



Genero®
report writer

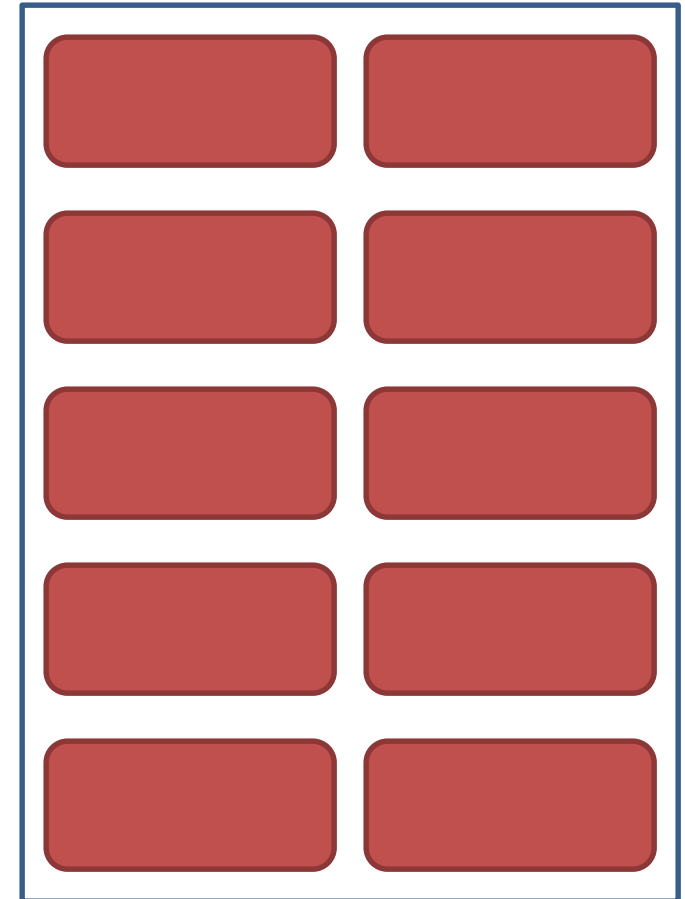
CREATING LABELS

Using Genero Report Writer
GRS 3.00

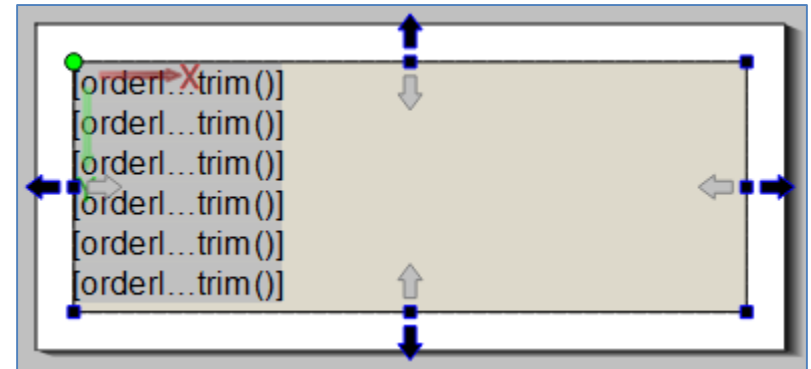
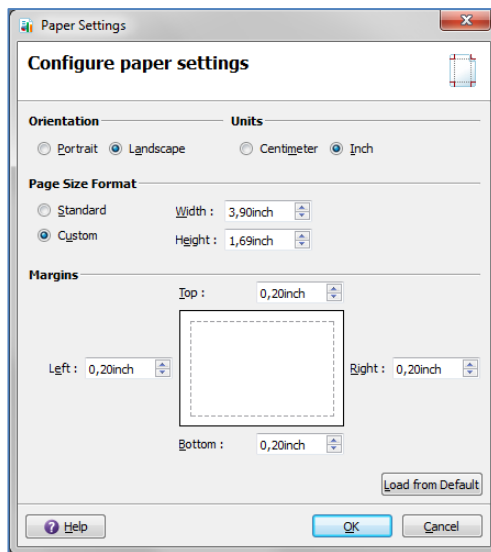
After this instruction, you will be able to:

- Create a report design document specifically for labels
- Add API calls for printing labels to your Java report application
- Use the visibility condition property to suppress blank report items

- Multiple **logical** pages printed on 1 **physical** page
- Typically used for name and address sticker labels
- A sheet of paper (physical page) will have many labels (logical pages) arranged in rows and columns
 - e.g. 2 columns, 5 rows

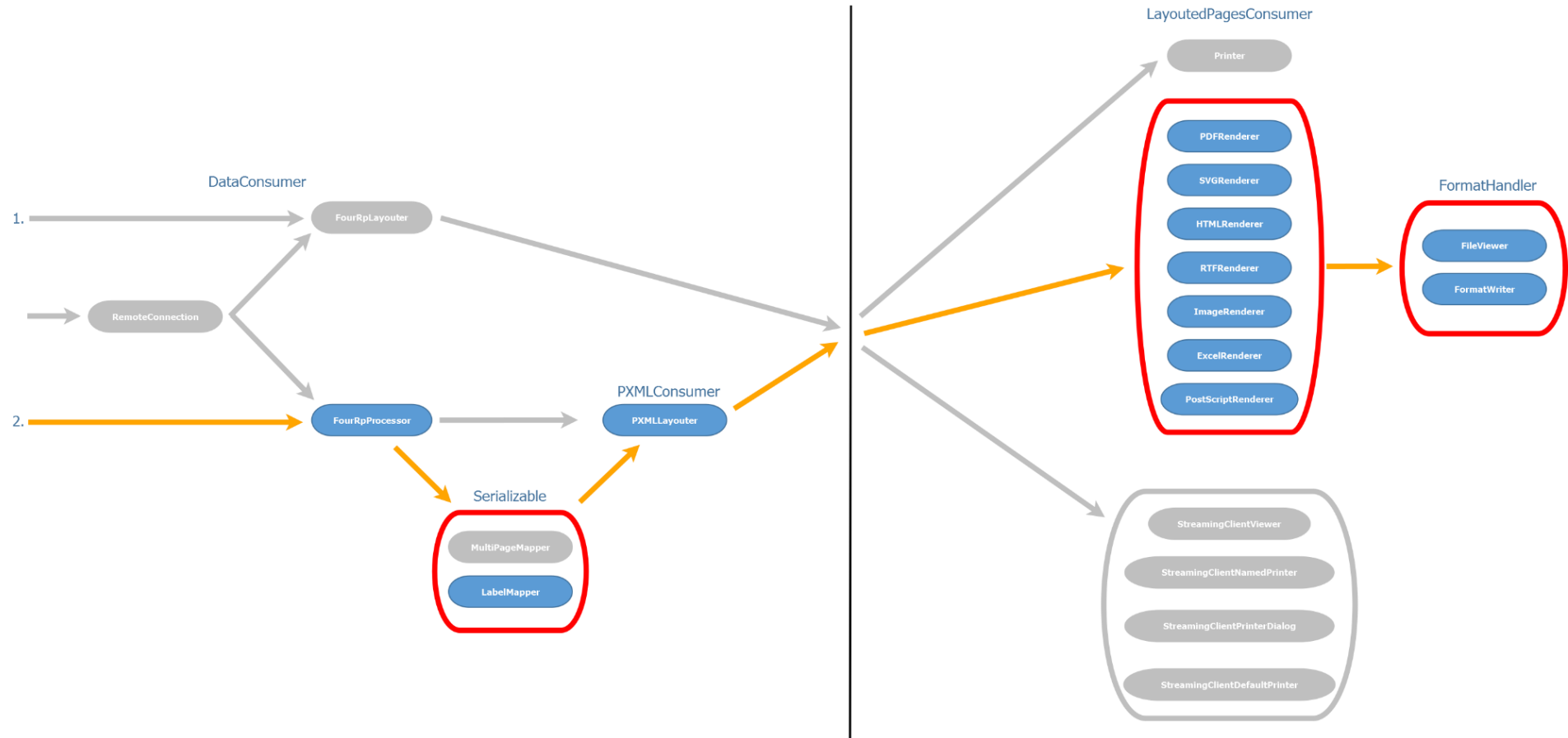


- The report design document represents a single label
 - Set page equal to size of actual label
 - File >> Report Properties >> Paper Settings...
 - Drag-and-drop report objects as usually



- Use specific 'LabelMapper' object
- Requires
 - FourRpProcessor object
 - conversion of a XML data stream to a PXML document
 - PXMLLayouter object
 - conversion of a PXML data stream to a layouted PXML document
- Call specific configuration APIs
 - Number of labels per row/column
 - Paper width/height
 - Paper margins

Processing pipe (labels)

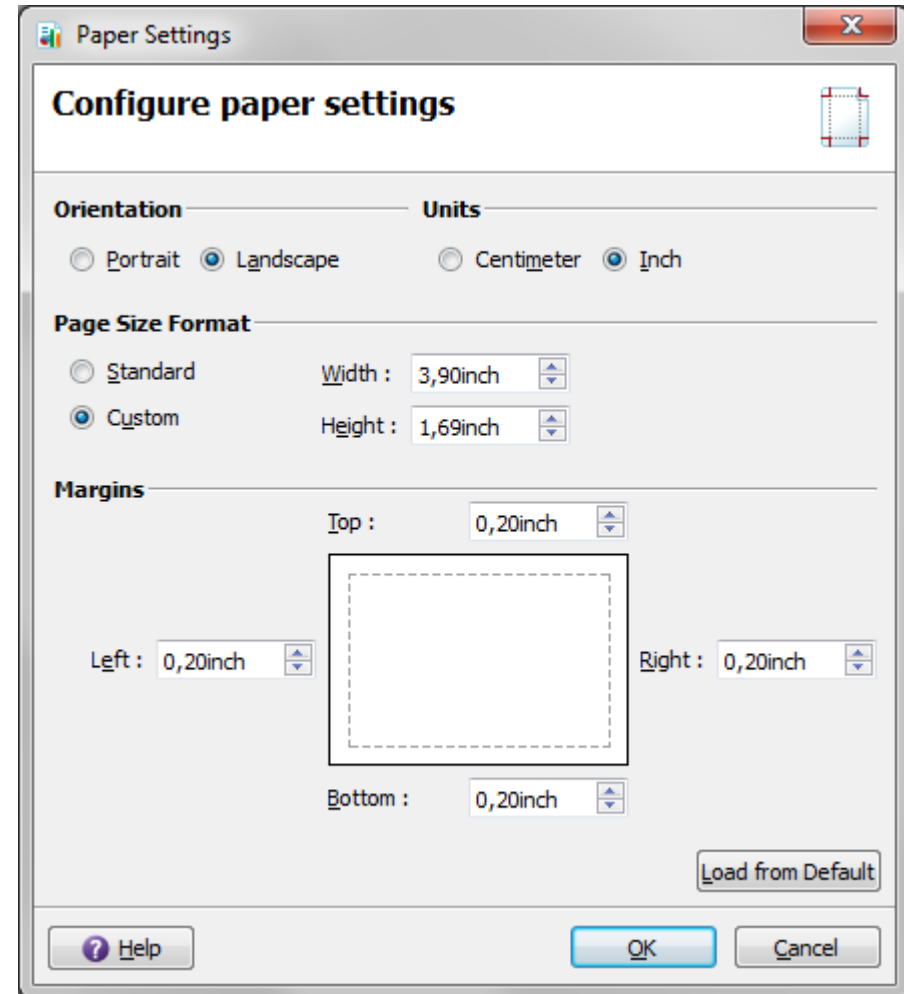


Java example (Label Report Application)

```
81
82 FormatHandler handler = new FormatWriter(outputFilename);
83 PDFRenderer renderer = new PDFRenderer(handler);
84
85 //Defining the label
86 PXMLConsumer labelLayout = new PXMLLayouter(renderer);
87 LabelMapper myLabelMapper = new LabelMapper(labelLayout);
88 myLabelMapper.setPaperTopMargin("15mm");
89 myLabelMapper.setPaperBottomMargin("15mm");
90 myLabelMapper.setPaperLeftMargin("5mm");
91 myLabelMapper.setPaperRightMargin("5mm");
92 myLabelMapper.setPaperWidth("a4width");
93 myLabelMapper.setPaperHeight("a4length");
94 myLabelMapper.setLabelsPerRow(2);
95 myLabelMapper.setLabelsPerColumn(3);
96 labelLayout = myLabelMapper;
97
98 FourRpProcessor report = new FourRpProcessor(designFile, labelLayout);
99
```

| | |
|---|---|
| <p>Columbus Arts 75038 2015-01-26T16:25:35.226+01:00</p> <p>Tablelamp Furniture 23.00 23.00</p> | <p>Columbus Arts 75038 2015-01-26T16:25:35.226+01:00</p> <p>Tablelamp Furniture 267.00 290.00</p> |
| <p>Columbus Arts 75038 2015-01-26T16:25:35.226+01:00</p> <p>Officechair Furniture 155.00 445.00</p> | <p>Columbus Arts 75038 2015-01-26T16:25:35.226+01:00</p> <p>Grandfather clock Furniture 329.00 774.00</p> |

- Product label / barcode printers typically have custom page sizes
 - Customize page size via [File >> Report Properties >> Paper Settings](#)
 - These printers may treat every label as a page so even though they may print in groups of 2's or 3's, you don't need to any special label code as for name and address labels



- Address label problems characterized by:
 - 3 to 5 lines of an address to be printed
 - No blank lines between lines of the address
 - At the end of the last non-blank line, print the customer's postcode

- Solution: 3 to 5 lines of address to be printed
 - Add a Vertical Box (Layout Node)
 - Add all the lines of the address as children of this node (dynamic layout)
- Solution: No blank lines
 - Set the **visibilityCondition** property of each address line so that it is not printed if the content is blank:
`header.addX.trim().length()>0`

- Solution: Print postcode at end of last non-blank line
 - Set the **value** property of each address line so that the postcode is appended if the lines after this one are all blank

Example for first address line:

```
header.add3.trim().length()+header.add2.trim().  
length()==0?header.add1.trim()+"  
"+header.postcode.trim():  
header.add1.trim()
```



- Exercise 1
 - Open the **‘Sales’ project**
 - Create a new report design for labels
 - Modify the ‘Sales.java’ source code to create labels and to use the new design (spread labels over 2 columns and 3 lines)
- Exercise 2
 - Open the **‘OrderReportJava’ demo project**
 - Check the ‘OrderLabels’ demo report design
 - Check the ‘OrderReportJava.java’ source code