



## Creating print layouts using KPrint

---

### Introduction

KPrint is a collection of Java classes that allows the user to generate print layouts within the SWT framework. The way a layout is created is comparable to the technique used to generate GUI layouts in SWT. You have some global container (Shell in SWT, PDocument in KPrint), on which you can place other elements (PBoxes).

### The KPrint layout concept

In SWT, you use Layouts to determine how the widgets are arranged on the window. KPrint has just one simple layout concept.

The elements you can place on the paper are text boxes, images, lines and whitespace. One element can either be on the right of the previous element or below the previous line of elements.

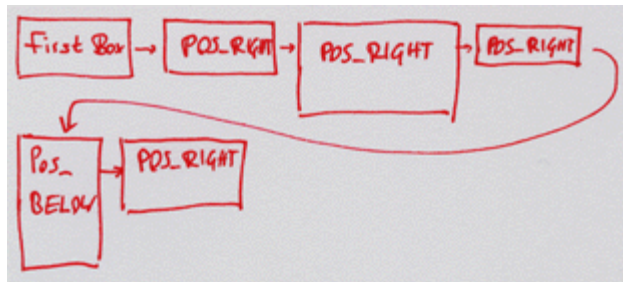


Figure 1: The position of an element is relative to the position of the previous element and determined by the style flag which can be PBox.POS\_RIGHT or PBox.POS\_BELOW.

This layout concept is both simple but powerful. There are some layouts that cannot be generated by this concept, but in most cases one can find a simpler solution that is possible to describe with the KPrint layout concept.

## This Text as a KPrint example

This text is created using the KPrint framework. It shows that KPrint can be used to layout long text passages, e.g. to print help system contents.

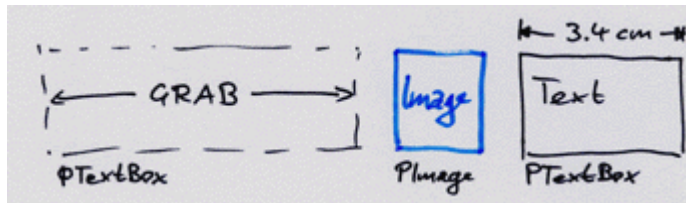


Figure 2: For right adjusted elements you need to put a grabbing PBox, e.g. a PHSpace or a PTextBox in front of them. For an example see the header of this document.

## Printing KTables

KPrint offers the PTable class that allows to print KTables. (for more information about KTable see [de.kupzog.ktable](http://de.kupzog.ktable), [www.kupzog.de/fkmk\\_uk](http://www.kupzog.de/fkmk_uk)) All you need to print a table is a KTableModel and a PTableBoxProvider. The KTableModel offers the data and the column size information. The box provider is comparable to a cell renderer. It creates a PBox for each table cell. You can use a default box provider which creates a PLittleTextBox or you can implement your own box provider with custom font, colors, borders or that provides PImageBoxes. See the PrintKTableExample class for an example how to print data from a KTableModel. You need the KTable.jar from [www.kupzog.de/fkmk\\_uk](http://www.kupzog.de/fkmk_uk) on your classpath to be able to compile this example.

Printing a Table works like that: you just add a PTable object to your document and set its table model and box provider. When the document is layouted, the layout function replaces the PTable object by PBoxes that are fetched from the box provider for each table cell.

## Printing SWT Tables

KPrint offers also the possibility to print the PTable SWT tables. It works pretty much like printing KTables, but you will use the SWTPTable class instead of PTable. Thanks to Onsel Armagan in Istanbul, Turkey for his feature.