

Continuous processing

Continuous processing is useful when you have an external process which generates some data records and you need to generate and deliver reports automatically, when records are generated. For example, you may have a software which generates invoices, orders, notices etc., and you want R-Tag to detect when a new record is created, print/export a report, and send it to the right recipients.

Let's say you have an online Orders system, which creates records in the Orders table.

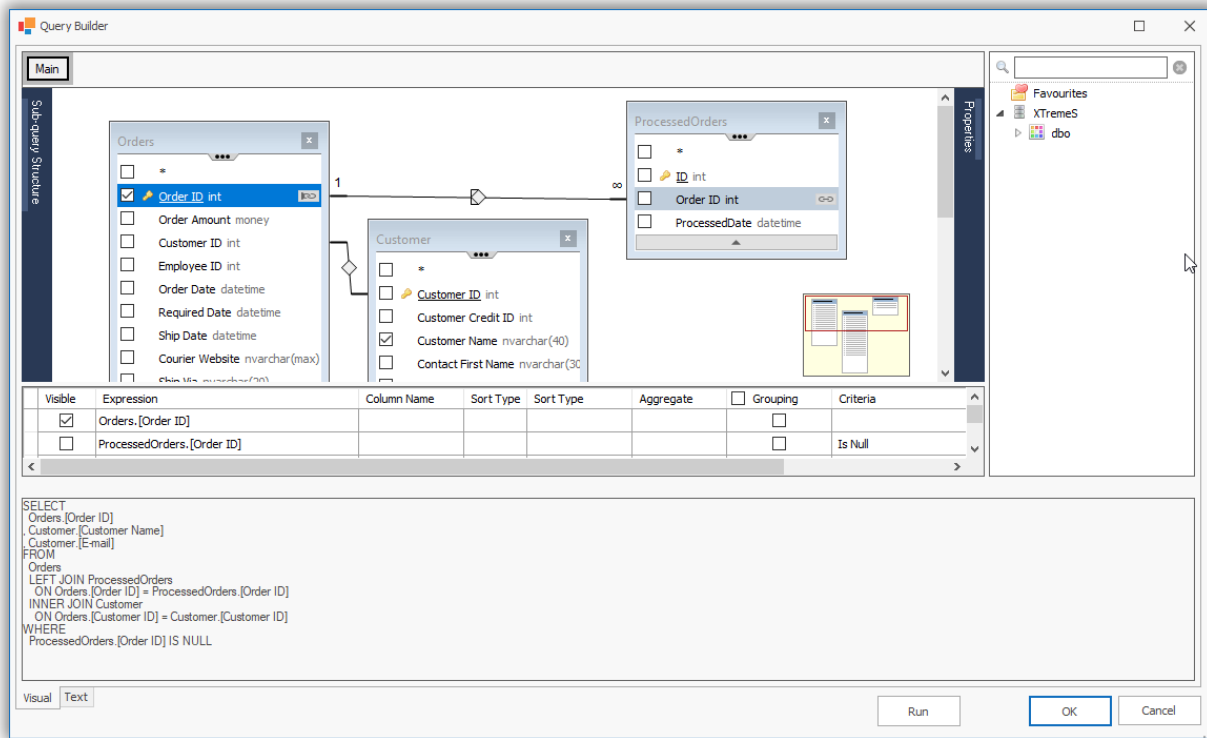
	Column Name	Condensed Type
🔑	[Order ID]	int
	[Order Amount]	money
	[Customer ID]	int
	[Employee ID]	int
	[Order Date]	datetime
	[Required Date]	datetime
	[Ship Date]	datetime
	[Courier Website]	nvarchar(MAX)
	[Ship Via]	nvarchar(20)
	Shipped	bit
	PO#	nvarchar(50)
	[Payment Received]	bit
	SSMA_TimeStamp	timestamp

When a new record is created you want to run a report and export the order info to a PDF and then send it as an attachment to the Customer's email address. It is easy to create a Data driven job, which will run every 5 seconds or 5 minutes and will retrieve a list of orders, run a report for each order, export it and send it by email. However, after processing, orders should be marked as processed so the next time the data driven job runs it will exclude processed orders. To handle this, create a table called ProcessedOrders.

	Column Name	Condensed Type
🔑	[Order ID]	int
	[Order Amount]	money
	[Customer ID]	int
	[Employee ID]	int
	[Order Date]	datetime
	[Required Date]	datetime
	[Ship Date]	datetime
	[Courier Website]	nvarchar(MAX)
	[Ship Via]	nvarchar(20)
	Shipped	bit
	PO#	nvarchar(50)
	[Payment Received]	bit
	SSMA_TimeStamp	timestamp

	Column Name	Condensed Type
🔑	ID	int
	[Order ID]	int
	ProcessedDate	datetime

Modify the data driven command to left join this table and to retrieve just orders without records in ProcessedOrders. You can use the R-Tag Query Builder.



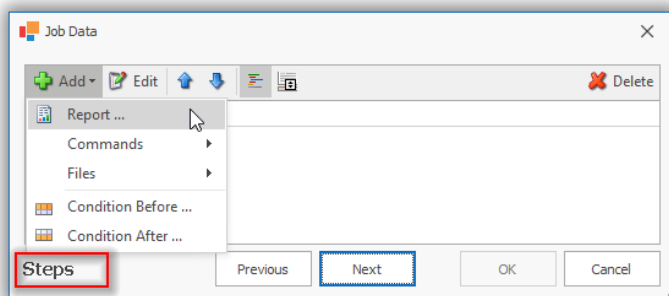
The data driven command will look something similar to this:

```

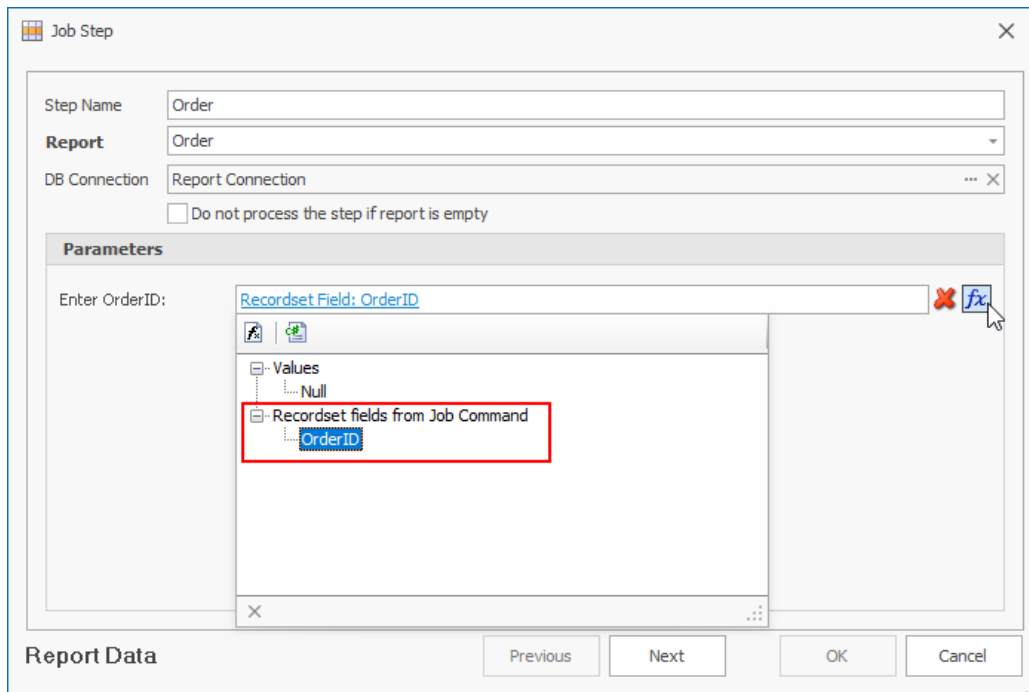
SELECT
    Orders.[Order ID] as OrderID
    , Customer.[Customer Name]
    , Customer.[E-mail]
FROM
    Orders
LEFT JOIN ProcessedOrders
    ON Orders.[Order ID] = ProcessedOrders.[Order ID]
INNER JOIN Customer
    ON Orders.[Customer ID] = Customer.[Customer ID]
WHERE
    ProcessedOrders.[Order ID] IS NULL
  
```

***Please note that the query is using alias for [Order ID] to avoid having spaces in name.**

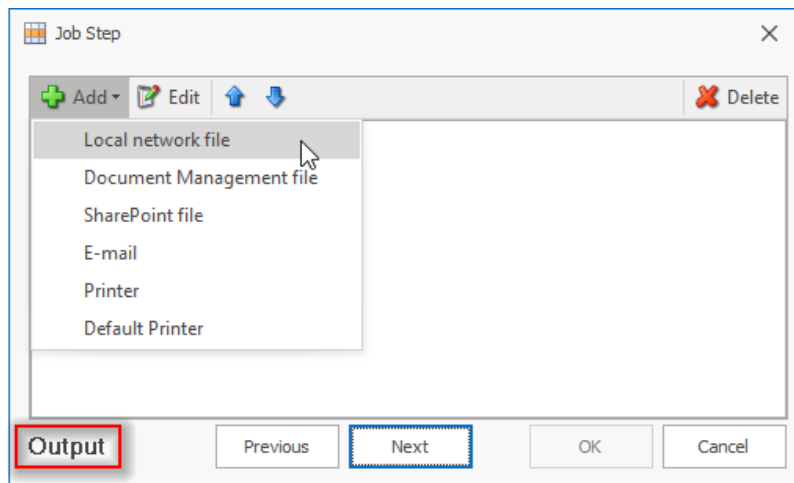
Navigate to “Steps” and add the report, which will be used to export each of the orders returned by the SQL above.



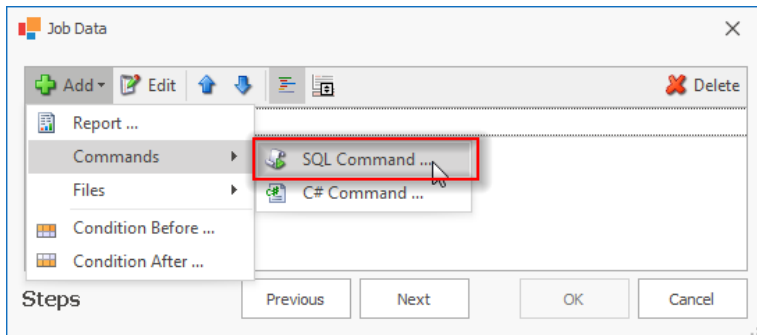
Report expects the parameter OrderID, and you can assign a field value from the current row by clicking *fx* and selecting the preferred field returned by the SQL query above.



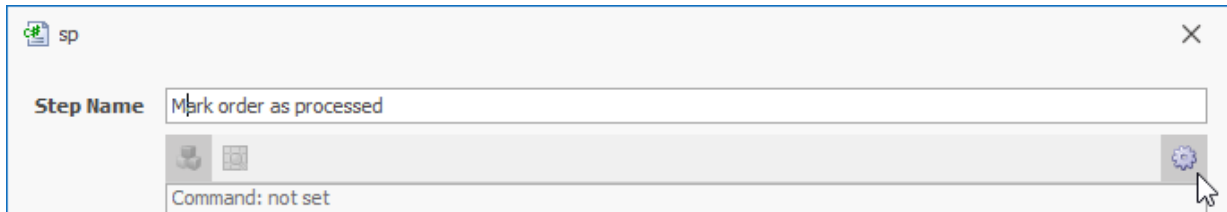
When executing the data driven job, R-Tag will run the query and will loop through all returned records. R-Tag will run the report Order and will use the value of the field OrderID to set the parameter OrderID for each record. The next step will be to define the local network file where the report should be exported and the email task to send the exported file to the email address in the *Customer[E-mail]* field.



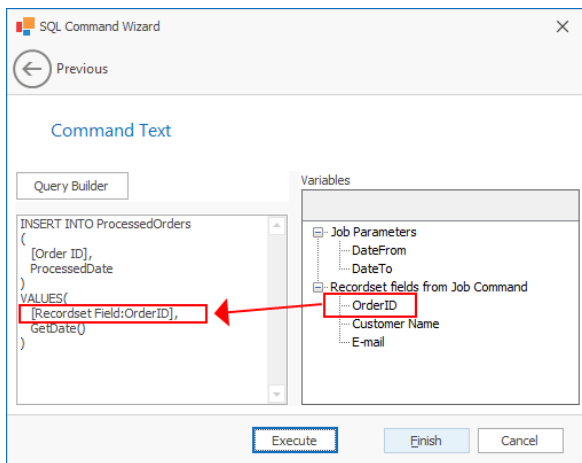
This step will run the report and deliver it to the customer. When it is executed the order is considered “processed”. After saving the report step, you will be back to Job Data “Steps” and you need to add a new SQL Command, which will run after the order is processed and will add a record to the *ProcessedOrders* table.



Insert the command name and click the Gear button to set the command.



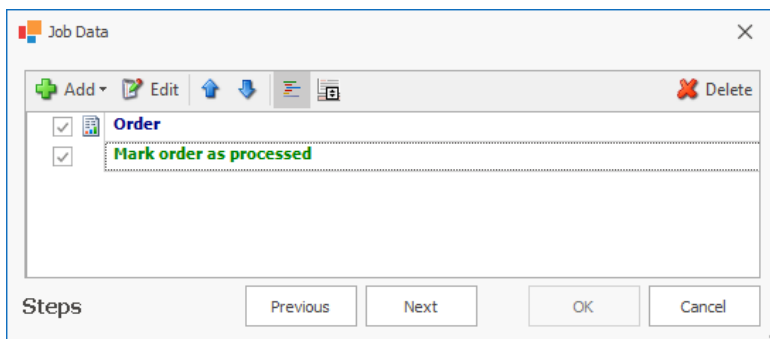
To insert a record using a text command use a command like:



```
INSERT INTO ProcessedOrders
(
    [Order ID],
    ProcessedDate
)
VALUES(
    [Recordset Field:OrderID],
    GetDate()
)
```

[Recordset Field:OrderID] could be inserted by double clicking the field OrderID in the Variables panel.

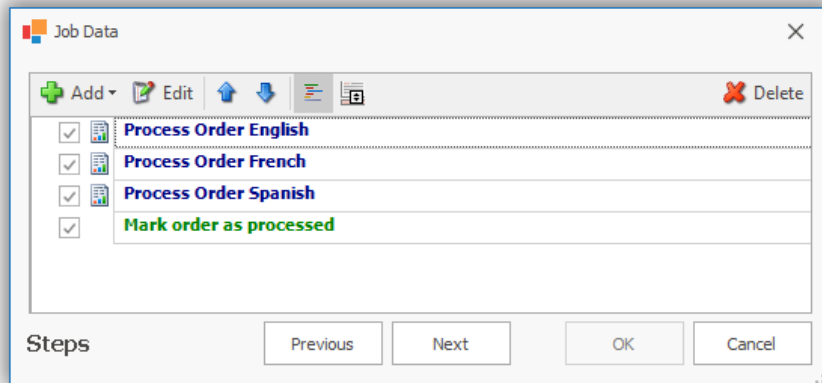
After saving the command the Steps screen will have 2 steps: "Order" and "Mark order as processed"



When R-Tag runs the data driven job it will only retrieve orders that have not been processed. Then it will process them and will mark them as processed so they will be ignored during the next run. The job can be scheduled using the short-term scheduler to run (for example) every 15 seconds so the new orders will be processed a maximum 15 seconds after they were created.

Advanced workflows

As you can see, the whole process looks like defining a workflow. In fact, it is a workflow. You can add other tasks such as a SQL Command to prepare some temporary tables before to run a report or a C# command to clean up some disk space. In addition, you can add conditions so you can handle more complicated scenarios. For example, you may have customers who prefer to receive their orders in language different than English, or it might be necessary to attach additional documents to certain orders. This can be handled using conditions. Let's say you have 3 different reports, in English, French and Spanish.

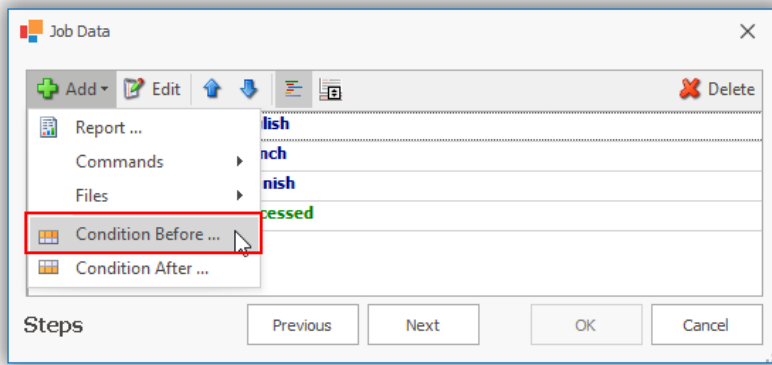


You can add a field for Customer's preferred language to the Data Driven query, and use this field in a condition to determine which step should be run. Here is how to do this:

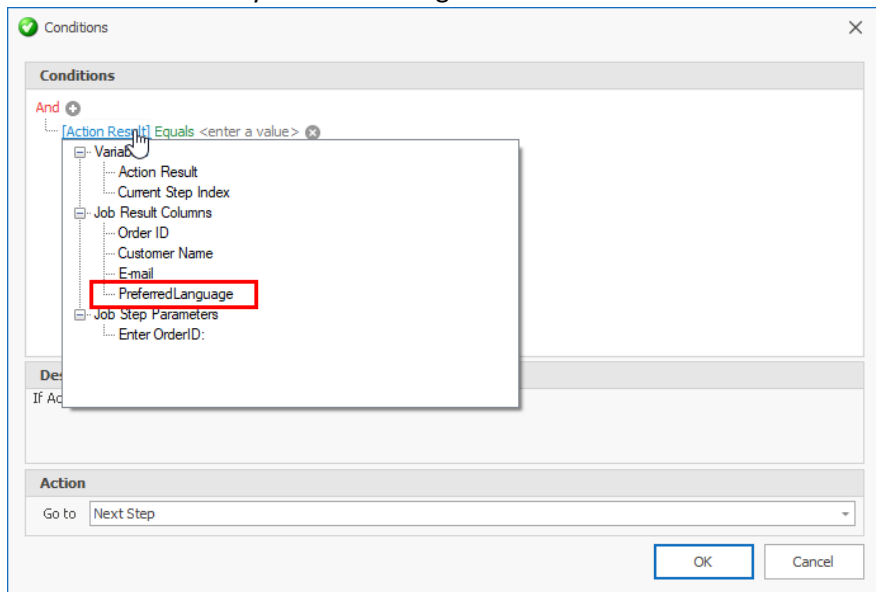
The Data Driven query will be changed to:

```
SELECT
  Orders.[Order ID] as OrderID
, Customer.[Customer Name]
, Customer.[E-mail]
, Customer.PreferredLanguage
FROM
  Orders
  LEFT JOIN ProcessedOrders
    ON Orders.[Order ID] = ProcessedOrders.[Order ID]
  INNER JOIN Customer
    ON Orders.[Customer ID] = Customer.[Customer ID]
WHERE
  ProcessedOrders.[Order ID] IS NULL
```

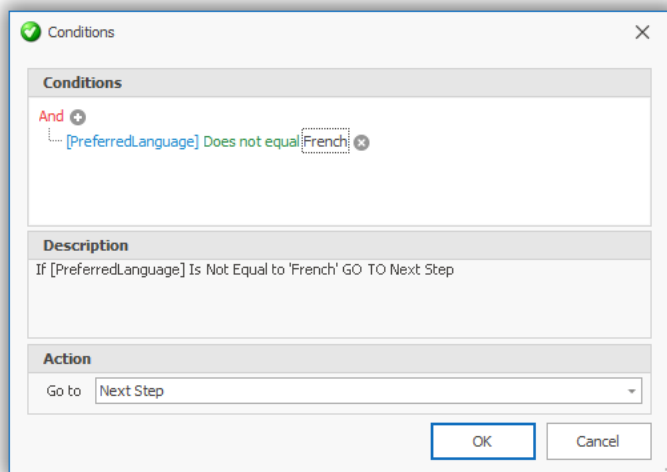
You can use the new field in a "Condition Before". Select the step for which you want to create a condition and click menu "Add" -> "Condition Before ..."



Inside the condition you can set a logical statement.

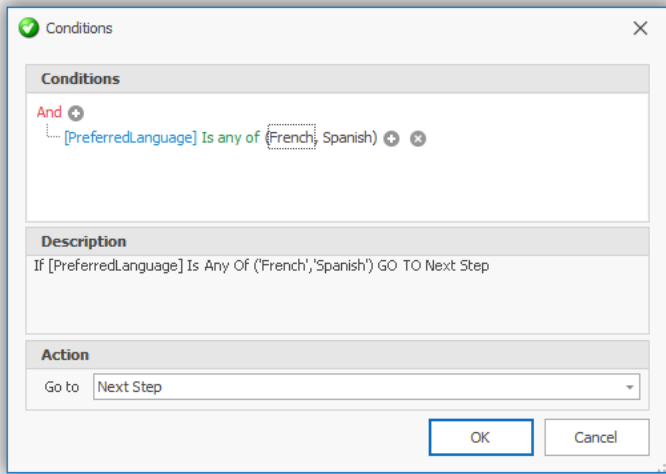


Here is a sample condition for French:

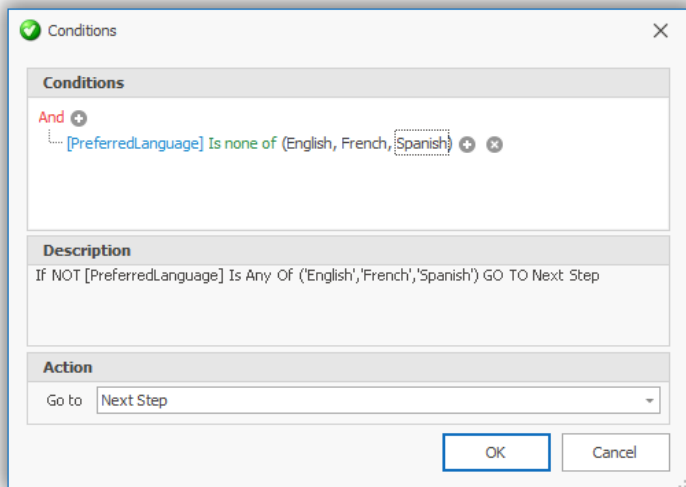


R-Tag will run the condition before the step to process French. If the preferred language is not 'French' it will not run the French step.

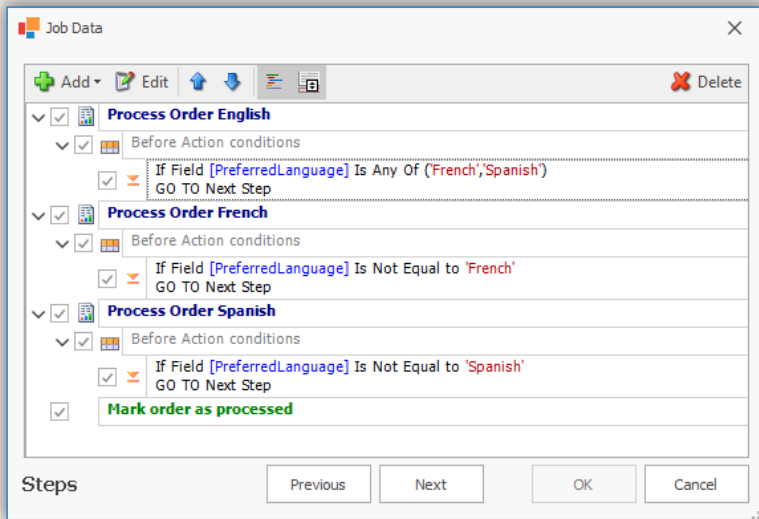
You can set Process Order English to be run for any other language except French and Spanish. In such a way you will be able to process specific languages (French and Spanish) and for all other languages you will use the English report.



Another option is to have one step for each language and to have a condition to run “Mark order as processed” just in case the order is one of the handled languages.

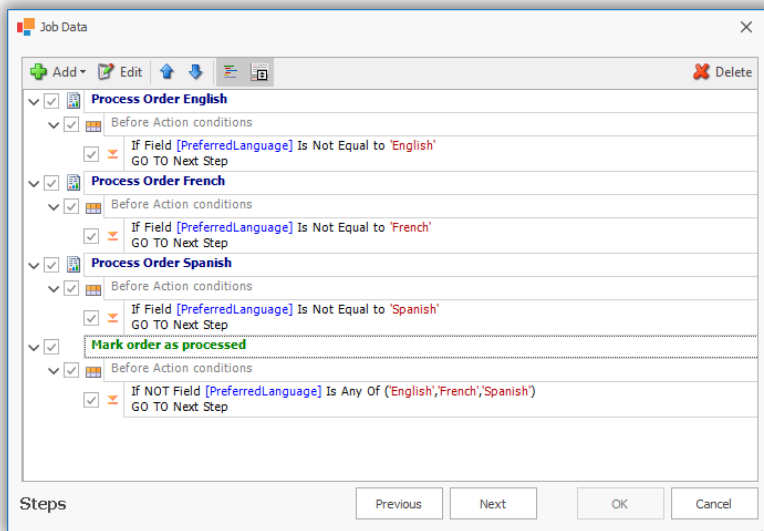


The final Steps screen will look like this:



The result is that for each row R-Tag will run just one of the steps to process order in French, Spanish and in English (for all languages different than French or Spanish) and will mark the order as processed.

The other option will process orders strictly in English, French or Spanish and if the language is different, the order will be **not** marked as processed and will be shown in the queue the next time (until you add a handler for the new language).



The same approach can be used to run different reports based on other factors, order type, products in the order etc.