

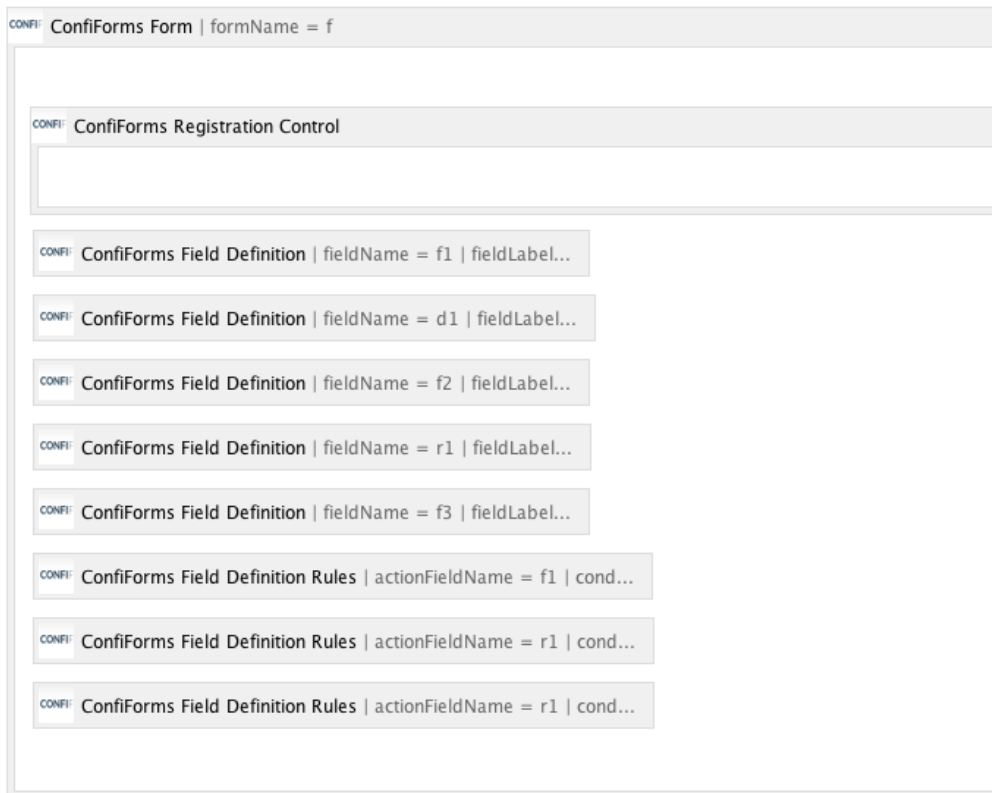
Creating dynamic and interactive forms in Confluence. How to create dependent fields in ConfiForms

 In this tutorial we will create a data entry form with dependent fields

- form has 5 fields
- the form will have 3 rules
 - one rule will set the values on 2 other fields
 - another rule will be to show some field based on the value selected. Here we need 2 rules, because we want to show some field based on the selection and hide otherwise. See the details below
- form uses standard registration control

 If you are new to ConfiForms, please take a few minutes to read the [Basic concepts](#) Guide, or go through the video tutorial on that page. 😊

Here is how the form looks in the Confluence editor



Total 5 fields

1. f1 - text field
2. d1 - dropdown field with 2 choices (with id=1 and label "one" and with id=2 and label "two")
3. f2 - text field
4. r1 - radio group field (with id=val1 and label "Value 1" and with id=val2 and label "Value 2")
5. f3 - text field

It is important to remember field names, as ConfiForms Field Definition rules macro expects you to know it

Here is how the form looks in the view mode

Registration

F1

Dropdown

F2

Radio 1 ☐ Value 1 ☐ Value 2

Save Cancel



Important note: on the screenshot above you see only **4 fields initially**. This is because of the rules defined. See below.

Let's see what is in rules

1. First rule defines that when the field f1 has the value "testme" it should do the following:

Field name *

Name of the field you want to track

Condition

Action will be executed when condition is met. [Same syntax as in filters](#) The scope is current entry/record

Action *

Action to perform

Values to set

Set values as key-value pairs, separated by & (as request parameters). Where 'key' is field name and 'value' is the value to set

- a. set the dropdown field (d1) to value "2"
- b. set text "value is set by field 1" to field f2

2. Another rule observes the value chosen in radio button

Field name *

Name of the field you want to track

Condition

Action will be executed when condition is met. [Same syntax as in filters](#) The scope is current entry/record

Action *

Action to perform

Actionable field name

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

shows the field 'f3' when field 'r1' has first option selected (the option is with id = "val1")

3. Third rule also observes the value chosen in the same radio button

Field name *

Name of the field you want to track

Condition

Action will be executed when condition is met. Same syntax as in filters The scope is current entry/record

Action *

Action to perform

Actionable field name

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

but hides the field 'f3' where second option is selected in the radio group control or when nothing is selected

The storage format for the form is listed below

```
<ac:structured-macro ac:name="conform">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:name="conform-entry-register">
        <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
        <ac:rich-text-body>
          <p> </p>
        </ac:rich-text-body>
      </ac:structured-macro>
      <ac:structured-macro ac:name="conform-field-definition">
        <ac:parameter ac:name="type">text</ac:parameter>
        <ac:parameter ac:name="fieldName">f1</ac:parameter>
        <ac:parameter ac:name="fieldLabel">F1</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:name="conform-field-definition">
        <ac:parameter ac:name="values">>false[1=one|2=two]</ac:parameter>
        <ac:parameter ac:name="type">select</ac:parameter>
        <ac:parameter ac:name="fieldName">d1</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Dropdown</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:name="conform-field-definition">
        <ac:parameter ac:name="type">text</ac:parameter>
        <ac:parameter ac:name="fieldName">f2</ac:parameter>
        <ac:parameter ac:name="fieldLabel">F2</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:name="conform-field-definition">
        <ac:parameter ac:name="values">>false[val1=Value 1|val2=Value 2]</ac:parameter>
        <ac:parameter ac:name="type">radio_group</ac:parameter>
        <ac:parameter ac:name="fieldName">r1</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Radio 1</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:name="conform-field-definition">
        <ac:parameter ac:name="type">text</ac:parameter>
        <ac:parameter ac:name="fieldName">f3</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Field 3</ac:parameter>
      </ac:structured-macro>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>
```

```

</p>
<p>
  <ac:structured-macro ac:name="confiorm-field-definition-rules">
    <ac:parameter ac:name="values">d1=2&f2=value is set by field 1</ac:parameter>
    <ac:parameter ac:name="condition">f1:testme</ac:parameter>
    <ac:parameter ac:name="action">Set value</ac:parameter>
    <ac:parameter ac:name="actionFieldName">f1</ac:parameter>
  </ac:structured-macro>
</p>
<p>
  <ac:structured-macro ac:name="confiorm-field-definition-rules">
    <ac:parameter ac:name="condition">r1:val1</ac:parameter>
    <ac:parameter ac:name="action">Show field</ac:parameter>
    <ac:parameter ac:name="fieldName">f3</ac:parameter>
    <ac:parameter ac:name="actionFieldName">r1</ac:parameter>
  </ac:structured-macro>
</p>
<p>
  <ac:structured-macro ac:name="confiorm-field-definition-rules">
    <ac:parameter ac:name="condition">r1:val2 OR r1:</ac:parameter>
    <ac:parameter ac:name="action">Hide field</ac:parameter>
    <ac:parameter ac:name="fieldName">f3</ac:parameter>
    <ac:parameter ac:name="actionFieldName">r1</ac:parameter>
  </ac:structured-macro>
</p>
<p> </p>
</ac:rich-text-body>
</ac:structured-macro>

```

Check out the video how the form looks like and behaves based on the values selected