



**Genero**®  
report writer

## REPORT OUTPUT OPTIONS #2

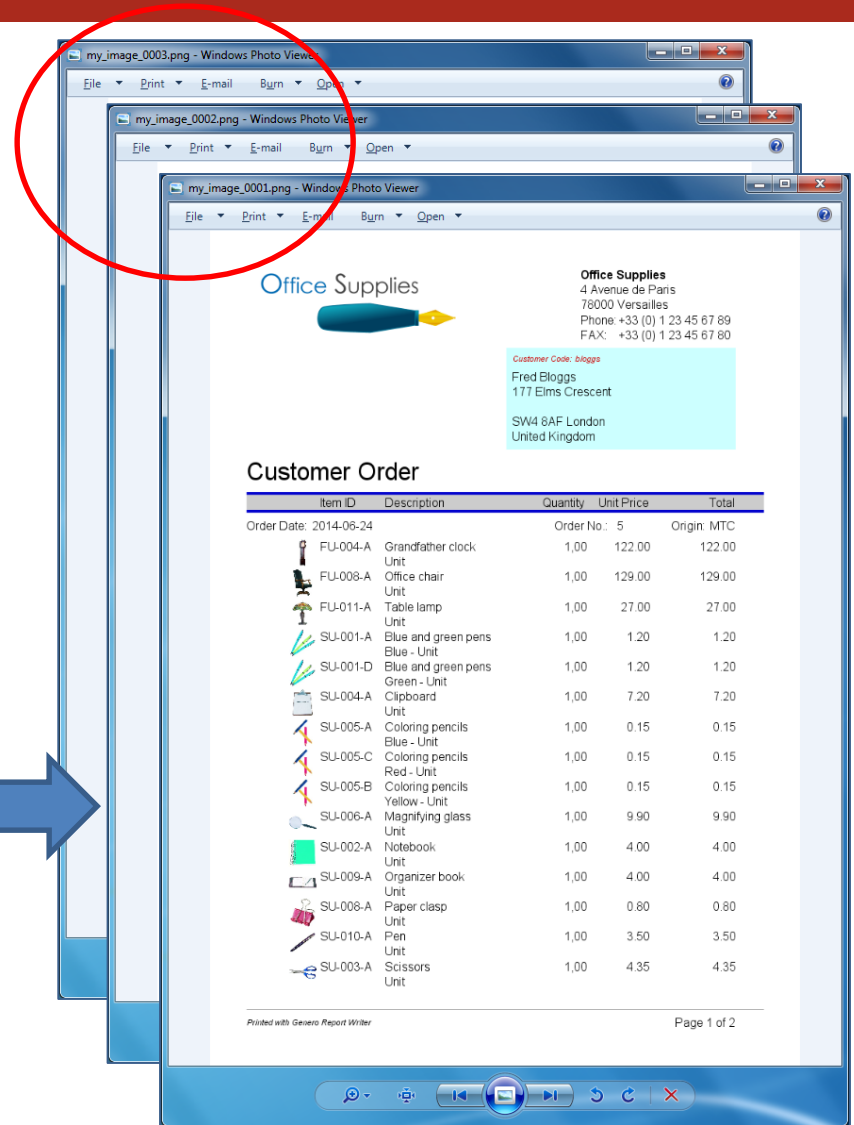
Using Genero Report Writer  
GRS 3.00

- ImageRenderer output
  - png output format
  - Print 3 pages: get 3 files
  - Prefix file name

```

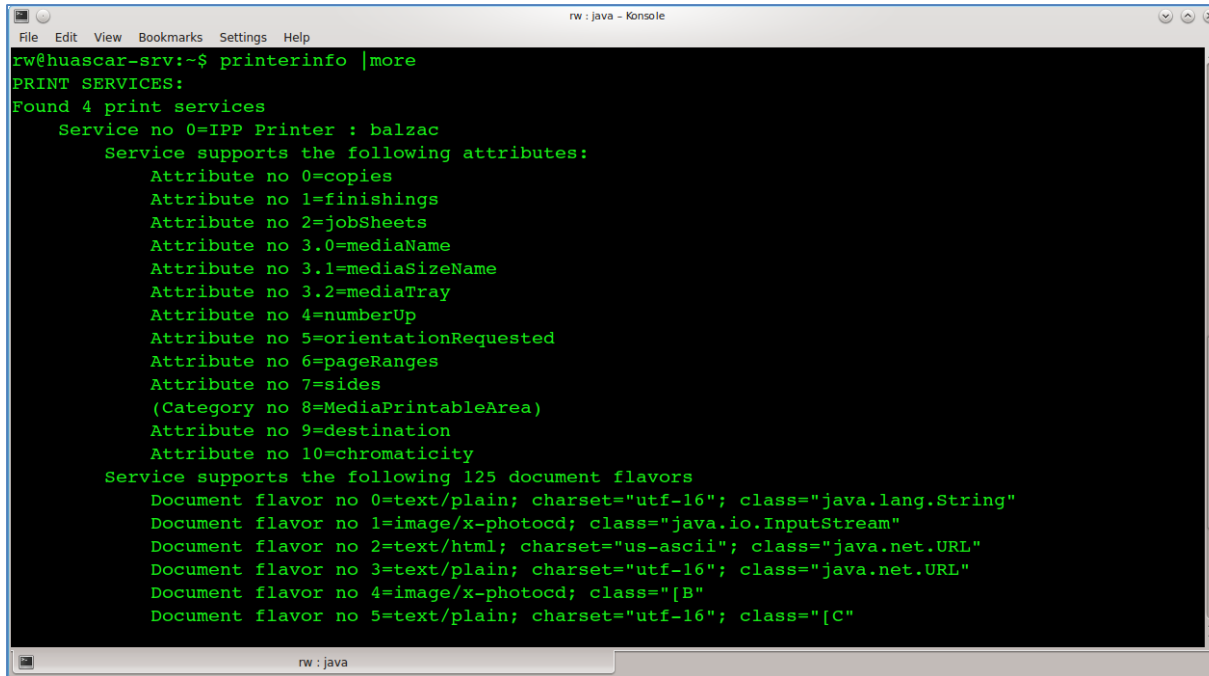
24  if (args.length == 0) {
25      designFile = "OrderReport.4rp";
26  } else {
27      designFile = args[0];
28  }
29  ImageRenderer renderer = new ImageRenderer();
30  //Configuring the Image output:
31  renderer.setFileType("png");
32  renderer.setFileNamePrefix("my_image_");
33  renderer.setFromPage(1);
34  renderer.setToPage(3);
35
36  FourRpLayouter report = new FourRpLayouter(designFile, renderer);
37
38  //Run report from XML file:
39  report.runFromXML("my_data_file.xml");
40

```



- Use graphical printers
  - Windows: Install printer driver
  - Linux/Unix: Install printer with CUPS
- Two print options
  - Silent Server printing
  - Print from the application opening the report document
- Printers to be used: printerinfo
  - In GREDIR/bin (Silent Server printing)

- Check available Server printers: printerinfo

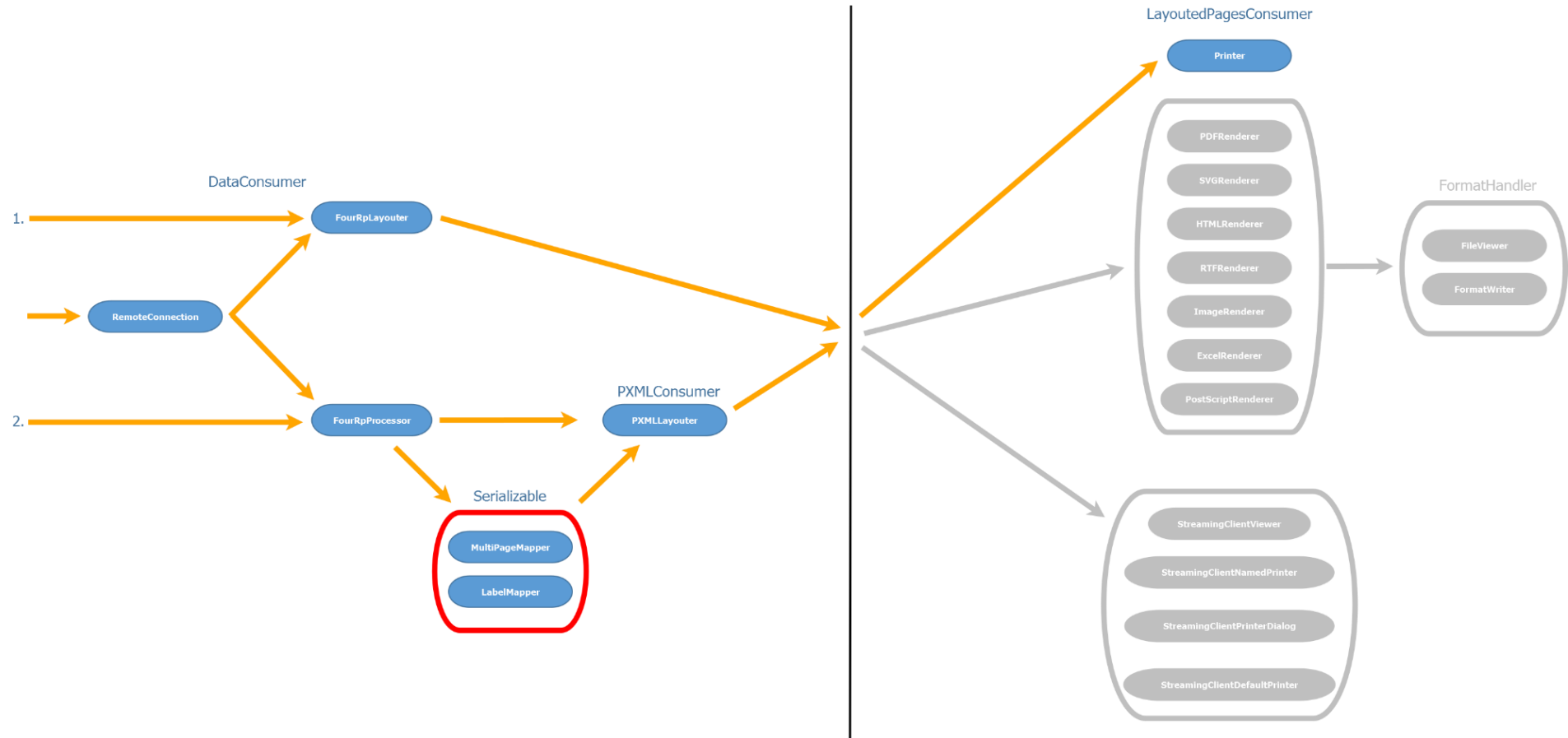


```
rw@huascar-srv:~$ printerinfo |more
PRINT SERVICES:
Found 4 print services
  Service no 0=IPP Printer : balzac
    Service supports the following attributes:
      Attribute no 0=copies
      Attribute no 1=finishings
      Attribute no 2=jobSheets
      Attribute no 3.0=mediaName
      Attribute no 3.1=mediaSizeName
      Attribute no 3.2=mediaTray
      Attribute no 4=numberUp
      Attribute no 5=orientationRequested
      Attribute no 6=pageRanges
      Attribute no 7=sides
      (Category no 8=MediaPrintableArea)
      Attribute no 9=destination
      Attribute no 10=chromaticity
    Service supports the following 125 document flavors
      Document flavor no 0=text/plain; charset="utf-16"; class="java.lang.String"
      Document flavor no 1=image/x-photocd; class="java.io.InputStream"
      Document flavor no 2=text/html; charset="us-ascii"; class="java.net.URL"
      Document flavor no 3=text/plain; charset="utf-16"; class="java.net.URL"
      Document flavor no 4=image/x-photocd; class="[B"
      Document flavor no 5=text/plain; charset="utf-16"; class="[C"
```

- Lists all print services that can be used
- Lists all print attributes that can be specified

- Steps
  - Use a ‘Printer’ object
  - Specify printer to print the report on
    - `printer.setPrinterName(<printer_name>);`
  - Specify print attributes
    - Use methods of implemented interface ‘PrintRequestAttributeSet’
  - Pass printer object to ‘FourRpLayouter’ object
    - `FourRpLayouter report = new FourRpLayouter(designFile, printer);`

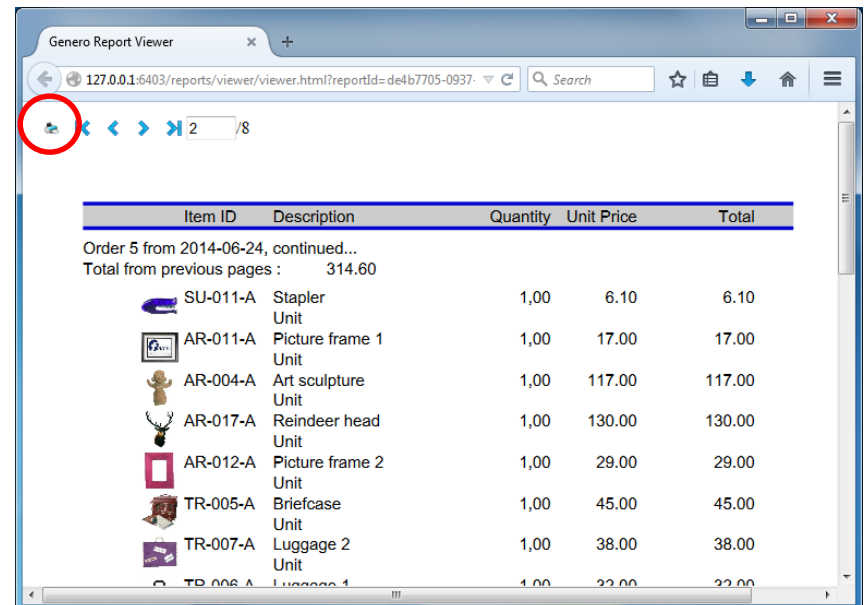
# Reminder: Processing pipe



- Example

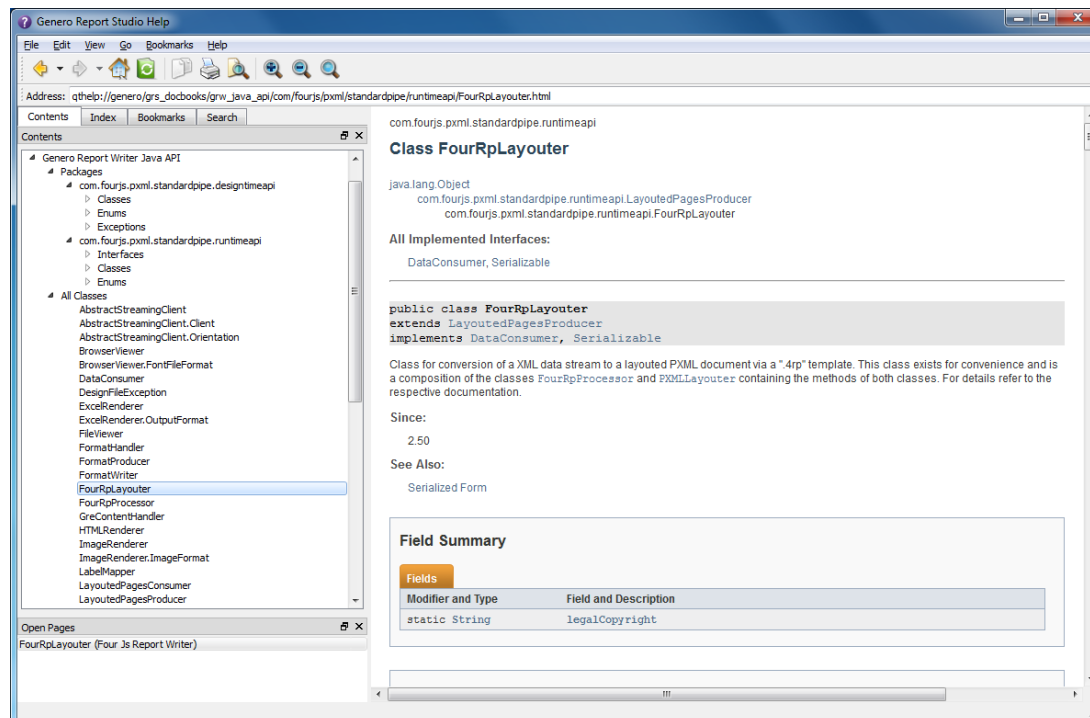
```
13 //For printing configuration:
14 import javax.print.attribute.standard.Sides;
15 import javax.print.attribute.standard.PageRanges;
16 import javax.print.attribute.standard.OrientationRequested;
17 import javax.print.attribute.standard.Copies;
18
28 String designFile;
29 if (args.length == 0) {
30     designFile = "OrderReport.4rp";
31 } else {
32     designFile = args[0];
33 }
34 Printer printer = new Printer();
35 printer.setPrinterName("balzac");
36
37 //Configuring the print output:
38 printer.add(Sides.TWO_SIDED_LONG_EDGE);
39 printer.add(new PageRanges(1,2));
40 printer.add(OrientationRequested.REVERSE_PORTRAIT);
41 printer.add(new Copies(2));
42
43 FourRpLayouter report = new FourRpLayouter(designFile, printer);
44
45 //Run report from XML file:
46 report.runFromXML("my_data_file.xml");
47
```

- Open report document in appropriate application
- Use print option of the application
- SVG Browser output
  - Specific print button to print the whole report





- Java classes used in GRE are defined in Java Archives located in GREDIR/lib/jars
- For details, refer to the documentation





- Open the **‘Sales’ project using the modified ‘Sales.java’ generating more data**
- Modify the ‘Sales.java’ source code to change the orientation of the list report (from ‘portrait’ to ‘landscape’)
- Run the list report in Excel output format
- Configure the Excel output format to only generate 1 sheet
- Modify the ‘Sales.java’ source code to generate a PDF output and display several report pages on a same physical page (multipage report)
- Modify the ‘Sales.java’ source to print the report on a server printer (if available)