**Using report parameters**

Report parameters are used in many different ways inside a report. They can be used in the *where* condition of an SQL query or to provide additional data to the report (i.e. the a value for a title or the name of user that executed the report).

A parameter is defined by a name and a Class, which is a Java class type like java.lang.String or java.lang.Integer. Any Java class is a valid parameter class. In example a parameter of type java.sql.Connection may be used to populate a subreport, while a simple java.lang.Boolean parameter may be used to show or hide a section of the report.

Whatever you decide to use the parameters for, they represent the best way communication channel between the report engine and the execution environment (which is your application).

A parameter can have a default value which is defined by means of the default expression property. This expression is evaluated by JasperReports only when a value for the parameter has not been provided by the user at run time.

## Built-in parameters

All reports contain a set of built-in parameters, parameters available by default that contain some run time information. Some the most important are the REPORT\_CONNECTION, which holds the JDBC connection used to run the SQL query of the report (if the report is filled using a JDBC connection), the REPORT\_DATA\_SOURCE which contains, if available, the data source used to fill the report, the REPORT\_LOCALE which contains the Locale used to fill the report and so on.

Some built-in parameters are specific of some query languages, in example when using the Hibernate query language, the reports automatically includes the parameter HIBERNATE\_SESSION that holds the Hibernate session used to run the HQL query.

The built-in parameters can not be modified or deleted.

## Using parameters in an SQL query

Parameters can be used in SQL queries to filter records in a where condition or to add/replace pieces of raw SQL or even to pass the entire SQL string to execute.  
In the first case the parameters are used as standard SQL parameters, in example:

**SELECT** \* **FROM** ORDERS **WHERE** ORDER\_ID = $P{my\_order\_id}

In this example my\_order\_id is a parameter that contains the ID of the order to read. This parameter can be passed to the report from the application that is running it to select only a specific order. Please note that the parameter here is a real SQL parameter, meaning that the query will be executed using a prepared statement like:

SELECT \* FROM ORDERS WHERE ORDER\_ID = ?

and the value of the parameter my\_order\_id will then passed to the statement. Let's consider this query:

**SELECT** \* **FROM** ORDERS **ORDER** **BY** $P!{my\_order\_field}

In this case my\_order\_field can not be treated as an SQL parameter. JasperReports will consider this parameter like a kind of place holder (note the special syntax $P!{}) which will be replaced with the text value of the parameter (which in this case may be for instance "ORDERDATE DESC"). With the same logic, a query can be fully passed using a parameter. The query string would look like:

$P!{my\_query}

The number of parameters in a query is arbitrary. When passing a value using the $P!{} syntax, the value of the parameter is taken as is, the user is responsible of the correctness of the passed value (SQL escaping is not performed by JasperReports in this case).

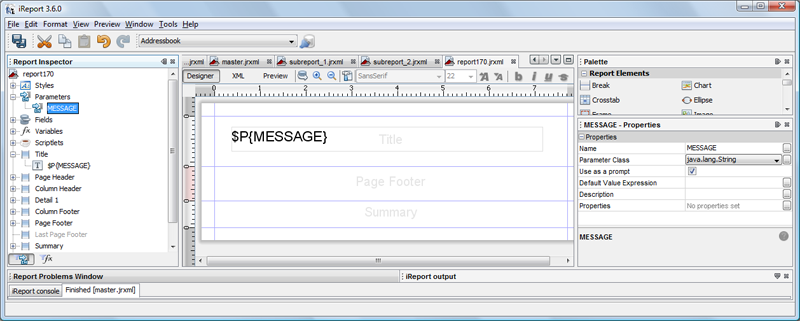
When using a parameter in a query, to allow iReport to execute the query to retrieve the available fields, a default value must be set for the parameter.

## Parameters prompt

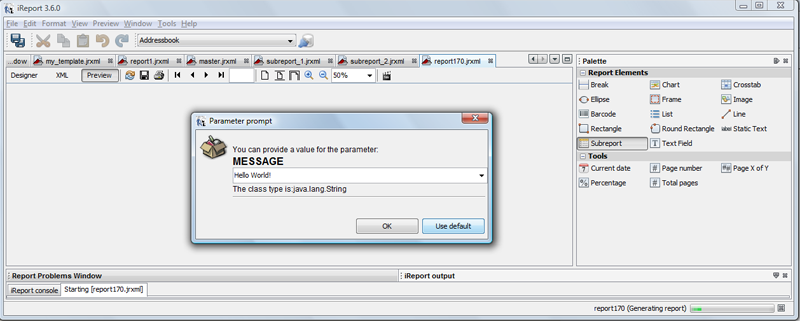
If a parameter is set to be used as a prompt, when a report is executed, iReport asks for the value of the parameter. Here is a simple example.

Create a report with a parameter called MESSAGE of type String set to be used as a prompt (the property Use as a prompt must be checked).

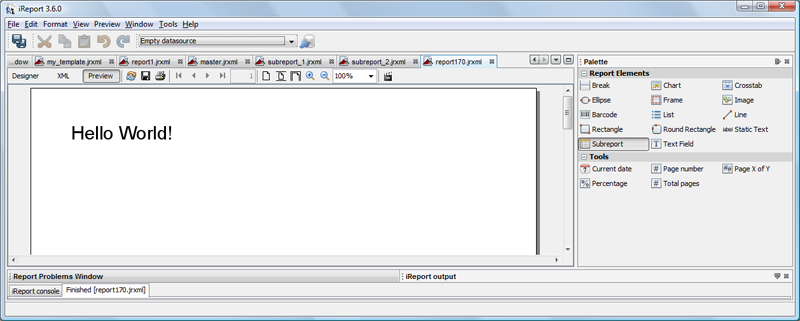
Drag the parameter from the report inspector inside the title band. iReport creates a textfield to display the parameter value.



Run the report using an empty data source by clicking the preview button. The parameter prompt dialog will appear asking for a value for the MESSAGE parameter.



Set a value for the parameter (i.e. Hello World!). The message will be printed in the title band.



iReport provides input dialogs for parameters of type String, Date, Time, Number and Collection.

# Using report variables

## Introduction

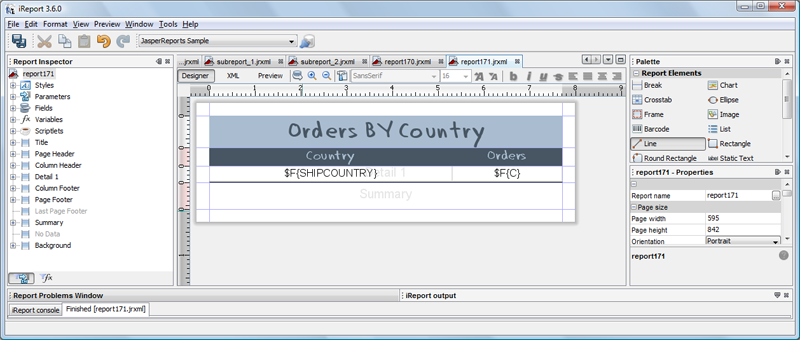
## Variables are used to perform calculations. They are managed using the report inspector and the property sheet just like the parameters.

## Built-in variables

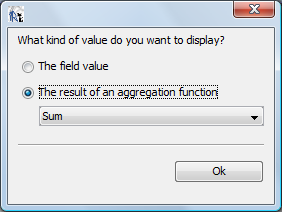
All reports contain a set of built-in variables. Their value changes during the report execution. Built-in variables include the PAGE\_NUMBER, which holds the current page of the report and the REPORT\_COUNT which holds the number of records currently processed. The built-in variables can not be modified or deleted.

## Create a total

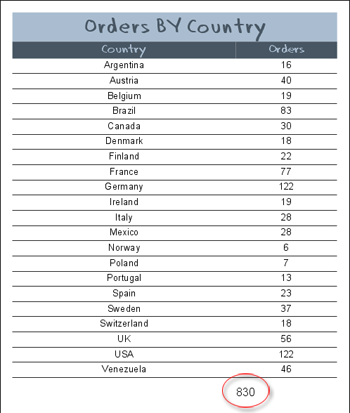
The fastest way to create a calculation is to use drag a field inside a band (like the summary) and use the calculation wizard. The following figure shows a simple report that lists a set of countries (SHIPCOUNTRY) and the number of orders placed in that country (C).



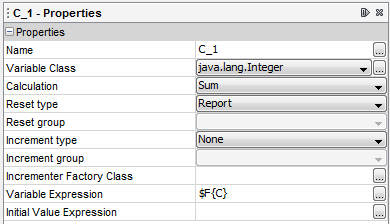
By dragging the field C inside the summary band, iReport asks what value to show. It can be just the value of C (which in this band will be just the last value assumed by the field C) or the result of an aggregation function like the sum.



Select the sum and press ok. Running the report, we can see that the value printed is, in effect, the sum of all orders in all countries.



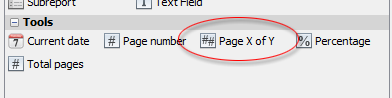
How does it happen? When we asked to print the sum of C, iReport created for use a variable which has been added to the variables (the name is C\_1). This variable is of type Integer (since the sum is a sum of Integer values), the calculation type is Sum and the expression used to update the variable value (with the sum calculation) is $F{C} which is an expression to get the value of the field C. For each record of the report, JasperReports gets the value of the field C (actually it evaluates the variable expression), and add that value to the set of collected values to perform the calculation. The variable C\_1 is displayed in a textfield in the summary band.

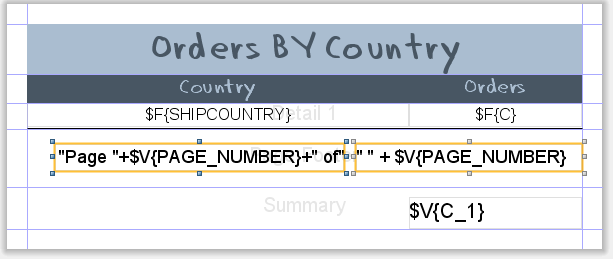


We used a simple wizard here, but the variable C\_1 can be created from scratch adding the variable from the Variables node and configuring it as we have just seen.

### Page X of Y

To add the Page X of Y in a report, just drag the Tool Page X of Y from the palette into any band (for example, in the page footer).





This tool creates two textfielda showing the same variable: PAGE\_NUMBER. The first textfield shows the current page, the second the total pages of the report. This is possible because the evaluation time of the two textfield is different, in particular the first textfield has evaluation time set to Now so PAGE\_NUMBER contains the value of the current page, the second to Report (at this evaluation time, JasperReports has reached the end of the report, so PAGE\_NUMBER contains the last page number).  
 The evaluation time of a textfield is very important because allows us to print the value assumed by a variable at different times. With this idea we can put the total sum of orders seen in the previous sample in the title band and get the correct value by setting the evaluation time of that textfield to Report (this is automatically done by iReport when a field is dragged in the title and the user chooses to show the result of an aggregation function).

**Parameters** are simple input to JasperReports. You have to define parameters in the JasperReport before using them. You can display the value of the variable, you can use it as part of boolean expressions and you can even pass it to subreports. This can be an input to SQL query.

**Fields** are simple variable definitions. You can think of these as instance variables of the datasource object thats passed in to the report or they can be key names if the datasource is a Map. If you configure JasperReport to create the dataset based on SQL, then Fields are the column names of the ResultSet. You will use Fields to display the resultset of an executed SQL query.

**Variables** are another kind of variables that live within Jasper Report, they are not inputs. They are used to calculate sum or average of certain Field (defined above). You can perform many other predefined calculation functions on the Fields using Variables.