

Data Export tricks and hints

- [Export formats](#)
- [What about flattened views?](#)
- [What if I need internal or transformed data in CSV/Excel format?](#)
- [Customising exports via ConfiForms Views](#)
- [Exports are not working for some users](#)

Export formats

ConfiForms provides a way to export data in 4 different formats (or in 6 if you count flattened views as separate types)

- CSV (+flattened view) - comma separated values
- XLS (+flattened view) - Microsoft Excel format
- JSON - JSON format
- XML - dataset exported in an XML format

First 2 are for end-users mainly and second 2 are for further data processing, as it depends how the data for complex fields is exported. JSON and XML formats export internal representation of the data, that means you will see IDs for dropdowns, GUIDs for smart classifiers, page IDs for pages and blogposts, date and time fields exported as timestamps (epoch) and also IDs for database-backed fields

Let's consider an example. Here is a form with 3 fields

The screenshot shows a configuration interface for a ConfiForms form. At the top, a header bar reads "CONFI: ConfiForms Form | formName = f". Below this, a "CONFI: ConfiForms Registration Control" section contains a large empty text input field. Underneath, three "CONFI: ConfiForms Field Definition" blocks are listed, each with a label and a truncated ID: "mytf | fieldLab...", "mydf | fieldLab...", and "mydate | fieldL...".

- mytf - text field
- mydf - dropdown field (contains 3 choices)

ID	LABEL
1	one
2	two
3	three

- mydate - is of type date

Export is enabled by default in the Admin UI for the form

[Register](#) [Form: f](#)

My text field	My Dropdown Field	My date field
No data stored		

[CSV](#) | [XLS](#) | [JSON](#) | [XML](#) | [RAW](#) | Records: 0 Size: 7 B

We will add couple of records and will show how the exported result for each format look like

[Register](#) [Form: f](#)

My text field	My Dropdown Field	My date field	
confirms tutorial	three	Jun 07, 2016	Edit Delete
hello export	one	Jun 04, 2016	Edit Delete

[CSV](#) | [XLS](#) | [JSON](#) | [XML](#) | [RAW](#) | Records: 2 Size: 725 B

Exported as CSV

```
Id, Created By, Owned By, Created, Created, My text field, My Dropdown Field, My date field  
f4ea4796-0dd2-455c-a1b4-926209260a02, sash, sash, 1465197019719, 2016-06-06 03:10, hello export, one, "Jun 04, 2016"  
66cee179-2f33-4fd1-b8b8-bb889043e06f, sash, sash, 1465197038331, 2016-06-06 03:10, confirms tutorial, three, "Jun 07, 2016"
```

Exported as XLS



Exported as JSON

```

{
  "list": {
    "entry": [
      {
        "recordId": 1,
        "deleted": 0,
        "createdBy": "sash",
        "created": 1465197019719,
        "id": "f4ea4796-0dd2-455c-alb4-926209260a02",
        "fields": {
          "mydate": 1465012800000,
          "class": "fields",
          "mytf": "hello export",
          "mydf": 1
        },
        "ownedBy": "sash"
      },
      {
        "recordId": 2,
        "deleted": 0,
        "createdBy": "sash",
        "created": 1465197038331,
        "id": "66cee179-2f33-4fd1-b8b8-bb889043e06f",
        "fields": {
          "mydate": 1465272000000,
          "class": "fields",
          "mytf": "confifirms tutorial",
          "mydf": 3
        },
        "ownedBy": "sash"
      }
    ]
  }
}

```

Exported as XML

```

<list>
  <entry>
    <id>f4ea4796-0dd2-455c-alb4-926209260a02</id>
    <recordId>1</recordId>
    <ownedBy>sash</ownedBy>
    <createdBy>sash</createdBy>
    <created>1465197019719</created>
    <deleted>0</deleted>
    <fields class="fields">
      <mytf>hello export</mytf>
      <mydf>1</mydf>
      <mydate>1465012800000</mydate>
    </fields>
  </entry>
  <entry>
    <id>66cee179-2f33-4fd1-b8b8-bb889043e06f</id>
    <recordId>2</recordId>
    <ownedBy>sash</ownedBy>
    <createdBy>sash</createdBy>
    <created>1465197038331</created>
    <deleted>0</deleted>
    <fields class="fields">
      <mytf>confifirms tutorial</mytf>
      <mydf>3</mydf>
      <mydate>1465272000000</mydate>
    </fields>
  </entry>
</list>

```

In the examples above you can see the difference in how the data differs (for non-text fields, see "mydf" and "mydate" fields)



Form Admin UI also provides a way to see the structure of the form dataset, providing RAW data format.

This is mainly used for troubleshooting and data recovery

What about flattened views?

These are used for forms which have multi-select fields and you want to export each choices as aseparate row

Consider the following example

|

The screenshot shows a configuration view for a form. At the top, there is a header bar with the text "CONFI: ConfiForms Form | formName = f". Below this, there is a large empty rectangular area. Underneath that, there are four separate configuration blocks, each starting with "CONFI:" and followed by a label and a field name. The labels are "ConfiForms Registration Control", "ConfiForms Field Definition | fieldName = mytf | fieldLab...", "ConfiForms Field Definition | fieldName = mydf | fieldLab...", and "ConfiForms Field Definition | fieldName = mydate | fieldL...". The last block is partially cut off at the bottom.

We have added one more field of type multi-select into the form (described above)

It is called "mymf" and of type "Multi-select"

Edit 'ConfiForms Field Definition' Macro

Field name *

 Must be unique within the form defined and alphanumeric

Field label *

Required *

Field type

id	value
A1	select A <input type="text" value="select A"/> <input type="button" value="✎"/> <input type="button" value="✕"/>
B2	select B <input type="text" value="select B"/> <input type="button" value="✎"/> <input type="button" value="✕"/>
	select C

Select macro

We changed our dataset to

My text field	My Dropdown Field	My date field	My multi field	
confiforms tutorial	three	Jun 07, 2016	<input type="text" value="select A"/> <input type="text" value="select C"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
hello export	one	Jun 04, 2016	<input type="text" value="select B"/>	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

[CSV \(flatten\)](#) | [XLS \(flatten\)](#) | [JSON](#) | [XML](#) | [RAW](#) | Records: 2 Size: 991 B

It still contains 2 fields and the exported view as CSV looks like this:

```
Id, Created By, Owned By, Created, Created, My text field, My Dropdown Field, My date field, My multi field
f4ea4796-0dd2-455c-alb4-926209260a02, sash, sash, 1465197019719, 2016-06-06 03:10, hello export, one, "Jun 04, 2016",
select B
66cee179-2f33-4fd1-b8b8-bb889043e06f, sash, sash, 1465197038331, 2016-06-06 03:10, confiforms tutorial, three, "Jun
07, 2016", select A select C
```

However, when "flattened view" is requested it will split the choices of multi-select field into separate rows

```
Id, Created By, Owned By, Created, Created, My text field, My Dropdown Field, My date field, My multi field
f4ea4796-0dd2-455c-alb4-926209260a02, sash, sash, 1465197019719, 2016-06-06 03:10, hello export, one, "Jun 04, 2016",
select B
66cee179-2f33-4fd1-b8b8-bb889043e06f, sash, sash, 1465197038331, 2016-06-06 03:10, confiforms tutorial, three, "Jun
07, 2016", select A
66cee179-2f33-4fd1-b8b8-bb889043e06f, sash, sash, 1465197038331, 2016-06-06 03:10, confiforms tutorial, three, "Jun
07, 2016", select C
```

See last 2 rows, they are practically identical with one exception, last field: each selection for multi-select field is on its own row

What if I need internal or transformed data in CSV/Excel format?

This is also possible, but will require a bit of custom work

Consider how the link to export data is constructed in form AdminUI (but also in other views where export is enabled)

```
Version 2.18.15 and later  
https://wiki.vertuna.com/plugins/servlet/confiform/export?t=csv&pageId=8159450&fd=f:8159450&fields=&filter=
```

```
Version 2.18.14 and older  
https://wiki.vertuna.com/ajax/confiforms/export.action?t=csv&pageId=8159450&fd=f:8159450&fields=&filter=
```

It expects:

Parameter name	Value
t	Could be: <ul style="list-style-type: none">• csv• flatten.csv• xml• json• xls• text• flatten.xls
pageId	Page Id of the page where ConfiForms is configured
fd	form name and again page ID where form is defined (some backward compatibility parameter, but still is required). It should be given in a format formname:pageId In the example above we have a form named "f" and it is located on the page with id "8159450"

<p>fields</p>	<p>Field names to export. This is interesting, as you can limit the fields in your export and also use the techniques from Accessing field values and properties to customize the export.</p> <p>For example:</p> <p>to export just user-defined fields, without metadata fields</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>Version 2.18.15 and later https://wiki.vertuna.com/plugins/servlet/confiform/export?t=csv&pageId=8159450&fd=f:8159450&fields=mytf,mydf,mydate,mymf&filter=</p> <p>Version 2.18.14 and older https://wiki.vertuna.com/ajax/confiforms/export.action?t=csv&pageId=8159450&fd=f:8159450&fields=mytf,mydf,mydate,mymf&filter=</p> </div> <p>result:</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>My text field,My Dropdown Field,My date field,My multi field hello export,one,"Jun 04, 2016",select B confiforms tutorial,three,"Jun 07,2016",select A select C</pre> </div> <p>to export date field as timestamp in CSV format and dropdown field's ID (not label)</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <p>Version 2.18.15 and later https://wiki.vertuna.com/plugins/servlet/confiform/export?t=csv&pageId=8159450&fd=f:8159450&fields=mydf.id,mydate.timestamp&filter=</p> <p>Version 2.18.14 and older https://wiki.vertuna.com/ajax/confiforms/export.action?t=csv&pageId=8159450&fd=f:8159450&fields=mydf.id,mydate.timestamp&filter=</p> </div> <p>result:</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>mydf.id,mydate.timestamp 1,146501 3,146527</pre> </div> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0; background-color: #e6f2ff;"> <p> For fields, you can use Virtual functions and format or transform the values as needed!</p> </div> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0; background-color: #ffe6e6;"> <p> Please note that functions that use <i>commas</i> (,) in the parameters could not be used at the moment</p> </div>
<p>filter</p>	<p>Same as ConfiForms Filters. To limit the exported dataset</p>
<p>headers</p>	<p>Since ConfiForms version 2.15.3 you can supply the "headers" parameter as comma separated list of names to use for headers in exported CSV/Excel file</p> <p>Quick example:</p> <div style="border: 1px solid #ccc; padding: 5px; margin: 5px 0;"> <pre>headers=My ID,Column2</pre> </div>
<p>sort</p> <p>SINCE 2.18.19</p>	<p>Sorting in ConfiForms</p>
<p>limit</p> <p>SINCE 2.18.19</p>	<p>Numeric value to limit the number of records returned. Missing parameter or limit=0 sets the limit to "no limit"</p>

Customising exports via ConfiForms Views

Same customizations as explained in the table above for "**fields**" parameter could be managed/configured on ConfiForms views (TableView, CardView, ListView, etc) macros using

Insert 'ConfiForms TableView' M

ConfiForms data in CSV, XLS, XML and JSON format

Comma separated list of fields to export

You can put rich properties, When left empty the default list fields list is taken

Exports are not working for some users

For form non-admin users (see here who is form administrator and who is not - [Confluence page permissions and ConfiForms](#)) you need to explicitly allow data exports. This is a security measure to prevent forms data from being exported by anyone when you don't want to.

To enable exports for non admin users you need to allow exports in the Form Definition for your form

Insert 'ConfiForms Form Definition' Macro

Allow data export and enable search

If you want to enable data export and access via REST APIs for form's non-admin users (You will still need to set "Enable export" parameter on the TableView macro). This also enables form's data to be searchable via Confluence search

 Preview

And then you can enable the option to export the data in the corresponding views

For example, this is how you do it for the ConfiForms TableView

Insert 'ConfiForms TableView' Macro

values

Enable export

ConfiForms data in CSV, XLS, XML and JSON format

 **Preview**