

How to setup a form which copies (backups) it's content to another form (with files)

i In this tutorial we will setup 2 data entry forms with ConfiForms on 2 different pages, where one form will be just a proxy, and the data (including attachments) will be passed to another form. This is done using ConfiForms IFTTT macro with action to "Create ConfiForms Entry"

The data is passed to another form (and deleted from original) only when:

- we set a field "Backup?" to true

See form configuration below for fields and types

! 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 form configuration (we will have 2 forms on different pages. Different pages is important here to demonstrate how the attachment /file type get's transferred)

So, a form with 4 fields

- text field
- another text field
- file type field
- checkbox type field (this will be the condition to backup or not the contents)

Form in page edit mode:

CONFI: ConfiForms Form | formName = f

CONFI: ConfiForms Field Definition | fieldName = t | fieldLabel ...

CONFI: ConfiForms Field Definition | fieldName = t1 | fieldLabel...

CONFI: ConfiForms Field Definition | fieldName = backup | fieldL...

CONFI: ConfiForms Field Definition | fieldName = f | fieldLabel ...

CONFI: ConfiForms Registration Control

In view mode the form looks like this

Registration

TextField 1

Another TextField

Backup?

☐

File

No file chosen



[Close](#)

Full solution (storage format is here)

```
<ac:structured-macro ac:macro-id="19608453-440e-4c19-8b2a-db952f81f567" ac:name="conform" ac:schema-version="1">
  <ac:parameter ac:name="formName">f</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="807fa60b-7cbe-4fe3-99c8-900b10f841b6" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">t</ac:parameter>
        <ac:parameter ac:name="fieldLabel">TextField 1</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="ac51aec4-6715-4e9d-832f-1994d1095519" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">t1</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Another TextField</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="75e1f0e2-024f-4d89-a897-a40c4baaa03c" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">backup</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Backup?</ac:parameter>
        <ac:parameter ac:name="type">checkbox</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="e29cec22-5e91-4eac-9d1a-c8964ba6cd21" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">f</ac:parameter>
        <ac:parameter ac:name="fieldLabel">File</ac:parameter>
        <ac:parameter ac:name="type">file</ac:parameter>
      </ac:structured-macro>
      <ac:structured-macro ac:macro-id="1d943281-7e87-4ef8-8d41-b140a8f4d9f9" ac:name="conform-entry-register" ac:schema-version="1">
        <ac:parameter ac:name="atlassian-macro-output-type">INLINE</ac:parameter>
      </ac:structured-macro>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>
```

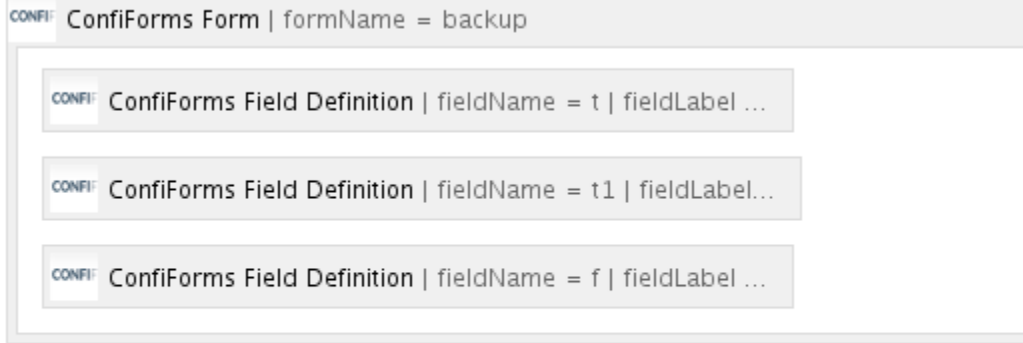
```

        <p> </p>
    </ac:rich-text-body>
</ac:structured-macro>
</p>
<ac:structured-macro ac:macro-id="3b48467f-e106-44e1-9669-7db873797b41" ac:name="confiorm-ifttt" ac:schema-
version="1">
    <ac:parameter ac:name="condition">backup:true</ac:parameter>
    <ac:parameter ac:name="action">Create ConfiForms Entry</ac:parameter>
    <ac:parameter ac:name="event">onCreated</ac:parameter>
    <ac:parameter ac:name="title">t=[entry.t]&t1=[entry.t1]</ac:parameter>
    <ac:parameter ac:name="who">backup:5341692</ac:parameter>
    <ac:rich-text-body>
        <p>
            <ac:structured-macro ac:macro-id="6177d529-fc27-4fdd-b575-ad5650f252eb" ac:name="confiorm-field" ac:
schema-version="1">
                <ac:parameter ac:name="fieldName">f</ac:parameter>
            </ac:structured-macro>
        </p>
    </ac:rich-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="162e023d-e833-4199-adc8-a4f092f127a0" ac:name="confiorm-ifttt" ac:schema-
version="1">
    <ac:parameter ac:name="condition">backup:true</ac:parameter>
    <ac:parameter ac:name="action">Create ConfiForms Entry</ac:parameter>
    <ac:parameter ac:name="event">onModified</ac:parameter>
    <ac:parameter ac:name="title">t=[entry.t]&t1=[entry.t1]</ac:parameter>
    <ac:parameter ac:name="who">backup:5341692</ac:parameter>
    <ac:rich-text-body>
        <p>
            <ac:structured-macro ac:macro-id="eccce14f-6aa8-4bf8-bf7f-6d5beadb89ee" ac:name="confiorm-field" ac:
schema-version="1">
                <ac:parameter ac:name="fieldName">f</ac:parameter>
            </ac:structured-macro>
        </p>
    </ac:rich-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="91d5efa6-8ab4-47c6-8f1d-9cc01a9c0321" ac:name="confiorm-ifttt" ac:schema-
version="1">
    <ac:parameter ac:name="condition">backup:true</ac:parameter>
    <ac:parameter ac:name="action">Delete ConfiForms Entry</ac:parameter>
    <ac:parameter ac:name="event">onCreated</ac:parameter>
    <ac:parameter ac:name="title">id:[entry.id]</ac:parameter>
    <ac:parameter ac:name="who">f:5341690</ac:parameter>
    <ac:rich-text-body>
        <p> </p>
    </ac:rich-text-body>
</ac:structured-macro>
<ac:structured-macro ac:macro-id="31e6bb23-b378-45a1-84a4-cf1050f33d96" ac:name="confiorm-ifttt" ac:schema-
version="1">
    <ac:parameter ac:name="condition">backup:true</ac:parameter>
    <ac:parameter ac:name="action">Delete ConfiForms Entry</ac:parameter>
    <ac:parameter ac:name="event">onModified</ac:parameter>
    <ac:parameter ac:name="title">id:[entry.id]</ac:parameter>
    <ac:parameter ac:name="who">f:5341690</ac:parameter>
    <ac:rich-text-body>
        <p> </p>
    </ac:rich-text-body>
</ac:structured-macro>
</ac:rich-text-body>
</ac:structured-macro>

```

Second form is very simple, just 3 fields:

- text field
- another text field
- file type field



Storage format for the second form

```

<ac:structured-macro ac:macro-id="55303a18-9a34-4f7f-a130-99fff66ced0f" ac:name="conform" ac:schema-version="1">
  <ac:parameter ac:name="formName">backup</ac:parameter>
  <ac:rich-text-body>
    <p>
      <ac:structured-macro ac:macro-id="f9f990de-a265-445f-83ff-142909b51ab3" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">t</ac:parameter>
        <ac:parameter ac:name="fieldLabel">TextField 1</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="4a8b19ea-a23c-45ea-b31a-53ae4297e108" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">t1</ac:parameter>
        <ac:parameter ac:name="fieldLabel">Another textfield</ac:parameter>
        <ac:parameter ac:name="type">text</ac:parameter>
      </ac:structured-macro>
    </p>
    <p>
      <ac:structured-macro ac:macro-id="1c8fb1b0-4696-4edc-b96b-efaa628230ee" ac:name="conform-field-definition" ac:schema-version="1">
        <ac:parameter ac:name="fieldName">f</ac:parameter>
        <ac:parameter ac:name="fieldLabel">f</ac:parameter>
        <ac:parameter ac:name="type">file</ac:parameter>
      </ac:structured-macro>
    </p>
  </ac:rich-text-body>
</ac:structured-macro>

```

Let's start with important bits about IFTTT setups, which will help us to implement this solution

For this 1st main form we will have **4 IFTTT handlers**

2 for each event, and we will track record creation and record updates. and will fire the IFTTT action only when "Backup?" checkbox field is checked

CONFIF: ConfiForms IFTTT Integration Rules | event = onCreated | action = Create ConfiForms Entry | condition = backup:true | title ...

CONFIF: ConfiForms Field | fieldName = f

CONFIF: ConfiForms IFTTT Integration Rules | event = onModified | action = Create ConfiForms Entry | condition = backup:true | title...

CONFIF: ConfiForms Field | fieldName = f

CONFIF: ConfiForms IFTTT Integration Rules | event = onCreated | action = Delete ConfiForms Entry | condition = backup:true | title ...

CONFIF: ConfiForms IFTTT Integration Rules | event = onModified | action = Delete ConfiForms Entry | condition = backup:true | title...

See first 2 IFTTT integration rules, they are identical but are binded to different events, their fire when condition

backup:true

is met, and they create a new record in our second form and, very important bit, they also say that field "f" should be included into action. Field "f" on our form is a file field type and this bit tells IFTTT action to transfer the File to a new place

Here is how the setup for those both IFTTT handlers looks like (they differ only in the event type they are binded to)

Edit 'ConfiForms IFTTT Integration Rules' Macro

Create / Update ConfiForms Ent

Fire IFTTT action only when this condition/filter is met

backup:true

If empty then IFTTT action is always executed when an event is occurred. [Same syntax as in filters](#)

Parameters to set on the entry

t=[entry.t]&t1=[entry.t1]

As key=value pairs, separated by &. Where key is the field name in the form where this entry should be created and value is the value to be set. Could be constructed dynamically. You can reference record owner as [owner]; record modifier, as [modifier]; any record field as [entry.FIELD_NAME], you can reference page watchers as [watchers]

Preview

Visible only in PREVIEW

How to configure ConfiForms IFTTT macro

The following fields are defined in the form **f** and can be referenced from IFTTT macro body using '\${FIELD_NAME}' (**deprecated now**) or using '[entry.field_name]' notation. See more on [ConfiForms wiki](#) for detailed help on how to reference complex field properties

- id
- createdBy
- ownedBy
- created
- dateCreatedFormatted
- t
- t1
- backup
- f

Select macro

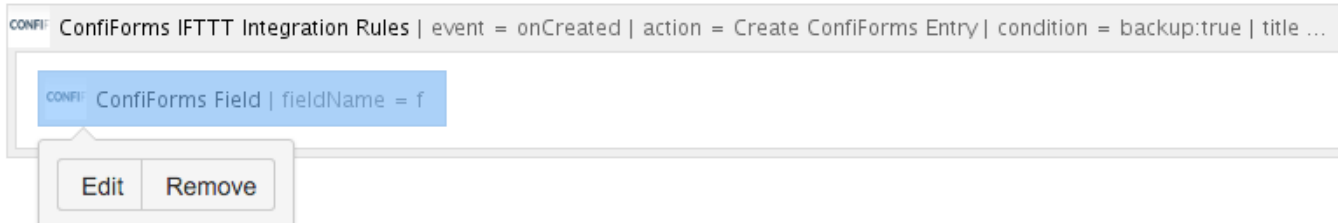
Save

Cancel

So, we fire this event only when "backup:true", meaning that checkbox to backup the record is set to "checked" and we copy values from original entry to the new one

```
t=[entry.t]&t1=[entry.t1]
```

And include the ConfiForms Field macro which references a file (in our case field type is called "f") inside the IFTTT macro body



Last 2 IFTTT Integration Rules macros deal with cleanup in the original form (also binded to 2 events: create and modify).

When the same condition is met the cleanup routine is executed

Edit 'ConfiForms IFTTT Integration Rules' Macro

Delete ConfiForms Entry

Fire IFTTT action only when this condition/filter is met

backup:true

If empty then IFTTT action is always executed when an event is occurred. [Same syntax as in filters](#)

Delete by filter

id:[entry.id]

Delete ConfiForms entry (or entries) by given filter. [Same syntax as in filters](#) The scope is all records in the form. Could be constructed dynamically. You can reference record owner as [owner]; record modifier, as [modifier]; any record field as [entry.FIELD_NAME], you can reference page watchers as [watchers]

Reference Form and its Location

Preview

Visible only in PREVIEW

How to configure ConfiForms IFTTT macro

The following fields are defined in the form **f** and can be referenced from IFTTT macro body using '**FIELD_NAME**' (**deprecated now**) or using '[entry.field_name]' notation. See more on [ConfiForms wiki](#) for detailed help on how to reference complex field properties

- id
- createdBy
- ownedBy
- created
- dateCreatedFormatted
- t
- t1
- backup
- f

Select macro

Save Cancel

So, the original record in original form get's deleted by given filter when "backup:true" condition is met



For IFTTT macro configurations which work with creating or deleting records in the form one very important configuration parameter is left unset
It is the form name and form's location (page)

So, scroll down for each IFTTT macro defined, and for the ones which copy the data the "Reference form and location" will be the backup form and it's location (pageId)

Edit 'ConfiForms IFTTT Integration Rules' Macro

If empty then IFTTT action is always executed when an event is occurred. [Same syntax as in filters](#)

Parameters to set on the entry

t=[entry.t]&t1=[entry.t1]

As key=value pairs, separated by &. Where key is the field name in the form where this entry should be created and value is the value to be set. Could be constructed dynamically. You can reference record owner as [owner]; record modifier, as [modifier]; any record field as [entry.FIELD_NAME], you can reference page watchers as [watchers]

Reference Form and its Location (pageId)

backup:5341692

Form and page ID separated by ":"
(Example: formName:pageId)

Preview

Visible only in

How to configure
The following fields from IFTTT macro or using '[entry.t1]' [wiki](#) for detailed

- id
- createdBy
- ownedBy
- created
- dateCreated
- t
- t1
- backup
- f

Select macro

For the IFTTTs which clean up the data in original form you will need to reference this form

Edit 'ConfiForms IFTTT Integration Rule'

If empty then IFTTT action is always executed when an event is occurred. [Same syntax as in filters](#)

Delete by filter

Delete ConfiForms entry (or entries) by given filter. [Same syntax as in filters](#) The scope is all records in the form. Could be constructed dynamically. You can reference record owner as [owner]; record modifier, as [modifier]; any record field as [entry.FIELD_NAME], you can reference page watchers as [watchers]

Reference Form and its Location (pageId)

Form and page ID separated by ":"
(Example: formName:pageId)

 Preview

[Select macro](#)

Where do you see the pageId?

The most easiest way is to go to the page edit mode and see the "pageId" parameter value in your browser bar

And the solution is ready!

First form takes inputs from a user, stores it inside it's own storage and when a backup is requested then the data is copied to another form (including files!) and deleted from original place