


Conditional validation rules against dataset

 In this tutorial you will learn how to set up validation rules to validate against the data stored

 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. 😊

Often there is a need to setup some checks to ensure the limits on some values.

In the example below we have a requirement to deny registrations if the dataset has already 3 records of the choice B

This validation is possible to automate in ConfiForms by using the "ConfiForms Field Definition Rules" macro and using it's "Action" selected to "Validation Rule (dataset)"

We start with the form, which has only one field of type Dropdown, a registration Control macro to enable new registrations and "ConfiForms Field Definition Rules" macro with rules to check the dataset

validation per field value with dropdown

|



The screenshot displays the configuration interface for a ConfiForms form. It shows a hierarchy of macros:

- CONFI: ConfiForms Form | formName = f** (parent macro)
- CONFI: ConfiForms Field Definition | fieldName = myddfield | fie...** (child macro)
- CONFI: ConfiForms Field Definition Rules | actionFieldName = myddfield...** (child macro)
- CONFI: ConfiForms Registration Control** (child macro)

The interface includes a search bar and a list of macros, with the 'ConfiForms Field Definition Rules' macro highlighted.

The dropdown field looks like this:

Edit 'ConfiForms Field Definition' Macro

Field type

Track changes

Sorting

id	value
A	this is A <input type="text"/> <input type="text"/>
B	this is B <input type="text"/> <input type="text"/>
C	this is C <input type="text"/> <input type="text"/>

Field description (optional)

Select macro

has 3 values: A, B and C

And the rules for validation is as follows:

Edit 'ConfiForms Field Definition Rules' Macro

Name of the field you want to track
(optional for validation rules)

Condition

Action will be executed when condition is met. [Same syntax as in filters](#) The scope is current entry/record (for dataset validation the scope is all records). You can reference current records fields via [entry.field_name]

Action *

Action to perform

Message to show when validation matches

Select macro

we check for values "B" and count them using built-in variable `[count]`



Variable has changed from [count] to `_count` as in [Accessing field values and properties](#)

Condition will match when a dataset has 3 records (where field value for field "myddfield" is "B") in it already

When it does the message to use will be shown

Storage format (configuration) for the whole solution:

```
<ac:structured-macro ac:macro-id="e15f05e9-90a9-4725-bd84-af49bedf9a8c" ac:name="confiform" ac:schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="56f6d25e-8791-4b69-9f83-93d3a9a85e29" ac:name="confiform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">myddfield</ac:parameter>
        <ac:parameter ac:name="fieldLabel">My drop down field</ac:parameter>
        <ac:parameter ac:name="values">>false[A=this is A|B=this is B|C=this is C|]</ac:parameter>
        <ac:parameter ac:name="type">select</ac:parameter>
      </ac:structured-macro>
    </p>
    <ac:structured-macro ac:macro-id="a855e900-ea9c-44e7-b838-dfe48529a95d" ac:name="confiform-field-definition-rules" ac:schema-version="1">
      <ac:parameter ac:name="condition">myddfield:B AND _count:>3</ac:parameter>
      <ac:parameter ac:name="fieldName">There are already 3 items of B value</ac:parameter>
      <ac:parameter ac:name="action">Validation rule dataset</ac:parameter>
      <ac:parameter ac:name="actionFieldName">myddfield</ac:parameter>
    </ac:structured-macro>
    <p> <ac:structured-macro ac:macro-id="639bf3c1-fa34-43c7-8dce-e8ab9f483f76" ac:name="confiform-entry-register" ac:schema-version="1">
      <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
      <ac:rich-text-body>
        <p> </p>
      </ac:rich-text-body>
    </ac:structured-macro> </p>
  </p> </p>
</ac:rich-text-body>
</ac:structured-macro>
```

More advanced storage to validate ANY option for max 3 items

```
<ac:structured-macro ac:macro-id="e15f05e9-90a9-4725-bd84-af49bedf9a8c" ac:name="confiform" ac:schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="56f6d25e-8791-4b69-9f83-93d3a9a85e29" ac:name="confiform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">myddfild</ac:parameter>
        <ac:parameter ac:name="fieldLabel">My drop down field</ac:parameter>
        <ac:parameter ac:name="values">>false[A=this is A|B=this is B|C=this is C|]</ac:parameter>
        <ac:parameter ac:name="type">select</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="a855e900-ea9c-44e7-b838-dfe48529a95d" ac:name="confiform-field-definition-rules" ac:schema-version="1">
        <ac:parameter ac:name="condition">myddfild:[entry.myddfild] AND _count:>3</ac:parameter>
        <ac:parameter ac:name="fieldName">There are already 3 items of '[entry.myddfild.label]' value</ac:parameter>
        <ac:parameter ac:name="action">Validation rule dataset</ac:parameter>
        <ac:parameter ac:name="actionFieldName">myddfild</ac:parameter>
      </ac:structured-macro>
    <p> <ac:structured-macro ac:macro-id="639bf3c1-fa34-43c7-8dce-e8ab9f483f76" ac:name="confiform-entry-register" ac:schema-version="1">
      <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
      <ac:rich-text-body>
        <p> </p>
      </ac:rich-text-body>
    </ac:structured-macro> </p>
  </ac:rich-text-body>
</ac:structured-macro>
```



Of course you can have as many validation rules as you want/need, with different conditions (which span across many fields if necessary)