



isCOBOL™ Evolve

isCOBOL IDE Report designer tutorial

Copyright © 2016 Veryant LLC.

All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution and recompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Veryant and its licensors, if any.

Veryant and isCOBOL are trademarks or registered trademarks of Veryant LLC in the U.S. and other countries. All other marks are property of their respective owners.

Summary

isCOBOL IDE settings	4
Preliminary Steps	5
Creating a new Report	5
Report Sections Description	6
Report header	6
Report footer	6
Group header/footer	6
Detail section	6
Section Controller	7
Drawing the Report	8
Drawing the Report using Data Pool	11
Fill report logic	15
“BeforeDoPrint”	15
“AfterDoPrint”	17
Execute the report from a cobol program	19
Print to the preview	19
Print to the default printer	20
Print to a printer with the printer selection	21
Print to an html file on the disk	22
Add a group section to the report	23

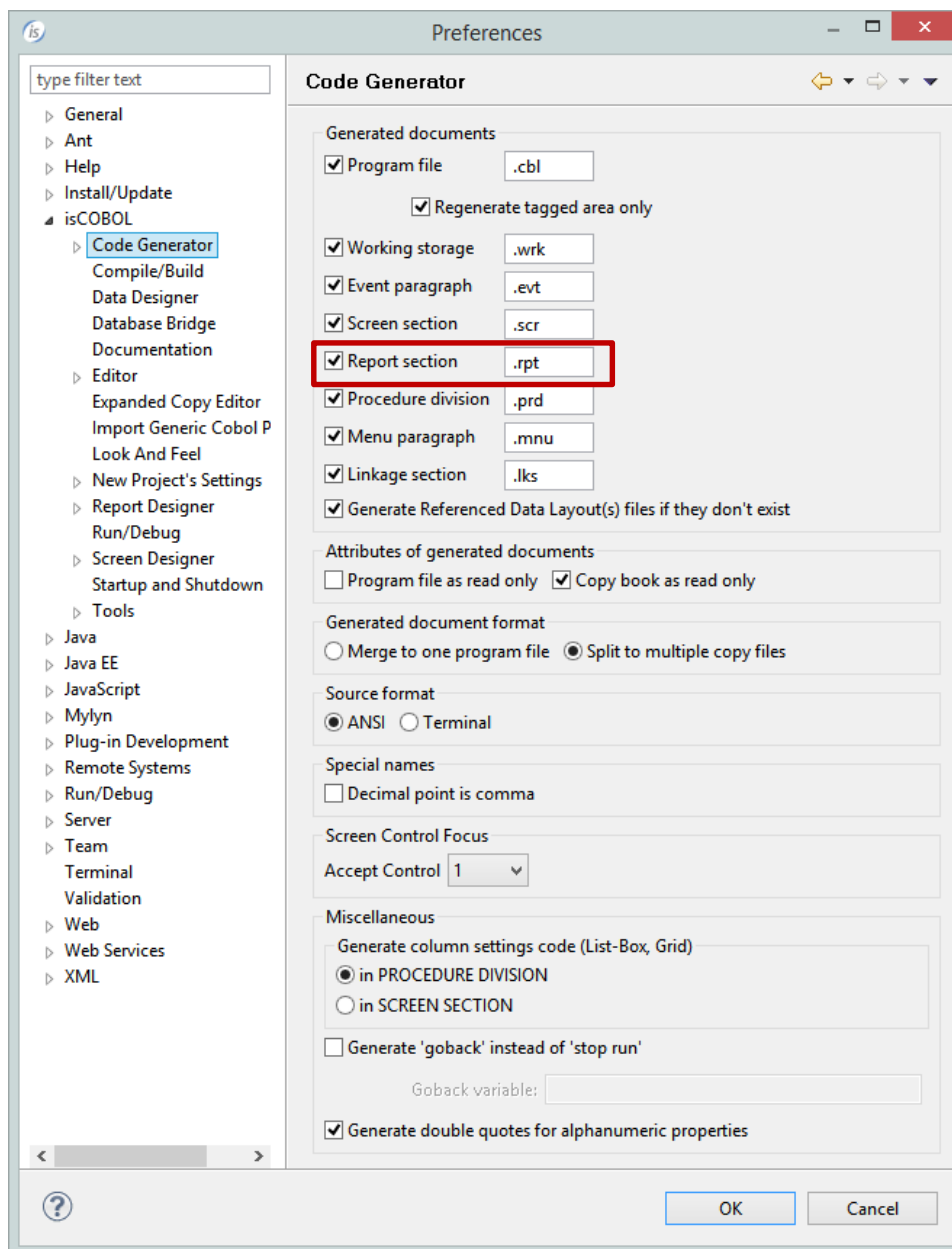
isCOBOL IDE settings

To use the report designer into the IDE we need to use the following setting for the code generation:

- Report section

To activate this option you must go to the menu “Window” and select “Preferences”. When you have the settings popup window comes up select “isCOBOL – Code Generator” (See Figure 1)

Figure 1. isCOBOL IDE Code Generator configurator



Preliminary Steps

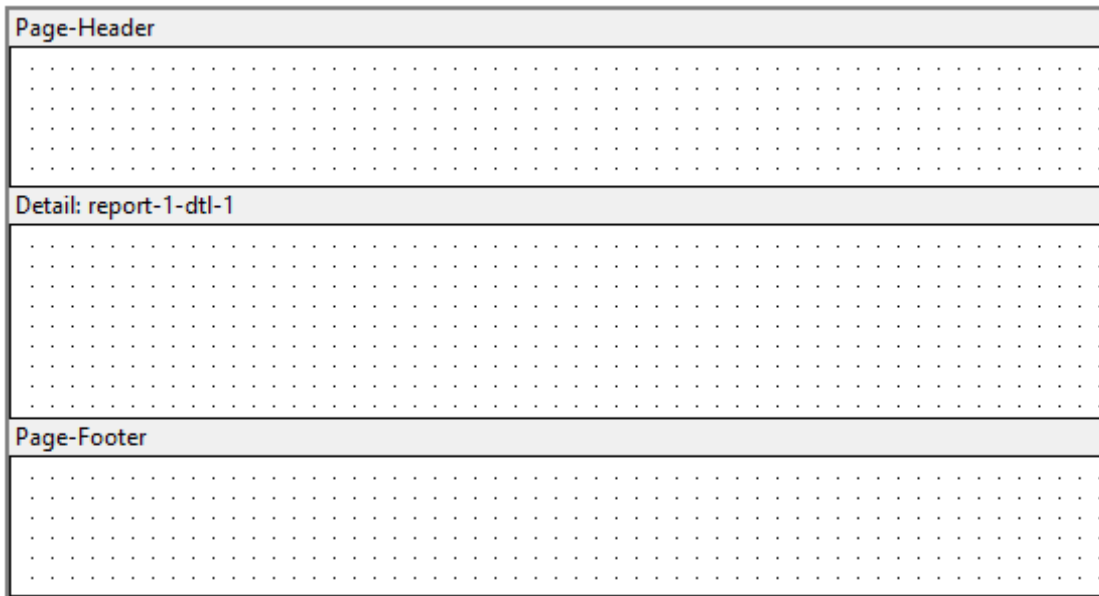
- Create a new Screen program or open an existing one
- Add the files used inside the report to this program's dataset.

Creating a new Report

To create a new Report, right click on the program name in the isCOBOL Explorer area and choose New / Report from the pop-up menu. Choose the name of your report and click Finish.

The new empty Report looks like this (like shown in Figure 2):

Figure 2. empty report



By default, a new Report has three sections: a page header, a detail section, and a page footer. With the Section Controller interface (described below) , you can add/remove report headers and footers, multiple detail sections, and group sections that define breakpoints for the information in the Report.

You can define the components and appearance of your report header or footer in the corresponding Property window.

Report Sections Description

Report header

The report header is printed once at the beginning of a Report.

Report footer

The report footer is printed once at the end of a Report. You can define the components and appearance of your report header or footer in the corresponding Property window.

Group header/footer

A group header prints at the beginning of a marked breakpoint, and a group footer prints at the end of a marked breakpoint. You may have multiple group headers or group footers in a Report, representing multiple breakpoints. Where multiple breakpoints are named, the first one named is the primary breakpoint, and subsequent breakpoints are always secondary to the breakpoint preceding them. When multiple group footers appear in a Report, they appear in reverse order in the Report Designer, with the secondary group footers appearing above the primary group footer. You add a group header or a group footer to the Report via the Section Controller dialog.

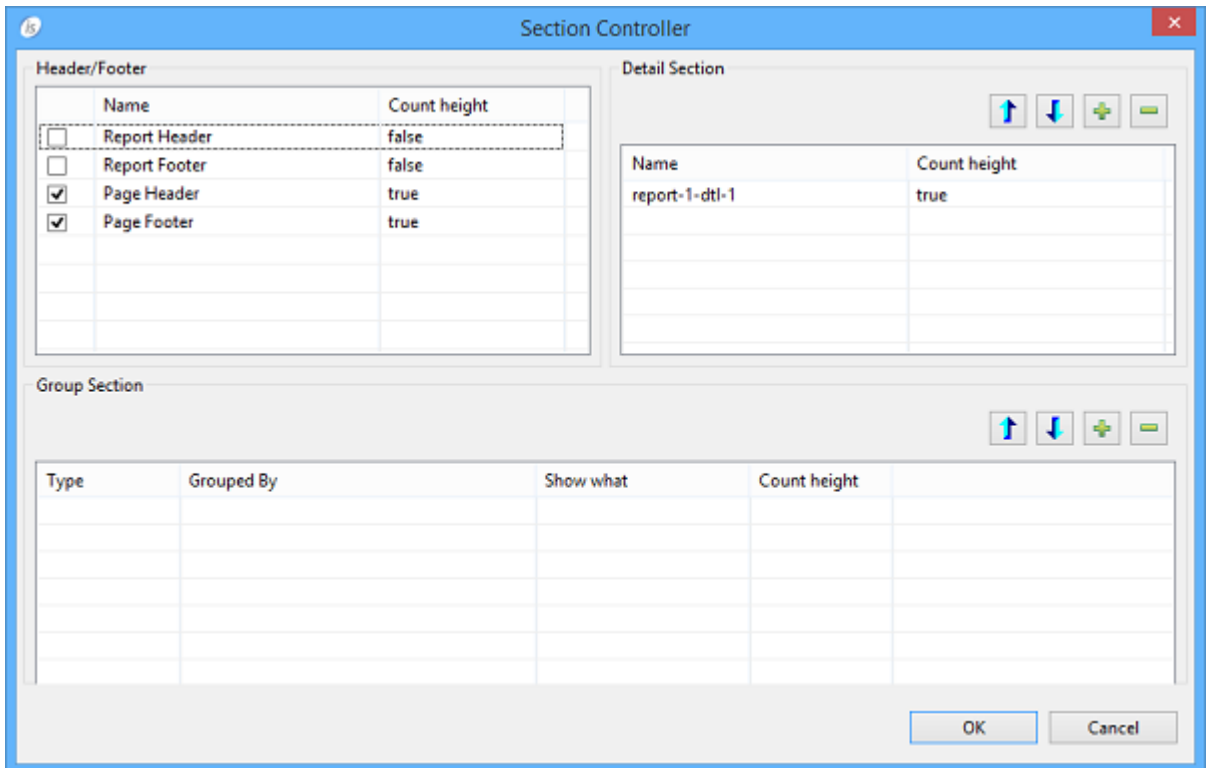
Detail section

The detail section prints repeatedly, providing the basic reporting function. This section constitutes the main body of the Report. Code for the detail section is generated into an ".rpt" COPY file. By default, one detail section appears in the Report Composer window when you create a new Report. You may have multiple detail sections in a Report, for example, if you wish to present your basic detail information in two parts and you choose a different font or background color for each part. You add detail sections via the Section Controller dialog.

Section Controller

Right-click in the Report Designer and choose Section Controller from the pop-up menu. Use this dialog (See Figure 3) to define what headers and footers should be printed.

Figure 3. Section controller

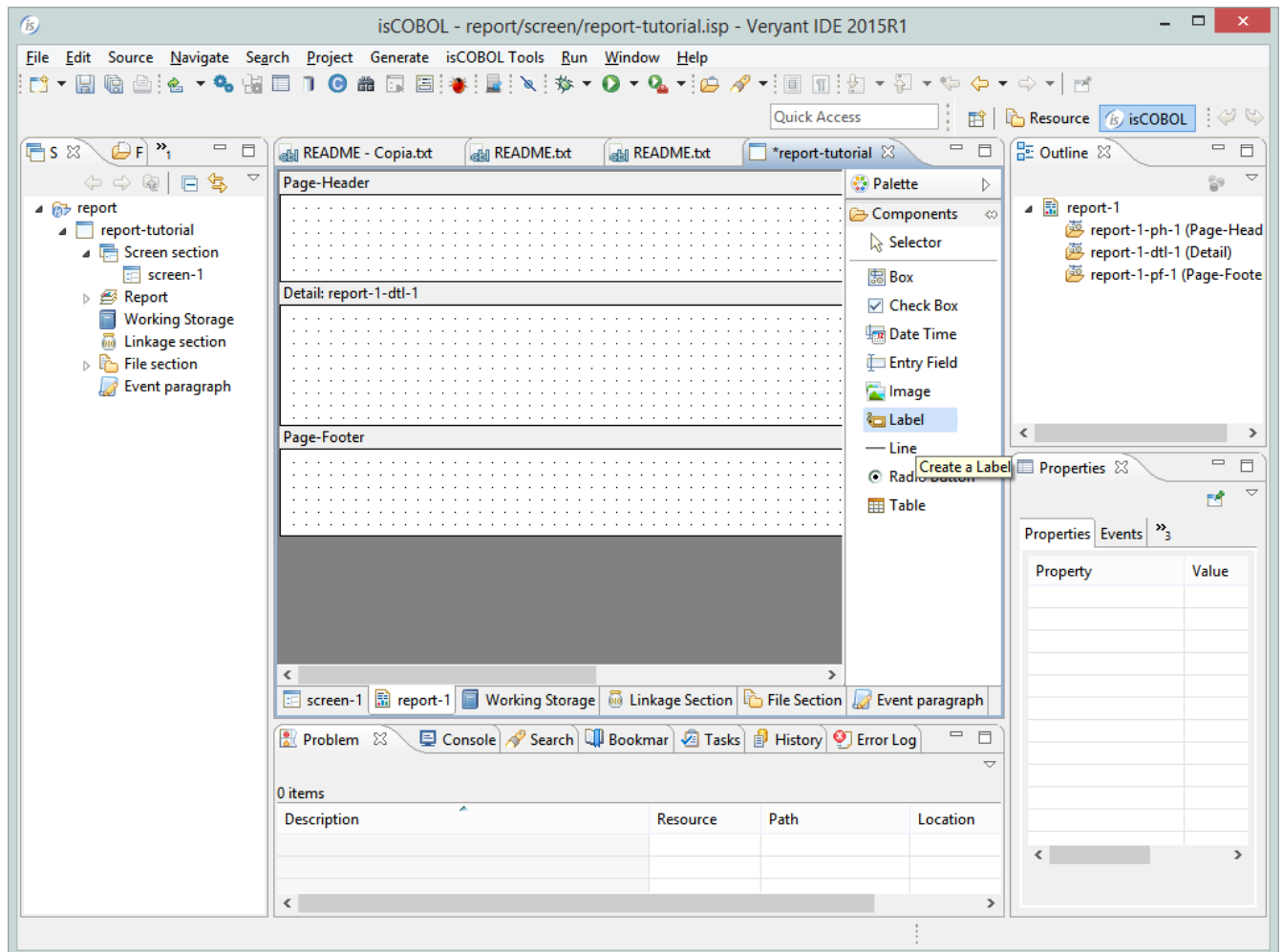


Drawing the Report

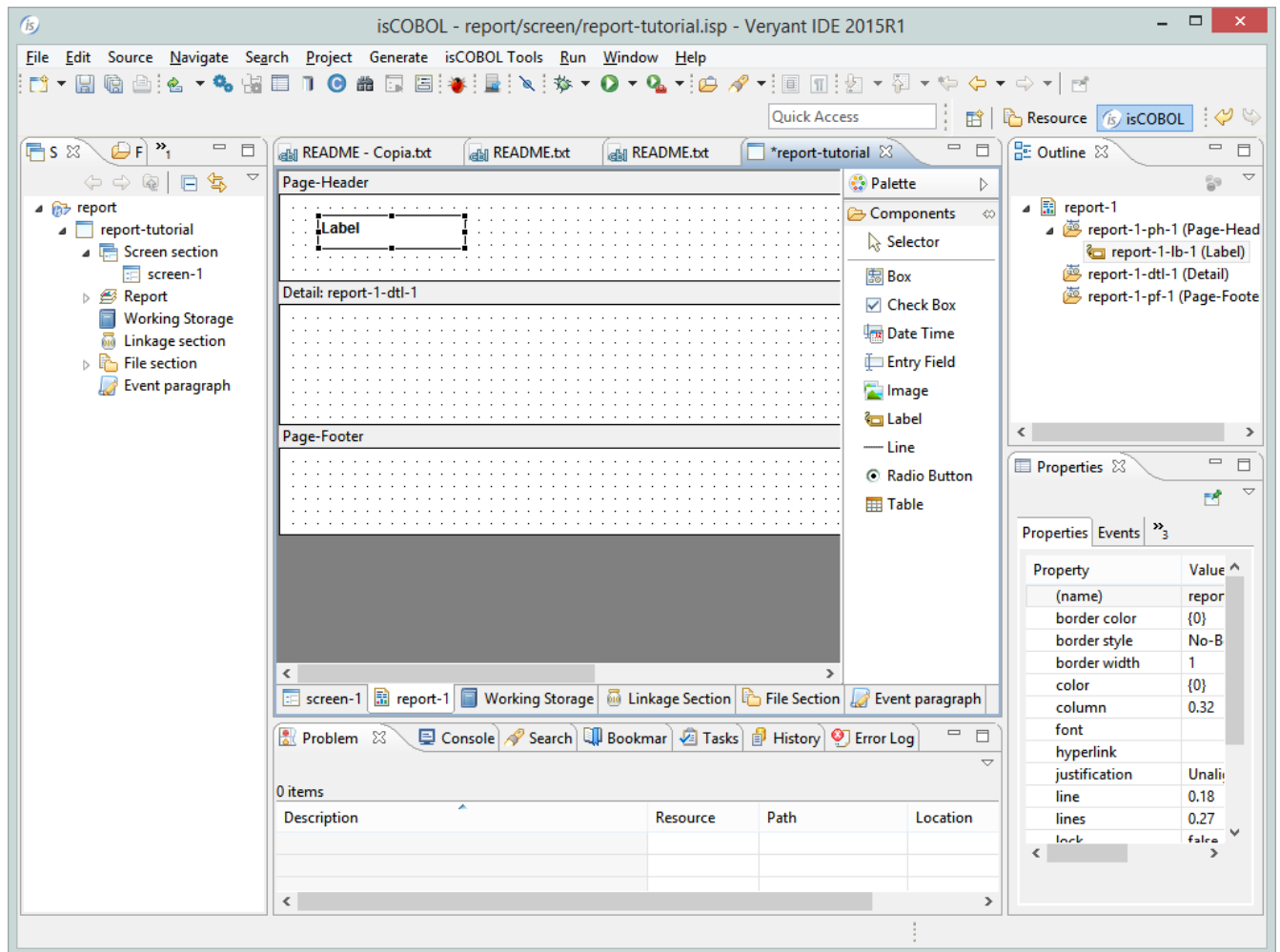
To draw a control on the Report:

1. Click and release to select the control you want to use. All the controls are on the Controls Palette. Make sure you release the mouse button before going to the next step

Figure 4. Report Designer



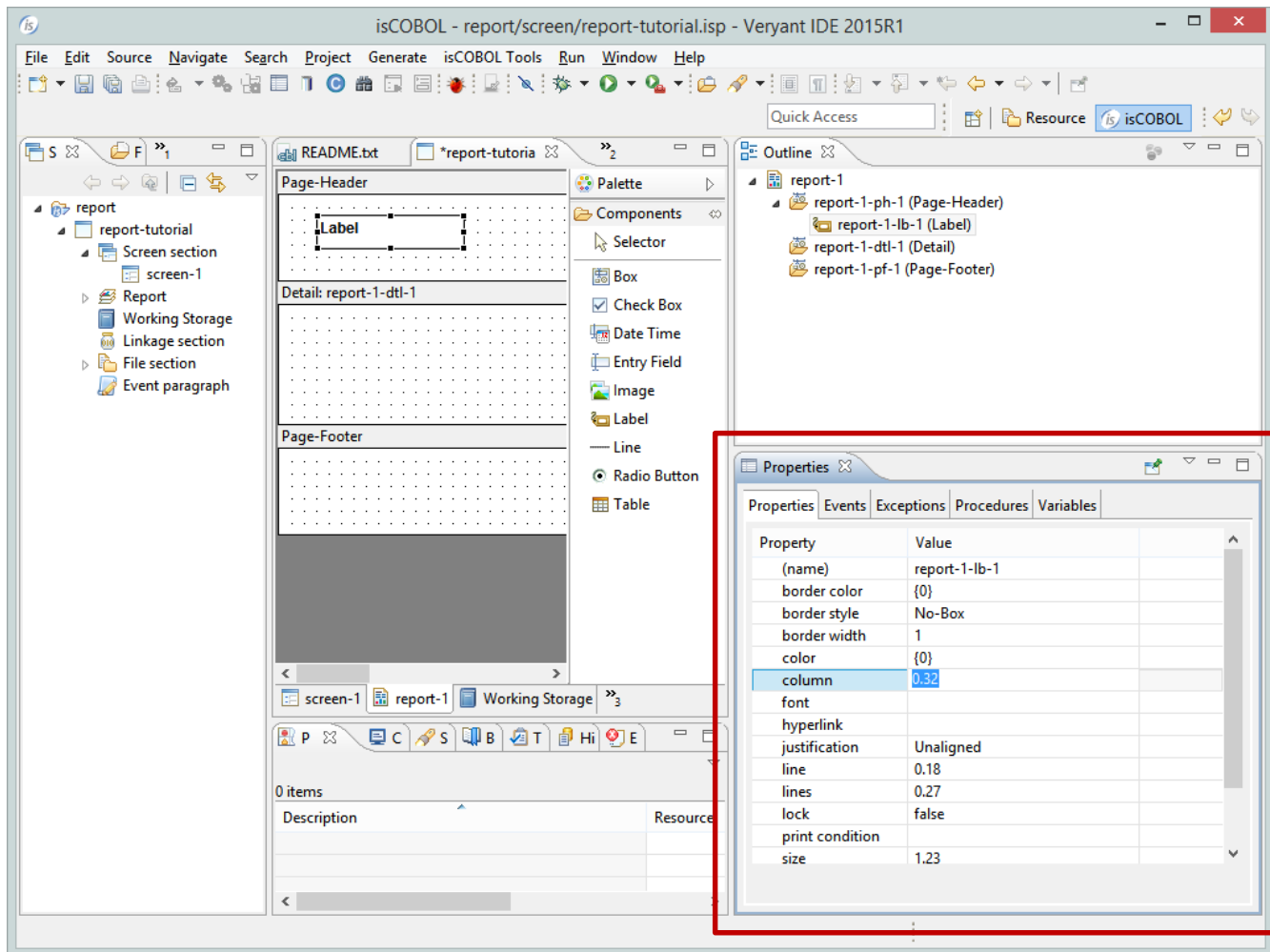
2. Click and keep the left mouse button pressed on the position you want to place the control, this is going to be the upper left corner of your control.
3. Drag the mouse to the lower right corner of your control (it will feel like you are sizing the control). Make sure you do not release the left mouse button in between step 2 and step 3.
4. Release the left button and see that the control has been successfully drawn

Figure 5. Report Designer after dragging a label

By drawing the control you automatically set LINE, COL, LINES and SIZE properties. All other properties are set to a default value.

To change the property values for one or more controls draw a rectangle around them on the screen:

1. Select the control(s).
 - a. The control is automatically selected when you draw a rectangle around it.
 - b. To select more than one control, hold Ctrl and click on the desired controls.
2. Change the properties values in the Properties view (See Figure 6).

Figure 6. Report Control properties

To easily change LINE, COL, LINES and SIZE

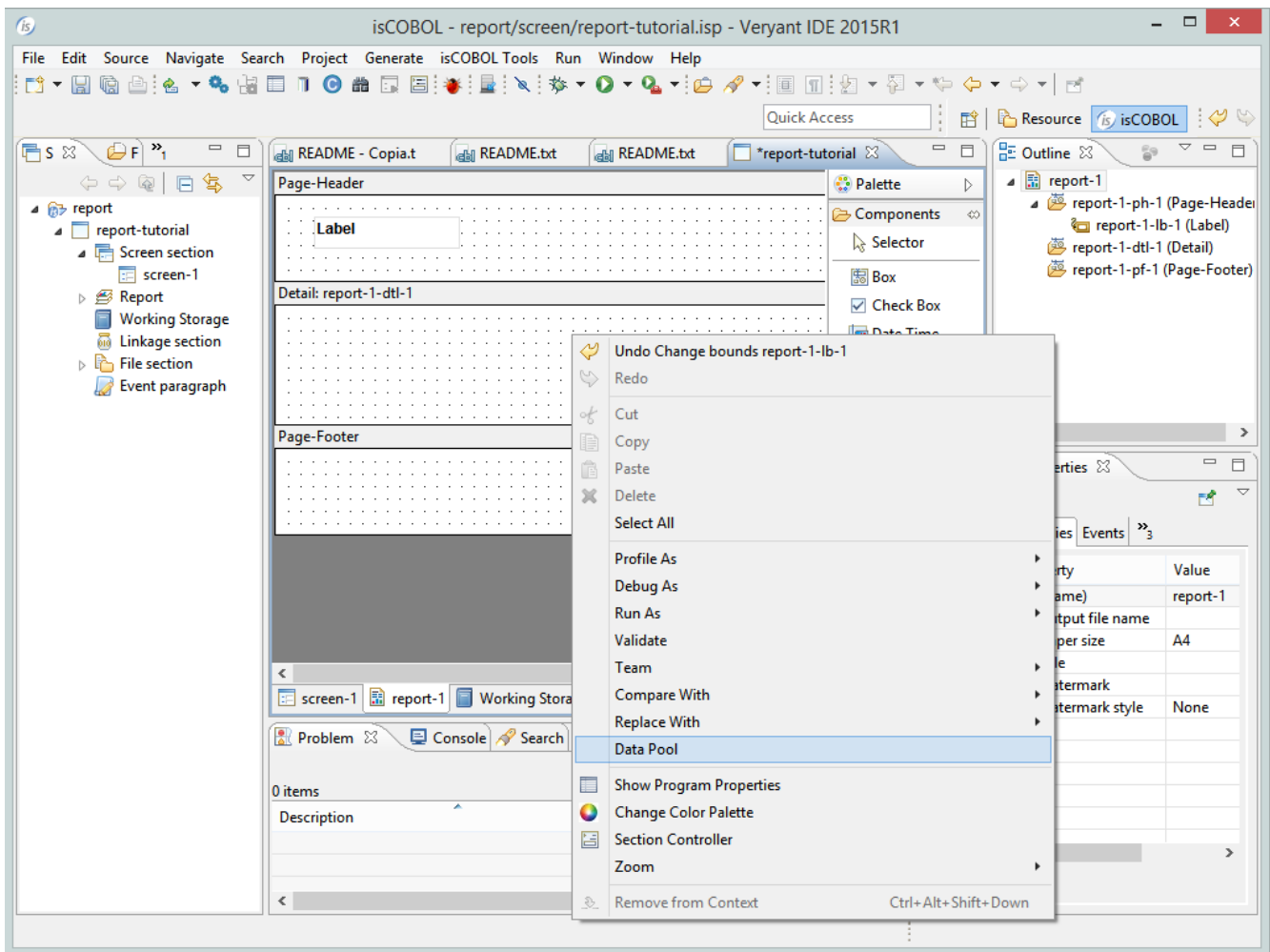
1. Select the control
2. Use the arrow keys to move the control on the Report (LINE and COL are updated)
 - a. Hold SHIFT and use the arrow keys to change the control dimensions (LINES and SIZE are updated)

Drawing the Report using Data Pool

The Data Pool is a special feature from IDE, with this feature you can draw a control and associate it to a working variable or a file's field in a simple way.

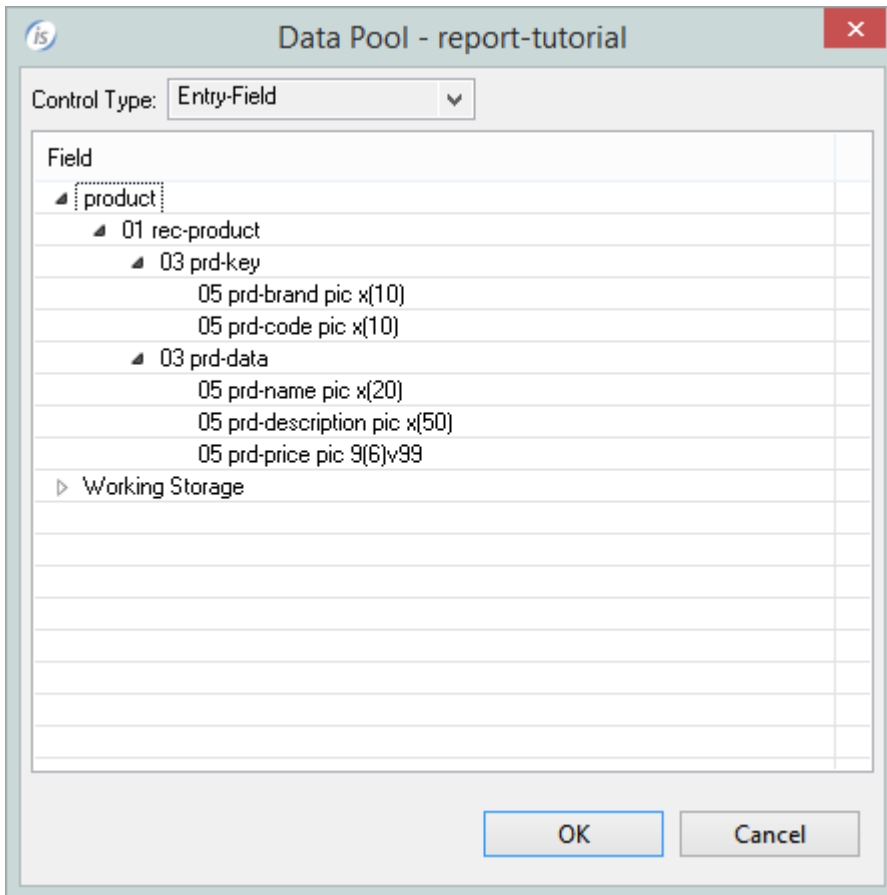
To activate this, right-click in the Report Designer and Data Pool from the pop-up menu (Se Figure 7).

Figure 7. Data Pool activation



Use this dialog (like shown in Figure 8)

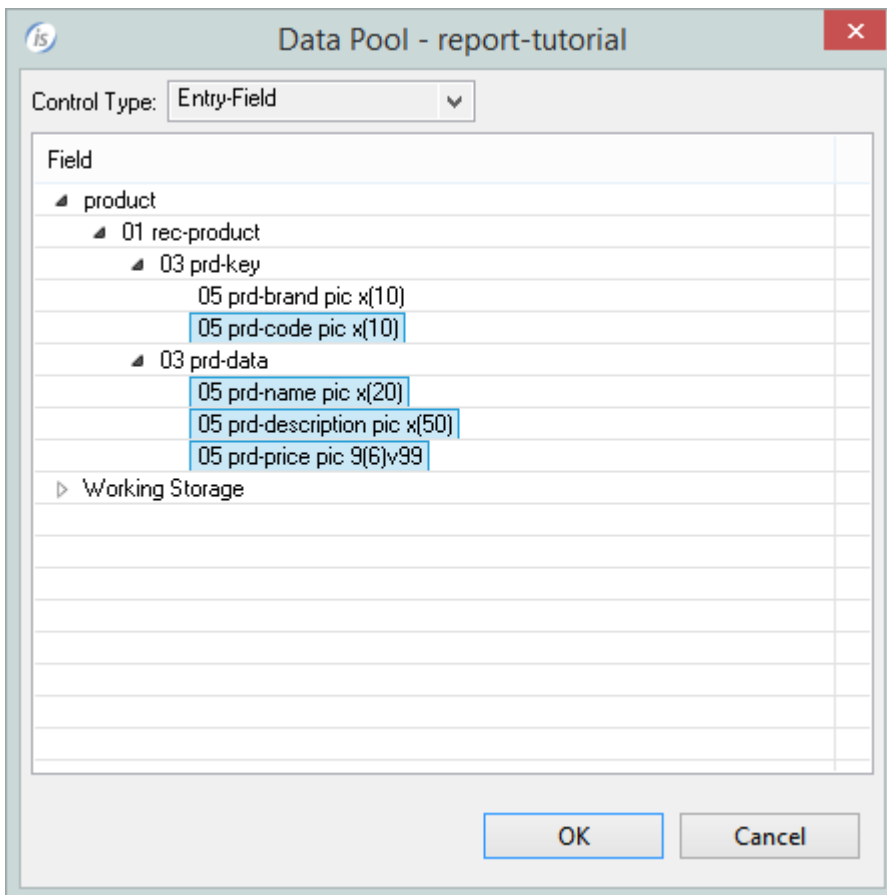
Figure 8. Data Pool



to define the kind of control to give the report by using the "Control Type".

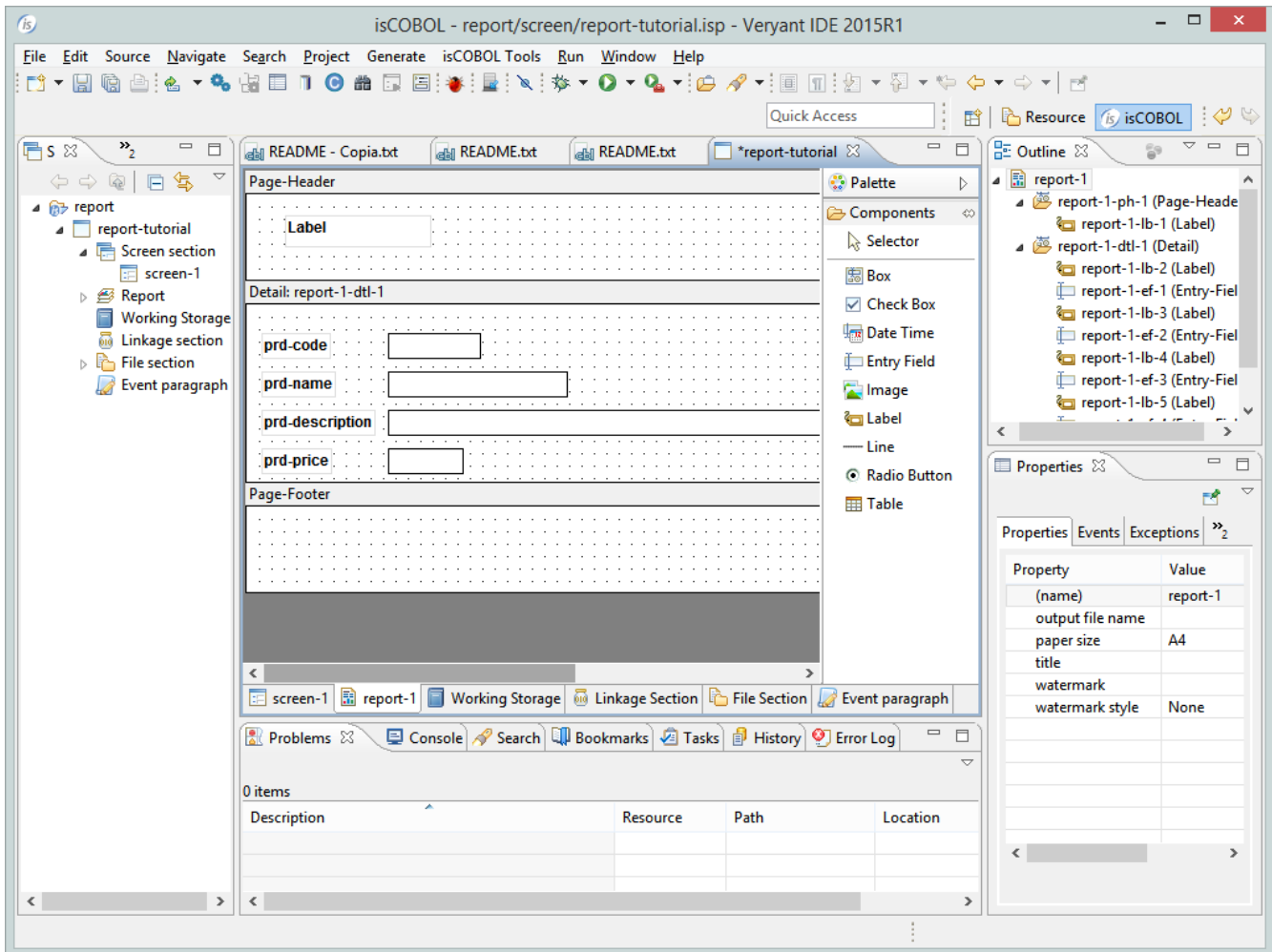
Select the field to put on the report and drag it on designer

Figure 9. Data Pool selection



This is the result after the drag & drop

Figure 10. the report after the drag & drop



Fill report logic

The next step is to add logic to retrieve the data to fill the report in our program.

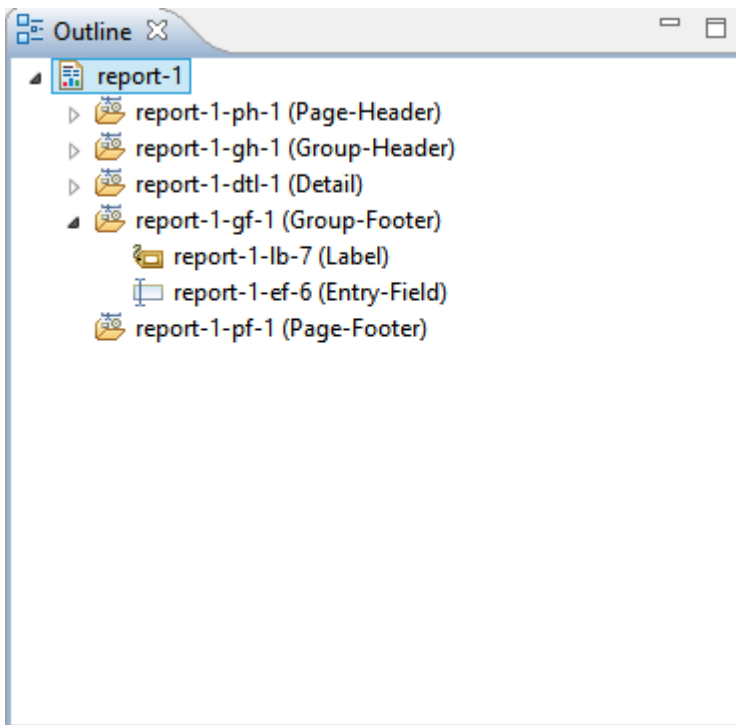
To make this we need to use 2 different entry-points of the report: “BeforDoPrint” and “AfterDoPrint”.

“BeforDoPrint”

This is a special paragraph that is executed automatically when the print of the report is requested. This paragraph is used for the first part of the “read” logic.

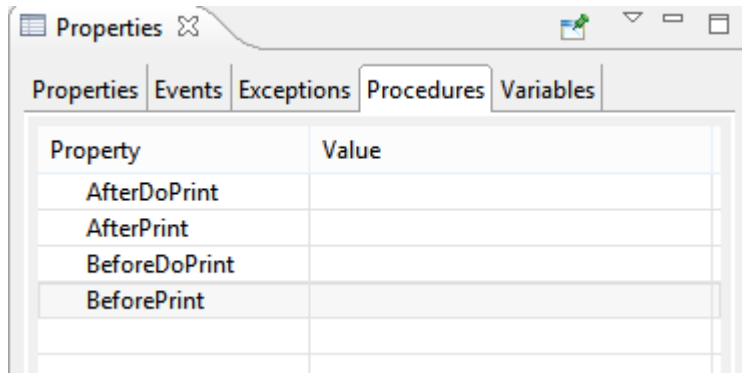
To create it select from the “Outline” the report (See Figure 11)

Figure 11. Report outline



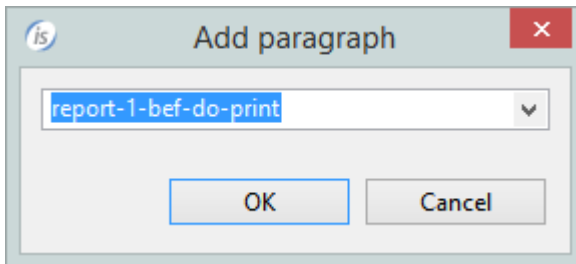
Now in the “properties” window select the “procedures” tab (See Figure 12)

Figure 12. Report Procedures



And click on the “BeforeDoPrint” row and then click on the button “...”. The popup window for the selection of the paragraph’s name will now appear. (See Figure 13).

Figure 13. Paragraph’s name selection



Press ok and go to the “Event paragraph”.

Now you can write your source code. In this way

```
report-1-bef-do-print.  
  move low-value to prd-key  
  start product key not < prd-key  
  invalid  
    move 0 to report-1-doprintrtn-loop  
  not invalid  
    move 1 to report-1-doprintrtn-loop  
  perform ACCESS-NEXT-RECORD  
END-START  
.  
ACCESS-NEXT-RECORD.  
  set print-record to false  
  perform until print-record  
    read product next no lock  
    at end  
      move 0 to report-1-doprintrtn-loop  
      exit perform  
  end-read  
*   This is my custom print condition  
  if print-condition = true  
    continue  
  else  
    set print-record to true  
  end-if  
*  
  end-perform.
```

Where report-1-doprintrtn-loop is a variable declared automatically by the IDE. The first part of the name (report1) is variable and is the name of the report, se second part is fixed. When we set this variable to 0 the print loop is interrupted, when we set it to 1 the print loop continues.

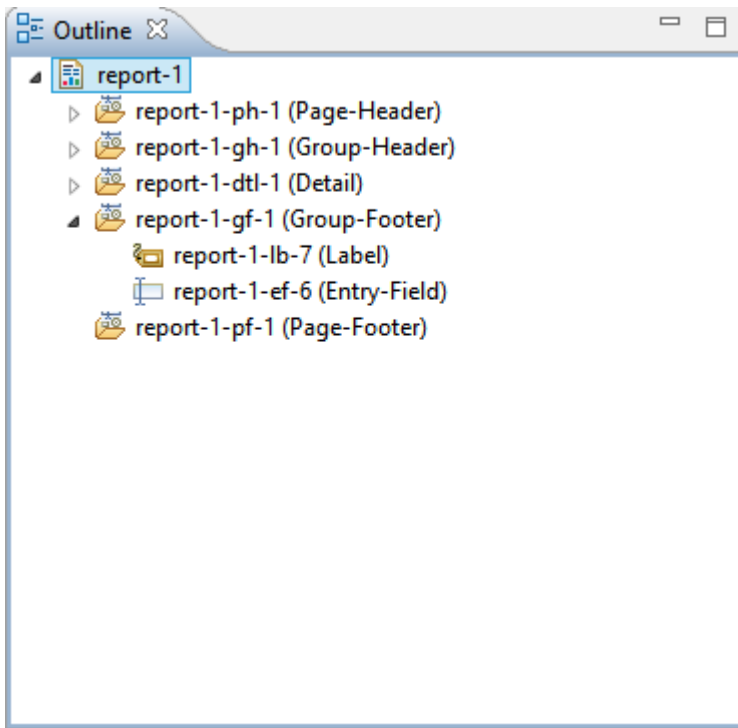
ACCESS-NEXT-RECORD is a customer paragraph that contains part of the “read” logic. In this case we have to introduce a perform, that using a simple condition, is used to discard dome records.

“AfterDoPrint”

This is a special paragraph that is executed automatically after the print of each detail.

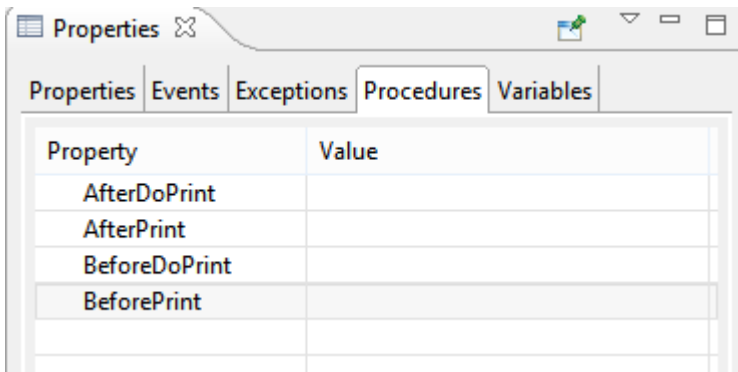
To create it select the report from the “Outline” window (See Figure 14)

Figure 14. Report outline



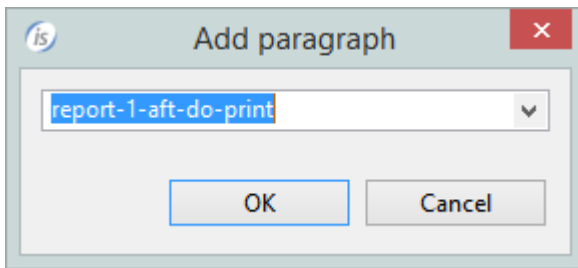
Now in the "properties" window select the "procedures" tab (See Figure 15)

Figure 15. Report Procedures



And click on the "AfterDoPrint" row and then click on the button "...". The popup window for the selection of the paragraph's name will appear. (See Figure 16).

Figure 16. Paragraph's name selection.



Press ok and go to the "Event paragraph".

Now you can write your source code. In this way

```
report-1-aft-do-print.  
    PERFORM ACCESS-NEXT-RECORD  
    .
```

Execute the report from a cobol program

When your program's logic needs to produce the report you can make a perform to some specific paragraph generated automatically by the IDE.

The isCOBOL IDE provides 4 different kinds of production of the report:

- Print to the preview
- Print to the default printer
- Print to a printer with the printer selection
- Print to an html file on the disk

Print to the preview

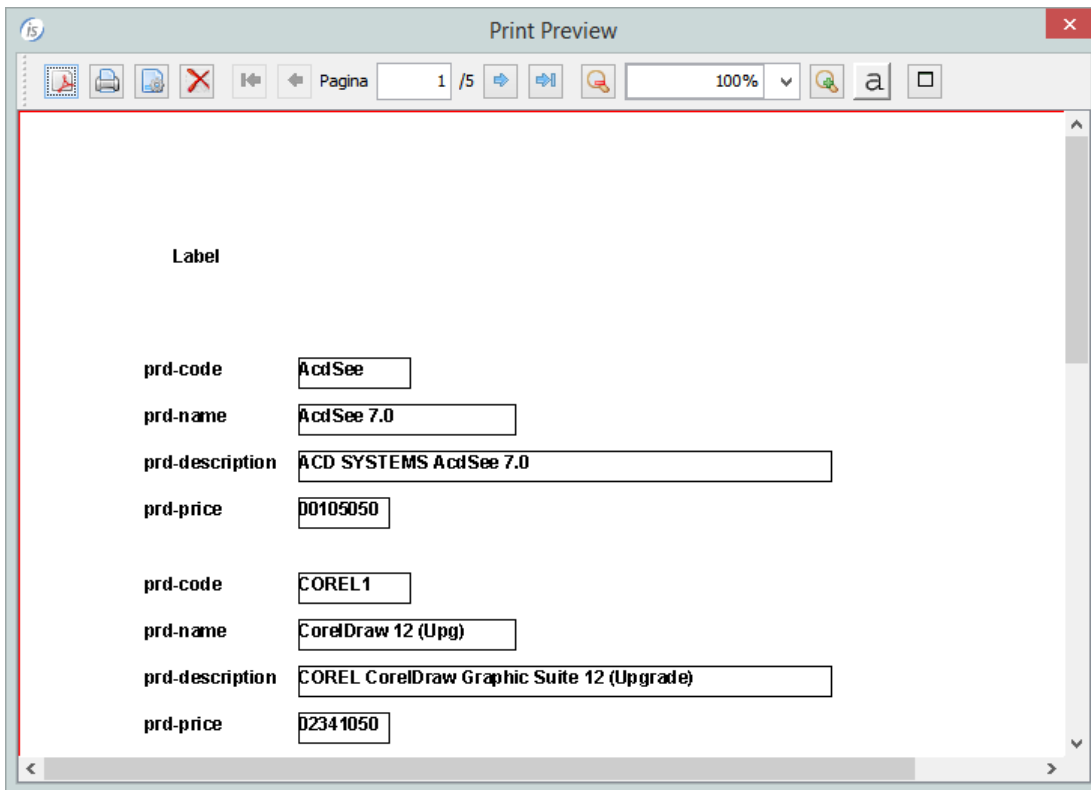
Opens the Report in a Print Preview (See Figure 17). From the preview we can print the report or we can save it as a PDF or Excel file.

To call the print preview from a cobol program we need to make a perform to the following paragraph

```
perform is-report-1-preview
```

where report-1 is the name of the report

Figure 17. Report preview



Print to the default printer

Prints the Report on the active printer.

To call the print from the cobol program we need to make a perform to the following paragraph

```
perform is-report-1-print
```

where report-1 is the name of the report

Print to a printer with the printer selection

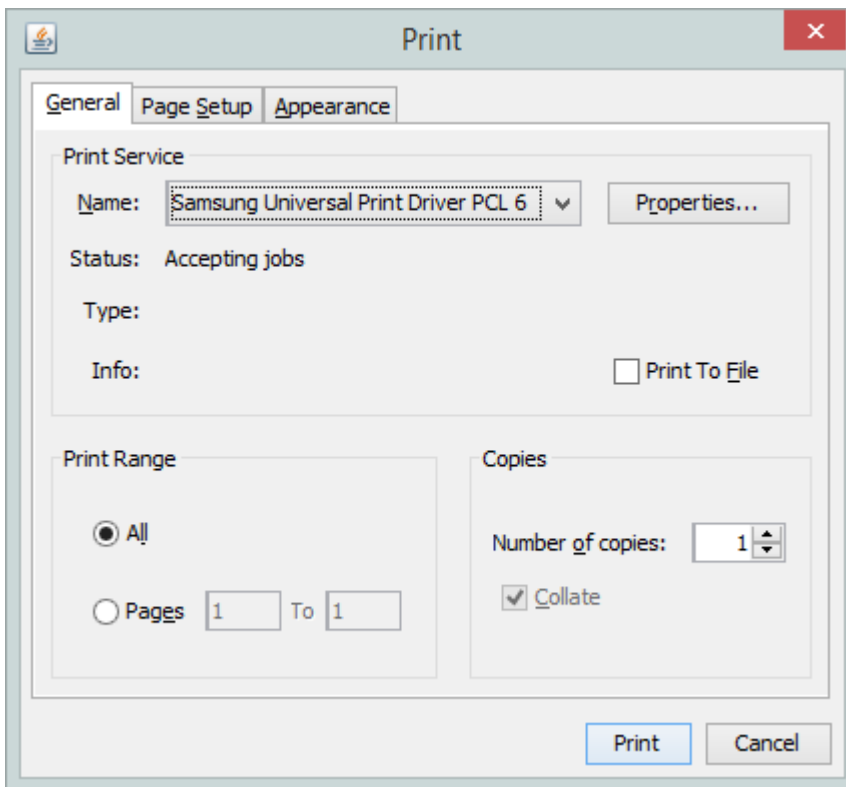
This shows the Choose Printer dialog (See Figure 18), then prints the Report on the chosen printer.

To call the print from a cobol program we need to make a perform to the following paragraph

```
perform is-Report-1-Setup-Print
```

where report-1 is the name of the report.

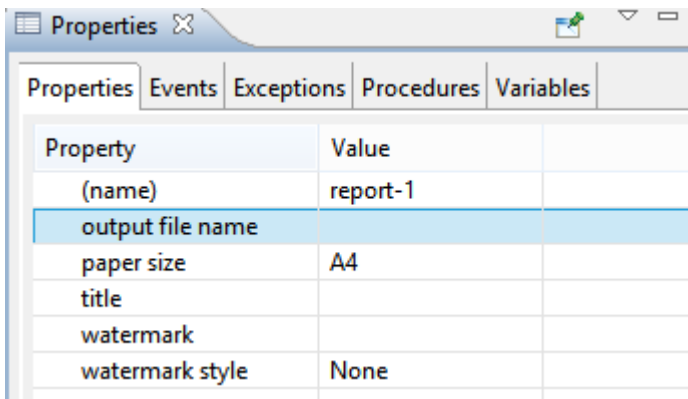
Figure 18. Report's printer selection



Print to an html file on the disk

Saves the Report to an HTML file on the disk. It uses the “output file name” (See Figure 19) to specify the name of the HTML file. If you only specify the name, the file is created following the rules of file-prefix.

Figure 19. “Output file name” property



To create the HTML file from a cobol program we need to make a perform to the following paragraph

```
perform is-Report-1-Print-ToFile
```

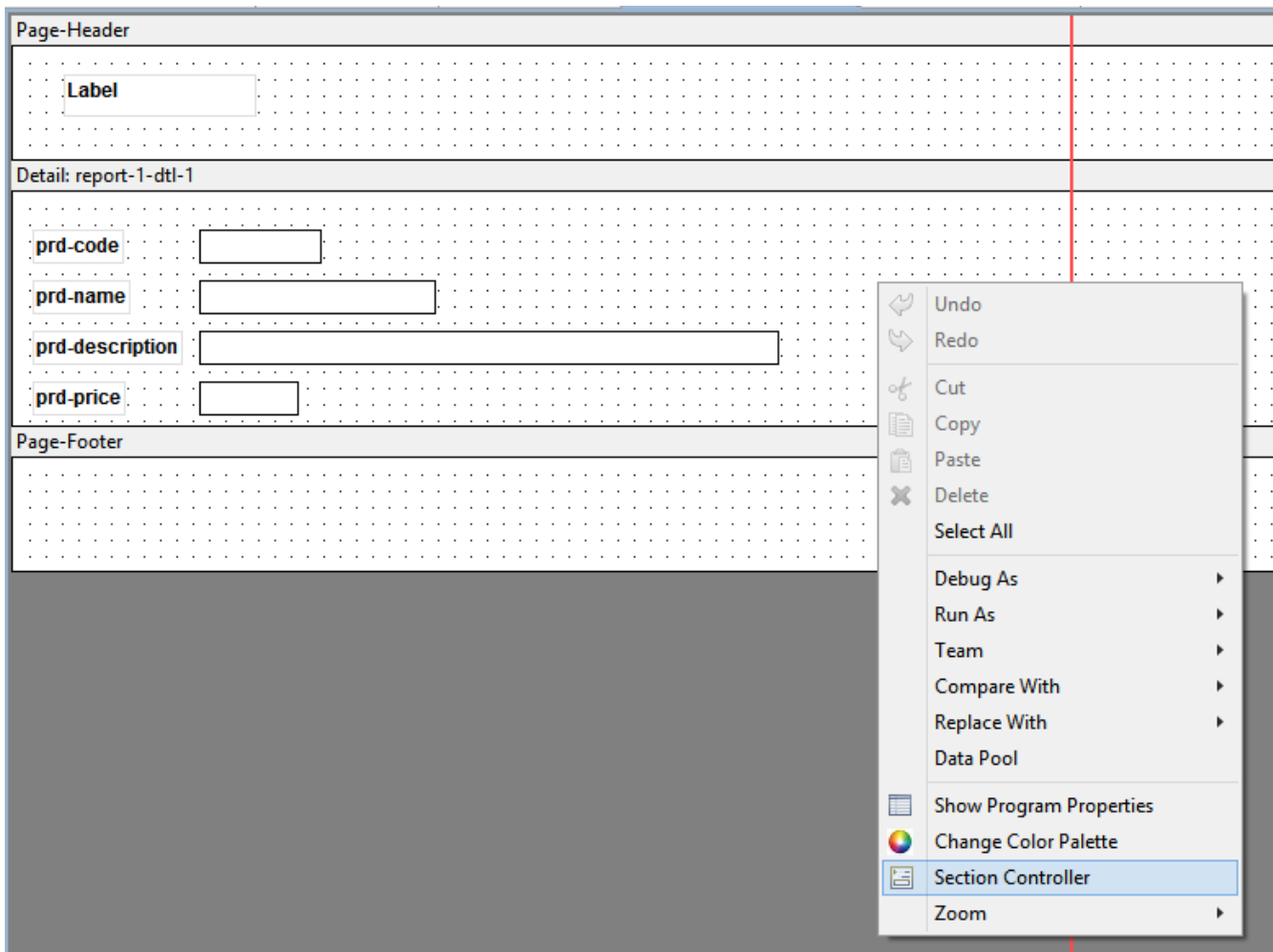
where report-1 is the name of the report

Add a group section to the report

In some cases it is useful adding a group logic to a report. To make this we can use the “section controller” feature.

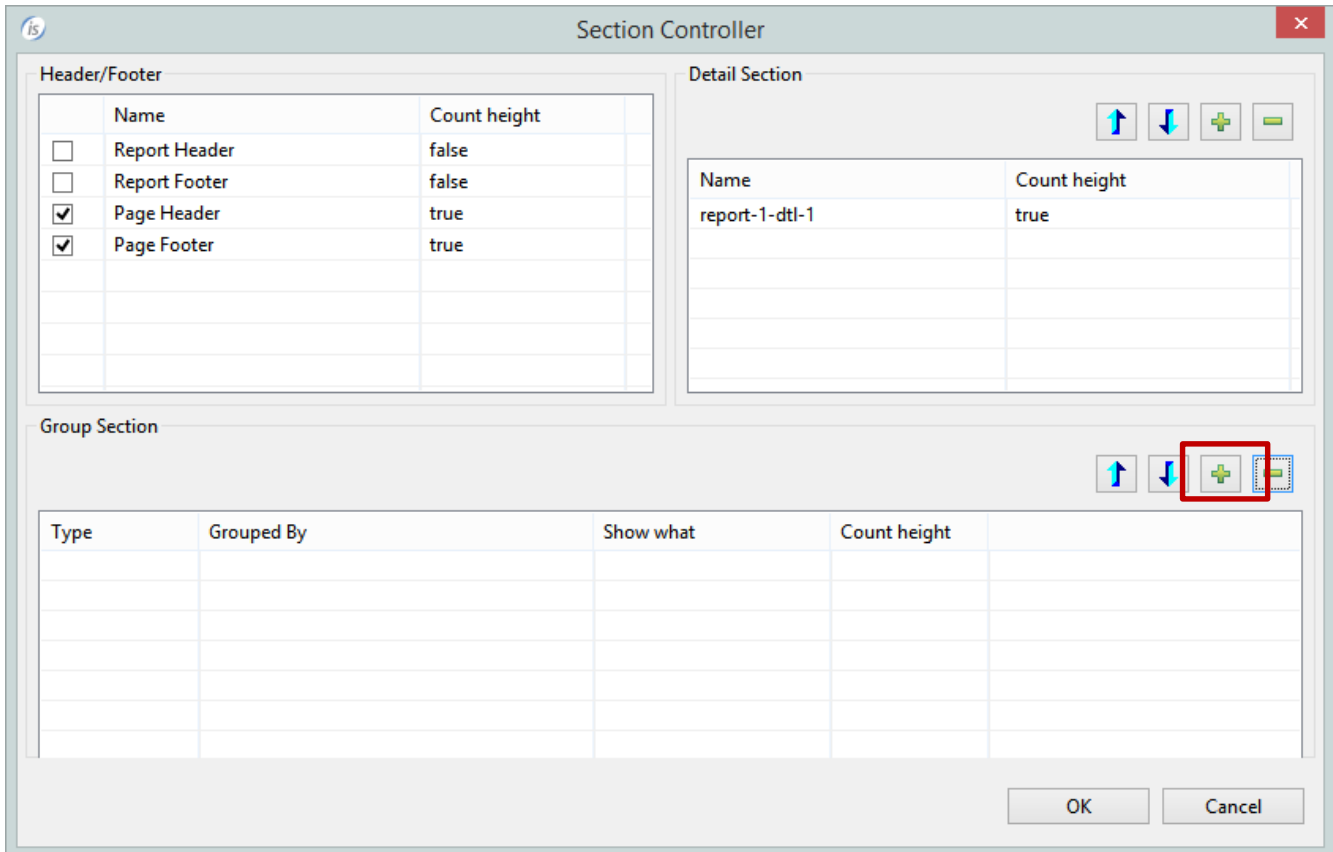
To activate right click on the report and select “Section controller” on the popup menu (See Figure 20).

Figure 20. Section controller activation



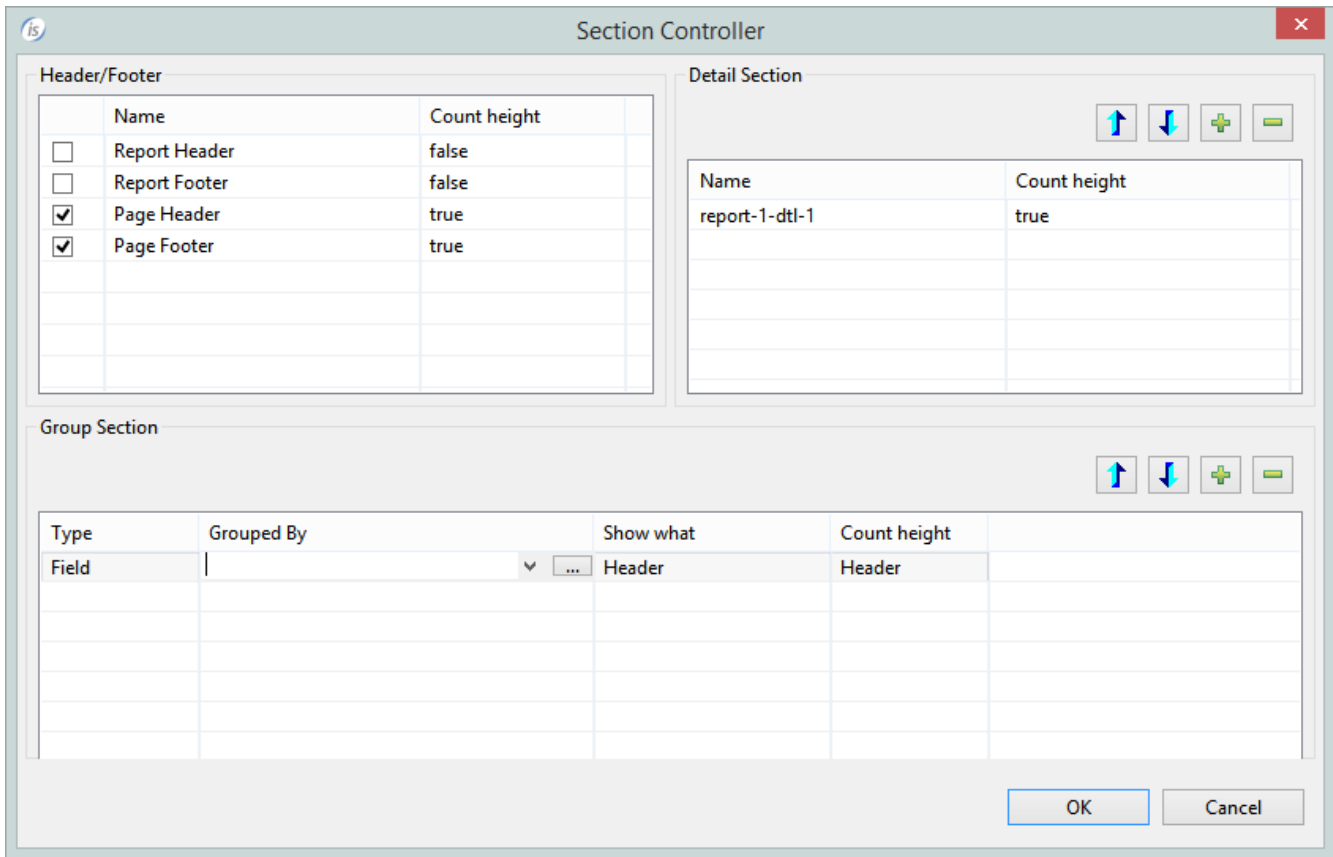
When the section controller window appears, press the "PLUS" button to add the variable used for the grouping (See Figure 21).

Figure 21. Section Controller



After that the first group section is created (See Figure 23).

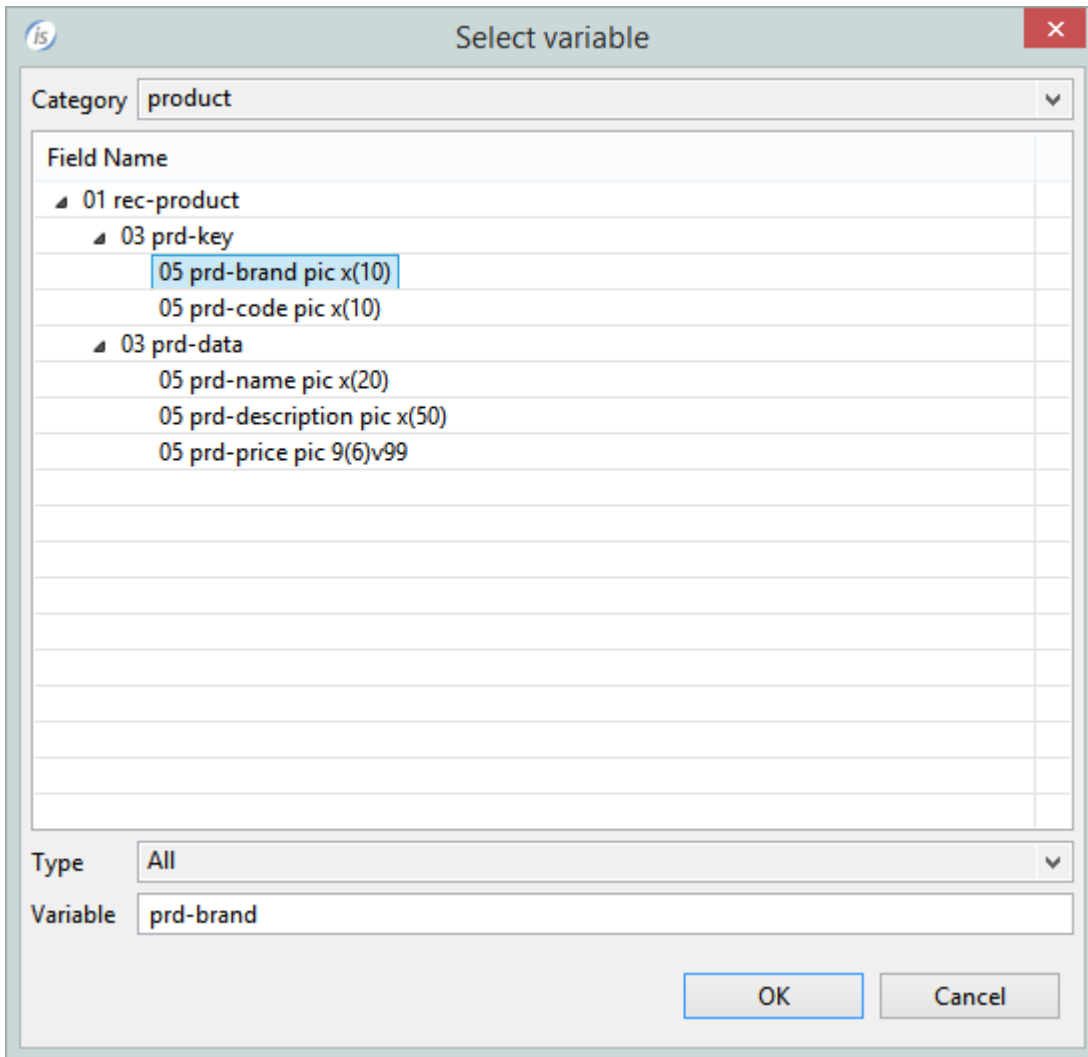
Figure 23. Creation of a group



Double click on the column "Grouped By" to specify the variable.

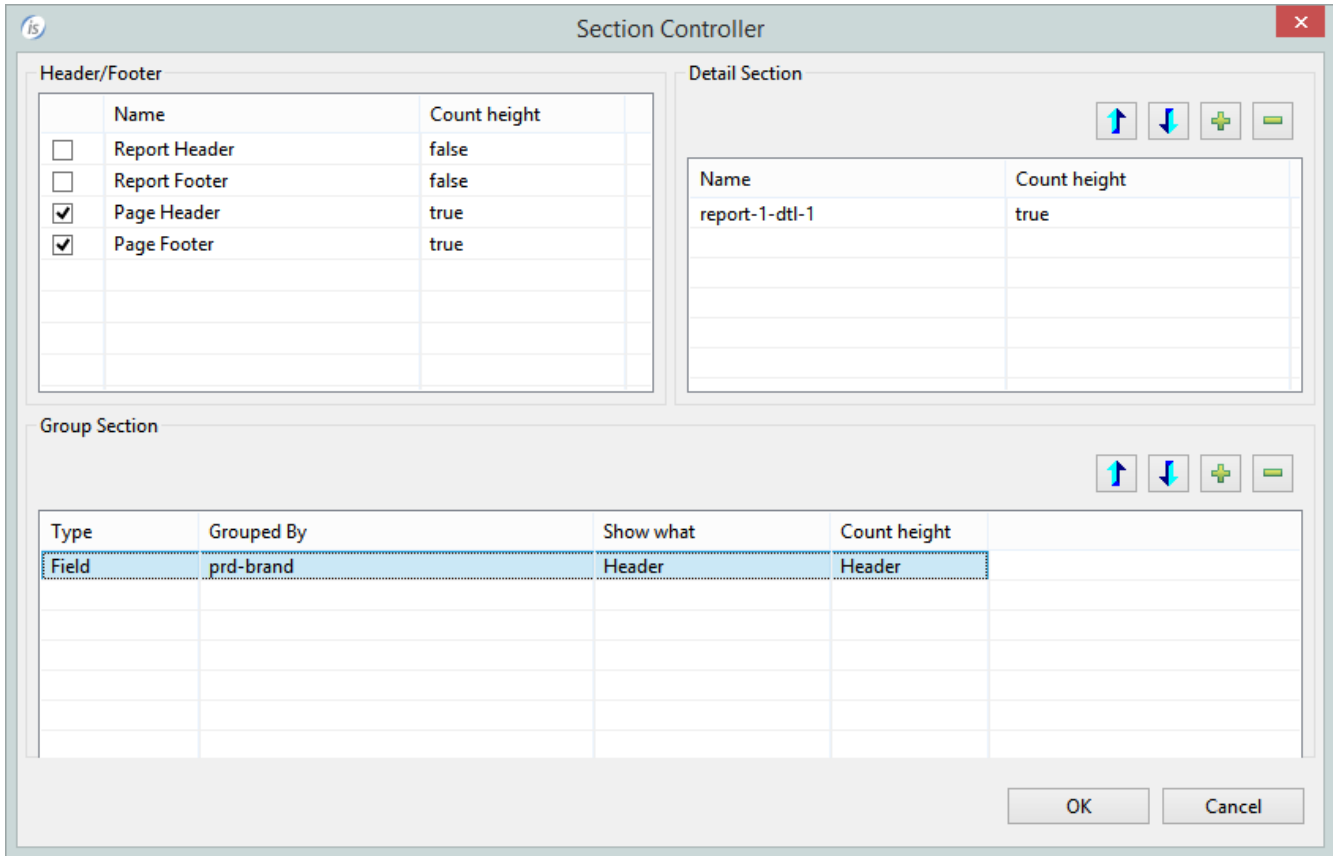
Select the field, in this case prd-brand (See Figure 25),

Figure 25. Variable selection from “product” file



and press ok.

Figure 26. Section controller with group



Press ok to close the section controller and return to the report

Now in the report you have a new section (See Figure 27)

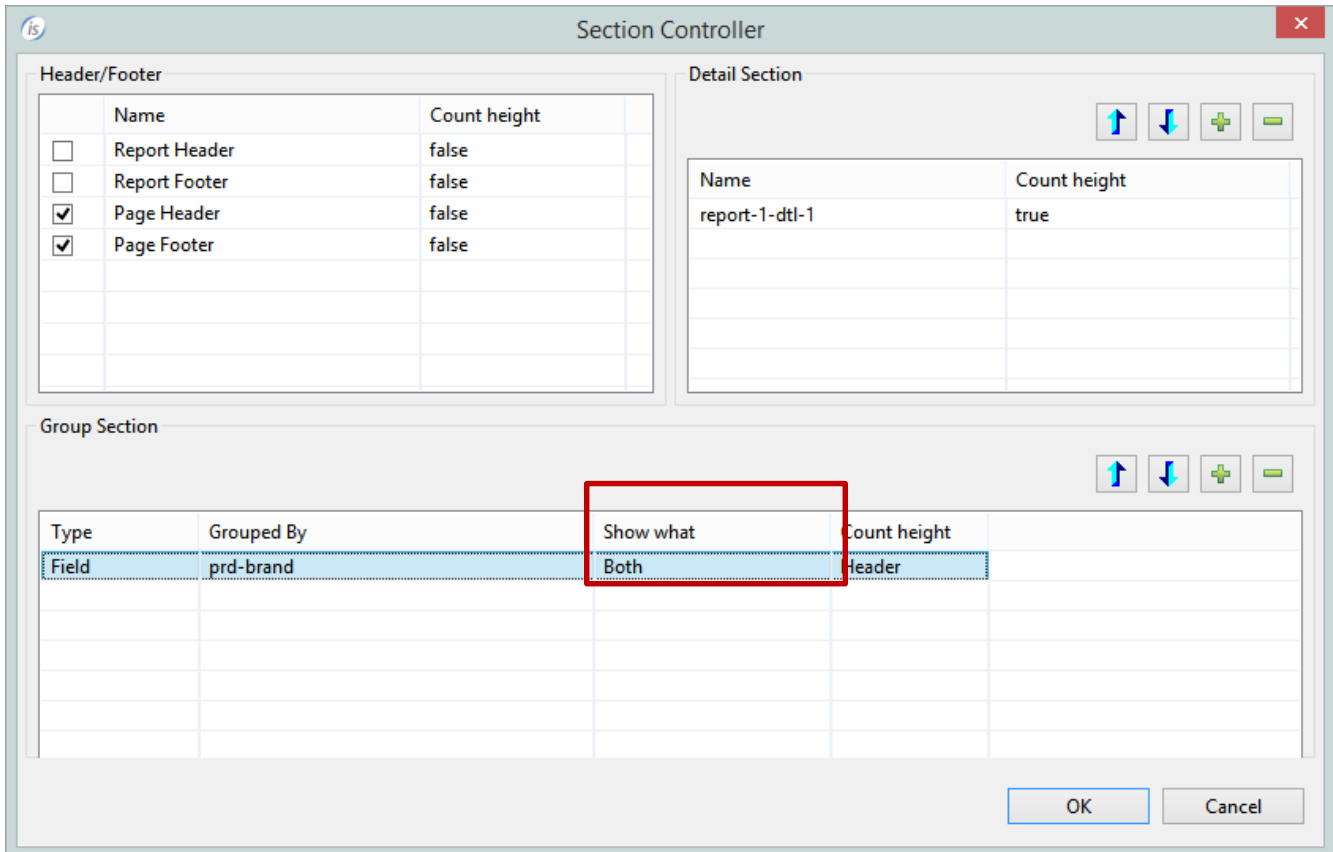
Figure 27. Report with the new Group-Header section

The screenshot displays a report layout with several sections. At the top is the 'Page-Header' section containing a 'Label' field. Below it is a 'Group-Header: prd-brand (Field)' section, which is highlighted with a red rectangular border. This section is currently empty. Underneath is the 'Detail: report-1-dtl-1' section, which contains four data fields: 'prd-code', 'prd-name', 'prd-description', and 'prd-price', each with a corresponding input box. At the bottom is the 'Page-Footer' section, which is also empty. A vertical red line is present on the right side of the report area.

This section is printed when the content of the field "prd-brand" changes.

If I want to add a new section at the end of the group I need to return to the “Section Controller” and select “Both” into the “Show what” column (See Figure 28).

Figure 28. Section controller, group section settings



Now the report has the following section (See Figure 29)

Figure 29. Report with the new Group-Header and Group-Footer sections

The screenshot shows a report designer interface with a dotted grid background. A vertical red line is positioned on the right side of the grid. The report is divided into several sections:

- Page-Header:** Contains a single field labeled "Label".
- Group-Header: prd-brand (Field):** Contains a single field labeled "Brand".
- Detail: report-1-dtl-1:** Contains four fields: "prd-code", "prd-name", "prd-description", and "prd-price".
- Group-Footer: prd-brand (Field):** This section is currently empty.
- Page-Footer:** This section is currently empty.

Now we want to add a new field to the Group-footer section. This new field will show the sum of all prices of each brand's product .

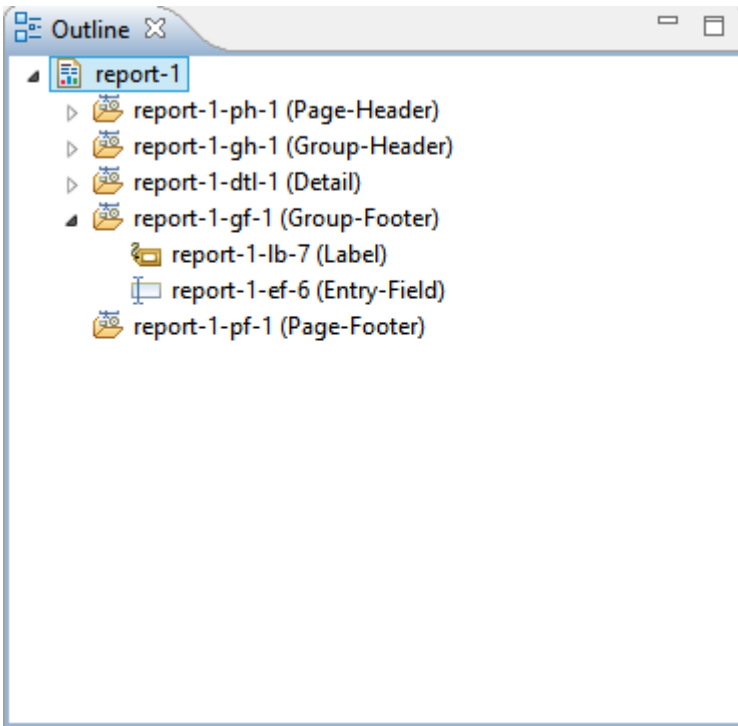
To make this we need to introduce a working variable that contains the total, like (for example)

```
77 brand-price pic 9(7)v99.
```

To have this variable initialized we use the "BeforPrint" entry-point of the report. This is a special paragraph that is executed automatically when the print of the report is requested.

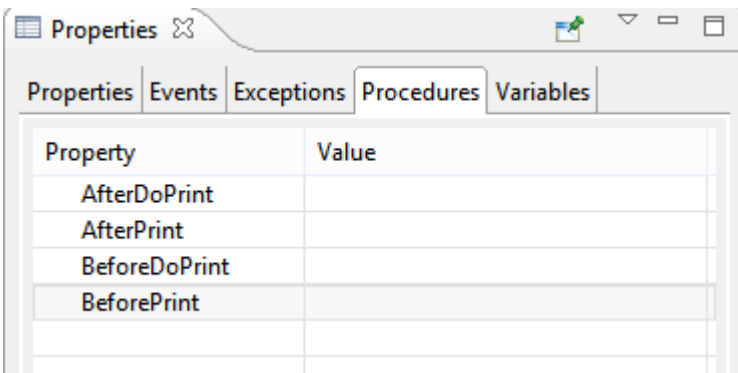
To create it select from the “Outline” the report (See Figure 30)

Figure 30. Report outline



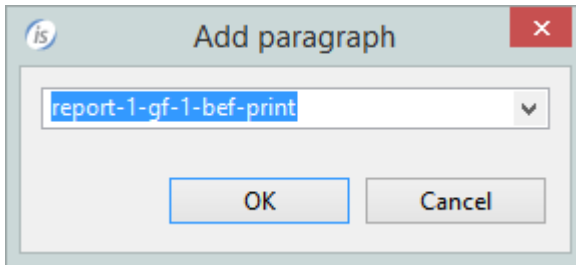
Now in the “properties” window select the “procedures” tab (See Figure 31)

Figure 31. Report Procedures



And click on the "BeforePrint" row and then click on the button "...". The popup window for the selection of the paragraph's name will appear (See Figure 32).

Figure 32. Selection of paragraph name



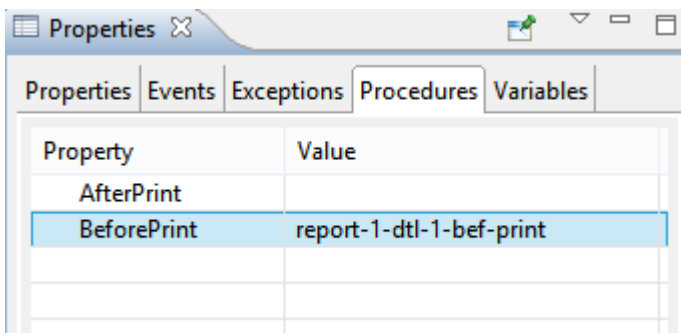
Press ok and go to the "Event paragraph".

Now you can write your source code. In this way

```
report-1-gh-1-bef-print.  
    move zero    to brand-price
```

Now we need to increase the brand price for each brand's record . To make this we go to the report designer and select the "detail section" and go to the property to create the BeforePrint procedure (See figure 32)

Figure 32. Report Procedures with the BeforePrint



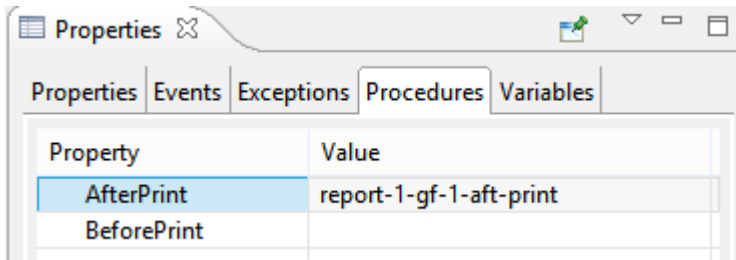
This paragraph is executed before the printing of each detail.

In this case we use this paragraph to increase the brand price

```
report-1-dtl-1-bef-print.  
  add prd-price to brand-price
```

Now we need to set to zero this variable. To make this we use the “AfterPrint” entry-point of the group-footer section (See Figure 33)

Figure 33. Report Procedures



In this case

```
report-1-gf-1-aft-print.  
  move zero to brand-price
```

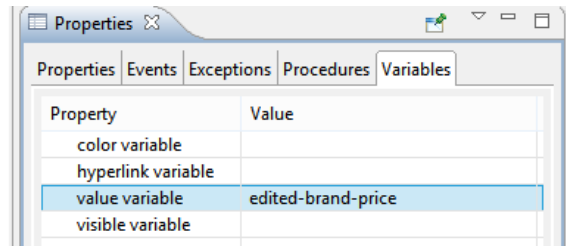
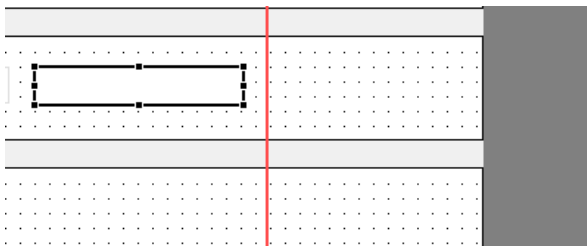
Now we can add the brand price to the report, but we cannot use this variable in the report. We need to use a variable with edited picture.

We declare the following variable into the working storage:

```
77 edited-brand-price pic z,zzz,zz9.99.
```

and link this variable into the report designer (See Figure 34).

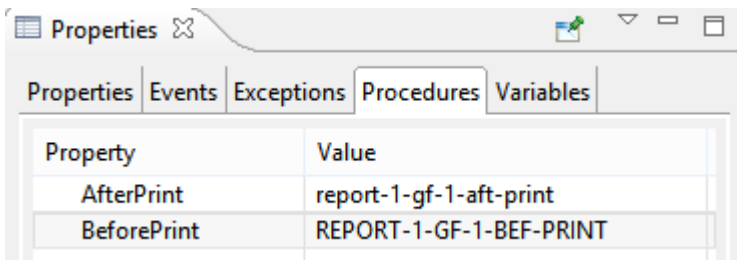
Figure 34. Value variable settings



Now we need to move the brand-price variable to the edited-brand-price-variable.

We can make this by using the "BeforePrint" entry-point of the group-footer section (See Figure 36)

Figure 36. Report Procedures



This paragraph is executed before the printing of each brand group. In our case

```
report-1-gf-1-bef-print.  
    move brand-price to edited-brand-price  
    .
```