Using and configuring ConfiForms Dynamic Filter Control

ⓘ In this tutorial we will demonstrate how to use a new feature of ConfiForms to construct dynamic filters over your stored data. The macro is called ConfiForms Dynamic Filter

⚠ 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. 🙂

As usual, let's start with a simple form. The form will have 3 fields

Field label	Field name	Type
My Text Field	mytffield	text
My Dropdown Field	mydropdownfield	advanced dropdown
My Text Area	mytextarea	textarea

Storage format

```
<ac:structured-macro ac:macro-id="45a7667b-79d4-4896-9830-705af8285cb8" 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="580584ec-6e33-4fb0-bc98-8f50d55cfd01" ac:name="confiform-field-
definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">mytffield</ac:parameter>
        <ac:parameter ac:name="fieldLabel">My Text Field</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="2a7dd6b1-30c4-4602-91eb-8753d391ddc1" ac:name="confiform-field-
definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">mydropdownfield</ac:parameter>
        <ac:parameter ac:name="fieldLabel">My Dropdown Field</ac:parameter>
        <ac:parameter ac:name="values">false[1=one|2=two|3=three|4=four|]</ac:parameter>
        <ac:parameter ac:name="type">select2</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="7399797b-9c13-461b-a8eb-3a763673d5b3" ac:name="confiform-field-
definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">mytextarea</ac:parameter>
        <ac:parameter ac:name="fieldLabel">My TextArea</ac:parameter>
        <ac:parameter ac:name="type">textarea</ac:parameter>
      </ac:structured-macro>
      <ac:structured-macro ac:macro-id="746fdd65-348b-4c8c-be60-b48044c67761" 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>
```

Something like this

We will add some records into the form

My Text Field	My Dropdown Field	My TextArea
Something else	four	
Record #2	two	no comments to this
This is one	one	ConfiForms is number #1 add-on! :-)

Now it is time to bring in the *ConfiForms Form Filter Control* macro

Here is the macro, with the default settings it takes all the fields from the form and present them in a card view. You can easily change the layout of this, see below

> ⊘ The macro has been renamed from ConfiForms Dynamic Filter Control to **ConfiForms Form Filter Control** macro
>
> { filter
>
> **Macro suggestions**
>
> ConfiForms **ConfiForms Form Filter Control**

> ⓘ ConfiForms Form Filter Control macro supports custom layouts. If you wish to have a **free text search with custom layout** then add a ConfiForms Field macro inside the Filter Control macro body and set it's field name parameter to *
>
> (wildcard)

My Text Field

My Dropdown Field ▾

My TextArea

Filter

But it does not work without a "view"... without "views" macros, such as CalendarView, ListView, TableView or CardView.

Trying to use it without a view will result in a warning message

Filter

Could not find a ConfiForms view macro to apply the filter. Make sure you have ConfiForms ListView, CalendarView or TableView/CardView on the same page (showing the data from the same form); and for **TableView/CardView you have enabled filter support**.

So we need a "view"

We will use the default ConfiForms TableView

CONFI ConfiForms Dynamic Filter Control | formName = f

|

CONFI ConfiForms TableView | formName = f

> **CONFI** ConfiForms Field | fieldName = mytffield
>
> **CONFI** ConfiForms Field | fieldName = mydropdownfield
>
> **CONFI** ConfiForms Field | fieldName = mytextareafield

However, for TableView and CardView you *MUST* enable support for filter in macro properties

Edit 'ConfiForms TableView' Macro

No limit when is blank

☐ Enable instant search field over the table data

☑ Enable support for ConfiForms Filter Control

Resulting table will be wrapped into a container which ConfiForms Filter Control will be able to find and filter

Pagination

[▼]

Select number of records per page

☐ Enable export

ConfiForms data in CSV, XLS, XML and JSON format

Comma separated list of fields to export

⟳ **Preview**

✓ **Visible only in PREVIEW**

Configuration helper for ListViews, TableViews, CardViews, PlainView, CalendarViews and Registration Control macros
Show

My Text Field	My Dropdown Field
This is one	one
Record #2	two
Something else	four

Select macro   Save   Cancel

Now we can use the filter

My Text Field	My Dropdown Field
Record #2	two

Here is an example using the Dynamic filter with custom layout and using it together with a ListView

ConfiForms Dynamic Filter Control | formName = f | filterFor = list

Select TYPE: ConfiForms Field | fieldName = mydropdownfield

ConfiForms ListView | formName = f | enableFilterSupport = true

ConfiForms Field | fieldName = mytffield

ConfiForms Field | fieldName = mydropdownfield

(i) Info

ConfiForms Field | fieldName = mytextarea

1. We have put ConfiForms Field macro inside the Dynamic Filter to force own layout and use only the fields we want
2. We Enabled Filter for "list" (listview) in Dynamic Filter
3. We created ListView with custom layout (ListView and CalendarView do not require filter support to be explicitely set, it is there by default. In TableView and CardView it is off by default to support clean tables and compatibility with COnfluence Chart macro)

Here is how it looks like on the page

Select TYPE: one ▼

Filter

This is one

one

(i) ConfiForms is number #1
add-on! :-)

Thank you for you time and reading this far. The tutorial now ends, but there is a bit more to explore in Dynamic Filter macro, such as:

- You can reduce the number of choices in dropdown fields (and all other choice-enabled fields) to show only available values
- When you enable inline search on "views" you will actually see what filter is used to filter
- You can set "views" to show nothing initially (useful for large dataset). To do this you need to write some filter which will not match anything. One to suggest is to use internal field _today and compare it agains 0

```
_today:<0
```

Which is always false, as _today holds the current timestamp
More about filter can be found here: ConfiForms Filters
- You can add a form reset button (setting in Dynamic Filter macro) to ease the clean up of fields to start over