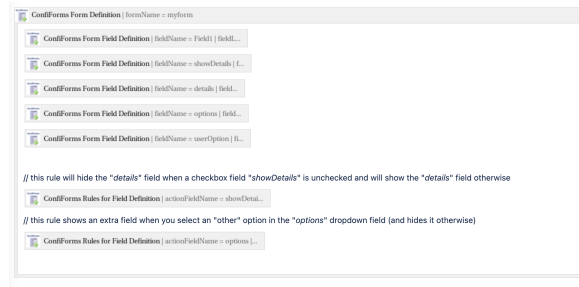


How to show and hide fields in the form conditionally

Simple tutorial on how to use [ConfiForms Field Definition Rules#Showfield](#) and it's counterpart [ConfiForms Field Definition Rules#Hidefield](#) ConfiForms Rules for Field Definitions to show and hide fields conditionally

Consider a form that has few fields like this



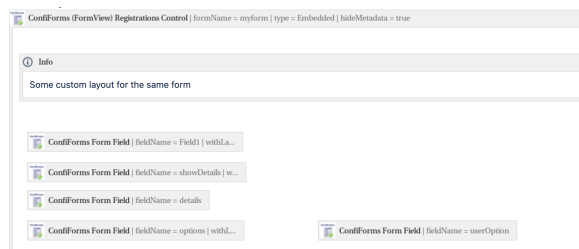
And the standard layout for the form

Implemented like



Custom layout for the same form

Implemented like



Configuration

As you can see, our form has 5 fields and 2 rules. Actually there are 4 rules, as each of the 2 rules has a "reverse rule" automatically created. That works for show/hide rules very well and you do not need to create an extra reverse rule yourself.

Our 5 fields are:

- Field1 - simple text field (does not really needed here, but added to show a simplest field possible - no rules, no behaviour)
- showDetails - is a checkbox field and that manages the visibility of the other field named "details"
- details - textarea field that is shown only when the checkbox "showDetails" is checked
- options - is a dropdown field with 4 options, 4th option manages the visibility of the "userOption" field
- userOption - is a text field that is shown only when someone has selected an option with ID=4 in the "options" field

Now let's see the 2 rules that add the dynamics to our form:

Full configuration for the form in storage format

```
<ac:structured-macro ac:macro-id="a356f4a8-bdfc-4f79-bc45-56c1abc82106" ac:name="conform" ac:schema-version="1">
  <ac:parameter ac:name="formName">myForm</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="94923a9b-bc51-4272-a4f3-e1711836ea84" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">Field1</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="faa26a33-b268-415f-bda2-ca26354f8237" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">showDetails</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Let me provide some details</ac:parameter>
        <ac:parameter ac:name="type">checkbox</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="cfde56ca-1d74-40f5-a437-105c7760d598" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="
```

```
// this rule will hide the "details" field when a checkbox field "showDetails" is unchecked and will show the "details" field otherwise
ConfiForms Rules for Field Definitions | actionFieldName = showDetails...

// this rule shows an extra field when you select an "other" option in the "options" dropdown field (and hides it otherwise)
ConfiForms Rules for Field Definitions | actionFieldName = options |...
```

Rule 1	Rule 2
First rule is "bound" to field "showDetails" and tracks for changes	Second rule is "bound" to field "options" and tracks for changes
It has a condition to check if showDetails field has value "false"	It checks for "options" field to have a value 4 selected
We set "actionable field" as "details", as this is the field we want to manage visibility for	Please note that the same could be achieved by using the field "showDetails" as the actionable field (checking the label of a dropdown instead)
	We set "actionable field" as "userOption", as this is the field for

Edit 'ConfiForms Rules for Field Definitions'

Rules for ConfiForms Field Definitions Documentation

Field name (or regular expression)

showDetails

Name of the field you want to track for changes (leave blank when you configure validation rules, as these rules are applied on form submit). Parameter supports regular expressions to affect multiple fields with the same rule. Can be given as comma separated list of fields

Condition

showDetails:false

Action will be executed when the condition is met. Same syntax expected as in filters. Empty value means it matches everything. The scope is **current entry/record** (for dataset validation the scope is all records). You can reference the values from the

```
fieldName">details</ac:parameter>
    <ac:parameter ac:name="fieldLabel">Details</ac:parameter>
    <ac:parameter ac:name="type">textarea</ac:parameter>
  </ac:structured-macro>
</p>
<p>
  It checks for "options" field to have a value 4 selected
  Please note that the same could be achieved by using the field "showDetails" as the actionable field (checking the label of a dropdown instead)
  We set "actionable field" as "userOption", as this is the field for
  <ac:parameter ac:name="fieldLabel">Choose an option</ac:parameter>
  <ac:parameter ac:name="values">false|Option 1|Option 2|3=Option 3|4=Let me provide my option|</ac:parameter>
  <ac:parameter ac:name="type">select</ac:parameter>
  </ac:structured-macro>
</p>
<p>
  Name of the field you want to track for changes (leave blank when you configure validation rules, as these rules are applied on form submit). Parameter supports regular expressions to affect multiple fields with the same rule. Can be given as comma separated list of fields
  <ac:parameter ac:name="fieldLabel">My option</ac:parameter>
  <ac:parameter ac:name="type">select</ac:parameter>
  </ac:structured-macro>
</p>
<br/>
</p>
<p>
  // this rule will hide the "details" field when a checkbox field "showDetails" is unchecked and will show the "details" field otherwise
  ConfiForms Rules for Field Definitions | actionFieldName = showDetails...
  // this rule shows an extra field when you select an "other" option in the "options" dropdown field (and hides it otherwise)
  ConfiForms Rules for Field Definitions | actionFieldName = options |...
```

Edit 'ConfiForms Rules for Field Definitions'

Rules for ConfiForms Field Definitions Documentation

Field name (or regular expression)

options

Name of the field you want to track for changes (leave blank when you configure validation rules, as these rules are applied on form submit). Parameter supports regular expressions to affect multiple fields with the same rule. Can be given as comma separated list of fields

Condition

options:4

Action will be executed when the condition is met. Same syntax expected as in filters. Empty value means it matches everything. The scope is **current entry/record** (for dataset validation the scope is all records). You can reference the values from the

```

  </p>
  <br/>
  </p>
  <p>
    // this rule will hide the "<em>details</em>" field when a checkbox field "<em>showDetails</em>" is unchecked and will show the "<em>details</em>"
    ConfiForms Rules for Field Definitions | actionFieldName = showDetails...
    ConfiForms Rules for Field Definitions | actionFieldName = options |...
```

Action to execute *

Hide field

Choose the action type to perform, see [more details on each action type in our documentation](#)

Actionable field name

details

Could be list of field names (comma separated) you want the action to be performed on

Value to match

If set, then only a field option holding this value is taken into action (Useful for radio/checkbox groups). You can use reference to other fields, via [entry.field_name] as usual

- ☒ With reverse rule (for 'Set field readonly' rule checking this option will unset the readonly state)

When checked, ConfiForms will try to create a reverse rule for given rule (works with Show/Hide field/container, 'Set field readonly' and 'Validate if exists in other Form' rules ONLY)

And we set "with reverse rule" to tell ConfiForms to create a reverse rule for us automatically. Reverse condition will be created and reverse action will be show)

You can set multiple fields in the "actionable field name" parameter, by using comma separated value list: field1,anotherfield,field3

You can do the same with 1st parameter in the ConfiForms Rules for Field Definition macro - and "bound" your rule to track changes in multiple fields (1 rule on multiple fields). Similarly to "actionable field name" parameter you can provide a comma separated list of field names

Conditions are written as filters, more on ConfiForms filters you can find here: [ConfiForms Filters](#) and [ConfiForms Filters by example](#)

Action to execute *

Show field

Choose the action type to perform, see [more details on each action type in our documentation](#)

Actionable field name

userOption

Could be list of field names (comma separated) you want the action to be performed on

- ☒ With reverse rule (for 'Set field readonly' rule checking this option will unset the readonly state)

When checked, ConfiForms will try to create a reverse rule for given rule (works with Show/Hide field/container, 'Set field readonly' and 'Validate if exists in other Form' rules ONLY)

```
field otherwise</p>
</ac>
<ac:structured-
macro ac:macro-id="
7299078c-2038-4881-976f-
48dce42b8cc5" ac:name="
ConfiForm-Field-Definition-
rules" ac:schema-version="
1">
  <ac:
parameter ac:name="
condition">options:4</ac:
parameter>
  <ac:
parameter ac:name="
actionField">userOption</ac:
parameter>
  <ac:
parameter ac:name="
actionField">options</ac:
parameter>
  <ac:
parameter ac:name="
withReverseRule">true</ac:
parameter>
</ac:
structured-macro>
</p>
<p>
  <br/>
</p>
</ac:rich-text-
```

```
body>  
  </ac:structured-  
macro>
```

To import via Atlassian [Confluence](#)
[Source Editor](#)