

# Composite Field

[CompositeFieldDefinition](#) renders a complex field that display a composition of fields based on standard field definition.

Class: `info.magnolia.ui.form.field.definition.CompositeFieldDefinition`

Here is an example coming from the sample dialog show room module.

Configured Composite

Node name	Value
configuredComposite	
fields	(individual field definition)
name	
class	<code>info.magnolia.ui.form.field.definition.TextFieldDefinition</code>
label	<code>dialog.devShow.allControls.tabLayout.simpleSwitch.fields.text.label</code>
type	String
level	
link	
class	<code>info.magnolia.ui.form.field.definition.CompositeFieldDefinition</code>
label	<code>dialog.devShow.allControls.tabLayout.configuredComposite.label</code>

## Storing strategy

By default the `transformerClass` is set to `CompositeTransformer`.

Assume that your `CompositeField` is called '`composite`' and contains two fields: a text field called '`simpleText`' and a date field called '`simpleDate`'. The values will be stored as following:

Node name	Value
<code>compositesimpleText</code>	some text value
<code>compositesimpleDate</code>	2006-05-01T21:47:58.230+02:00

## Configuration

You can use all [common field properties](#) and the following:

Property	Description	Default value
<code>fields</code>	Parent node containing the single field that composed the composite field.	
<code>field1</code>	Parent node of one field composing the composite field.	
<code>class</code>	Class definition of the field1	
...	Continue with field1 definition	

...	Add any number of field definition that should composed the composite field.	
class	CompositeFieldDefinition class name. Mandatory.	
transformerClass	Custom Transformer definition. Optional.	
layout	Define the field layout. Layout of the fields could be vertical or horizontal. Optional.	horizontal