

# Fields

## The <field> Element

Before we dive into the details of defining the various UI elements I would like to present XML elements and general concepts that apply throughout. This saves me a lot of work in writing and you a lot of repetitive reading and maybe a tree or two.

The UI elements are generally laid out top to bottom in the order they appear in the XML file. The only exception to this rule is the title, which always appears at the very top. The layout pattern for the input fields is as follows: If a description is defined, it appears first, using the full available layout width. The input field is placed beneath the description. With fields such as the text field or the combo box, the label is placed to the left and the input field to the right. Fields such as radio buttons and check boxes are somewhat indented and have the label text appear to their right.

Each UI element is specified with a <field> tag. The <field> tag has some common attributes:

Name	Required	Description	Value
type	yes	Used to specify what kind of field you want to place.	Any valid field type
label_position	no	Used to change the default position of the label.	west, westonly, both, eastonly, east <b>default: west</b>
label_align	no	Used to change the default alignment of the label.	left, center, right <b>default: left</b>
label_indent	no	Controls whether or not the label is indented.	true, false <b>default: false</b>
control_position	no	Used to change the default position of the field's control.	west, westonly, both, eastonly, east <b>default: east</b>
control_align	no	Used to change the default alignment of the control.	left, center, right <b>default: left</b>
control_indent	no	Controls whether or not the control is indented.	true, false <b>default: false</b>
variable	yes, if input field	The variable that should be substituted with the user input.	A variable name.
conditionid	no	The ID of a condition in install.xml which must evaluate to true in order to display the field.	A valid condition id in install.xml.
tooltip	no	The ID of a string to be displayed as the tooltip for the GUI components in this field.	The id of a string from langpacks.
omitFromAuto	no	Whether to omit the field variable's value from being written to the auto-install.xml record initiated from the FinishPanel (if used).	true, false <b>default: false</b>

readonly	no	<p>The <code>readonly</code> attribute reverses the <code>displayHidden</code> logic described below - <code>readonly="true"</code> means to display a field read-only in general in case it is not disabled by a general field condition.</p> <div data-bbox="602 239 1321 590" style="border: 1px solid #ccc; padding: 10px;"> <div style="background-color: #f9f9f9; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> <span><b>Example of the <code>readonly</code> field attribute</b></span> <span><a href="#">Expand</a> <a href="#">source</a></span> </div> <pre style="margin-top: 5px;"> &lt;panel id="panel.dbsettings"&gt;   &lt;field ... readonly="true"&gt;   &lt;/field&gt;   ... &lt;/panel&gt;</pre> </div>	true, false <b>default:</b> false
readonlyCondition	no	<p>The <code>readonlyCondition</code> attribute enhances and replaces the <code>readonly</code> condition and makes the related behavior dependent on another condition. Thus, the related field gets displayed read-only only if it is not disabled by a general field condition and the condition defined by <code>readonlyCondition</code> gets true.</p> <div data-bbox="602 758 1321 1192" style="border: 1px solid #ccc; padding: 10px;"> <div style="background-color: #f9f9f9; padding: 5px; display: flex; justify-content: space-between; align-items: center;"> <span><b>Example of the <code>readonlyCondition</code> field attribute</b></span> <span><a href="#">Expand</a> <a href="#">source</a></span> </div> <pre style="margin-top: 5px;"> &lt;panel id="panel.dbsettings" &gt;   &lt;field ...   readonlyCondition="isReadOnlyField"&gt;   &lt;/field&gt;   ... &lt;/panel&gt;</pre> </div>	true, false <b>default:</b> false

displayHidden	no	<p>Enables to "grey out" the field not matching its specific condition. Makes just sense if a conditionid has been defined.</p> <p>Setting displayHidden attribute in the that specific field within the panel be shown regardless of what the field's conditionid evaluates to. But if the conditionid evaluates to false, display the field as disabled instead of hiding it.</p> <div data-bbox="602 333 1321 905" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <div data-bbox="618 359 1013 485"> <p><b>Example of the displayHidden field attribute</b></p> </div> <div data-bbox="1089 380 1219 474"> <p><a href="#">Expand source</a></p> </div> </div> <pre data-bbox="618 491 1224 856"> &lt;panel id="a.panel"&gt;   &lt;field type="check" variable="b" conditionid="b.cond" displayHidden="true"&gt;     &lt;spec id="b.id" true="true" false="false"/&gt;   &lt;/field&gt;   &lt;field type="check" variable="c" conditionid="c.cond"&gt;     &lt;spec id="c.id" true="true" false="false"/&gt;   &lt;/field&gt; &lt;/panel&gt; </pre> </div>	true, false <b>default:</b> false
displayHiddenCondition	no	<p>The displayHiddenCondition attribute enhances and is an alternative to the displayHidden condition and makes the related behavior dependent on another condition. Thus, the related field gets displayed read-only only if a general condition on that field disables it for displaying and the condition defined by displayHiddenCondition gets true.</p> <div data-bbox="602 1098 1321 1753" style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <div style="display: flex; justify-content: space-between; align-items: center;"> <div data-bbox="618 1123 1013 1283"> <p><b>Example of the displayHiddenCondition field attribute</b></p> </div> <div data-bbox="1089 1144 1219 1239"> <p><a href="#">Expand source</a></p> </div> </div> <pre data-bbox="618 1289 1224 1703"> &lt;panel id="b.panel"&gt;   &lt;field type="check" variable="b" conditionid="b.cond" displayHiddenCondition="isDisplayHiddenField"&gt;     &lt;spec id="b.id" true="true" false="false"/&gt;   &lt;/field&gt;   &lt;field type="check" variable="c" conditionid="c.cond"&gt;     &lt;spec id="c.id" true="true" false="false"/&gt;   &lt;/field&gt; &lt;/panel&gt; </pre> </div>	true, false <b>default:</b> false

**Examples**

In the following example, the the label and the control will be shown on it's own row.

```
<field type="text" variable="value1">
  <spec txt="The label" id="" size="20" set="default value overriding current values
of 'value1'" label_both="both" control_position="both" />
</field>
```

In the following example, the label is right aligned to the control.

```
<field type="text" variable="value1">
  <spec txt="The label" id="" size="20" set="default value overriding current values
of 'value1'" label_align="right" />
</field>
```

## Nested elements of <field>

### The <description> Element

Almost all fields allow a description element (exception: directory and file chooser). The description is part of the data within the field element. There can only be one description per field. If you add more than one description, the first description is used and the others ignored. There are three attributes used with this tag. The text is specified through the `txt` or the `id` attribute. The details on using them are described below. The attributes are all optional but you must specify text to use, either directly or through the `id` attribute. In addition, you can set the text justification to `left`, `center` and `right` with the `align` attribute.

Name	Required	Description	Value
txt	yes, if no id	Specified the default text to use in absence of a language package.	Any text
id	yes, if no txt	Specifies text description through use of a language package.	A valid id
align	no	Sets the justification of the description.	left, center, right <b>default: left</b>

The following example illustrates the general pattern for field specification:

```
<field type="text" variable="myFirstVariable">
  <description align="left" txt="A description" id="description1"/>
  .
  .
  .
</field>
```

A very frequently used pattern is for the definition of text. Where ever text is needed (labels, descriptions, static text, choices etc.) it can be specified in place using the `txt` attribute. This is convenient if you are only supporting a single language. However, if you would like to separate your text definitions from the panel specification or if you need to support multiple languages you might want to use the `id` attribute instead to only specify an identifier. You can then add multiple XML files with the same name as this spec file (`userInputSpec.xml`) appended with an underscore '\_' and the the appropriate three letter ISO3 language code. The content of those files must conform to the specification for IzPack language packages. For more details on this topic see the chapter on language packages under advanced features. `id` defines an identifier that is also defined in the language package, together with the localized text to use. It is possible to use both the `txt` and the `id` attribute. In this case the text from the language package is used. If for some reason the language package is not available or the `id` is not defined there, the text specified with `txt` is used as default.

All input fields can be pre-set with a value of your choice. Although the details vary a bit from field type to field type, the `set` attribute is always used to accomplish this. The `set` attribute is of course optional. Please note that if you use the `set` attribute, you would have to keep in mind, that the `UserInputPanel` will be rendered after each user input. Thus it will be set back to the default value if you don't use a condition to handle that. IzPack generates builtin conditions for every variable used as target for a field to be able to check if there's any input.

### The <spec> Element

All fields that take user input use a `<spec>` tag to define the details of the input field. In the some cases the content of this tag is rather simple. Input fields with a more complex nature tend to have accordingly complex content in this tag. Since the details vary widely, they are explained with

each input field.

Any number of `<createForPack name=a pack name />` tags can be added to the `<panel>` and `<field>` sections. This tag has only one attribute and no data. The attribute is `name` and specifies the name of one of the installation packs that you have defined. Here is how it works: if no `<createForPack ...>` tag exists in a section, the entity is always created. However, if the tag exists, the entity is only created if one or more of the listed packs are selected for installation. As mentioned before, if you are using this feature, make sure the user input panel shows up after the packs panel.

Also, any number of `<createForUnselectedPack name=a pack name />` tags can be added to the `<panel>` and `<field>` sections. This tag has only one attribute and no data. It works exactly like `createForPack` except that once added user input panel will appear for only NOT Selected packs. As mentioned earlier, you need to make sure that the user input panel shows up after the packs panel for this feature to work.

There is a possibility to use variables in those elements where the text is supplied via `txt` attribute. This includes static fields and input fields (spec, description). The text can contain unlimited number of variables that will be substituted. Variable substitution also works with language packs, just use variables in your language pack, and they will be still substituted properly.

Attributes of the `<spec>` element:

Name	Required	Description	Value
id	yes, if no txt	unique identifier, used for finding field label translations	string
txt	yes, if no id	default label text, overridden by existing translations	string
set	no	initial value represented in the user input field according to its layout overriding previous values of the assigned variable (regardless whether the variable is set to a different value)	string
default	no	default value represented in the user input field according to its layout being used if there is no previous value of the assigned variable (if the variable is unset)	string
size	no	field size in characters for "text" fields	integer

## Example

In the following example, the variables: `name1`, `name2`, `name3` will be substituted.

```
<field type="text" variable="value1">
  <description align="left" txt="Configuration for $name1 and $name2" id="" />
  <spec txt="The value for $name3:" id="" size="20" set="default value" />
</field>
```

## Nested Constraints for Visibility

Name	Required	Description	Values
<code>createForPack</code>	no	If this tag is present the appropriate user input panel gets activated just if the mentioned pack is selected by the user in the <a href="#">PacksPanel</a> .	Valid pack names, see <a href="#">PacksPanel</a> .
<code>createForUnselectedPack</code>	no	If this tag is present the appropriate user input panel gets activated just if the mentioned pack is selected by the user in the <a href="#">PacksPanel</a> .	Valid pack names, see <a href="#">PacksPanel</a> .
<code>os</code>	no	Operating system-specific constraints.	Valid OS constraints, see <a href="#">OS Restrictions</a> .