

# R-Tag Report Manager

R - Tag

July 2011

## External Data Source

What you can do with data source.

## Contents

1.	Introduction: How datasource can be helpful.....	3
2.	Set data source .....	4
2.1.	Set Command data.....	4
2.2.	SqlCommand Wizard.....	4
2.2.1.	Welcome screen.....	4
2.2.2.	Database connection and command type .....	4
2.1.3	Stored procedure settings .....	4
2.1.4	Command text.....	5

## 1. Introduction: How datasource can be helpful

DataSource will provide the option to use data push method. Usually this is useful when you want to set report datasource from an application and data is already prepared. However, there are some cases where push method can speed up the report execution significantly.

Some reporting engines do not support multiple recordsets. The best example is Crystal Reports. In order to get multiple recordsets you will need to create sub-reports. This will make it possible to show the data, but it is costly since you have separate calls to the database for the main report and for each sub-report.

Example: You have a report, which prints an invoice for a customer. The report will show customer name and address on the top and there will also be a place to show the account balance. The invoice details will be printed below the customer data.

ProductName	Quantity	UnitPrice	Discount	Total
Jack's New England Clam Chowder	10	\$7.70	0	\$77.00
Manjimup Dried Apples	35	\$42.40	15	\$1,261.40
Louisiana Fiery Hot Pepper Sauce	15	\$16.80	15	\$214.20
Total				\$1,552.60

There will be 2 calls to the database because the main report will retrieve customer's information and there will be a sub-report for the invoice details. Furthermore, customer's balance should be calculated in the main report because it will be printed before the details section. **This means that the details will be queried twice: once to calculate the total and second time to show them in the sub-report.** If you have a system, which accepts partial payments this calculation can be really costly because you will need to calculate the balance of each item in order to decide if it will be included in the details section and you will do this twice, once for the customer's balance and a second time for the details.

Using push data method will have 3 advantages:

- One call to the database
- One calculation for the details section items
- Complex queries – will be explained later

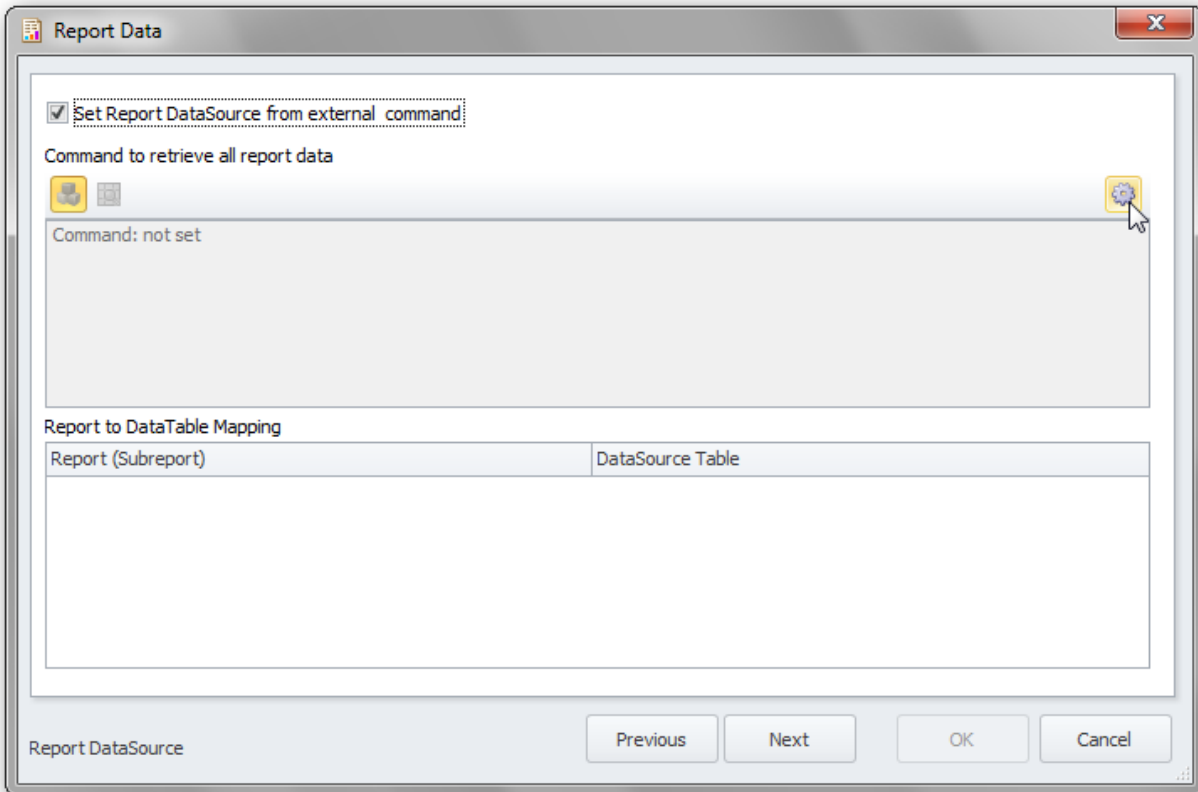
## 2. Set data source

### 2.1. Set Command data

First step of the process is to define the command, which will retrieve data.

In page “Report DataSource” check the checkbox “Set Report DataSource from external program”. If this check box is not checked the controls on the page will be disabled and Report Manager will use the data source defined in the report file.

Next step is to define the external data source: Click the button  to invoke SQL command wizard.



The wizard will guide you through the steps necessary to set the SQL command.

### 2.2. SQLCommand Wizard

#### 2.2.1. Welcome screen

First screen is the welcome screen. You can hide it by checking the check box “Do not show this screen again”.

#### 2.2.2. Database connection and command type

This is the place where you will set the connection information and the command type.

#### 2.1.3 Stored procedure settings

In case the command type is stored procedure you will set the stored procedure’s name and parameter values here

### 2.1.4 Command text

In case the command type is text you will set the query text here. Some databases support complex queries with more than one batch. For example in SQL server you can use text like this for the command:

```
DECLARE @OrderTotal MONEY

SELECT od.ProductName,od.Quantity,od.UnitPrice,od.Discount
INTO #temp
FROM OrderDetails od
WHERE od.OrderID={@OrderID}

SELECT @OrderTotal = SUM(od.UnitPrice*(1-od.Discount)*od.Quantity)
FROM #temp od

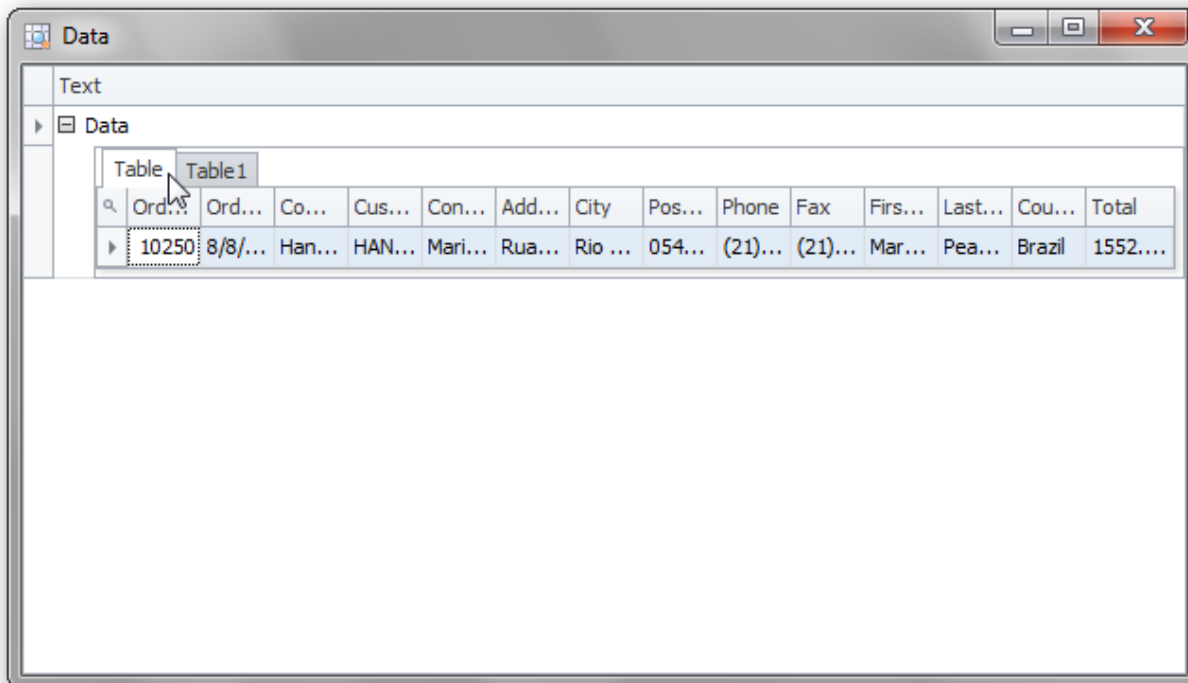
SELECT o.OrderID, o.OrderDate,c.CompanyName, c.CustomerID,
c.ContactName, c.[Address], c.City, c.PostalCode,
c.Phone, c.Fax, e.FirstName, e.LastName, c.Country,
@OrderTotal as Total
FROM Orders o
      INNER JOIN Customers c ON c.CustomerID = o.CustomerID
      INNER JOIN Employees e ON e.EmployeeID=o.EmployeeID
WHERE o.OrderID={@OrderID}

SELECT * FROM #temp
```

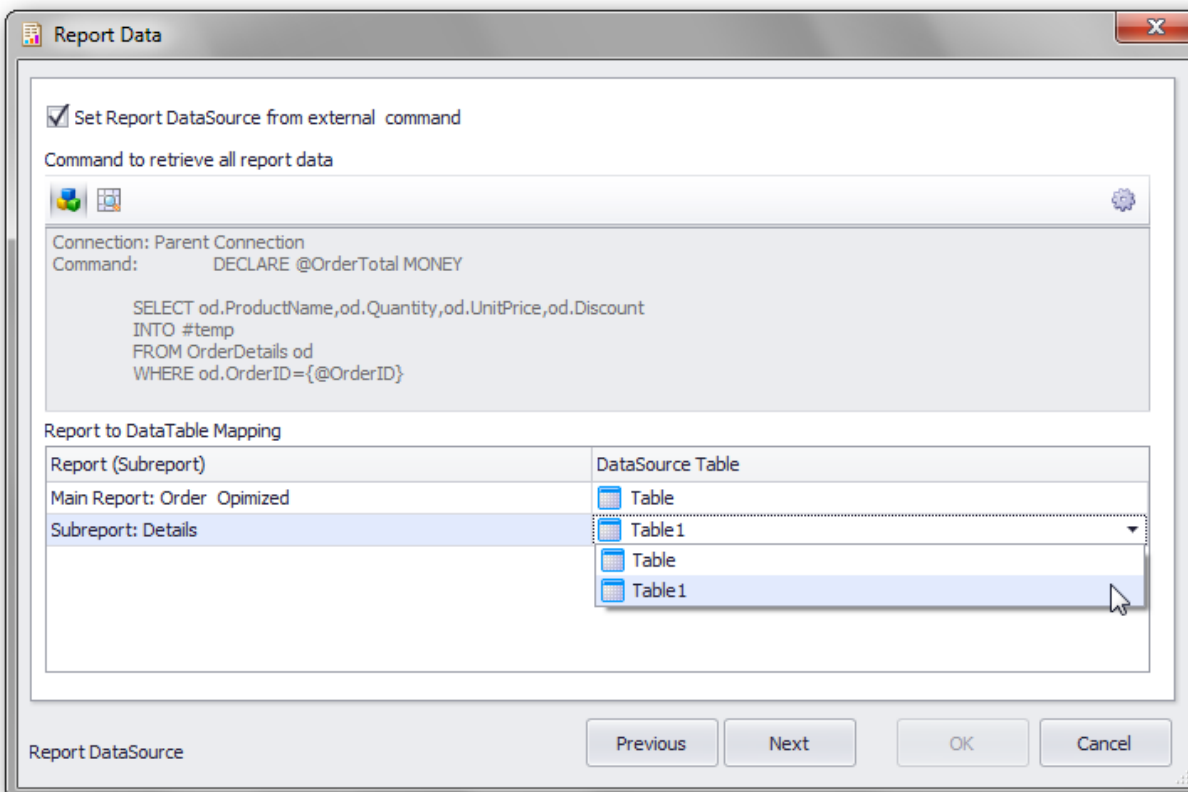
**As you can see text command in SQL server can define internal variables and temp tables, which can be handy in case you don't have rights to create stored procedures, but you can browse the data.**

This command will insert the details for the order in temporary table #temp; calculate the total for the records in that table; return the calculated total together with the customer information associated with the order and return the data from the #temp table in a separate recordset.

The result will be provided in 2 data tables



and you will be able to assign each table as a datasource for the main report or one of the subreports



As it is now, in Crystal reports, if an external datasource is used parameter values are not available directly. There is a workaround: add parameter value to the recordset and use the recordset field. For example if you want to show the value for the parameter OrderID on the report you will need to add it to the returned recordset and use the recordset field.