the cursor
- *insertHyperlinkAtCursor* - Insert the given hyperlink at the cursor
- *raiseEvent* - Call a JavaScript function with the name specified in the value attribute. An event is also fired to the [ELJBean](#) using this configuration file. The following values will be assigned to the Java TextEvent sent with the event:
 - *Action Command* - `TextEvent.CUSTOM_ACTION`
 - *Extra String* - The string specified in the **value** attribute
 - *Extra Object* - A Java Map containing the attributes of the element selected when the custom toolbar button was clicked.

The event sent to the bean will also have the value `TextEvent.CustomAction.RAISE_EVENT` added to the extra *int* property of the event.

When the event has been handled (by using the event method `setHandled(boolean b)`, where `b = true`), the javascript function defined in value will be called.

- *customPropertiesDialog* - Fires an event for the [ELJBean](#) using this configuration file. The following values will be assigned to the TextEvent sent with the event:
 - *Action Command* - `TextEvent.SHOW_CUSTOM_PROPERTIES`
 - *Extra String* - The string specified in the **value** attribute
 - *Extra Object* - A Java Map containing the attributes of the element selected when the custom toolbar button was clicked.

value

The value of this attribute depends on the value specified in the **action** attribute:

- *insertHTMLAtCursor* - value will be a string of HTML
- *InsertHyperlinkAtCursor* - value will be a URL

- *raiseEvent* - value will be the name of a JavaScript function
- *customPropertiesDialog* - value will be the Extra String field of the TextEvent sent with the ELJBean event.



When using the insertHTMLAtCursor action the HTML to be inserted must be URL encoded in the configuration file. For example, <p>HTML to insert<p> becomes %3Cp%3EHTML%20to%20insert%3C/p%3E.

imageURL

The URL of the image to be placed on the toolbar button. The image should be of a .gif format and be a size of sixteen (16) pixels high and sixteen (16) pixels wide.



This URL can be relative or absolute. Relative URLs are relative to the location of the page in which EditLive! is embedded.

Optional Attributes

enableintag

This attribute defines in which tags the function should be enabled. For example, when set to *td* the function will be enabled when the cursor is within a <td> tag (i.e. a table cell).

Examples

The following example demonstrates how to define a custom toolbar button for use within EditLive!. The button defined in this example will insert HTML to insert at the cursor; note that the value in the example below is URL encoded.

```
<editLive>
...
<toolbars>
...
  <toolbar name="Example">
    <customToolbarButton
      name="customItem1"
      text="Custom Item"
      imageURL="http://www.someserver.com/image16x16.gif"
      action="insertHTMLAtCursor"
      value="%3Cp%3EHTML%20to%20insert%3C/p%3E" />
    </toolbar>
  ...
</toolbars>
...
</editLive>
```

The following example demonstrates how to define a custom properties dialog which is launched from a custom toolbar button for use within EditLive!. The custom properties dialog will be available for use when the cursor is inside any <td> tag.

```
<editLive>
...
<toolbars>
...
  <toolbar name="Example">
    <customToolbarButton
      name="customProperties1"
      text="Custom td Properties"
      imageURL="http://www.someserver.com/image20x20.gif"
      action="customPropertiesDialog"
      value="customTDFunction"
      enableintag="td" />
    </toolbar>
  ...
</toolbars>
...
</editLive>
```

Remarks

The <customToolbarButton> element can appear multiple times within the <toolbar> element.

The `<customToolBarButton>` element must be a complete tag; it cannot contain a tag body. Therefore the tag must be closed in the same line. See the example below:

```
<customToolBarButton name=... />
```

Text assigned to the value attribute must be [URL encoded](#) as it is in the example above.

<customComboBoxItem> Configuration File Element

This element specifies the properties for a developer defined custom combo box item for use within Tiny EditLive!. The custom combo box item must be listed within a `<customToolBarComboBox>` element and will therefore appear on one of the toolbars within EditLive!.

Configuration Element Tree Structure

```
<editLive>
<toolbars>
<toolbar>
<customToolBarComboBox>
<customComboBoxItem .../>
```

```
<editLive>
...
<toolbars>
...
<toolbar>
  <customToolBarComboBox>
    <customComboBoxItem .../>
  </customToolBarComboBox>
</toolbar>
...
</toolbars>
...
</editLive>
```

Required Attributes

name

The name which uniquely defines this custom combo box item within the `<customToolBarComboBox>` element. This means that there cannot be two `<customComboBoxItem>` elements with the same name within one `<customToolBarComboBox>` element.

text

The text to represent this item within the combo box it is to be listed in.

action

The action which this menu item performs when clicked on.

Note: This attribute has the following possible values:

- *insertHTMLAtCursor* - Insert the given HTML at the cursor
- *insertHyperlinkAtCursor* - Insert the given hyperlink at the cursor
- *raiseEvent* - Call a JavaScript function with the name specified in the value attribute. An event is also fired to the [ELJBean](#) using this configuration file. The following values will be assigned to the Java TextEvent sent with the event:
 - *Action Command* - `TextEvent.CUSTOM_ACTION`
 - *Extra String* - The string specified in the **value** attribute
 - *Extra Object* - A Java Map containing the attributes of the element selected when the custom combo box item was clicked.

The event sent to the bean will also have the value `TextEvent.CustomAction.RAISE_EVENT` added to the extra int property of the event.

When the event has been handled (by using the event method `setHandled(boolean b)`, where `b = true`), the javascript function defined in value will be called.

- *customPropertiesDialog* - Fires an event for the [ELJBean](#) using this configuration file. The following values will be assigned to the TextEvent sent with the event:
 - *Action Command* - `TextEvent.SHOW_CUSTOM_PROPERTIES`
 - *Extra String* - The string specified in the **value** attribute
 - *Extra Object* - A Java Map containing the attributes of the element selected when the custom combo box item was clicked.

value

The value of this attribute depends on the value specified in the **action** attribute:

- *insertHTMLAtCursor* - value will be a string of HTML
- *InsertHyperlinkAtCursor* - value will be a URL
- *raiseEvent* - value will be the name of a JavaScript function
- *customPropertiesDialog* - value will be the Extra String field of the TextEvent sent with the ELJBean event.

Note: When using the *insertHTMLAtCursor* **action** the HTML to be inserted must be URL encoded in the configuration file. For example, `<p>HTML to insert<p>` becomes `%3Cp%3EHTML%20to%20insert%3C/p%3E`.

Examples

The following example demonstrates how to define a custom combo box item for use within a custom combo box which exists on the EditLive! Command Toolbar. The combo box item defined in this example will insert HTML to insert at the cursor; note that the value in the example below is URL encoded.

```
<editLive>
...
<toolbars>
...
  <toolbar name="command">
    <customToolbarComboBox name="customCombo">
      <customComboBoxItem
        name="customItem1"
        text="Custom Item"
        action="insertHTMLAtCursor"
        value="%3Cp%3EHTML%20to%20insert%3C/p%3E" />
      <customToolbarComboBox>
    </toolbar>
  </toolbars>
...
</editLive>
```

Remarks

The `<customComboBoxItem>` element can appear multiple times within the `<customToolbarComboBox>` element.

The `<customComboBoxItem>` element must be a complete tag, it cannot contain a tag body. Therefore the tag must be closed in the same line. See the example below:

```
<customComboBoxItem name=... />
```

Text assigned to the value attribute must be [URL encoded](#) as it is in the example above.