Using ConfiForms Field Definition Rules macro to reduce number of choices in a dropdown field based on the value selected in another field

> ⓘ In this tutorial you will learn how to Use ConfiForms Field Definition Rule to reduce a number of choices in a dropdown field.
>
> We will consider 2 setups - one which uses database and another, which uses other form.
>
> This tutorial is aplicable not only to dropdown fields, but also to:
>
> - radio group fields
> - advanced dropdowns
> - multi-select fields
> - checkbox groups fields

> ⊘ 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. 🙂

Let's start with form design. It will have just 2 fields

- a checkbox field called "includeActiveOnly"
- a dropdown field called "choices" with values loaded from
  - a database table
  - another form

This means that we will have 2 variants of the same form (or 2 forms you may say)

Dropdown choices from a database table

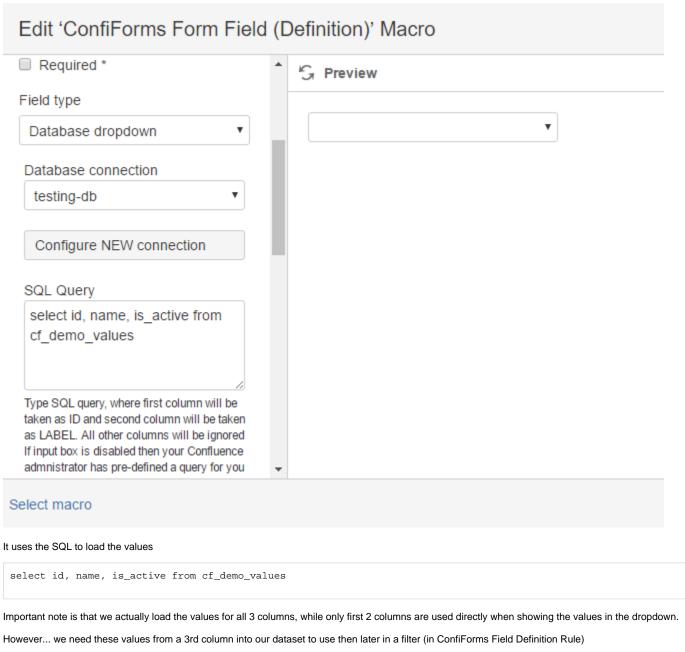Let's start with the one which has a dropdown field with the values loaded from a database table

A table we connect as a dropdown source has the following structure

ID	NAME	IS_ACTIVE
1	This one	true
2	Another	true
3	Some inactive	false
4	Quattro	true
5	Obsolete	false

This is stored in a table called "cf_demo_values"

```
create table cf_demo_values (id INTEGER NOT NULL, name CHARACTER(50) NOT NULL, is_active boolean);
```

Our "choices" dropdown is configured as follows

Edit 'ConfiForms Form Field (Definition)' Macro

☐ Required *

Field type

| Database dropdown ▼ |

Database connection

| testing-db ▼ |

| Configure NEW connection |

SQL Query

| select id, name, is_active from cf_demo_values |

Type SQL query, where first column will be
taken as ID and second column will be taken
as LABEL. All other columns will be ignored
If input box is disabled then your Confluence
admnistrator has pre-defined a query for you

↺ **Preview**

| ▼ |

Select macro

It uses the SQL to load the values

```
select id, name, is_active from cf_demo_values
```

Important note is that we actually load the values for all 3 columns, while only first 2 columns are used directly when showing the values in the dropdown.

However... we need these values from a 3rd column into our dataset to use then later in a filter (in ConfiForms Field Definition Rule)
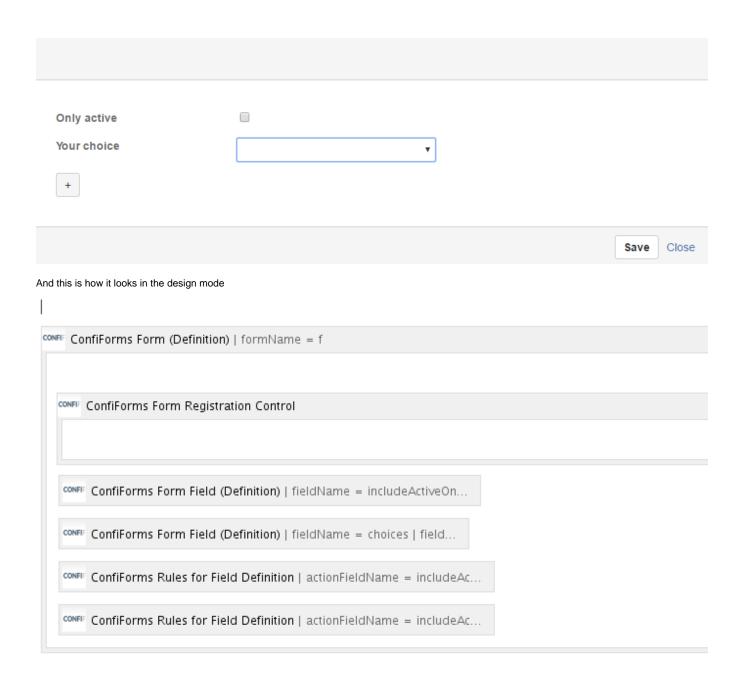
> ⊘ Important!
>
> When SQL is loaded into ConfiForms the names for columns which have underscore (_) in their names are removed
>
> So, in our case the column "is_active" becomes "isactive"
>
> This is very important when you use it in a filter later, see below

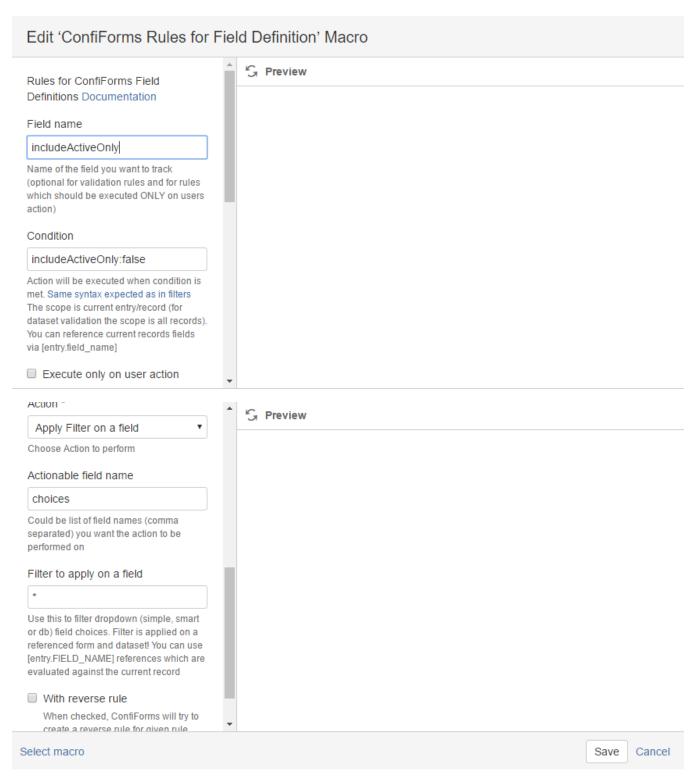Then form in the view mode looks like this

Only active	☐
Your choice	[▼]

[+]

Save   Close

And this is how it looks in the design mode

|

CONFI ConfiForms Form (Definition) | formName = f

> CONFI ConfiForms Form Registration Control
>
> CONFI ConfiForms Form Field (Definition) | fieldName = includeActiveOn...
>
> CONFI ConfiForms Form Field (Definition) | fieldName = choices | field...
>
> CONFI ConfiForms Rules for Field Definition | actionFieldName = includeAc...
>
> CONFI ConfiForms Rules for Field Definition | actionFieldName = includeAc...

Let's see how the 2 last macros (Field Definition Rules) are configured

Edit 'ConfiForms Rules for Field Definition' Macro

Rules for ConfiForms Field
Definitions Documentation

Field name

includeActiveOnly

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

Condition

includeActiveOnly:true

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 *

Apply Filter on a field ▼

Choose Action to perform

Actionable field name

choices

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

Filter to apply on a field

isactive:true

Use this to filter dropdown (simple, smart
or db) field choices. Filter is applied on a
referenced form and dataset! You can use
[entry.FIELD_NAME] references which are
evaluated against the current record

☐ With reverse rule

When checked, ConfiForms will try to

🔄 **Preview**

🔄 **Preview**

Select macro   Save   Cancel

See an important bit on the filter field - we take only "active" rows. This "isactive" field value comes from an SQL, from a field "is_active"

This rule fires when checkbox is checked

Here is how the 2nd Rule is configured

Edit 'ConfiForms Rules for Field Definition' Macro

Rules for ConfiForms Field
Definitions Documentation

Field name

```
includeActiveOnly
```

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

Condition

```
includeActiveOnly:false
```

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

```
Apply Filter on a field          ▼
```

Choose Action to perform

Actionable field name

```
choices
```

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

Filter to apply on a field

```
*
```

Use this to filter dropdown (simple, smart
or db) field choices. Filter is applied on a
referenced form and dataset! You can use
[entry.FIELD_NAME] references which are
evaluated against the current record

☐ With reverse rule

When checked, ConfiForms will try to
create a reverse rule for given rule

⟳ Preview

⟳ Preview

Select macro   Save   Cancel

In a filter we tell it to take ALL (this rule fires when checkbox is not checked)

Dropdown choices from another form

This is a 2 version of the form, when your dropdown field is actually takes the values from another ConfiForms Form (using a "smart" field, a "smart" dropdown)
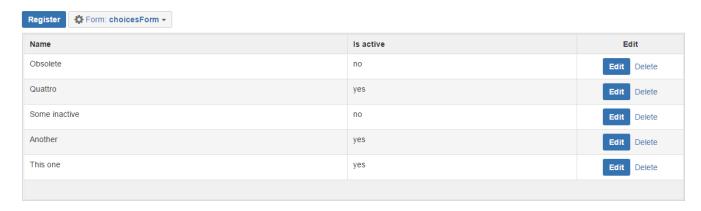
In the edit mode the form looks exactly the same as in the 1st version.

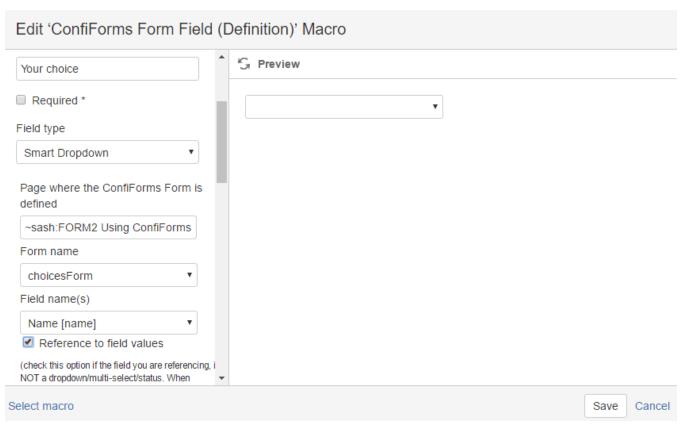But the difference is in how the "choices" field is configured

It is configured to take the vlues from another ConfiForms Form with the following structure

- Field to store name (field types is "text")
- Field to store the status (active or inactive), with field type "checkbox"

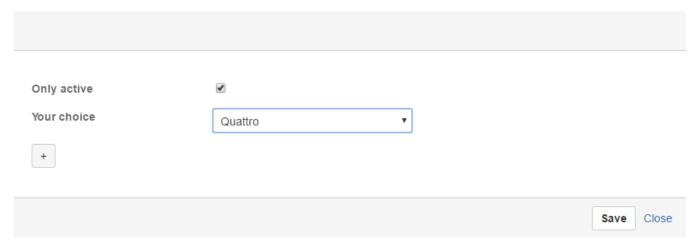Same set of values as in the database version

Register	Form: choicesForm ▾		
Name	**Is active**		**Edit**
Obsolete	no		Edit Delete
Quattro	yes		Edit Delete
Some inactive	no		Edit Delete
Another	yes		Edit Delete
This one	yes		Edit Delete

And the "choices" field is configured like this:

Edit 'ConfiForms Form Field (Definition)' Macro

Your choice

☐ Required *

Field type

Smart Dropdown ▾

Page where the ConfiForms Form is defined

~sash:FORM2 Using ConfiForms

Form name

choicesForm ▾

Field name(s)

Name [name] ▾

☑ Reference to field values

(check this option if the field you are referencing, i
NOT a dropdown/multi-select/status. When

↺ Preview

▾

Select macro   Save   Cancel

Taking the values from a "choiceForm" and showing those in teh dropdown

As we have the same field name to show if the choice is active or not the Field Definition Rules stay the same as in the version 1 of the form.

Giving us the same result in the end

| Only active | ☑ |
| Your choice | Quattro ▾ |

+

Save   Close

Showing only active choices and serving them from another ConfiForms Form

Complete solution for version 1 of the form

```
<ac:structured-macro ac:macro-id="29ea6581-a183-4fac-b595-8161170b3d06" 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="ec132d3d-37c5-48d8-8a11-afc352b2e01a" 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="a35a3f59-9af3-4b51-a083-e3666d6e033a" ac:name="confiform-field-
definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">includeActiveOnly</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Only active</ac:parameter>
        <ac:parameter ac:name="type">checkbox</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="94905715-ee3a-440c-9556-86db790db4a9" ac:name="confiform-field-
definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">choices</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Your choice</ac:parameter>
        <ac:parameter ac:name="values">select id, name, is_active from cf_demo_values</ac:parameter>
        <ac:parameter ac:name="extras">7422036e-9a6f-4455-92d9-7140131fb9da</ac:parameter>
        <ac:parameter ac:name="type">dbselect</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="afcef165-b323-4b02-adb0-9cfd00d61da2" ac:name="confiform-field-
definition-rules" ac:schema-version="1">
        <ac:parameter ac:name="condition">includeActiveOnly:true</ac:parameter>
        <ac:parameter ac:name="fieldName">choices</ac:parameter>
        <ac:parameter ac:name="values">isactive:true</ac:parameter>
        <ac:parameter ac:name="action">Apply Filter on a field</ac:parameter>
        <ac:parameter ac:name="actionFieldName">includeActiveOnly</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="310c818c-9460-4d7b-a780-ba66ba4c5dda" ac:name="confiform-field-
definition-rules" ac:schema-version="1">
        <ac:parameter ac:name="condition">includeActiveOnly:false</ac:parameter>
        <ac:parameter ac:name="fieldName">choices</ac:parameter>
        <ac:parameter ac:name="values">*</ac:parameter>
        <ac:parameter ac:name="action">Apply Filter on a field</ac:parameter>
        <ac:parameter ac:name="actionFieldName">includeActiveOnly</ac:parameter>
      </ac:structured-macro>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>
```