

Jasper Reports Manual

Add-on version: 2.1.11

Description	3
Prerequisites	3
Configuration	3
Client-Side Configuration	3
Server-Side Configuration	4
Limitations	5
Installation	5
Uninstallation	6
Usage	6
Dashboard widgets export	6
Exporting a single widget using default dataset	9
Using template parameters	10
Connecting widget data with jasper template	11
Downloading report	12
Downloading report for all selected widgets	12
Downloading report for a single widget	13
Showing "No Results" for empty widgets	13
Page breaks in template	15
Add-On Architecture	15
Fonts Installation	16

Description

Jasper Reports add-on provides a way to export Pivot, Table or BloX widgets to PDF on a pixel perfect level and Excel

The Add-on generates reports based on an xml template containing report markup (created in [TIBCO Jaspersoft Studio](#)) and under the hood uses the JasperReports Library Community Edition

The Current version of The Add-on is compatible with Report Manager add-on version 1.11.4 and above.

Prerequisites

Users should be familiar with Jasper reports and be able to create Jasper JRXML templates using the [Jaspersoft Studio](#).

Configuration

The add-on can be configured using the next files:

- Client-side configuration: /plugins/jasperReports/config.6.js
- Server-side configuration: /external-plugins/apiPlugins/plugins/jasper-reports/v1/config.js

Client-Side Configuration

The following properties can be configured:

replaceOrigExportWithJasper - when the value is TRUE, there will be no additional “Jasper download” menu created on the dashboard or widget level and the native export to pdf menu item will produce a Jasper generated report.

useDefaultDataset - indicates if default dataset should be used, data from default dataset is available only in the Details band of report. See more details about this property in the [Exporting a single widget](#) chapter of this document.

Server-Side Configuration

The following properties can be configured:

tempFilePath - the path to folder containing final reports and other temporary files needed for export

exportServerConf - java server configuration used for export

host - java server host

port - java server port,

threadsNumber - the number of reports that can be generated simultaneously (should whether match the number of cpu cores available for "external-plugins" pod OR be less)

javaRuntime - java runtime config

xms - initial heap size

mxm - maximum heap size

statusServerConf - NodeJS server configuration used for tracking export statuses

host - server host name

port - server port number

Example configuration file:

```
1  module.exports = {
2    tempFilePath: '/opt/sisense/storage/external-plugins/tmp',
3    exportServerConf: {
4      host: 'localhost',
5      port: 55111,
6      threadsNumber: 1,
7      javaRuntime: {
8        xms: '1024m',
9        mxm: '4096m',
10     },
11   },
12   statusServerConf: {
13     host: 'localhost',
14     port: 55110,
15   },
16 };
```

In most cases you will not need to modify the configuration and use the default one.

Note: For optimal performance of the jasper report generation we recommend having 2 cpu cores and 4GB memory dedicated to the external-plugins kubernetes pod. You can increase the memory in the kubernetes deployment configuration yaml file. Ask your system administrator to help. After these changes you can set **threadsNumber** equal to 2 and **javaRuntime.xmx** equal to '4096m' to increase performance for concurrent reports generation.

Limitations

- The add-on is supported only for Pivot, Table, and BloX widgets.
- Data formatting that is set on the widget level is not presented in the exported file. If you need to apply formats (like currency, date format etc.), you can do it on the Jasper template itself using the JasperSoft Studio.
- Uploading and configuring templates is not available:
 - for users with role "viewer";
 - if a dashboard is shared with the user/group with "Can View" rule.
- In case of replacing original PDF export with Jasper - scheduling is not supported (for scheduled reports the original PDF export is used).
- Exporting a dashboard configured with multiple templates doesn't support pagination in the final report because each template is exported to a separate report and then all the reports are being concatenated.

Installation

1. Open a bash window
2. Copy the Sisense.JasperReports.zip file to the Linux server
3. Run the command below to unarchive the Add-on installation file:
- unzip Sisense.JasperReports.zip
4. Go to folder with scripts:
- cd Sisense.JasperReports.{version}/scripts
5. Run script with one of the keys described below in the Script Flags Description chapter.
For the first time installation please use -i flag.
For the upgrade of installed add-on use -u flag. (Please be aware that configuration backup will be made during upgrade)
 - All required pods will be restarted automatically. This may take a few minutes.
6. Refresh the dashboard

Script Flags Description

- a. To run script command use:
 - `bash ./l-addon-script.sh {command-key}` - replace {command-key} with one of the keys listed from -h command.
- b. Show available commands:
 - `bash ./l-addon-script.sh -h` - shows a list of available command keys and their description.
- c. Common commands:
 - `bash ./l-addon-script.sh -i` - install add-on or rewrite existing add-on files.
 - `bash ./l-addon-script.sh -r` - remove add-on from the system. Configuration in Mongo and the Configuration page will persist.
 - `bash ./l-addon-script.sh -u` - update the currently installed add-on by removing it and then installing a new version.

Uninstallation

To remove the plugin:

1. Open a bash window
2. Go to folder with scripts (you should be in a folder with add-on files):
- cd Sisense.JasperReports.{version}/scripts
3. To uninstall please run:
- bash ./l-addon-script.sh -r
All required pods will be restarted automatically. This may take a few minutes.

Usage

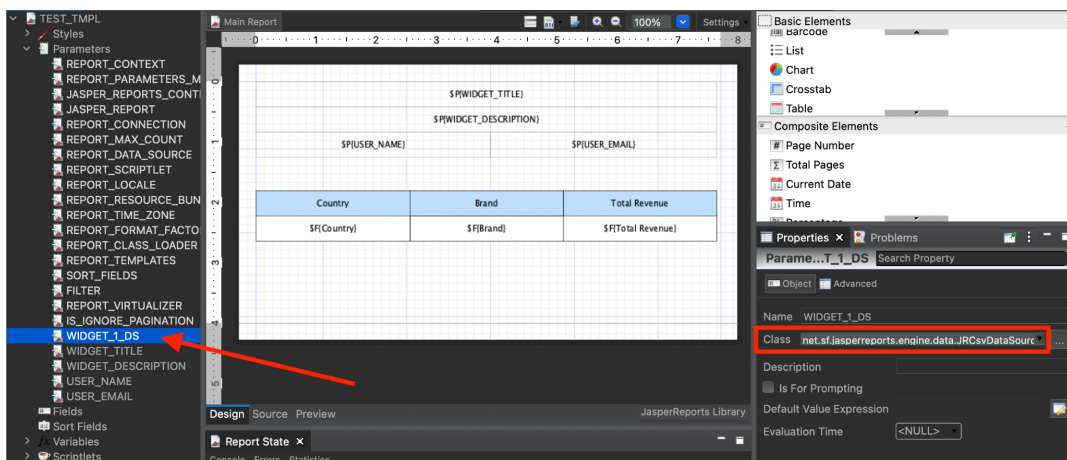
Jasper templates are created using Jaspersoft Studio. You can find the community edition of the Jaspersoft Studio [here](#).

The Jasper Reports add-on sends some of the widgets (title, description) and user information and the widgets data to the Jasper library.

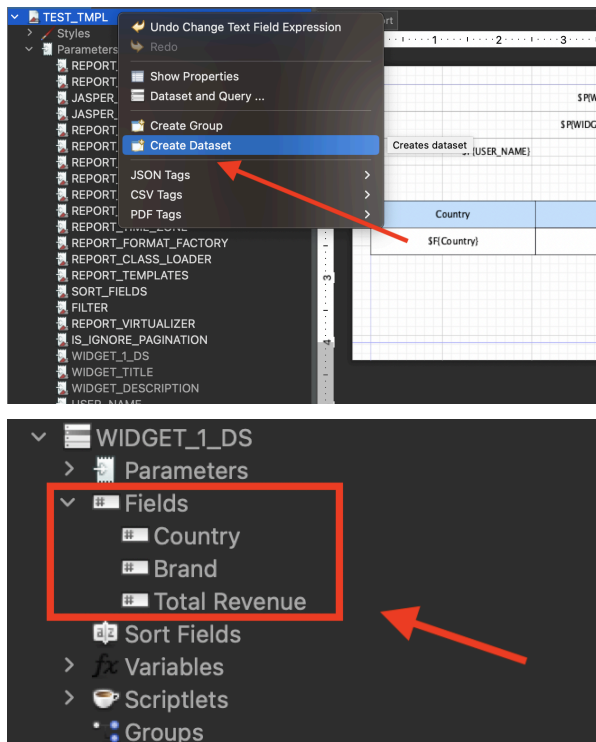
Dashboard widgets export

The Add-on allows you to pick some of the widgets of a dashboard and use the widgets data during Jasper report generation. This is the main use case. It is suitable when we need to use data from multiple widgets in one template.

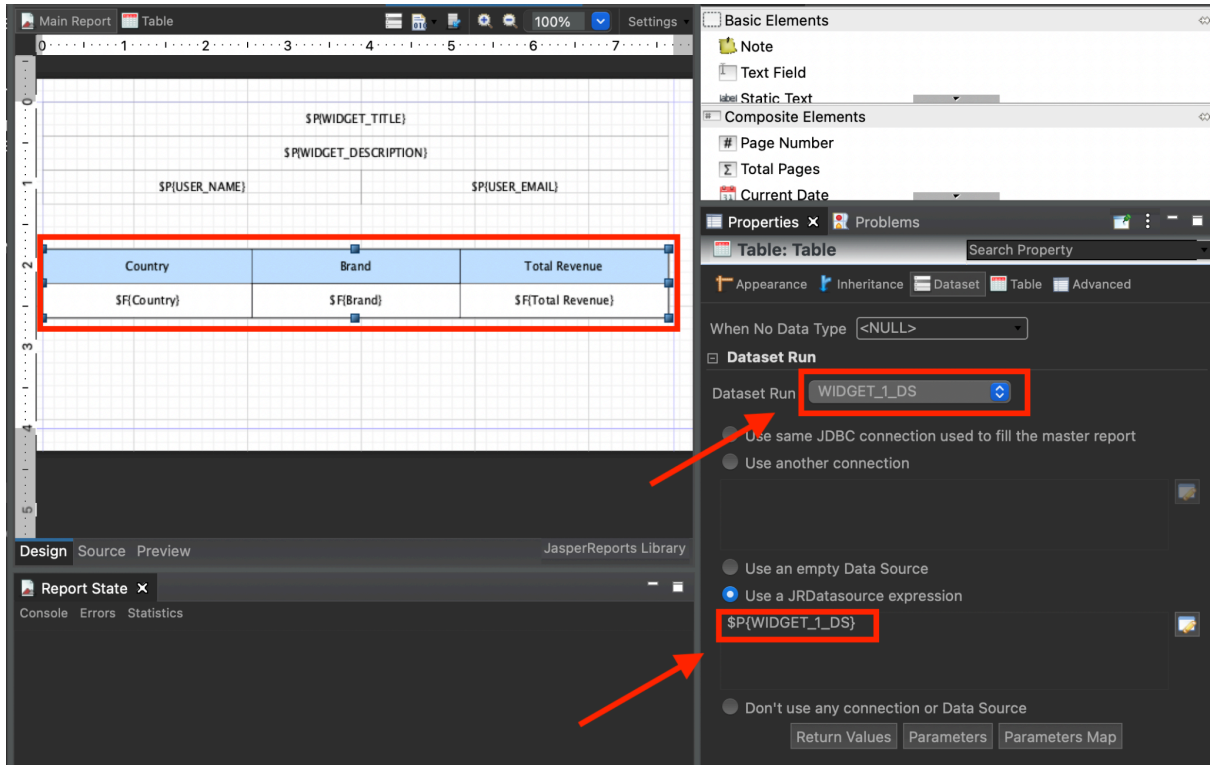
1. For each of the exported widgets, create corresponding parameter of type "net.sf.jasperreports.engine.data.JRCsvDataSource" in template root level "Parameters" section. During report exporting the actual widget CSV data will be stored in this parameter.



- For each of the exported widgets, create a corresponding dataset containing the same set of fields as the widget has. We can think of a dataset as the data provider for tables or charts. But the dataset itself doesn't contain any data. The actual data is fetched from the parameter created in step 1. So it's required for parameters and corresponding datasets to have the same names. The name can contain letters (lower and upper case), underscores and dashes only. Spaces are not allowed.

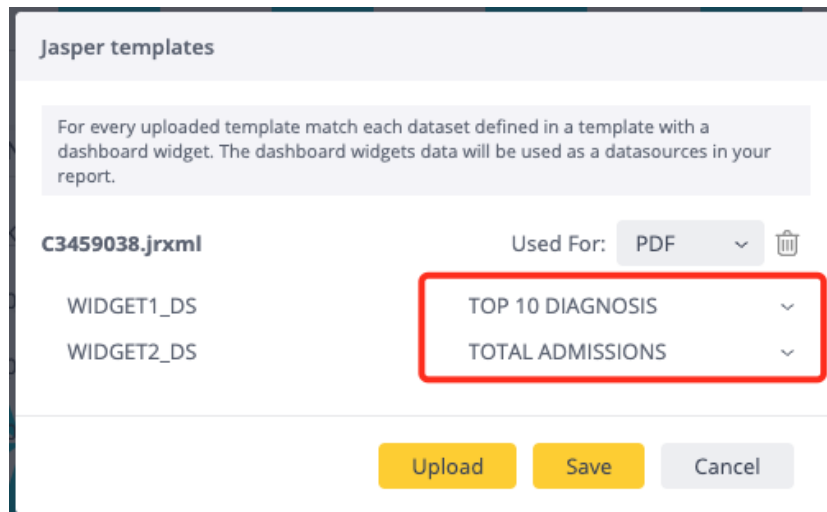


- For each of the table/chart Basic Elements, choose the corresponding dataset. Then select "Use a JRDataSource expression" and set the expression to be the corresponding parameter name.

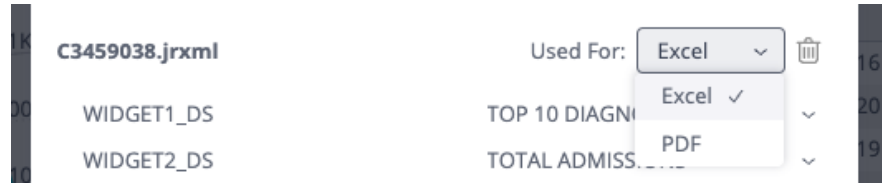


4. Upload template(s):

- a. Go to the target dashboard.
- b. Click on "Options" (three dots) icon in the dashboard header.
- c. In the dropdown menu click "Jasper Templates".
- d. In the popup click the "Upload" button and upload the template(s) one by one from your file system. Once the template is uploaded the Add-on parses it and shows the existing dataset(s)/parameter(s) names that you defined in the template.
- e. Match each existing dataset with the corresponding widget (only supported widget types appear in the dropdown of available widgets)



- f. By Default, template is used for PDF export, Select "Excel" option to use the template for the Excel export via Jasper.



- g. Click "Save" to save the configuration.

Exporting a single widget using default dataset

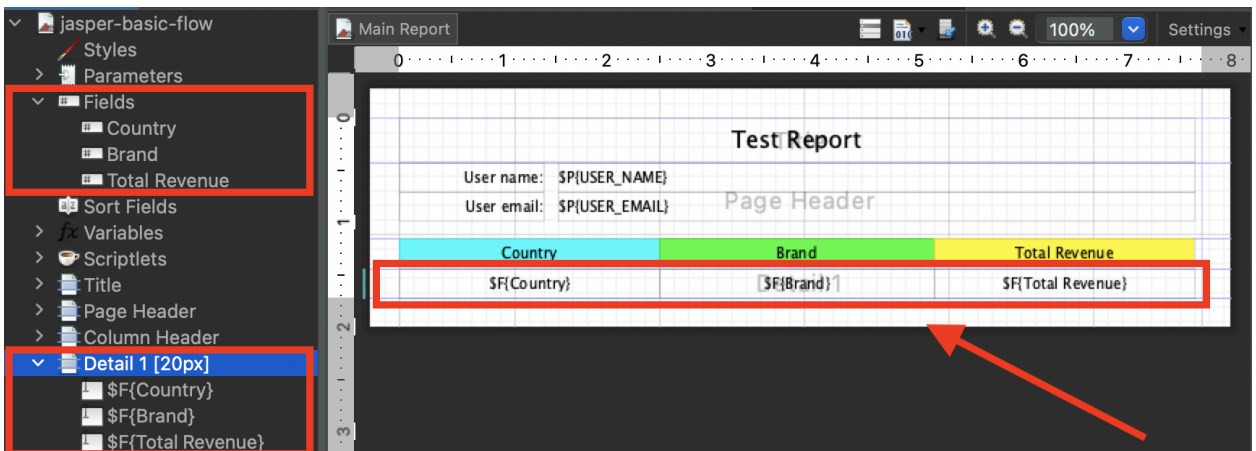
If you don't need to configure datasets in your template, you can use the default dataset functionality.

1. Set the "useDefaultDataset" property to "true" in the client side add-on configuration.

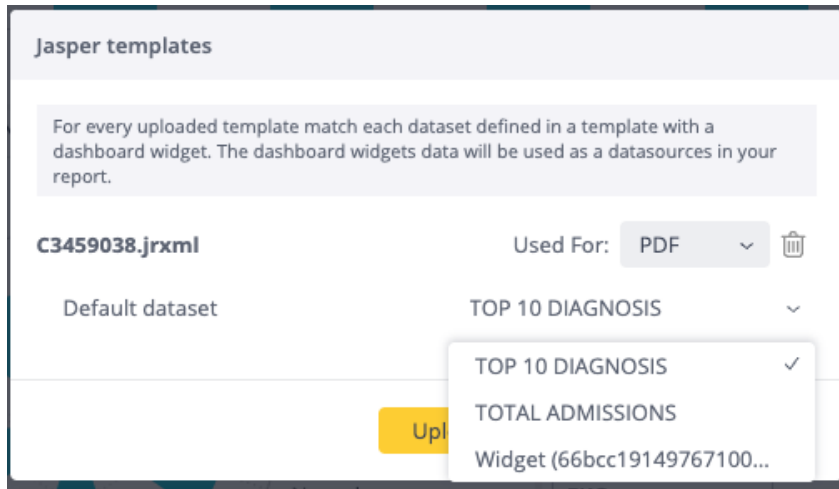
```

JS config.6.js M X
1  const config = {
2    ...replaceOrigExportWithJasper: false,
3    ...useDefaultDataset: true,
4  };
5
6  module.exports = config;
7
  
```

2. In the template root "Fields" section create fields that will be the same as on the widget metadata panel and put them into the "Details" band in the template. All the data in this band will be iterated and shown in the resulting report.



3. Select the corresponding widget to match the "Default dataset" in "Jasper Templates" configuration. Because "useDefaultDataset" is set to TRUE in step 1, you will always get the "Default dataset" option in the "Jasper templates" pop-up.



Using template parameters

You can embed and use the following information in your Jasper template:

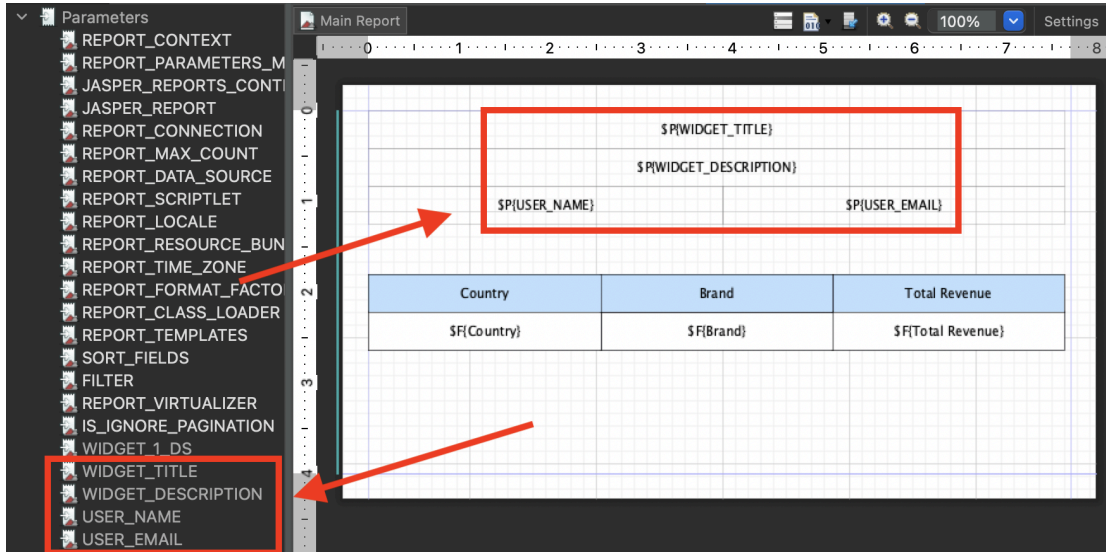
- Username and email
- Widget title and description

Widget title and widget description parameters only make sense in case the "useDefaultDataset" configuration property is "true" and the template contains only one widget data (more details in "Exporting a single widget" section above).

To use the parameters in your template, add them to the Parameters list in the JasperSoft Studio:

- To show the user name, add a "USER_NAME" parameter.
- To show the user email, add a "USER_EMAIL" parameter.
- To show the widget title, add a "WIDGET_TITLE" parameter.
- To show the widget description, add a "WIDGET_DESCRIPTION" parameter.

You can drag and drop the new parameters into the template.



Connecting widget data with jasper template

When you trigger the Jasper report generation, The Add-on sends widgets queries to the server-side part of the add-on. Then the widget query is used to fetch the data from Sisense. Then Jasper needs to match the widget data fields with the template fields. To achieve this, follow these steps:

- To show the widget data, add a field and change the field title to match the widget field title, as defined in the widget metadata panel.
For example, you can create in the following widget two fields in Jasper Category and Cost.

SISENSE Data Analytics Puls

Pivot

Rows

- Category

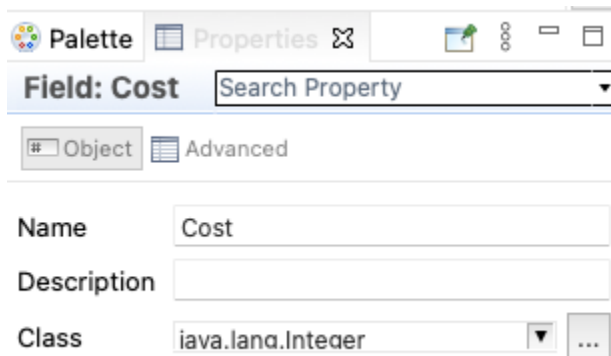
Values

- Cost

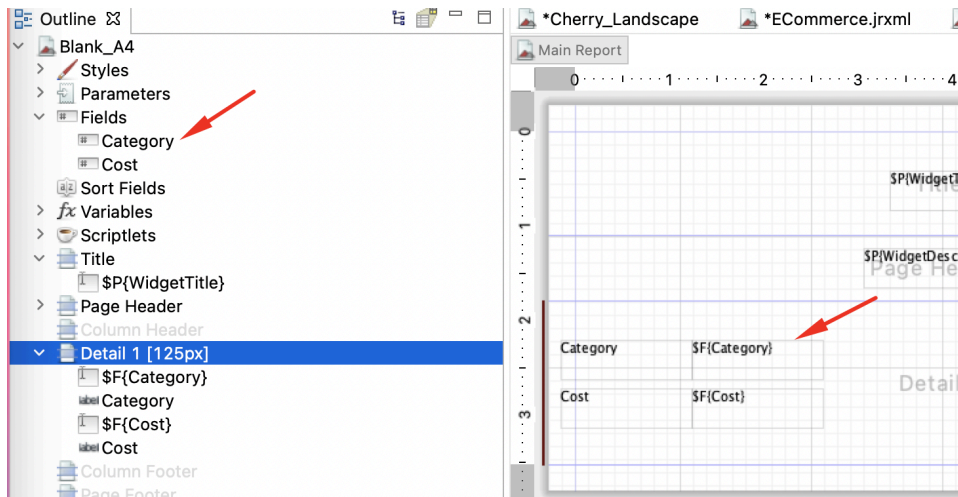
Columns

Category	Cost
Apple Mac Desktops	2,266,128.43
Apple Mac Laptops	4,210,312.11
Calculators	109,638.16
Camcorders	3,013,462.6
Camera Flashes	1,442,058.97
Car Amplifiers	498,600.08
Car Speakers and Subwoofers	646,753.82
Cell Phones	33,812,455.7
Digital Cameras	14,105,555.09
DVD Players	333,070.71
Film Cameras	2,278,451.22
GPS Devices	3,407,012.89
Hard Drives	1,403,957.73

- Change the field class to match the type of the field in the cube. For example, the Cost field is an integer field.



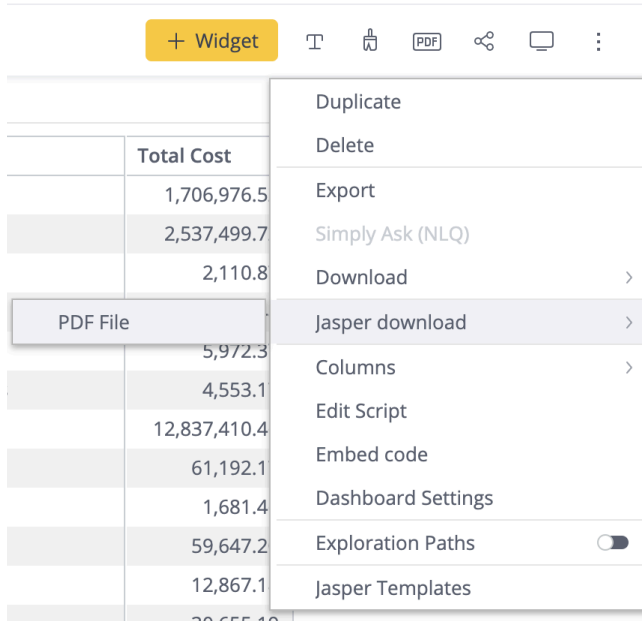
- After the fields are created, drag and drop the field into the template.



Downloading report

Downloading report for all selected widgets

- Click on the "Options" icon of the dashboard header.
- Select "Jasper download" => "PDF File" in the dropdown menu.



The "Jasper download" menu item is available if:

- Dashboard has at least one uploaded template;
- There is at least one widget of supported type (Pivot, Table or BloX) configured for export in the "Jasper templates" popup.

Downloading report for a single widget

- Click on the "Options" icon of the widget header.
- In the dropdown menu select "Jasper download" => "PDF File".

The option is available if:

- dashboard has at least one uploaded template;
- template is configured for this particular widget in "Jasper templates" popup.

Note: Downloading a report for a single widget makes sense in case you have configured only one template for a single widget. But if a widget is selected for several templates for the given dashboard, then all the templates where this widget is present will be generated as a PDF file and concatenated together.

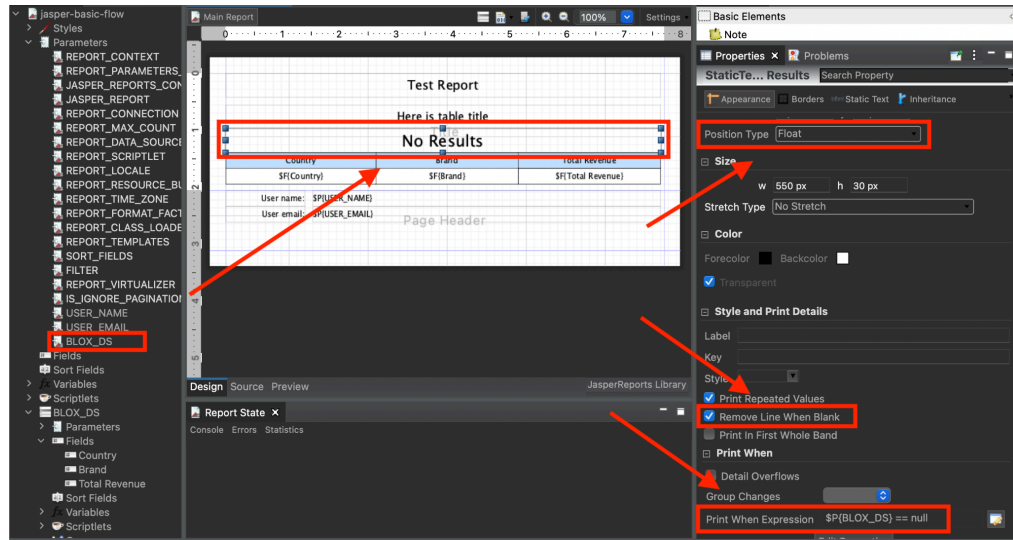
Showing "No Results" for empty widgets

You can configure your template to show "No Results" message or icon in case there is no data in table.

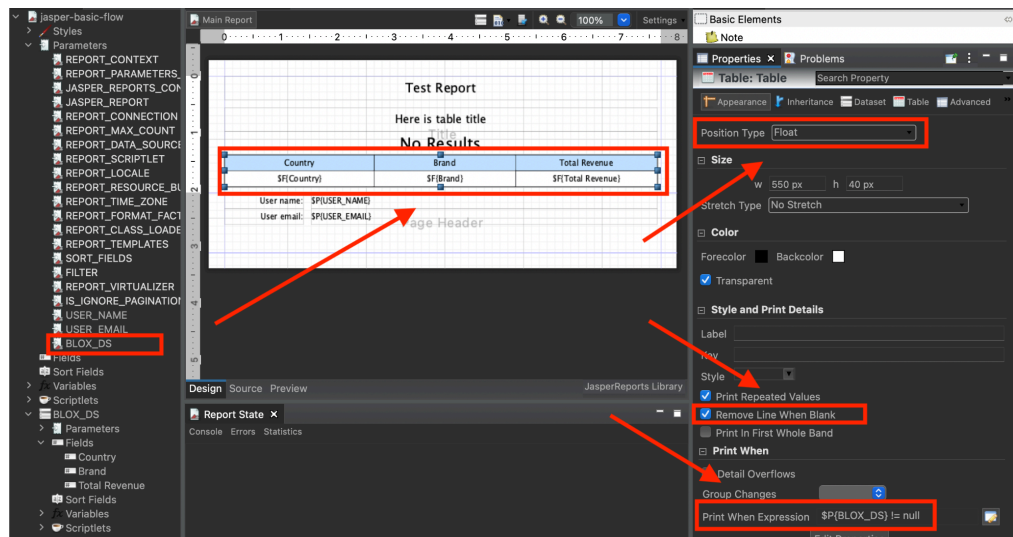
- Create additional element right before your table. This element will be shown in case the widget doesn't contain any data. Click on this element => "Properties" tab will be opened on the right:
 - Set the element position to "float". We need it to let the elements consider the

real space the elements take. For example in final report the table length will depend on the content, so the "float" position will let next elements to go right after the end of the table.

- Set "Remove Line When Blank" property. This property will remove the empty space in case this element will be conditionally hidden.
- Set "Print When Expression" to conditionally hide/show the element. In this expression check if table corresponding parameter is null. It means there is no data and we should show "No Results" message.



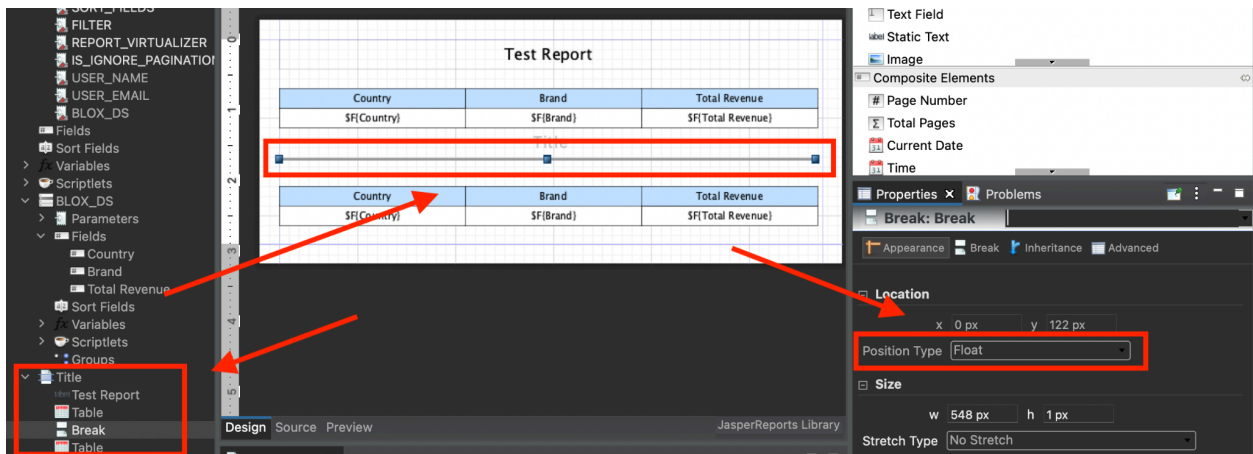
- Do the same for your table element that is located right after `No Results` element, except one thing:
 - Set "Print When Expression" to conditionally hide/show the element. In this expression check if table corresponding parameter is NOT null. It means there is some data and we should show the table.



Page breaks in template

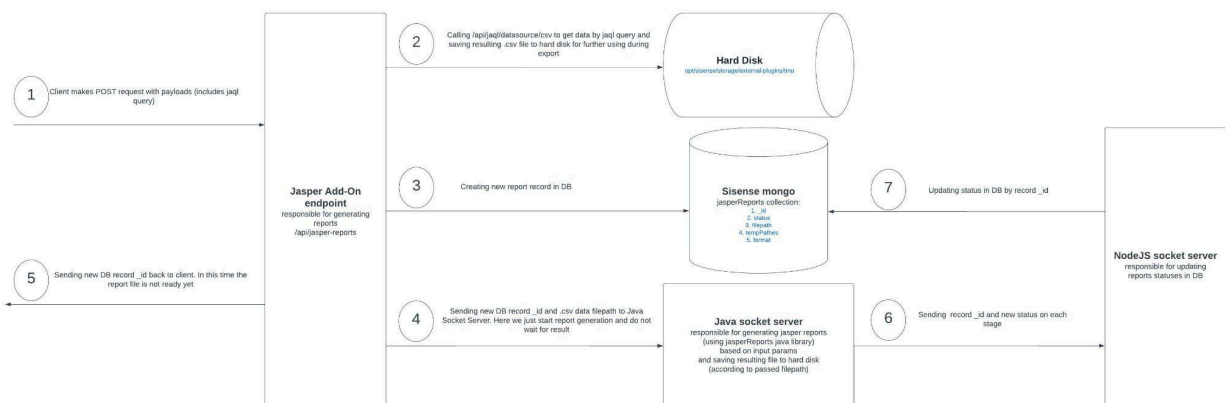
Lets imagine you want second table to start from new page. You can achieve this using "Break" element.

- Put "Break" element between your tables in template.
- Make sure "Break" element also is situated between tables in "Title" band on the left.
- Set "float" position for your "Break" element



Add-On Architecture

The architecture of the add-on is presented on the diagram below



Fonts Installation

By default Plugin supports a limited number of fonts. To extend it please consider the recommendations below.

1. Manual how to create a "font extension" ".jar" file using "Jaspersoft Studio":
<https://community.jaspersoft.com/wiki/custom-font-font-extension>
2. For installing fonts you have to create at least one "font extension" as described above and put it to location setup in "javaFilesPathes.libDirPath" from
"/opt/sisense/storage/external-plugins/apiPlugins/plugins/jasper-reports/v1/constants.js".
3. The place where you can download True Type Fonts files: <https://fonts.google.com/>
Also you can use other fonts.

Note: as Microsoft Excel uses fonts from OS we recommend to use fonts in your jasper template that are available in Excel. This ensures the font consistency in target OS where the excel file will be used.