

ConfiForms Form to allow only one entry per user per month

This short tutorial shows you how to setup a form with conditional validation, which allows only one record per user per month

Consider a very simple form with just one field of type "Date", we name it "mydate"

The screenshot shows the 'ConfiForms Form (Definition) | formName = f' editor. It contains a 'ConfiForms Form Registration Control' and a 'ConfiForms Form Field (Definition) | fieldName = mydate | fieldL...'. Below the field definition is a 'ConfiForms Rules for Field Definition | condition = mydate.formatDa...' rule.

Just one field and one rule. Everything is in default state, the important bit here is the ConfiForms Rules for Field Definition macro and how it is set up

The screenshot shows the 'Edit ConfiForms Rules for Field Definition' macro configuration screen. It includes a 'Field name' input field, a 'Condition' input field with the text 'mydate.formatDate(MM):[entry.my', and a checkbox labeled 'Execute only on user action'. A 'Preview' button is visible on the right.

Rules for ConfiForms Field Definitions [Documentation](#)

Field name

Name of the field you want to track (optional for validation rules and for rules which should be executed ONLY on users action)

Condition

mydate.formatDate(MM):[entry.my

Action will be executed when condition is met. [Same syntax expected 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]

☐ Execute only on user action

Action *

Validation rule (dataset) ▼

Choose Action to perform

Message to show when validation rule matches

Already exists for this month

Validation rule

Regular Expression to use for fields input validation (when expression matches (or is EMPTY) a message will be shown)

So, we have a rule to run a validation action across the dataset and this will be executed when the condition is met.

The condition is set as follows:

```
mydate.formatDate(MM):[entry.mydate.formatDate(MM)] AND createdBy:[user.name] AND _count:>1
```

We use [Virtual functions](#) here to transform a date field into a month, we use context variable `_count` and we use context variable called `[user.name]` to access the current user login

Alternative way of writing the condition (and more modern), using the context variable `_user`

```
mydate.formatDate(MM):[entry.mydate.formatDate(MM)] AND createdBy:[entry._user] AND _count:>1
```

Complete configuration in storage format is available [here](#)

```
<ac:structured-macro ac:macro-id="e3152962-e681-4234-bed7-44d892da4907" ac:name="confiform" ac:schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <ac:structured-macro ac:macro-id="8dcdefd1-41ab-4847-83b1-77ca5f8b2b8e" ac:name="confiform-entry-register"
ac:schema-version="1">
      <ac:rich-text-body>
        <p> </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  <p>
    <ac:structured-macro ac:macro-id="82642dea-938e-45c7-a735-d2f6cf7bd96f" ac:name="confiform-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">mydate</ac:parameter>
      <ac:parameter ac:name="fieldLabel">My date</ac:parameter>
      <ac:parameter ac:name="type">date</ac:parameter>
    </ac:structured-macro>
  </p>
</p>
  <ac:structured-macro ac:macro-id="71d71948-d0ae-42d2-a66c-e452e8f85bd2" ac:name="confiform-field-
definition-rules" ac:schema-version="1">
    <ac:parameter ac:name="condition">mydate.formatDate(MM):[entry.mydate.formatDate(MM)] AND createdBy:
[user.name] AND _count:>1</ac:parameter>
    <ac:parameter ac:name="fieldName">Already exists for this month</ac:parameter>
    <ac:parameter ac:name="action">Validation rule dataset</ac:parameter>
  </ac:structured-macro>
</p>
</ac:rich-text-body>
</ac:structured-macro>
```