

Working with Dependent DB dropdowns

Quick demo, showing how to work with Database dropdown fields and make connection/dependency/filter on the choices they provide

Let's start...

We have 2 tables with dependent data

One-to-many from **contractor_categories** to **contractor_subcategories**



If you are starting with dropdown fields then start by looking at <https://wiki.vertuna.com/display/CONFIFORMS/Working+with+database+fields+in+ConfiForms> tutorial, which explains the basics a bit more

Dropdown fields use the following SQL queries

```
SELECT id, name FROM contractor_categories

SELECT id, name, contractor_categories_id FROM contractor_subcategories
```

DB setup

```
create table contractor_categories (id int, name varchar(100));
create table contractor_subcategories (id int, name varchar(100), contractor_categories_id int);

insert into contractor_categories values(1, 'Software Development');
insert into contractor_categories values(2, 'Cleaning');
insert into contractor_categories values(3, 'Security');

insert into contractor_subcategories values(4, 'General cleaning', 2);

insert into contractor_subcategories values(5, 'Entrance', 3);
insert into contractor_subcategories values(6, 'Night shifts only', 3);
insert into contractor_subcategories values(7, 'Basement and perimeter', 3);
```

Form configuration

Register

"Dependency" is defined using "ConfiForms Field Definition Rule" with action to filter choices

Edit 'ConfiForms Rules for Field Definition' Macro

Apply Filter on a field ▼

Choose Action to perform

Actionable field name

subcategory

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

Filter to apply on a field

contractorcategoriesid:[entry.categ

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

☐ With reverse rule

↺ Preview

Select macro

Save

Cancel

Rule is set to be connected to changes on "category" field (not shown on the screenshot, but it is the first parameter in the rules macro)

And important bit is the filter:

contractorcategoriesid:[entry.category.id]

Why "contractorcategoriesid" and not "contractor_categories_id" as in the SQL defined? Because the way the field name is defined, allowing no non-alphanumeric characters in it's name. SQL query results become the dataset similar to internal datasets stored within ConfiForms.

Storage format for the whole solution (you can import it using the free add-on from Atlassian, called Source editor, <https://marketplace.atlassian.com/plugins/com.atlassian.confluence.plugins.editor.confluence-source-editor/server/overview>)

```

<ac:structured-macro ac:macro-id="14859127-f36f-4e7f-9de8-219a6790b6e3" ac:name="conform" ac:schema-version="1">
  <ac:parameter ac:name="formName">f1</ac:parameter>
  <ac:rich-text-body>
    <ac:structured-macro ac:macro-id="fdff034e-b509-4a3f-ae55-ed872aa6d0be" ac:name="conform-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="2da9717f-57f5-4eda-9670-57123708c625" ac:name="conform-field-definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">category</ac:parameter>
      <ac:parameter ac:name="fieldLabel">Contractor Category</ac:parameter>
      <ac:parameter ac:name="values">SELECT id, name FROM contractor_categories</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="415940f1-d144-445d-9b72-9821d0310930" ac:name="conform-field-definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">subcategory</ac:parameter>
      <ac:parameter ac:name="fieldLabel">Contractor Sub Category</ac:parameter>
      <ac:parameter ac:name="values">SELECT id, name, contractor_categories_id FROM contractor_subcategories</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="d6b863fc-69cb-45e3-bbeb-5c76578158f7" ac:name="conform-field-definition-rules" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">subcategory</ac:parameter>
      <ac:parameter ac:name="values">contractorcategoriesid:[entry.category.id]</ac:parameter>
      <ac:parameter ac:name="action">Apply Filter on a field</ac:parameter>
      <ac:parameter ac:name="actionFieldName">category</ac:parameter>
    </ac:structured-macro>
  </p>
</ac:rich-text-body>
</ac:structured-macro>

```

Optional example, where we show the same form, but the SQLs used have "as" operators and the name of the fields are changed in the resulting dataset. Therefore the field definition rule is changed as well. See below

```

<ac:structured-macro ac:macro-id="14859127-f36f-4e7f-9de8-219a6790b6e3" ac:name="confiorm" ac:schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <ac:structured-macro ac:macro-id="fa517b7f-eabf-4755-bfa2-c3fce043bcc4" ac:name="confiorm-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="b2f153ba-f8ac-455b-b63b-ea07252f149a" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">category</ac:parameter>
      <ac:parameter ac:name="fieldLabel">Contractor Category</ac:parameter>
      <ac:parameter ac:name="values">SELECT id as "myid", name as "myname" FROM contractor_categories</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="77520462-dc4e-44d5-94f0-19fce73e80db" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">subcategory</ac:parameter>
      <ac:parameter ac:name="fieldLabel">Contractor Sub Category</ac:parameter>
      <ac:parameter ac:name="values">SELECT id, name, contractor_categories_id as "subcatid" FROM
contractor_subcategories</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="229b7d63-5896-4b43-b1ec-c06adfdc2a3f" ac:name="confiorm-field-
definition-rules" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">subcategory</ac:parameter>
      <ac:parameter ac:name="values">subcatid:[entry.category.id]</ac:parameter>
      <ac:parameter ac:name="action">Apply Filter on a field</ac:parameter>
      <ac:parameter ac:name="actionFieldName">category</ac:parameter>
    </ac:structured-macro>
  </p>
</ac:rich-text-body>
</ac:structured-macro>

```

The query for second dropdown is chnaged like this:

```
SELECT id, name, contractor_categories_id as "subcatid" FROM contractor_subcategories
```

Renaming "contractor_categories_id" to "**subcatid**"

Apply filter definition rule is changed as well to follow the rename

```
subcatid:[entry.category.id]
```

```

<ac:structured-macro ac:macro-id="14859127-f36f-4e7f-9de8-219a6790b6e3" 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="fa517b7f-eabf-4755-bfa2-c3fce043bcc4" 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="b2f153ba-f8ac-455b-b63b-ea07252f149a" ac:name="confiform-field-
definition"
    ac:schema-version="1">
    <ac:parameter ac:name="fieldName">category</ac:parameter>
    <ac:parameter ac:name="fieldLabel">Contractor Category</ac:parameter>
    <ac:parameter ac:name="values">SELECT id as "myid", name as "myname" FROM contractor_categories</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="77520462-dc4e-44d5-94f0-19fce73e80db" ac:name="confiform-field-
definition"
    ac:schema-version="1">
    <ac:parameter ac:name="fieldName">subcategory</ac:parameter>
    <ac:parameter ac:name="fieldLabel">Contractor Sub Category</ac:parameter>
    <ac:parameter ac:name="values">SELECT id, name, contractor_categories_id as "subcatid" FROM
      contractor_subcategories
    </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="229b7d63-5896-4b43-b1ec-c06adfdc2a3f" ac:name="confiform-field-
definition-rules"
    ac:schema-version="1">
    <ac:parameter ac:name="fieldName">subcategory</ac:parameter>
    <ac:parameter ac:name="values">subcatid:[entry.category.id]</ac:parameter>
    <ac:parameter ac:name="action">Apply Filter on a field</ac:parameter>
    <ac:parameter ac:name="actionFieldName">category</ac:parameter>
  </ac:structured-macro>
</p>
</ac:rich-text-body>
</ac:structured-macro>

```