

JTableReporter v1.0 Guide

Overview

JTableReporter is a Java library aimed to aid application developers make nice reports out of JTable instances.

In this guide, you will find example to learn how to use the most of the JTableReporter functionality and features.

It should be easy to integrate the most of the solutions in a Java/Swing program. This guide is far from complete, but it will be updated on a regular basis.

JTableReporter is a highly portable Java product.

Why JTableReporter ?

Whenever you use JTable, you will almost certainly like to have a printed report. The default print() method of your JTable instance will produce the report which is not what you want. JTable print() method uses the rendering suitable for the screen not for the paper layout, and you get the report with scrollbars, fonts you dislike etc.

That is why we have developed JTableReporter.

JTableReporter provides:

- Printing after column resizing
- Printing after row sorting
- Proportional column width
- Adjustable Date formatting
- User defined font properties
- Numeric field right alignment
- User defined decimal field format
- PORTRAIT / LANDSCAPE printing supported

Installation

After downloading **jtr-1_0-eval.zip**, or **jtr-1_0-full.zip** unzip it to the folder of your choice, and add **jtr-1_0.jar** to the CLASSPATH.

Running a demo

Demo program is within **jtr-1_0.jar**.

To activate it from shell type

```
java -jar jtr-1_0.jar
```

and a window with JTable will pop up.

Demo program shows a simple JTable with two String fields shown as randomized name and surname, and an integer, double and boolean field one of each. You can rearrange column order, or resize columns.

After clicking on the "Print" button the common print dialog will show up.

Confirm the printing and the report is here!

Demo program source version is included in jtr-1_0-eval.zip and jtr-1_0-full.zip.

Programmers HowTo

Once you have instanced JTable, it is very easy to generate a nice report. Just take a look at the following code snippet, and everything will become clear. To use JTableReporter you have to add to your Java source:

```
include com.logotet.tablereporter.JTableReporter;
```

The example assumes that you have a JButton instance named printButton, and JTable instance named table. All you have to do is put some of the code into addActionListener() method as in this example:

```
printButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        PriinterJob job = PrinterJob.getPrinterJob();
        PageFormat pageFormat = job.defaultPage();
        pageFormat.setOrientation(PageFormat.PORTRAIT);

        JTableReporter tp = new JTableReporter(table, pageFormat);
        //table is the assumed instance of your JTable
        tp.setDecimalDigits(2);
        tp.setColoredRows(true);
        tp.setDatePattern("MM-dd-yyyy");
        tp.setReportFont(new Font("Times New Roman", Font.PLAIN, 10));
        tp.setCheckMark(true);
        tp.setGrid(false);
        tp.setPageText("This is the page no.");
        Book book = tp.getPageable();
        job.setPageable(book);
        if (job.printDialog()) {
            // Print the job if the user didn't cancel printing
            try {
                job.print();
            } catch (Exception pe) {
                pe.printStackTrace();
            }
        }
    }
});
```

The code shown in bold an is optional, if you are not satisfied with JTavbleReporter default behaviour. Other parts of the code are common use of Java printing features, and will not be explained here.

To obtain proper information of your paper format you need to issue:

```
PrinterJob job = PrinterJob.getPrinterJob();
PageFormat pageFormat = job.defaultPage();
pageFormat.setOrientation(PageFormat.PORTRAIT);
// or PageFormat.LANDSCAPE
```

and then make the JTableReporter instance.

```
JTableReporter tp = new JTableReporter(table, pageFormat);
```

If you are happy with JTableReporter default options you can jump to :

```
Book book = tp.getPageable();
job.setPageable(book);
if (job.printDialog()) {
// Print the job if the user didn't cancel printing
    try {
        job.print();
    } catch (Exception pe) {
        pe.printStackTrace();
    }
}
```

and the report is ready.

If you want to do a little customization, you can try:

- Setting number of decimal digits

```
tp.setDecimalDigits(2);
```

Sets the number of digits behind the decimal point. All Double and Floats are printed aligned right, with decimal point and 2 decimal digits as default. You can change this if you like.

- Setting alternate colored rows

```
tp.setColoredRows(true);
```

By default, all rows are printed on white background. If you set this option to **true** rows will be printed on white and light gray background alternatively.

- Setting number of decimal digits

```
tp.setDatePattern("MM-dd-yyyy");
```

The default SimpleDateFormat is used on all your Date fields. If you prefer different format use SimpleDateFormat to specify it.

Note: NO SimpleDateFormat syntax check is performed!

- Setting report font

```
tp.setReportFont(new Font("Times New Roman", Font.PLAIN, 10));
```

Changes the Font used for your report. The font is used throughout the report. Header line is printed in bold, therefore the Font.PLAIN is recommended for the rest of the report.

- Setting check mark option

```
tp.setCheckMark(true);
```

Boolean fields can be printed as true/false text centered. If you setCheckMark to true, the check mark will be printed.

- Setting grid option

```
tp.setGrid(false);
```

The grid is not printed by default, but you can set the option to true if you like it.

- Setting the bottom page text

```
tp.setPageText("This is the page no.");
```

"Page" text is printed at the bottom of each page along with the page number. The text can be changed using this option.

- Setting the page orientation

Page orientation is set using the PageFormat instance

```
pageFormat.setOrientation(PageFormat.PORTRAIT);  
// or PageFormat.LANDSCAPE
```

If not set, your default printer setting is used.

How JTableReporter works

TableModel considerations

As you can have your own TableModel assigned to JTable instance with custom ColumnModel, different CellRenderers etc., it is important to know that JTableReporter will NOT use any of those. Only the TableModel is used to obtain data, and if your cell Object is other than Double, Float, Long, Integer, Boolean, Date and String, it's toString() method is used to print.

Column width and position

All columns from your JTable will be printed. Their width is proportionally adjusted to the report width. Report width depends on your default printer setting.

The JTable columns position retains while printing.

Future JTableReporter release will allow more features regarding column sizing and positioning.

System requirements

JavaTM Runtime Environment 1.4 or later.

The software is fully tested under Linux and WinXP.

Licence

The evaluation/demo version of JTableReporter cannot be deployed for commercial, non-commercial or any other purpose in any possible way. You may not build any software, applets or applications for distribution with the evaluation version of JTableReporter. It cannot be distributed in any form or method, electronic or otherwise. The evaluation version is to be solely used by you to evaluate the suitability of the JTableReporter for your needs.

Once purchased, you are granted the license to use JTableReporter on a single developer computer only. You may build applications from JTableReporter which can be distributed royalty free as per your needs, provided all the other conditions in this license agreement are adhered to.

JTableReporter evaluation is a fully functional, never-expiring, version of JTableReporter. It simply adds a watermark to generated reports in its unlicensed form. To eliminate the watermark, simply purchase and download a full version.

Disclaimer

Logotet disclaim all warranties and conditions, either expressed or implied, including, but not limited to, implied warranties of merchantability, fitness for a particular purpose, title, and non-infringement, with regard to JTableReporter, and the provision of or failure to provide support services.

Contact Us

Purchase

Be it an individual purchase inquiry, an institutional purchase, or any other purchase related communication, we would like to receive it on office@logotet.com with "JTableReporter" as Subject line.

Visit us at <http://www.logotet.com> to complete your purchase.

Support

We are committed to provide support. We provide free pre-purchase technical support to assist you to help in successful evaluation of JTableReporter.

You can contact us for any of your queries for example,

- If you are searching for a bean with a specific feature.
- If you want us to help you find if your specific need can be satisfied by our beans.
- Integrating our beans with your source code or other coding related queries.

If you want to learn more about our products, please send us an e-mail and we will respond to you.

Forward your support related mails to office@logotet.com with "JTableReporter" as Subject line.

Feedback

We would be pleased to get your feedback as well as suggestions about our products.

Forward your feedback or suggestions related mails to office@logotet.com with "JTableReporter" as Subject line.

Future releases

We plan further development, but it mainly depends on your satisfaction. We plan to add more features and to make JTableReporter a true Java reporting tool. Your suggestions are welcome!

Software consulting services

"Logotet"

Majora jagodica 8,
11040 Belgrade,
Serbia and Montenegro
office@logotet.com
www.logotet.com