Custom Forms Configuration

Video Tutorial on Custom Forms

https://www.youtube.com/embed/Zn4NepG0Uhs

Use cases

There are 3 basic use cases (and 1 deprecated), how does the bussiness object hold the reference to the data in custom form structure. There is a reason behind the diffenrence and it is important to understand them.

A) formvalues id is stored in the object

The main and basic use case is storing the ID of formvalues in the bussiness object, so when some client needs to show form values of the object, it just reads the formvalue id from the object and then call API method ListItemValue forminfo = formvalueid. The field in the object has usually DB constraint to formvalues table. This solution is best when it is essential feature of the object. Typically the formvalues are created and set into the object immidiatelly after its creation.

This way is used in folder, project, activtiy - they all have formvaluesid field, so when you get folder detail, you can ask for form item values by this id.

This is the only way, which is also supported in reporting, also the only way how to get these information into timeline.

B) remoteid is used to the object

Another way how to bind form data to an object is to save the ID of the object into remoteid field in formvalues. This allows us storing the reference to the form without actual need for the object moidfication. Sometimes we do not want to modify the structure of the object (it is not worth it) or we can not (pairing form values with external objects, which we can not modify). In such a case, we just set the ID of the object into field remoteid. When doing this, it is also mandatory to fill field remote_id_type, so later on we know, what kind of object does the formvalues belong to (tasks for example have com.atollon.task, if I want to pair the form with some external subject - like linkedin profile, structure in external system - we use linkedin.profile etc.).

C) combination of context and formid

This is very specific usage and it is used very rarely. The problem which this addresses is, that sometimes you need more customforms on one context. In such a scenario, as unique reference for form values you can use combination of context and formid. See example bellow:

The client needed to save dynamical custom form on the applicant. The form was his accomodation history and for each accomodation they needed to save few basic data. So for this scenario we implemented the custom app, where you configure formID.

So the client creates new form by calling CreateCustomForm with formID and context, and then fills this values with new id from form values. So when displaying the existing forms, client list all formvalues by context + formID and then for each form value render the form as usual (so the form then ask formitemvalues by formvalue id as forminfo).

D) forminfo directly contains the id of the object !DEPRECATED!

We used to have 2 types of forms on folder/project/activity. One by template (case A) and one by type. That means that into formitemvalues we used to save directly ID of folder/project/activity. That means that those formitemvalues did not even have a formvalue. We dislike it and it is no longer supported. Theoretically, if we would like to have something like this nowdays, it would be better to use case B. But so far we do not need it and I do not see any usage in the near future.

Formvalues vs form

This paragraph with focus on why dataset for form (formvalues) is not same as form. todo

Atollon provides Forms functionality that allows Atollon administrators add new custom fields to existing Contact Folders, Simple Folders, Projects or Activities and Milestones.

Custom Forms Features

Custom Form Fields

Edit Field

You can add new Edit Field into Form to add simple text, date field, inteager or numeric field into Custom Form. Set it in Edit Fields variables: Constraint.

Checkbox

This field is used to set Yes/No data.

Memo Field

It is possible to add multi-line plain-text notes to Custom Form.

Listbox

This it the only multi-select component for fixed values. You can switch whether the Listbox is multi-select or simple-select.

Combo

Simple option selector for fixed values.

How to Create New Custom Form?

How to Assign Custom Form to Contact Type?

You can assign the Custom Form to Contact Type eithery by assigning it to Contact Folder Type or Contact Folder Template.

How to Assign Custom Form to Project?

You can assign the Custom Form to Project either by assigning it to Project Type or Project Template. Both Forms may be used at the same time.

How to Copy Custom Form?

1. You must create blank destination form (for create ACL)

2. Go to psql for get ID of used forms

```
select id,form_name from forms;
```

3. move copyFormIntoFormSql.sh into /tmp (Available down at Attachments)

4. Start backup-db (what if)

5. Edit copyFormIntoFormSql for name of target DB

6. Start Form copy as a asp user

```
sh copyFormIntoFormSql.sh soucre_form_id destination_form_id
```

Migration values under proper formValues identifier

The new features, like timeline, Context Dashboard component etc. need values to be saved under formValuesId in all cases (in Activity, it used to be directly under the id of Activity). So if we want to enable new features (and generally migrate the data to newer version, since the old way wont be supported anymore), we need to fullfill next steps:

1. Make sure, every Folder, Project and Activity has its own formValues. This is by new function wrote by Zima
   <wsdl:CreateMissingFormValuesFPA>
   <server>$INSTANCE</server>
   <session>$SESSION</session>
   </wsdl:CreateMissingFormValuesFPA>
   But this function will only fix 10 folders, 10 projects and 10 activities (because of the timeout). So we need to call it more times, here is proposal for bash solution:#!/bin/bash

```
#usage:  $ SESSION=6C8DB8900FCE6B9E93908A9339C365386D357BA13508 INSTANCE=mbluetest ./fv.sh
for ((i=1;i<=1000;i++));
do
   echo $i
   cat << EOF | curl --header "Content-Type: text/xml;charset=UTF-8" --header
"SOAPAction: atollon#CreateMissingFormValuesFPA" -k --data @- https://`hostname`/soap

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:wsdl="http://atollon.com/enterprise/wsdl">
   <soapenv:Header/>
   <soapenv:Body>
      <wsdl:CreateMissingFormValuesFPA>
	      <server>$INSTANCE</server>
	      <session>$SESSION</session>
      </wsdl:CreateMissingFormValuesFPA>
   </soapenv:Body>
</soapenv:Envelope>
```

```
EOF

done
```

2. Migrate the values of needed context. We have three different SQL queries for this, one for folder, one for project and one for activity. This one is for folder

```
-- Folder
UPDATE formitemvalues SET forminfo = tmp.s_formvalues
FROM (
SELECT spt.id,
       spt.name,
       s.id,
       s.name,
       s.formvalues s_formvalues,
       fiv.id fiv_id
FROM subpro_type spt
JOIN subject s ON s.subjecttype = spt.id
JOIN formitemvalues fiv ON fiv.forminfo = s.id
WHERE spt.id = 1360000
) as tmp
WHERE formitemvalues.id = tmp.fiv_id;


-- Project
UPDATE formitemvalues SET forminfo = tmp.p_formvalues
FROM (
SELECT spt.id,
       spt.name,
       p.id,
       p.name,
       p.formvalues p_formvalues,
       fiv.id fiv_id
FROM subpro_type spt
JOIN project p ON p.projecttype = spt.id
JOIN formitemvalues fiv ON fiv.forminfo = p.id
WHERE spt.id = 4912918000
) as tmp
WHERE formitemvalues.id = tmp.fiv_id;

-- Activity
UPDATE formitemvalues SET forminfo = tmp.a_formvalues
FROM (
SELECT spt.id,
       spt.name,
       at.id,
       at.name,
       a.formvalues a_formvalues,
       fiv.id fiv_id
FROM subpro_type spt
JOIN activity a ON a.activitytype = spt.id
JOIN tree at ON at.id = a.id
JOIN formitemvalues fiv ON fiv.forminfo = a.id
WHERE spt.id = 948362000 -- ?
) as tmp
WHERE formitemvalues.id = tmp.fiv_id;
```

All of those scripts are migrating values only for a specific folder/project/actiivty type. To change all values, here is a script:

```
UPDATE formitemvalues SET forminfo = tmp.a_formvalues
FROM (
SELECT
      a.formvalues a_formvalues,
      fiv.id fiv_id
FROM activity a
JOIN formitemvalues fiv ON fiv.forminfo = a.id
) as tmp
WHERE formitemvalues.id = tmp.fiv_id;
```

3. The last part is not necesarry, depends on finder component we are using to present the form. If it is formView, the ID of form is directly in the configuration of component, so we do not need to migrate formId. But in case we are using calssical project detail or folder detail or component, which uses the formId from the folder/project/activity, we need to make sure all formIds are properly configured. That means somehting like:

```
UPDATE activity set formid = X where activitytype = Y -- X is formID, Y is id of ActivityTy
```

There was a problem on some data, that the date format is broken (historically) and when we try to proceed the update of formValues, the new constraint on the database prevents us from moving the values. So you also need to convert the dates.
So when during UPDATE formitemvalues SET forminfo -- type 102 fails on "formitemvalues.value format check failed", we need to
convert formitemvalues.value date to format YYYY-MM-DD 00:00:00

```
begin;
UPDATE formitemvalues SET value = value::timestamp::date::text || ' 00:00:00'
FROM (
SELECT fiv.id
FROM formitemvalues fiv
LEFT JOIN formitems fi ON fi.id = fiv.formitem
WHERE NOT(CASE
              WHEN fi.typ = 101 THEN fiv.value ~ E'^-?\\d*\\.?\\d+$'
              WHEN fi.typ = 102 THEN fiv.value ~ E'^(?:[1-9]\\d{3}-(?:(?:0[1-9]|1[0-2])-
(?:0[1-9]|1\\d|2[0-8])|(?:0[13-9]|1[0-2])-(?:29|30)|(?:0[13578]|1[02])-31)|(?:[1-
9]\\d(?:0[48]|[2468][048]|[13579][26])|(?:[2468][048]|[13579][26])00)-02-29)[ T]00:00:00$'
              ELSE TRUE
          END)
AND fi.typ = 102
) as tmp
WHERE formitemvalues.id = tmp.id;

SELECT fiv.value, fi.typ, fi.id
FROM formitemvalues fiv
LEFT JOIN formitems fi ON fi.id = fiv.formitem
WHERE NOT(CASE
             WHEN fi.typ = 101 THEN fiv.value ~ E'^-?\\d*\\.?\\d+$'
             WHEN fi.typ = 102 THEN fiv.value ~ E'^(?:[1-9]\\d{3}-(?:(?:0[1-9]|1[0-2])-(?:0[1-
9]|1\\d|2[0-8])|(?:0[13-9]|1[0-2])-(?:29|30)|(?:0[13578]|1[02])-31)|(?:[1-
9]\\d(?:0[48]|[2468][048]|[13579][26])|(?:[2468][048]|[13579][26])00)-02-29)[ T]00:00:00$'
             ELSE TRUE
         END);
rollback;
--commit;
```

Troubleshooting

Some records ain't displaying in report results

It may happen due data inconsistency when someone change form on folder.

How do I know?

This returns non-zero result

```
select formvalues.id, formitems.formid, formvalues.form from formitemvalues left join
formitems on formitemvalues.formitem = formitems.id left join formvalues on
formitemvalues.forminfo = formvalues.id where formitems.formid != formvalues.form group by
formvalues.id, formitems.formid, formvalues.form;
```

How do I fix?

With this hopefully

```
update formvalues set form = smt.formid from (select formvalues.id, formitems.formid,
formvalues.form from formitemvalues left join formitems on formitemvalues.formitem =
formitems.id left join formvalues on formitemvalues.forminfo = formvalues.id where
formitems.formid != formvalues.form group by formvalues.id, formitems.formid, formvalues.form)
as smt where formvalues.id = smt.id;
```

Initialise a newly added form field for old objects created before

```
oknostudio=# select * from formitems where name like 'Zpln%';;
-[ RECORD 1 ]--------+-----------------
id                   | 517877101
name                 | ZplnomocnenValue
idinform             | 666
caption              |
mandatory            | 0
readonly             | 0
taborder             | 0
defaultvalue         |
pos_left             | 0
pos_top              | 0
width                | 0
height               | 0
editmask             |
parent               | 0
typ                  | 5
valuesfrommodule     | 0
valuesfrommoduleid   |
valuesfrommoduleacl  |
valuesfrommodulename |
modulename           |
modulefunc           |
modulerowname        |
listboxvalues        |
buttontype           | 0
listmultiselect      | 0
componentname        |
formid               | 511358101
fontcolor            |
componentcolor       |
font                 |
savevalue            | 1
restriction          | 0
componentproperty    |
dataprovider         |
dataprovidertype     |
treehandle           | 131021101
```

```
oknostudio=# select * from subpro_type where name like 'Smlouva OKN';
-[ RECORD 1 ]---+-----------------
id              | 405077101
aclh            | PROJ000000067072
treenode        | 131025101
name            | Smlouva OKN
comment         |
itype           | 3
masterproject   | f
guiaction       |
defaulttemplate |
onlytypes       | <PROJECTTYPES/>
subjectlinktype | 1
masterrequired  | f
formid          |
numbering       | 377875101
copyform        | 0
copyitem        | 0
dimension       | 0
fixtype         | 0
require_refid   | f
systemtype      | 7000
categories      | {}
```

```
oknostudio=# select * from formitemvalues where formitem = 517877101;;
     id     | formitem |  forminfo  |         value
```

```
------------+-----------+------------+----------------------
 1041368101 | 517877101 | 1024718101 |
 1041358101 | 517877101 | 1027547101 |
 1041287101 | 517877101 | 1041274101 |
 1040659101 | 517877101 | 1040609101 |
 1040094101 | 517877101 | 1040052101 |
 1039934101 | 517877101 | 1039879101 |
 1036331101 | 517877101 | 1032946101 | p.Karel Dupal
 1036090101 | 517877101 | 1032978101 | p. Karel Dupal
 1035348101 | 517877101 | 1008257101 |
 1035018101 | 517877101 |  903811101 | sdfksdjflksdjklfjdsk
  545489101 | 517877101 |  517579101 | Petr Novák
  553356101 | 517877101 |  553333101 |
(12 rows)
```

Revision #3
Created 23 January 2020 14:13:39 by Jan Safka
Updated 26 July 2021 17:36:51