

Merging the data from forms with different structure sharing a common key

We are often asked if TableViewMerger macro can deal with forms of a different structure and how to show the data from such forms in a single table (and build charts using this data later on if needed)

And up until version of 1.51.3 we were not been able to do it very nicely.

Well, the TableViewMerger macro does a great job merging and aggregating the data from different forms of the same or very similar structure, but there was no good way to merge the data from forms of a different structure (having different fields)

Imagine 2 forms, with a structures like this (it is important to have a form sharing a **field with the same name** for this to work!)

Form 1 (name "f")

- field with name "k" of type text
- field with name "name" of type text

Form 2 (name "f1")

- field with name "k" of type text
- field with name "age" of type number

Source code for the forms:

Form 1

```
<ac:structured-macro ac:macro-id="5bfd0a-c03b-4091-8a98-c9b13f3a50e0" 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="34c5ade4-26cf-4ba0-8ddb-6baebfba7f98" ac:name="confiorm-entry-register"
ac:schema-version="1">
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  <p>
    <ac:structured-macro ac:macro-id="1be25592-a765-422d-855d-ee343d684c57" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">k</ac:parameter>
      <ac:parameter ac:name="fieldLabel">k</ac:parameter>
      <ac:parameter ac:name="type">text</ac:parameter>
    </ac:structured-macro>
  </p>
  <p>
    <ac:structured-macro ac:macro-id="14c9c2a4-d84e-47da-bcbe-1f6c7bab10c4" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">name</ac:parameter>
      <ac:parameter ac:name="fieldLabel">name</ac:parameter>
      <ac:parameter ac:name="type">text</ac:parameter>
    </ac:structured-macro>
  </p>
</ac:rich-text-body>
</ac:structured-macro>
```

Form 2

```
<ac:structured-macro ac:macro-id="02ed11a4-5d12-4b29-bde4-23397add42f9" ac:name="confiorm" ac:schema-version="1">
  <ac:parameter ac:name="formName">f1</ac:parameter>
  <ac:rich-text-body>
    <ac:structured-macro ac:macro-id="44693cal-e471-400f-9eca-d89ce54d8332" ac:name="confiorm-entry-register"
ac:schema-version="1">
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  <p>
    <ac:structured-macro ac:macro-id="5fd10460-4c37-44d7-a2fd-478ad3f1e5ee" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">k</ac:parameter>
      <ac:parameter ac:name="fieldLabel">k</ac:parameter>
      <ac:parameter ac:name="type">text</ac:parameter>
    </ac:structured-macro>
  </p>
  <p>
    <ac:structured-macro ac:macro-id="a18ad470-c5de-497d-aa04-fff248c811c6" ac:name="confiorm-field-
definition" ac:schema-version="1">
      <ac:parameter ac:name="fieldName">age</ac:parameter>
      <ac:parameter ac:name="fieldLabel">age</ac:parameter>
      <ac:parameter ac:name="type">number</ac:parameter>
    </ac:structured-macro>
  </p>
</ac:rich-text-body>
</ac:structured-macro>
```

As you can see, both forms have a commonly named field called "k"

We gonna use this field as a "key" field in our TableView Merger macro

What is important is that in TableViewMerger macro you specify the ConfiForms Field macros, but some are taken as keys (for grouping), while others are taken as expressions.

Fields taken as expressions are given with the format like (may have references to record field values or have functions, like documented here: [Supported math operators, formulas and functions](#)):

```
some label:(EXPRESSION HERE)

some other label:([entry.age])
```

while fields which dont have such format are taken as keys (to construct a grouping key, for aggregation)

Here is how our TableViewMerger macro will look like, which will merge the data from these 2 forms like this:

k	name	age
k1	sash	20

Where the Form 1 has a row with fields k1 and name "sash", and Form 2 has a row with k1 and field "age" having a value 20.

Form 1

name
sash

Form 2

age
20

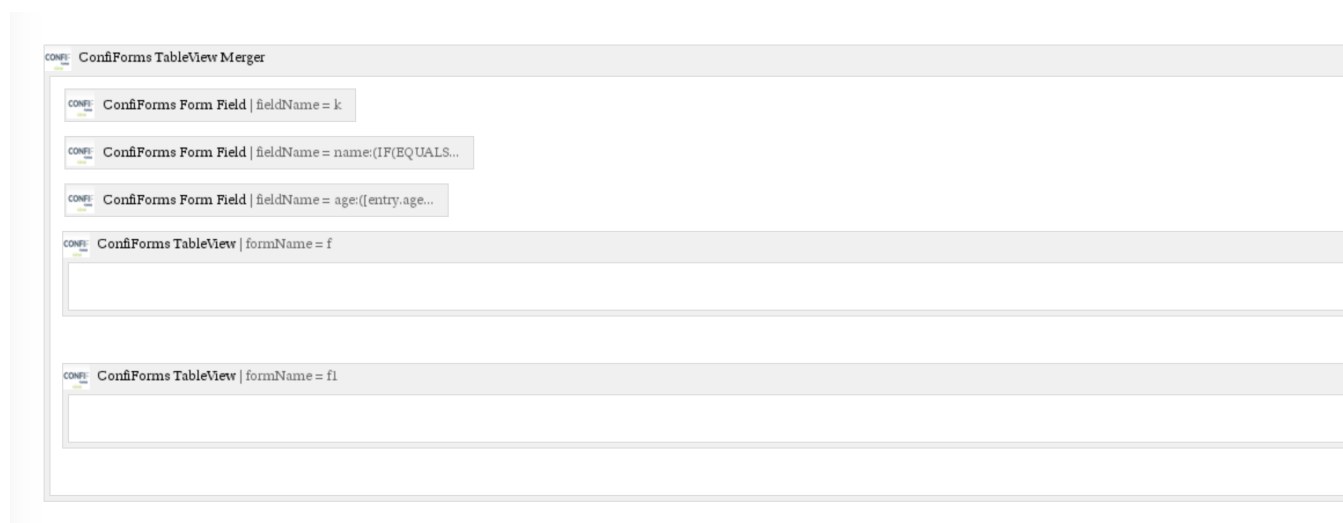
Below is the storage format for the solution:

```

<ac:structured-macro ac:macro-id="e98ccd03-5624-4344-ae4-33ca91263f0e" ac:name="confiform-table-merger" ac:
schema-version="1">
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="192bd752-15c5-4881-8343-efb7bcab064d" ac:name="confiform-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">k</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="591dlb6e-56fa-45b3-8824-ddf041cd5722" ac:name="confiform-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">name:(IF(EQUALS("[entry.name]", "0"), "", "[entry.name]))</ac:
parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="bc172d0e-4f4a-4dcb-bf19-eeaa934e6373" ac:name="confiform-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">age:([entry.age])</ac:parameter>
      </ac:structured-macro>
    </p>
    <ac:structured-macro ac:macro-id="c1467011-c101-4d31-a410-a2b5048b1d24" ac:name="confiform-table" ac:schema-
version="1">
      <ac:parameter ac:name="formName">f</ac:parameter>
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
    <ac:structured-macro ac:macro-id="b97110c0-ee54-4ca9-b69c-fb4abec0d21" ac:name="confiform-table" ac:schema-
version="1">
      <ac:parameter ac:name="formName">f1</ac:parameter>
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  </ac:rich-text-body>
</ac:structured-macro>

```

And this is how it looks like in the Confluence editor



It is very important to have a little bit of transformation for the name column, as if we leave it like "age" column we will end up with 0, instead of empty values

```
name:(IF(EQUALS("[entry.name]", "0"), "", "[entry.name]"))
```

So, we use [Supported math operators, formulas and functions](#) to evaluate the field value (it evaluates to 0 if not found and we replace the 0 with empty value, like "")

If you want just to merger the data from 2 forms, and enable edits over the data, you should not have any expressions in your fields. This way the data is merged together (and not aggregated), which enables support for edits.

Data is shown on own row, as it is stored in respected forms

See below:

k	name	age	Edit	Show
k1		20	Edit Delete	Show
k1	sash		Edit Delete	Show

With configuration like you see below:

```
<ac:structured-macro ac:macro-id="a10d8b74-c01f-434a-9aa7-984ef5887f5a" ac:name="confiorm-table-merger" ac:
schema-version="1">
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="d105acba-7d72-4a66-b256-961157d2e005" ac:name="confiorm-field" ac:
schema-version="1">
        <ac:parameter ac:name="hideIfEmpty">true</ac:parameter>
        <ac:parameter ac:name="fieldName">id</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="2d89f9e7-9985-4288-a29d-745e81425aa3" ac:name="confiorm-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">k</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <br/>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="c67b5b70-f48c-4c10-9525-c47e1a4378c6" ac:name="confiorm-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">name</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="e27a54e5-a29c-48e1-8b04-df79ae3ac8ac" ac:name="confiorm-field" ac:
schema-version="1">
        <ac:parameter ac:name="fieldName">age</ac:parameter>
      </ac:structured-macro>
    </p>
    <ac:structured-macro ac:macro-id="6f11658c-fb26-457e-9d57-ee7e101b7a4b" ac:name="confiorm-entry-edit" ac:
schema-version="1">
      <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
      <ac:rich-text-body>
        <p>
          <br/>
        </p>
      </ac:rich-text-body>
    </ac:structured-macro>
  </ac:rich-text-body>
</ac:structured-macro>
```

```

    </ac:rich-text-body>
  </ac:structured-macro>
  <ac:structured-macro ac:macro-id="d8bce7e0-dc6b-439c-9c9a-71ab60877b8c" ac:name="conform-entry-viewer" ac:
schema-version="1">
    <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
    <ac:rich-text-body>
      <p>
        <br/>
      </p>
    </ac:rich-text-body>
  </ac:structured-macro>
  <p>
    <br/>
  </p>
  <p>
    <br/>
  </p>
  <ac:structured-macro ac:macro-id="fcb79807-4c55-4ee0-bb8f-62b1c472835d" ac:name="conform-table" ac:schema-
version="1">
    <ac:parameter ac:name="formName">f1</ac:parameter>
    <ac:rich-text-body>
      <p>
        <br/>
      </p>
    </ac:rich-text-body>
  </ac:structured-macro>
  <ac:structured-macro ac:macro-id="c3341d60-eea3-452b-934c-21c3acd98d26" ac:name="conform-table" ac:schema-
version="1">
    <ac:parameter ac:name="formName">f</ac:parameter>
    <ac:rich-text-body>
      <p>
        <br/>
      </p>
    </ac:rich-text-body>
  </ac:structured-macro>
  <p>
    <br/>
  </p>
  <p>
    <br/>
  </p>
  </ac:rich-text-body>
</ac:structured-macro>

```