Space Details

Key:	Reporting
Name:	Pentaho Reporting Documentation - Latest
Description:	
Creator (Creation Date):	mdamour (Nov 27, 2006)
Last Modifier (Mod. Date):	admin (Nov 28, 2006)

Available Pages

- shape-field
- string-field

- 08. Functions and Expressions
  - Advanced Calculations
    - CreateGroupAnchorsFunction
    - CreateHyperLinksFunction
    - ElementColorFunction
    - ElementVisibilitySwitchFunction
    - HideElementByNameFunction
    - HidePageBandForTableExport
    - PageFunction
    - PageOfPagesFunction
    - PageTotalFunction
    - property-ref
    - TextFormatExpression
  - Item Calculations
    - AverageExpression
    - BSHExpression
    - CapitalizeStringExpression
    - ColumnDifferenceExpression
    - ColumnDivisionExpression
    - ColumnMaximumExpression
    - ColumnMinimumExpression
    - ColumnMultiplyExpression
    - ColumnSumExpression
    - ConvertToDateExpression
    - ConvertToNumberExpression
    - DateExpression
    - DateSpanExpression
    - MapIndirectExpression
    - MapStringExpression
    - PercentageExpression
    - ResourceBundleLookupExpression
    - ResourceMessageExpression
    - SubStringExpression
    - TimeDiffAndFormatFunction
    - TokenizeStringExpression
    - ToLowerCaseStringExpression
    - ToUpperCaseStringExpression
    - VariableDateExpression
  - Pentaho Charting Extensions
    - CategorySetCollectorFunction
    - Chart Properties And Hierarchy
      - AbstractChartExpression
        - CategoricalChartExpression

Advanced Reporting Guide

This page last changed on Nov 29, 2006 by admin.

When looking at any production, managerial, financial or operational report, each displays detail data, organizes it by certain categories and specifies which pieces of data gets presented on the first page, last page or on every page. In addition, these report types can contain:

- Analytical computations that are simple like performing subtotaling or averaging by group or complex like computing each detail line time's percentage of their group's total or highlighting line items or groups for exception reporting,
- Objects like lines or bands to emphasize totaling or other metrics
- Images for corporate logos or watermarks

This guide will explain:

- The report xml definition for a Pentaho Reporting document based on JFreeReport 0.8.7.
- Report layout and object placement
- Formatting
- The use of functions
- Configuration and layout

This guide contains the following sections:

01. Report Layout

This page last changed on Nov 29, 2006 by admin.

Advanced Reporting Guide 02. Report Settings

The report xml definition for Pentaho Reports contains the following major sections; Report Header, Report Footer, Page Header, Page Footer, Groups including each group along with their corresponding header and footer, and Details/Items.

Report Sections

Elements	Purpose
include	Allows a report to reference another xml document for other report sections. For instance, multiple reports can reference a single report with only a page header defined for easier report maintenance.
parser-config	Sets variables with corresponding values for a report to use at runtime. Thus, common values can be set throughout a report like common text, background color or fonts. Note: The parser-config only defines replacement values for the parser itself. There is no parser-config anymore once the report has been parsed.
configuration	Allows parameters to be set for special output handling or debugging purposes.
watermark	Printed before any other object is printed on a new page. This band can consume the complete space of the page and will never trigger a pagebreak. The watermark-band is intended to fill the page background.
reportheader	Determines content that prints on the first page of a report.
reportfooter	Determines content that prints on the last page of a report.
pageheader	Determines content that prints at the top of each page of a report.
pagefooter	Determines content that prints at the bottom of each page of a report.
groups	Determines how the report is grouped and the content of each group's header and footer.
items	Determines the fields and content will be the line items or details.
functions	Computes results based on predefined or customized mathematical expressions or complex

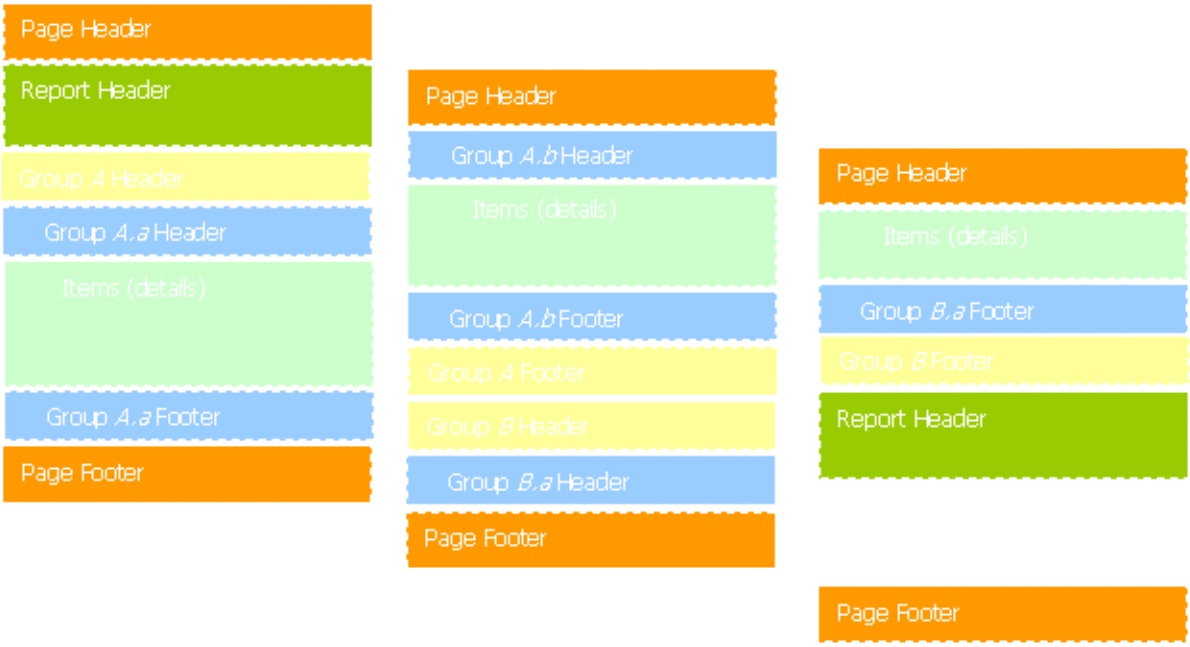
	scripts.

Base XML

Below is the base definition for a report.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<\!DOCTYPE report PUBLIC "-//JFreeReport//DTD report definition//EN//simple/version 0.8.5"
"http://jfreereport.sourceforge.net/report-085.dtd">
<report>
 <include>
 </include>

 <parser-config>
 </parser-config>

 <configuration>
 </configuration>

 <watermark>
 </watermark>

 <reportheader>
 </reportheader>

 <pageheader>
 </pageheader>

 <groups>
 </groups>

 <items>
 </items>

 <pagefooter>
 </pagefooter>

 <reportfooter>
 </reportfooter>

 <functions>
 </functions>

 <watermark>
 </watermark>
</report>
```

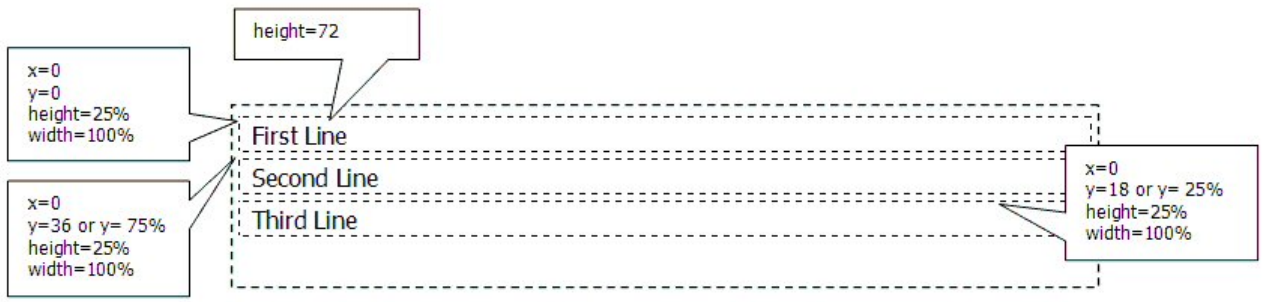
Example of Report Output



Position Objects

The position of an object in a section is declared by defining the x and y coordinates within a particular section. The top left starting point is x=0 and y=0 and is also the default position. When using an object, the mandatory attributes that need to be specified are the x (horizontal position), y (vertical position), height and width. These attributes are defined in 1/72 inch unit increments. These can be defined in units or percent. When using percent to define y and height, the container's height must be specified. The container's height is computed automatically. If a height is defined, it serves as a minimum-height. Therefore a band can increase its height beyond that size. Relative x and width values are resolved against the width of the parent (which corresponds to 100%). Relative y and height values are resolved against the used height of an container. For that computation, the reporting engine first computes the effective height of all absolutely positioned elements and uses that height as 100%. Then it computes the complete layout for all elements, with all relative sizes resolved against that height.

X & Y Coordinate Example



As with every element container, you may define default font settings for sub elements without their own font definition.

02. Report Settings

This page last changed on Nov 29, 2006 by admin.

01. Report Layout | Advanced Reporting Guide | 03. Report Header and Footer

In a report definition, report attributes set margins, page size and orientation.

Example Report Definition

```
<report rightmargin="72"
        topmargin="72"
        bottommargin="72"
        leftmargin="72"
        name="Quadrant For Region"
        orientation="portrait"
        pageformat="LETTER" >
```

<report>

Attributes	Description
pageFormats	Defines the page height and width based on predefined page types. Values: PAPER10X11 | PAPER10X13 | PAPER10X14 | PAPER12X11 | PAPER15X11 | PAPER7X9 | PAPER8X10 | PAPER9X11 | PAPER9X12 | A0 | A1 | A2 | A3 | A3_TRANSVERSE | A3_EXTRA | A3_EXTRATRANSVERSE | A3_ROTATED | A4 | A4_TRANSVERSE | A4_EXTRA | A4_PLUS | A4_ROTATED | A4_SMALL | A5 | A5_TRANSVERSE | A5_EXTRA | A5_ROTATED | A6 | A6_ROTATED | A7 | A8 | A9 | A10 | ANSIC | ANSID | ANSIE | ARCHA | ARCHB | ARCHC | ARCHD | ARCHE | B0 | B1 | B2 | B3 | B4 | B4_ROTATED | B5 | B5_TRANSVERSE | B5_ROTATED | B6 | B6_ROTATED | B7 | B8 | B9 | B10 | C4 | C5 | C6 | COMM10 | DL | DOUBLEPOSTCARD | DOUBLEPOSTCARD_ROTATED | ENV9 | ENV10 | ENV11 | ENV12 | ENV14 | ENVC0 | ENVC1 | ENVC2 | ENVC3 | ENVC4 | ENVC5 | ENVC6 | ENVC65 | ENVC7 | ENVCHOU3 | ENVCHOU3_ROTATED | ENVCHOU4 | ENVCHOU4_ROTATED | ENVDL | ENVINVITE | ENVISOB4 | ENVISOB5 | ENVISOB6 | ENVITALIAN | ENVKAKU2 | ENVKAKU2_ROTATED | ENVKAKU3 | ENVKAKU3_ROTATED | ENVMONARCH | ENVPERSONAL | ENVPRC1 | ENVPRC1_ROTATED | ENVPRC2 | ENVPRC2_ROTATED | ENVPRC3 | ENVPRC3_ROTATED | ENVPRC4 | ENVPRC4_ROTATED | ENVPRC5 |

	ENVPRC5_ROTATED | ENVPRC6 | ENVPRC6_ROTATED | ENVPRC7 | ENVPRC7_ROTATED | ENVPRC8 | ENVPRC8_ROTATED | ENVPRC9 | ENVPRC9_ROTATED | ENVPRC10 | ENVPRC10_ROTATED | ENVYOU4 | ENVYOU4_ROTATED | EXECUTIVE | FANFOLDUS | FANFOLDGERMAN | FANFOLDGERMANLEGAL | FOLIO | ISOB0 | ISOB1 | ISOB2 | ISOB3 | ISOB4 | ISOB5 | ISOB5_EXTRA | ISOB6 | ISOB7 | ISOB8 | ISOB9 | ISOB10 | LEDGER | LEGAL | LEGAL_EXTRA | LETTER | LETTER_TRANSVERSE | LETTER_EXTRA | LETTER_EXTRATRANSVERSE | LETTER_PLUS | LETTER_ROTATED | LETTER_SMALL | MONARCH | NOTE | POSTCARD | POSTCARD_ROTATED | PRC16K | PRC16K_ROTATED | PRC32K | PRC32K_ROTATED | PRC32K_BIG | PRC32K_BIGROTATED | QUARTO | STATEMENT | SUPERA | SUPERB | TABLOID | TABLOIDEXTRA
orientation	Determines the page orientation. Values: portrait | landscape
height	Determines the height of each page if pageFormat is not specified. Value is a whole number and each increment is equivalent to 1/72 of an inch.
width	Determines the width of each page if pageFormat is not specified. Value is a whole number and each increment is equivalent to 1/72 of an inch.
name	Report Name (optional)
leftmargin	Determines the left margin of each page. Value is a whole number and each increment is equivalent to 1/72 of an inch.
rightmargin	Determines the right margin of each page. Value is a whole number and each increment is equivalent to 1/72 of an inch.
topmargin	Determines the top margin of each page. Value is a whole number and each increment is equivalent to 1/72 of an inch.
bottommargin	Determines the bottom margin of each page. Value is a whole number and each increment is equivalent to 1/72 of an inch.
pagespan	Determines the number of pages the report width will extend.

03. Report Header and Footer

This page last changed on Feb 23, 2007 by uchile.

02. Report Settings | Advanced Reporting Guide | 04. Page Header and Footer

This section defines content that will print once at the beginning and end of the report. All objects contained within this section will inherit this attribute unless specified directly by the objects.

The report header can contain any band-element. The height of the report header is ignored, if the header and footer is printed on the same page.

Report Section	XML Syntax	Description
Report Header	<reportheader> </reportheader>	Prints on the first page.
Report Footer	<reportheader> </reportheader>	Prints on the last page.

Example

First Line
Second Line
Third Line

```
<reportheader height="72" alignment="left" font="Arial" fontsize="12">
  <label height="25%" width="100%" alignment="left" x="0" y="0">First Line</label>
  <label height="25%" width="100%" alignment="left" x="0" y="25%">Second Line</label>
  <label height="25%" width="100%" alignment="left" x=0 y="50%">Third Line</label>
</reportheader>
```

or

```
<reportheader alignment="left" font="Arial" fontsize="12">
  <label height="18" width="100%" alignment="left" x="0" y="0">First Line</label>
  <label height="18" width="100%" alignment="left" x="0" y="18">Second Line</label>
  <label height="18" width="100%" alignment="left" x=0 y="36">Third Line</label>
</reportheader>
```

Detailed Properties

- 01. Report Header Properties
- 02. Report Footer Properties

01. Report Header Properties

This page last changed on Nov 28, 2006 by admin.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fixedposition	Defines the absolute position on the page where the band should be printed. If the current cursor position is before that y-position, then empty space is added. If the current position is after that y-position, the band is printed on the next page (on that y-position).
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF export. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is used as a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false

fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number and each increment is equivalent to 1/72 of an inch. Objects don't inherit this setting. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
pagebreak-after-print	If set to true, the page breaks after the report header is printed. Type: Boolean Default value: false
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the report header will printed. Type: Boolean Default Value: true

02. Report Footer Properties

This page last changed on Nov 28, 2006 by admin.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fixedposition	Defines the absolute position on the page where the band should be printed. If the current cursor position is before that y-position, then empty space is added. If the current position is after that y-position, the band is printed on the next page (on that y-position). Type: Integer
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Value: false
font-encoding	Specifies the encoding to be used for the PDF export. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean

	Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number and each increment is equivalent to 1/72 of an inch. Objects don't inherit this setting. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
pagebreak-before-print	If set to true, the page breaks before the report footer is printed. Type: Boolean Default Value: false
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: True
vertical-alignment	Sets the vertical position of the text. Type: String Default Value: top Values: top | middle | bottom
visible	Sets whether the report header will printed. Type: Boolean Default Value: true

04. Page Header and Footer

This page last changed on Nov 29, 2006 by admin.

03. Report Header and Footer | Advanced Reporting Guide | 05. Groups

This section defines content that will print every page of the report. The page header can contain any band-element aka section. The page header/footer should not contain dynamic elements. The page header cannot create page breaks. If the combined content in the page header and page footer is larger than the space for a single page or no other content can fit on the page after the page header has printed and space for the page footer has been reserved, the report processing will fail.

As with every section, the font settings for this section are inherited to the sub elements without an own font definition.

Page

Report Section	XML Syntax	Description
Page Header	<pageheader> </pageheader>	Prints before any content is printed on the page.
Page Footer	<pagefooter> </pagefooter>	Prints after the last content for the page is printed. The page footer is always positioned at the bottom of a page, regardless how much space of the page is filled.

```
<pageheader height="72" alignment="left" font="Arial" fontsize="12">
  <label height="25%" width="100%" alignment="left" x="0" y="0">First Line</label>
  <label height="25%" width="100%" alignment="left" x="0" y="25%">Second Line</label>
  <label height="25%" width="100%" alignment="left" x=0 y="50%">Third Line</label>
<\pageheader>
```

or

```
<pagefooter alignment="left" font="Arial" fontsize="12">
  <label height="18" width="100%" alignment="left" x="0" y="0">First Line</label>
  <label height="18" width="100%" alignment="left" x="0" y="18">Second Line</label>
  <label height="18" width="100%" alignment="left" x=0 y="36">Third Line</label>
<\pagefooter>
```

Detailed Attributes

- 01. Page Header Attributes
- 02. Page Footer Attributes

01. Page Header Attributes

This page last changed on Nov 28, 2006 by admin.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false

height	Determines the minimum height. Value is a whole number and each increment is equivalent to 1/72 of an inch. Objects don't inherit this setting. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
onfirstpage	If set to false, content is not printed on first page. Type: Boolean Default Value: true
onlastpage	If set to false, content is not printed on last page. Type: Boolean Default Value: true
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the page header will printed. Type: Boolean Default Value: true

02. Page Footer Attributes

This page last changed on Nov 28, 2006 by admin.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole

	number and each increment is equivalent to 1/72 of an inch. Objects don't inherit this setting. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
onfirstpage	If set to false, content is not printed on first page. Type: Boolean Default Value: true
onlastpage	If set to false, content is not printed on last page. Default is true. Type: Boolean Default Value: true
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the report header will printed. Type: Boolean Default Value: true

05. Groups

This section defines content that will print for each group defined on the report. The tag encapsulates all groups. This tag helps to keep parsing simple. If no groups are defined, a default group is created and contains all elements of the report data row.

A fields list may only contain strings defining the names of the items which form a group. This is not limited to items from the data model, you may also enter expressions here.

If you define a subgroup, then you'll have to include all fields of the parent group and at least one new field. The field is not required to exist. If the field does not exist during the report processing, the value 'null' is used as replacement for the field's content.

The group ordering is defined by the contents of the <fields> tag of the groups. A less specific group is printed before any more specific group. A group is considered more specific than an other group, if it contains more fields in its field list.

Report Section	XML Syntax	Description
Groups	<groups> </groups>	There can be one or more groups definitions in the "groups" element. If no groups are defined, a default group is created to contain all data elements of the current report.
Group	<group></group>	Defines a group inside a groups element. (A group has a uniqe name and a list of fields.
Group Header	<groupheader> </groupheader>	Prints as the header for a group.
Group Footer	<groupfooter> </groupfooter>	Prints as the footer for a group

Example of Groups

The text below is sample data

```
1ST GROUP BY
BEGIN <first group by A>
    2ND GROUP BY
    BEGIN <second group by A.a>
      <details>
    2ND GROUP BY
    END <second group by A.a>
    2ND GROUP BY
```

```
    BEGIN <second group by A.b>
      <details>
    2ND GROUP BY
    END <second group by A.b>
1ST GROUP BY
END <first group by A>
1ST GROUP BY
BEGIN <first group by B>
    2ND GROUP BY
    BEGIN <second group by B.a>
      <details>
    2ND GROUP BY
    END <second group by B.a>
    2ND GROUP BY
    BEGIN <second group by B.b>
      <details>
    2ND GROUP BY
    END <second group by B.b>
1ST GROUP BY
END <first group by B>
```

```
<groups>
  <group name="FIRST GROUP ">
    <fields>
      <field>FIRST_GROUP_COLUMN_FIELD</field>
    </fields>
    <groupheader height="36" font="Arial" fontsize="10">
      <label height="18" alignment="left" width="100%" x="0" y="0">1ST GROUP BY</label>
      <message\-field height="18" alignment="left" width="100%" x="0" y="18">
        BEGIN $(FIRST_GROUP_COLUMN_FIELD)
      </message-field>
    </groupheader>
    <groupfooter height="36" font="Arial" fontsize="10">
      <label height="18" alignment="left" width="100%" x="0" y="0">1ST GROUP BY </label>
      <message-field height="18" alignment="left" width="100%" x="0" y="18">
        END $(FIRST_GROUP_COLUMN_FIELD)
      </message-field>
    </groupfooter>
  </group>
  <group name="SECOND GROUP ">
    <fields>
      <field>FIRST_GROUP_COLUMN_FIELD</field>
      <field>SECOND_GROUP_COLUMN_FIELD</field>
    </fields>
    <groupheader height="36" font="Arial" fontsize="10">
      <label height="18" alignment="left" width="90%" x="10%" y="0">2ND GROUP BY</label>
      <message-field height="18" alignment="left" width="90%" x="10%" y="18">
        BEGIN $(SECOND_GROUP_COLUMN_FIELD)
      </message-field>
    </groupheader>
    <groupfooter height="36" font="Arial" fontsize="10">
      <label height="18" alignment="left" width="90%" x="0" y="0">2ND GROUP BY</label>
      <message-field height="18" alignment="left" width="90%" x="10%" y="18">
        END $(SECOND_GROUP_COLUMN_FIELD)
      </message-field>
    </groupfooter>
  </group>
</groups>
```

Detailed Header/Footer Attribute Pages

- 01. Group Header Attributes
- 02. Group Footer Attributes

01. Group Header Attributes

This page last changed on Nov 28, 2006 by admin.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fixedposition	Defines the absolute postion on the page where the band should be printed. If the current cursor position is before that y-position, then empty space is added. If the current position is after that y-position, the band is printed on the next page (on that y-position).
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false

fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number and each increment is equivalent to 1/72 of an inch. Objects don't inherit this setting. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
pagebreak-before-print	If set to true, page break will occur before the group header is printed. Type: Boolean Default Value: false
pagebreak-after-print	If set to true, page break will occur after the group header is printed. Type: Boolean Default Value: false
repeat	Repeats the group header at the top of each new page if it spans over to another page. Repeated group headers are printed directly below the page header. Type: Boolean Default Value: false
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom

visible	Sets whether the report header will printed. Type: Boolean Default Value: true

02. Group Footer Attributes

This page last changed on Nov 28, 2006 by admin.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fixed-position	Defines the absolute postion on the page where the band should be printed. If the current cursor position is before that y-position, then empty space is added. If the current position is after that y-position, the band is printed on the next page (on that y-position).
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false

fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number and each increment is equivalent to 1/72 of an inch. Objects don't inherit this setting. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
pagebreak-after-print	If set to true, page break will occur before the group footer is printed. Type: Boolean Default Value: false
pagebreak-before-print	If set to true, page break will occur after the group footer is printed. Default is false. Type: Boolean Default Value: false
repeat	This will repeat the group footer if it spans over pages.
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the report header will printed. Type: Boolean Default Value: true

06. Items

This section defines content that will be printed for each data row.

Items Definition

Report Section	XML Syntax	Description
Details/Items	<items> </items>	Prints data rows. If no details/items band is defined, only printing is disabled. All calculations are performed regardless of the appearance of the items. Accepts the same attributes as all other bands. In the default case, the number of itembands printed corresponds to the number of rows in the tablemodel. If the tablemodel is empty, no itembands will be generated (but all other header and footer will be printed).

Items Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care. Type: Boolean

	Default Value: false
fixed-position	Defines the absolute postion on the page where the band should be printed. If the current cursor position is before that y-position, then empty space is added. If the current position is after that y-position, the band is printed on the next page (on that y-position).
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to bold. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the

	lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
pagebreak-after-print	Page break after printing details section. Type: Boolean Default Value: false
pagebreak-before-print	Page break before printing details section. Type: Boolean Default Value: false
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true

07. Report Objects

This page last changed on Nov 30, 2006 by mbatchelor.

06. Items | Advanced Reporting Guide | 08. Functions and Expressions

Each of the preceding report sections make references to report objects. This document defines each of the report objects, and then provides links to detailed attributes available for each report object.

Report Objects List

- anchor-field — Marks a specific point in an document as either the source or destination of a hypertext link.
- band — Defines a sub-band within an other band i.e. group header, report footer, etc.
- date-field — Returns the date from a specified field based on the current record.
- drawable-field — Returns a drawable object from a specified field on the current record.
- drawable-url-field — Returns a drawable object's URL from a specified field on the current record.
- drawableref — Returns a drawable object's loaded from the specified URL.
- image-field — Returns a image object from a specified field on the current record.
- imageref — Returns a image reference from a specified field on the current record.
- imageurl-field — Returns an image loaded from an URL found in the specified field on the current record.
- label — Writes static, immutable text.
- line — Draws a line.
- message-field — Writes multiple data types (text, string field, date field, and numeric field) into one object.
- number-field — Returns the numeric value from a specified field on the current record.
- rectangle — Draw a rectangle.
- resource-field — Used for localizing reports.
- resource-label — Used for localizing reports.
- shape-field — Returns a shape value from a specified field on the current record.
- string-field — Returns a string value from a specified field on the current record.

To Do

Tasks: Report Objects To Be Documented	

- Document resource-message (del)

Add Task:

anchor-field

This page last changed on Nov 30, 2006 by mbatchelor.

Marks a specific point in an document as either the source or destination of a hypertext link.
This allows you to create links from one hypertext document to another, as well as to different sections within the same document.

Attributes	Description
fieldname	Specify fieldname. The field must either contain URLs or Strings.
name	Used to reference the element later. Functions pick up elements by their name. Type: String
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

band

This page last changed on Nov 30, 2006 by mbatchelor.

Defines a sub-band within an other band i.e. group header, report footer, etc. This can be used to group objects and to trigger the visibility or change the style of the whole group using a single operation. Bands can have their own layout manager attached, which is also the only way to stack dynamic elements below each other in a safe way.

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: false
font-embedded	Specifies the font to embed into the PDF file. Type: String
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. Type: String
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false

fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

date-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns the date from a specified field based on the current record.

Detailed Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-format	When outputting to Excel, use another date/number format. The format is excel-specific and defined in the Excel-Help. If it is invalid, the resulting excel file might be unreadable. . If no format is specified here, the Java format string is used (as defined by the 'format' attribute). Type: String
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain java.util.Date objects.
format	Defines the formatting of the date field based on the Date Formatting Table. mmm dd, yyyy = Dec. 25, 2005 mm/dd/yy = 12/25/05 mm/dd/yy h:m = 12/25/05 8:00 am
font-embedded	Specifies the font to embed into the PDF file.

	Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is requred. If the height is smaller than the font size, no content is printed.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
nullstring	If the value read from the given field is null or is no instance of 'java.util.Date', replace with a predefined value.
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."

trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. This attribute is required. If no width is given, nothing gets printed.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

Date Formatting Table

Symbol	Meaning	Type	Example
G	Era	Text	"GG" -> "AD"
Y	Year	Number	"yy" -> "03" "yyyy" -> "2003"
M	Month	Text or Number	"M" -> "7" "M" -> "12" "MM" -> "07" "MMM" -> "Jul" "MMMM" -> "December"
D	Day in month	Number	"d" -> "3" "dd" -> "03"
H	Hour (1-12, AM/PM)	Number	"h" -> "3" "hh" -> "03"
H	Hour (0-23)	Number	"H" -> "15" "HH" -> "15"
K	Hour (1-24)	Number	"k" -> "3" "kk" -> "03"
K	Hour (0-11 AM/PM)	Number	"K" -> "15" "KK" -> "15"
M	Minute	Number	"m" -> "7" "m" -> "15" "mm" -> "15"
S	Second	Number	"s" -> "15" "ss" -> "15"

S	Millisecond (0-999)	Number	"SSS" -> "007"
E	Day in week	Text	"EEE" -> "Tue" "EEEE" -> "Tuesday"
D	Day in year (1-365 or 1-364)	Number	"D" -> "65" "DDD" -> "065"
F	Day of week in month (1-5)	Number	"F" -> "1"
W	Week in year (1-53)	Number	"w" -> "7"
W	Week in month (1-5)	Number	"W" -> "3"
A	AM/PM	Text	"a" -> "AM" "aa" -> "AM"
Z	Time zone	Text	"z" -> "EST" "zzz" -> "EST" "zzzz" -> "Eastern Standard Time"
'	Escape for text	Delimiter	"'hour' h" -> "hour 9"
''	Single quote	Literal	"ss''SSS" -> "45'876"

Date Formatting Table

This page last changed on Nov 28, 2006 by admin.

Symbol	Meaning	Type	Example
G	Era	Text	"GG" -> "AD"
Y	Year	Number	"yy" -> "03" "yyyy" -> "2003"
M	Month	Text or Number	"M" -> "7" "M" -> "12" "MM" -> "07" "MMM" -> "Jul" "MMMM" -> "December"
D	Day in month	Number	"d" -> "3" "dd" -> "03"
H	Hour (1-12, AM/PM)	Number	"h" -> "3" "hh" -> "03"
H	Hour (0-23)	Number	"H" -> "15" "HH" -> "15"
K	Hour (1-24)	Number	"k" -> "3" "kk" -> "03"
K	Hour (0-11 AM/PM)	Number	"K" -> "15" "KK" -> "15"
M	Minute	Number	"m" -> "7" "m" -> "15" "mm" -> "15"
S	Second	Number	"s" -> "15" "ss" -> "15"
S	Millisecond (0-999)	Number	"SSS" -> "007"
E	Day in week	Text	"EEE" -> "Tue" "EEEE" -> "Tuesday"
D	Day in year (1-365 or 1-364)	Number	"D" -> "65" "DDD" -> "065"
F	Day of week in month (1-5)	Number	"F" -> "1"
W	Week in year (1-53)	Number	"w" -> "7"
W	Week in month (1-5)	Number	"W" -> "3"
A	AM/PM	Text	"a" -> "AM" "aa" -> "AM"
Z	Time zone	Text	"z" -> "EST" "zzz" -> "EST" "zzzz" -> "Eastern Standard Time"
'	Escape for text	Delimiter	"'hour' h" -> "hour 9"

''	Single quote	Literal	"ss''SSS" -> "45'876"

drawable-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a drawable object from a specified field on the current record.

Attributes

Attributes	Description
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain org.jfree.ui.Drawable objects.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
name	Used to reference the element later. Functions pick up elements by their name. Type: String
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

drawable-url-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a drawable object's URL from a specified field on the current record.

Attributes

Attributes	Description
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain URLs or Strings.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: The height is required.
href	Sets a hyperlink for the section. Type: String
name	Used to reference the element later. Functions pick up elements by their name. Type: String
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

drawableref

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a drawable object's loaded from the specified URL. If the URL is a relative path, the URL will be resolved using the report definition's location. The core library allows loading WMF (Windows Meta Format) files using this field. Support for SVG images requires JfreeReport-Ext.

Attributes

Attributes	Description
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain org.jfree.ui.Drawable objects.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: The height is required.
href	Sets a hyperlink for the section. Type: String
name	Used to reference the element later. Functions pick up elements by their name. Type: String
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

image-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a image object from a specified field on the current record.

Attributes

Attributes	Description
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain java.awt.Image objects or org.jfree.report.ImageContainer objects.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
keepAspectRatio	Defines, whether the image's original aspect ratio should be preserved during scaling. This has no effect if no scaling is used. Type: Boolean Default Value: false
name	Used to reference the element later. Functions pick up elements by their name. Type: String
scale	Shrinks or expands the image to completely fill the element's defined bounds. If scaling is disabled and the unscaled image does not fit into the element's boundaries, the image is clipped at the element bounds. Type: Boolean Default Value: false
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object

	defined by a value or percentage.
Y	Defines the vertical starting position of an object defined by a value or percentage.

image-field-properties

This page last changed on Nov 30, 2006 by mbatchelor.

Attributes

Attributes	Description
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain java.awt.Image objects or org.jfree.report.ImageContainer objects.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
keepAspectRatio	Defines, whether the image's original aspect ratio should be preserved during scaling. This has no effect if no scaling is used. Type: Boolean Default Value: false
name	Used to reference the element later. Functions pick up elements by their name. Type: String
scale	Shrinks or expands the image to completely fill the element's defined bounds. If scaling is disabled and the unscaled image does not fit into the element's boundaries, the image is clipped at the element bounds. Type: Boolean Default Value: false
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.

Y	Defines the vertical starting position of an object defined by a value or percentage.

imageref

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a image reference from a specified field on the current record.

Attributes

Attributes	Description
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain java.awt.Image objects or org.jfree.report.ImageContainer objects.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
keepAspectRatio	Defines, whether the image's original aspect ratio should be preserved during scaling. This has no effect if no scaling is used. Type: Boolean Default Value: false
name	Used to reference the element later. Functions pick up elements by their name. Type: String
scale	Shrinks or expands the image to completely fill the element's defined bounds. If scaling is disabled and the unscaled image does not fit into the element's boundaries, the image is clipped at the element bounds. Type: Boolean Default Value: false
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object

	defined by a value or percentage.
Y	Defines the vertical starting position of an object defined by a value or percentage.

imageurl-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns an image loaded from an URL found in the specified field on the current record.

Attributes

Attributes	Description
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain java.awt.Image objects or org.jfree.report.ImageContainer objects.
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
keepAspectRatio	Defines, whether the image's original aspect ratio should be preserved during scaling. This has no effect if no scaling is used. Type: Boolean Default Value: false
name	Used to reference the element later. Functions pick up elements by their name. Type: String
scale	Shrinks or expands the image to completely fill the element's defined bounds. If scaling is disabled and the unscaled image does not fit into the element's boundaries, the image is clipped at the element bounds. Type: Boolean Default Value: false
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object

	defined by a value or percentage.
Y	Defines the vertical starting position of an object defined by a value or percentage.

label

This page last changed on Nov 30, 2006 by mbatchelor.

Writes static, immutable text.

Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care. For labels, using the dynamic width should seldom be necessary and can almost always be avoided by using carefully selected height values.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics.

	Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. The width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

line

This page last changed on Nov 30, 2006 by mbatchelor.

The reporting engine will construct the bounds for the line shape using the x1, y1 coordinates and the given width or height. If no width or height is given, the missing values will be computed from the difference between x2 and x1 and y2 to y1.

The values x1, y1 and x2, y2 define the start and end-point of the line within the element. This is used to specify the direction of the line.

Line shapes will always be scaled, without preserving the aspect ratio.

Attributes

Attributes	Description
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: If no height is specified, the height will be derived from the difference between y1 and y2.
name	Used to reference the element later. Functions pick up elements by their name. Type: String
stroke-style	Defines how the lines of the shape get drawn. Type: String Default Value: solid Predefined Values: solid | dashed | dotted | dot-dash | dot-dot-dash
weight	The weight defines the width of the pen used to draw the line. Type: Integer Default: 1
width	Set the horizontal width of the object. Default: If no width is specified, the width will be derived from the difference between x1 and x2.
x1	Defines the horizontal starting position of the line

	defined by a value or percentage.
x2	Defines the horizontal ending position of the line defined by a value or percentage.
y1	Defines the vertical starting position of the line defined by a value or percentage.
y2	Defines the vertical ending position of the line defined by a value or percentage.

message-field

This page last changed on Nov 30, 2006 by mbatchelor.

Writes multiple data types (text, string field, date field, and numeric field) into one object.

Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false

fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
nullstring	If a null value, replace with a predefined value.
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. Width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

Number Formatting Table

Symbol	Location	Localized?	Meaning
0	Number	Yes	Digit
#	Number	Yes	Digit, zero shows as absent
.	Number	Yes	Decimal separator or monetary decimal separator
-	Number	Yes	Minus sign
,	Number	Yes	Grouping separator
E	Number	Yes	Separates mantissa and exponent in scientific notation. *Need not be quoted in prefix or suffix.*
;	Sub-pattern boundary	Yes	Separates positive and negative sub-patterns
%	Prefix or suffix	Yes	Multiply by 100 and show as percentage
‰ (\u2030)	Prefix or suffix	Yes	Multiply by 1000 and show as per mille
¤ (\u00A4)	Prefix or suffix	No	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
'	Prefix or suffix	No	Used to quote special characters in a prefix or suffix, for example, "'#'#" formats 123 to "#123". To create a single quote itself, use two in a row: "# o''clock".

Date Formatting Table

Symbol	Meaning	Type	Example
G	Era	Text	"GG" -> "AD"
Y	Year	Number	"yy" -> "03"

			"yyyy" -> "2003"
M	Month	Text or Number	"M" -> "7" "M" -> "12" "MM" -> "07" "MMM" -> "Jul" "MMMM" -> "December"
D	Day in month	Number	"d" -> "3" "dd" -> "03"
H	Hour (1-12, AM/PM)	Number	"h" -> "3" "hh" -> "03"
H	Hour (0-23)	Number	"H" -> "15" "HH" -> "15"
K	Hour (1-24)	Number	"k" -> "3" "kk" -> "03"
K	Hour (0-11 AM/PM)	Number	"K" -> "15" "KK" -> "15"
M	Minute	Number	"m" -> "7" "m" -> "15" "mm" -> "15"
S	Second	Number	"s" -> "15" "ss" -> "15"
S	Millisecond (0-999)	Number	"SSS" -> "007"
E	Day in week	Text	"EEE" -> "Tue" "EEEE" -> "Tuesday"
D	Day in year (1-365 or 1-364)	Number	"D" -> "65" "DDD" -> "065"
F	Day of week in month (1-5)	Number	"F" -> "1"
W	Week in year (1-53)	Number	"w" -> "7"
W	Week in month (1-5)	Number	"W" -> "3"
A	AM/PM	Text	"a" -> "AM" "aa" -> "AM"
Z	Time zone	Text	"z" -> "EST" "zzz" -> "EST" "zzzz" -> "Eastern Standard Time"
'	Escape for text	Delimiter	"'hour' h" -> "hour 9"
''	Single quote	Literal	"ss''SSS" -> "45'876"

Number Formatting Table

This page last changed on Nov 28, 2006 by admin.

Symbol	Location	Localized?	Meaning
0	Number	Yes	Digit
#	Number	Yes	Digit, zero shows as absent
.	Number	Yes	Decimal separator or monetary decimal separator
-	Number	Yes	Minus sign
,	Number	Yes	Grouping separator
E	Number	Yes	Separates mantissa and exponent in scientific notation. *Need not be quoted in prefix or suffix.*
;	Sub-pattern boundary	Yes	Separates positive and negative sub-patterns
%	Prefix or suffix	Yes	Multiply by 100 and show as percentage
‰ (\u2030)	Prefix or suffix	Yes	Multiply by 1000 and show as per mille
¤ (\u00A4)	Prefix or suffix	No	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.
'	Prefix or suffix	No	Used to quote special characters in a prefix or suffix, for example, "'#'#" formats 123 to "#123". To create a single quote itself, use two in a row: "# o''clock".

number-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns the numeric value from a specified field on the current record.

Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-format	When outputting to Excel, use another date/number format. The format is excel-specific and defined in the Excel-Help. If it is invalid, the resulting excel file might be unreadable.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain instances of java.lang.Number.
format	Defines the formatting of the date field based on the Date Formatting Table. 123456789.1234 => $ ###,###,###.00 = $ 123,456,789.12 123456789.1234 => ###,###,### = 123,456,789 123456 => 00###### = 00123456
font-embedded	Specifies the font to embed into the PDF file. Type: String

font-encoding	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer The height is required.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
nullstring	If a null value or a non-number value is encountered, replace with a predefined value.
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout.

	Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

Number Formatting Table

Symbol	Location	Localized?	Meaning
0	Number	Yes	Digit
#	Number	Yes	Digit, zero shows as absent
.	Number	Yes	Decimal separator or monetary decimal separator
-	Number	Yes	Minus sign
,	Number	Yes	Grouping separator
E	Number	Yes	Separates mantissa and exponent in scientific notation. *Need not be quoted in prefix or suffix.*
;	Sub-pattern boundary	Yes	Separates positive and negative sub-patterns
%	Prefix or suffix	Yes	Multiply by 100 and show as percentage
‰ (\u2030)	Prefix or suffix	Yes	Multiply by 1000 and show as per mille
¤ (\u00A4)	Prefix or suffix	No	Currency sign, replaced by currency symbol. If doubled, replaced by international currency symbol. If present in a pattern, the monetary decimal separator is used instead of the decimal separator.

'	Prefix or suffix	No	Used to quote special characters in a prefix or suffix, for example, "'#'#" formats 123 to "#123". To create a single quote itself, use two in a row: "# o''clock".

rectangle

This page last changed on Nov 30, 2006 by mbatchelor.

Draw a rectangle.

Attributes

Attributes	Description
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
draw	Sets the outline of the object to be visible. The outline is drawn using the color and stroke defined for the element. Type: Boolean Default: false
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care. Type: Boolean Default: false
fill	Fills the rectangle with the color defined by the color attribute. Type: Boolean Default: true
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
name	Used to reference the element later. Functions pick up elements by their name. Type: String
reserved-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."

stroke-style	Defines how the lines of the shape get drawn. Type: String Default Value: solid Predefined Values: solid | dashed | dotted | dot-dash | dot-dot-dash
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
weight	Defines the width of the pen used to draw the outline of the rectangle. Type: Boolean Default Value: true
width	Set the horizontal width of the object. Width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

resource-field

This page last changed on Nov 30, 2006 by mbatchelor.

Used for localizing reports.

Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field should contain a string. That string must be a recognized key value of the used ResourceBundle, or nothing will be printed.
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic.

	Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: Height is required.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
nullstring	If a null value is encountered or the value read from the field could not be resolved using the defined resource bundle, replace with a predefined value.
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean

	Default Value: true
width	Set the horizontal width of the object.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

resource-label

This page last changed on Nov 30, 2006 by mbatchelor.

Used for localizing reports.

Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic. Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false

fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: Height is required.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. Width is required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

shape-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a shape value from a specified field on the current record.

Attributes

Attributes	Description
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
draw	Defines, whether the outline of the shape found in the defined field should be drawn. Type: Boolean Default Value: false
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. The field must contain java.awt.Shape objects.
fill	Defines, whether the shape should be filled. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
name	Used to reference the element later. Functions pick up elements by their name. Type: String
stroke-style	Defines how the lines of the shape get drawn. Type: String Default Value: solid

	Predefined Values: solid | dashed | dotted | dot-dash | dot-dot-dash
visible	Sets whether the object will be printed. Default Value: top Values: top | middle | bottom
weight	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

string-field

This page last changed on Nov 30, 2006 by mbatchelor.

Returns a string value from a specified field on the current record.

Attributes

Attributes	Description
alignment	Sets the horizontal justification of the text based on a defined width. Type: String Default Value: left Values: left | center | right
color	Sets the font color by the using RGB-triplet (#ffffff for white; #ff0000 for red, etc) or from list of the predefined colors below. Type: String Default Value: black Predefined Values: black | blue| cyan | darkGray | gray | green | lightGray | magenta | orange | pink | red | white | yellow
dynamic	A dynamic element increases the height of the element to fit the content of that element. Computing that dynamic height is expensive, so use it with care.
excel-wrap-text	When outputting to Excel, sets the text to word wrap. Type: Boolean Default Value: true
fieldname	The fieldname defines the name of the column of the tablemodel or the name of the function/expression/report property from where to read the data. Any value found in the column will be converted into a String using String.valueOf (...).
font-embedded	Specifies the font to embed into the PDF file. Type: Boolean Default Value: false
font-encoding	Specifies the encoding to be used for the PDF. Type: String
fontname	Sets the font name. Type: String
fontsize	Sets the font size. Type: Integer
fontstyle	Sets font style. This is a shortcut for defining fsbold and fsitalic.

	Type: String Values: plain | bold | italic | bolditalic
fsbold	Sets font to italics. Type: Boolean Default Value: false
fsitalic	Sets font to italics. Type: Boolean Default Value: false
fsstrikethr	Sets font to strikethrough. Type: Boolean Default Value: false
fsunderline	Sets font to underline. Type: Boolean Default Value: false
height	Determines the minimum height. Value is a whole number where each increment is equivalent to 1/72 of an inch or percentage. Height is required. Type: Integer Default: If no height is specified, then height will be determine by all the objects contained.
href	Sets a hyperlink for the section. Type: String
line-height	The height of a single text line in text elements. If the line-height is set and is greater than the font size, extra padding will be added between the lines making text more readable. Type: Integer
name	Used to reference the element later. Functions pick up elements by their name. Type: String
nullstring	If a null value, replace with a predefined value.
reserve-literal	Determines the text printed when the text does not fit completely into an element. Type: String Default value: "..."
trim-text-content	Defines, whether leading and trailing whitespaces of the generated lines get removed. Important for a clean layout. Type: Boolean Default Value: true
vertical-alignment	Sets the vertical position of the text. Default Value: top Values: top | middle | bottom
visible	Sets whether the object will be printed. Type: Boolean Default Value: true
width	Set the horizontal width of the object. Width is

	required.
x	Defines the horizontal starting position an object defined by a value or percentage.
y	Defines the vertical starting position of an object defined by a value or percentage.

08. Functions and Expressions

This page last changed on Nov 29, 2006 by admin.

07. Report Objects | Advanced Reporting Guide | 09. Displaying the Chart

Functions have access to the data row and can access other functions or expressions or the data source. Functions are stateful meaning they maintain their state during the report generation. For stateless user-defined computations, consider using an expression instead of functions, as expression are cheaper to compute and maintain when using huge reports. Function parameters are given by property elements.

The deplevel attribute can be used to prioritize the functions. Functions with an higher dependency level are executed before any function with lower dependency levels. Dependency levels lower than 0 are not allowed.

An expression is a stateless user-defined function. It can access the data row and the report properties (property-ref) to perform its task. Using the data row, an expression has access to the data source and other functions and expressions. Expression parameters are given by property elements.

Function/Expression Type	Description
Summary Calculations	These calculate values for display in groups, headers and footers.
Item Calculations	These calculate values for display in item (detail) bands.
Advanced Calculations	These are misc. advanced calculations.
Pentaho Charting Extensions	These are functions and expressions for generating charts within reports.

Advanced Calculations

This page last changed on Dec 01, 2006 by mbatchelor.

Available Advanced Calculations

- CreateGroupAnchorsFunction — Adds an anchor to a group to all elements with the name specified in *element*.
- CreateHyperLinksFunction — Adds hyperlinks to all elements (objects) with the name specified in *element*.
- ElementColorFunction — A function that alternates between true and false for each item within a group. The function value determines when to use an object's color.
- ElementVisibilitySwitchFunction — A function that alternates between true and false for each item within a group. The functions value affects a defined objects visibility.
- HideElementByNameFunction — This function hides all objects with a given name, as long as the defined field does not contain the element name.
- HidePageBandForTableExport — Hide the page header and footer if the current export type is not a pageable export (like printing, PDF or plain text).
- PageFunction — A report function that counts pages.
- PageOfPagesFunction — A report function that combines the PageFunction and PageTotalFunction.
- PageTotalFunction — Prints the total number of pages of an report.
- property-ref — Exposes values and properties that are available in the reporting object.
- TextFormatExpression — Uses a java.text.MessageFormat to format a string message.

CreateGroupAnchorsFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Adds an anchor to a group to all elements with the name specified in *element*. The link target is read from a specified field. The column referenced by this field should contain URLs or Strings.

Example

```
<function class="org.jfree.report.function.CreateGroupAnchorsFunction" name="Grp_Anchr">
  <properties>
    <property name="field">Country</property>
    <property name="group">Product</property>
  </properties>
<function>
```

CreateHyperLinksFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Adds hyperlinks to all elements (objects) with the name specified in *element*. The link target is read from a specified field. The column referenced by this field should contain URLs or Strings.

Example

```
<expression class="org.jfree.report.function.TextFormatExpression" name="URLCreateExpression">
  <properties>
     <property
name="pattern">http://localhost:8080/pentaho/ViewAction?&amp;solution=test&amp;path=reporting&amp;action=ope
     <property name="field[0]">CUSTOMERNAME</property>
  </properties>
</expression>

<function class="org.jfree.report.function.CreateHyperLinksFunction" name="AddHRefs">
  <properties>
     <property name="field">URLCreateExpression</property>
     <property name="element">URL-Field</property>
  </properties>
</function>

<items fontname="Arial" fontsize="9" fontstyle="plain" height="12" visible="true"
vertical-alignment="middle">
  <rectangle name="background" x="15%" y="0" width="70%" height="100%" color="#DFDFDF"
weight="0"/>
  <string-field height="12" width="55%" fieldname="CUSTOMERNAME" x="15%" alignment="left"
name="URL-Field" />
  <number-field height="12" width="15%" fieldname="QUANTITYORDERED" x="70%" alignment="right"
format="###,###.00" />
</items>
```

ElementColorFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A function that alternates between true and false for each item within a group. The function value determines when to use an object's color. If the function evaluates to true, the named object is painted with the color=True, else the object is painted with color=False.

Use a property tag to name an object contained in the Item section whose color should be affected by this function.

Example

```
<expression class="org.jfree.report.modules.misc.beanshell.BSHExpression" name="isVarNeg">
<properties>
   <property name="expression">
    Object getValue() {
      Object value = dataRow.get("VARIANCE");
      if (!(value instanceof Number)) {
        return Boolean.FALSE;
      }
      Number number = (Number) value;
      if (number.doubleValue() < 0) {
        return Boolean.TRUE;
      }
      return Boolean.FALSE;
    }
    </property>
 </properties>
</expression>

<function class="org.jfree.report.function.ElementColorFunction" name="changeColor">
  <properties>
    <property name="element">Variance Field</property>
    <property name="field">isVarNeg</property>
    <property name="colorTrue">red</property>
    <property name="colorFalse">black</property>
  </properties>
</function>

<items fontname="Arial" fontsize="10" vertical-alignment="middle">
  <string-field fieldname="POSITIONTITLE" height="15" vertical-alignment="middle" width="40%"
x="15%" y="0"/>
  <number-field alignment="right" fieldname="ACTUAL" format="$#,##0" height="15" width="15%"
x="55%" y="0"/>
  <number-field alignment="right" fieldname="BUDGET" format="$#,##0" height="15" width="15%"
x="70%" y="0"/>
  <number-field alignment="right" fieldname="VARIANCE" format="$#,##0" height="15"
name="Variance Field" width="15%" x="85%" y="0"/>
  <line color="gray" height="0" weight=".25" width="85%" x1="15%" x2="100%" y1="16" y2="16"/>
</items>
```

ElementVisibilitySwitchFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A function that alternates between true and false for each item within a group. The functions value affects a defined objects visibility. If the function evaluates to true, the named object is visible, else the element is invisible.

Objects do not define their own background color attribute. To create a background, a rectangle shape element must be created behind the element.

The ElementVisibilitySwitchFunction is used to trigger the visibility of an named object. If the object is in the background, it will alternate between an object being visible and not being visible.

The ElementVisibilitySwitchFunction defines two parameters:

- *element* defines the name of the element aka object in the item band that should be modified. The element must be named using the "name" attribute, only the first occurrence of that named element will be modified.

- *initial-state* defines the initial state of the function. (true or false) defaults to false. This is the reverts of the object's visibility (set to false to start with an visible object, set to true to hide the object in the first item row).

```
<function name="bkgrndTrggr" class="org.jfree.report.function.ElementVisibilitySwitchFunction">
  <properties>
     <property name="element">background</property>
     <property name="numberOfElements">1</property>
  </properties>

<items fontname="Arial" fontsize="9" fontstyle="plain" height="12" layout-cachable="false"
visible="true" vertical-alignment="middle">
  <rectangle name="background" x="15%" y="0" width="70%" height="100%" color="#DFDFDF"
weight="0"/>
  <string-field height="12" width="55%" fieldname="CUSTOMERNAME" x="15%" alignment="left"
name="URL-Field" />
  <number-field height="12" width="15%" fieldname="QUANTITYORDERED" x="70%" alignment="right"
format="###,###.00" />
</items>
```

HideElementByNameFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

This function hides all objects with a given name, as long as the defined field does not contain the element name.

Example

```
<function class="org.jfree.report.function.HideElementByNameFunction" name="LeaderVisibility">
  <properties>
    <property name="element">leader</property>
    <property name="field">recordID</property>
  </properties>
</function>

<band name="leader"  x="0" y="0">
  ...
</band>
```

HidePageBandForTableExport

This page last changed on Feb 23, 2007 by uchile.

Description

Hide the page header and footer if the current export type is not a pageable export (like printing, PDF or plain text).
This function accepts two parameters:

- *hidePageBands* defaults to true; hides the page header and footer for non-pageable exports.
- *disableRepeatingHeader* defaults to false; all group headers, which have the repeat flag set to true get that flag reset to false. On a page break, no repeated header will be printed. As this may be confusing to the user and not always appropriate, this property defaults to false and must be enabled explicitly.

Example

```
<function class="org.jfree.report.function.HidePageBandForTableExportFunction"
name="hidePageHeaderandPageFooter">
  <properties>
    <property name="hidePageBands">true</property>
    <property name="disableReportingHeader">true</property>
  </properties>
</function>
```

PageFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that counts pages. This method is only useable when a report processor is used, which generated PageEvents. The PageableReportProcessor is one of them.

As with all page dependent functions: The function will only be active, when the page events get fired, this usually only happens during the last pagination run and the printing. The function level will be negative when this happens.

Example

```
<function class="org.jfree.report.function.PagesFunction" name="Page"/>
```

PageOfPagesFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Definition

A report function that combines the PageFunction and PageTotalFunction. For example, it outputs the page as 3/14 which is page 3 of 14 total pages.

Restrictions for both classes apply to this one also.

Example

```
<function class="org.jfree.report.function.PageOfPagesFunction" name="PageofPages"/>
```

PageTotalFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Definition

Prints the total number of pages of an report. If a group is specified, this function expects the group to have the manual page break enabled.

This function will only work as expected in group mode if the named group has page break set to true.

Example

```
<function class="org.jfree.report.function.PagesTotalFunction" name="PageTotal"/>
```

property-ref

This page last changed on Dec 01, 2006 by mbatchelor.

Definition

Exposes values and properties that are available in the reporting object. These could have been created by the reporting engine (for example, report.name or report.date), or are placed in the reporting environment by a program or the Pentaho BI platform.

Accepts either "text" or "serialized-base64" encoding.

For example, these are variables that could be passed from a Pentaho action.

These property references need to be defined before it can be used in a report.

Examples

```
<property-ref name="report.date" />
```

```
<property-ref name="productname"/> <!--variable passed from Pentaho *.xaction-->
<property-ref name="username"/>    <!--variable passed from Pentaho *.xaction-->
```

TextFormatExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Uses a java.text.MessageFormat to format a string message. It will concatenate and format one or more values evaluated from an expression, function or report data source.

The TextFormatExpression uses the pattern property to define the global format-pattern used when evaluating the expression. The data row fields used to fill the expressions placeholders are defined in a list of properties where the property-names are numbers.

The property counting starts at "0".

The Syntax of the *pattern* property is explained in java.text.MessageFormat.

Example

```
<expression name="expr" class="org.jfree.report.function.TextFormatExpression">
 <properties>
   <property name="pattern">Invoice for your order from \{0, date, EEE, MMM d,
yyyy\}</property>
   <property name="fields\[0\]">printdate</property>
 </properties>
</expression>

<expression class="org.jfree.report.function.TextFormatExpression" name="leader.name">
  <properties>
    <property name="pattern">{0} {1}</property>
    <property name="field[0]">leader.firstName</property>
    <property name="field[1]">leader.lastName</property>
  </properties>
</expression>
```

Item Calculations

This page last changed on Dec 01, 2006 by mbatchelor.

Available Item Calculations

- AverageExpression — An expression that takes values from one or more fields and returns the average of them.
- BSHExpression — Uses script code to calculate values within a single row of a report.
- CapitalizeStringExpression — Capitalizes the first letter of all words found in the string.
- ColumnDifferenceExpression — An expression that takes values from two or more fields and returns the difference of them.
- ColumnDivisionExpression — An expression that takes values from two or more fields and divides all values.
- ColumnMaximumExpression — An expression that takes values from two or more fields and returns the the greatest value of them
- ColumnMinimumExpression — An expression that takes values from two or more fields and returns the the smallest value of them
- ColumnMultiplyExpression — An expression that takes values from two or more fields and and multiplies all values
- ColumnSumExpression — An expression that takes values from two or more fields and returns the the sum of all values
- ConvertToDateExpression — Converts string to a date type
- ConvertToNumberExpression — Converts string to a numeric type.
- DateExpression — Constructs a date using the specified parameters.
- DateSpanExpression — Computes the difference between two dates.
- MapIndirectExpression — Replaces various predefined values with values read from other fields.
- MapStringExpression — Replaces various predefined values with other constant values.
- PercentageExpression — Computes the percentage for a column in relation to a base column.
- ResourceBundleLookupExpression — Returns a translated value from a resource bundle.
- ResourceMessageExpression — Formats a string using a MessageFormat string read from an ResourceBundle. This is the expression-counterpart of the resource-message field.
- SubStringExpression — Returns a partial string.
- TimeDiffAndFormatFunction — Computes the Time between two date fields
- TokenizeStringExpression — Replaces all occurrences of a certain substring with an other string.
- ToLowerCaseStringExpression — Converts the string to all lower case letters
- ToUpperCaseStringExpression — Converts the string to all upper case letters
- VariableDateExpression — Constructs a date using the specified parameters and using values read from various fields.

To Do List

Tasks: Item Calculations	

- () Doc TimeDiffAndFormatFunction (del)
- () Doc ColumnDifferenceExpression (del)
- () Doc ColumnDivisionExpression (del)
- () Doc ColumnMaximumExpression (del)
- () Doc ColumnMinimumExpression (del)
- () Doc ColumnMultiplyExpression (del)
- () Doc ColumnSumExpression (del)
- () Doc ConvertToDateExpression (del)
- () Doc DateExpression (del)
- () Doc VariableDateExpression (del)
- () Doc DateSpanExpression (del)
- () Doc CapitalizeStringExpression (del)
- () Doc ToLowerCaseStringExpression (del)
- () Doc ToUppercaseStringExpression (del)
- () Doc TokenizeStringExpression (del)
- () Doc SubStringExpression (del)
- () Doc MapStringExpression (del)
- () Doc MapIndirectExpression (del)
- () Doc ResourceBundleLookupExpression (del)
- () ResourceMessageExpression (del)

Add Task:

AverageExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

An expression that takes values from one or more fields and returns the average of them.

Example

```
<expression name="Average\_Column" class="org.jfree.report.function.AverageExpression">
  <properties>
    <property name="field\[0\]">Jan\_Column</property>
    <property name="field\[1\]">Feb\_Column</property>
    <property name="field\[2\]">Mar\_Column</property>
  </properties>
</expression>
```

BSHExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Definition

Uses script code to calculate values within a single row of a report.

Examples

```
<expression class="org.jfree.report.modules.misc.beanshell.BSHExpression" name="totalPrice">
  <properties>
    <property name="expression">
     Object getValue() {
       if (dataRow == null) {
         return null;
       }
       Number OrderCount = (Number) dataRow.get(&quot;QUANTITYORDERED&quot;);
       Number SalePrice = (Number) dataRow.get (&quot;PRICEEACH&quot;);
       if (OrderCount == null || SalePrice == null) {
         return null;
       }
       float totalPrice = OrderCount.floatValue() * SalePrice.floatValue();
       return new Float (totalPrice);
     }
    </property>
   </properties>
</expression>

<expression class="org.jfree.report.modules.misc.beanshell.BSHExpression"
name="isVarianceNegative">
<properties>
   <property name="expression">
    Object getValue() {
      Object value = dataRow.get(&quot;VARIANCE&quot;);
      if (value instanceof Number == false) {
        return Boolean.FALSE;
      }
      Number number = (Number) value;
      if (number.doubleValue() < 0) {
        return Boolean.TRUE;
      }
      return Boolean.FALSE;
    }
    </property>
 </properties>
</expression>
```

CapitalizeStringExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Capitalizes the first letter of all words found in the string.

TODO
Document function/expression

ColumnDifferenceExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

An expression that takes values from two or more fields and returns the difference of them.

Details

TODO
Document Function/Expression

ColumnDivisionExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

An expression that takes values from two or more fields and divides all values.

TODO
Document Function/Expression

ColumnMaximumExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

An expression that takes values from two or more fields and returns the the greatest value of them

TODO
Document Function/Expression

ColumnMinimumExpression

This page last changed on Dec 01, 2006 by mbatchelor.

An expression that takes values from two or more fields and returns the the smallest value of them

TODO
Document Function/Expression

ColumnMultiplyExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

An expression that takes values from two or more fields and and multiplies all values

TODO
Document Function/Expression

ColumnSumExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

An expression that takes values from two or more fields and returns the the sum of all values

Details

TODO
Document Function/Expression

ConvertToDateExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Converts string to a date type

Details

TODO
Document Function/Expression

ConvertToNumberExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Converts string to a numeric type.

Example

```
<expression name="C_Txt_Nmbr" class="org.jfree.report.function.ConvertToNumberExpression">
  <properties>
    <property name="field[0]">Invoice</property>
  </properties>
</expression>
```

DateExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Constructs a date using the specified parameters.

Details

TODO
Document Function/Expression

DateSpanExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Computes the difference between two dates.

TODO
Document function/expression

MapIndirectExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Replaces various predefined values with values read from other fields.

TODO
Document function/expression

MapStringExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Replaces various predefined values with other constant values.

TODO
Document function/expression

PercentageExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Definition

Computes the percentage for a column in relation to a base column.

The function understands two parameters.

- *dividend* parameter is required and denotes the name of an Item Section field which is used as dividend.
- *divisor* parameter is required and denotes the name of an Item Section field which is used as divisor.

If either the dividend or the divisor are not numeric, the expression will return null.

The formula used is as follows, Percent := divident / divisor.
If the flag *useDifference* is set, the difference between base and subject is used instead. Percent := (divisor - divident) / divisor

Example

```
<expression name="saved" class="org.jfree.report.function.PercentageExpression">
  <properties>
    <property name="dividend">article.price</property>
    <property name="divisor">article.reducedPrice</property>
    <property name="useDifference">true</property>
  </properties>
</expression>
```

ResourceBundleLookupExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Returns a translated value from a resource bundle.

TODO
Document function/expression

ResourceMessageExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Formats a string using a MessageFormat string read from an ResourceBundle. This is the expression-counterpart of the resource-message field.

TODO
Document function/expression

SubStringExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Returns a partial string.

TODO
Document function/expression

TimeDiffAndFormatFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Computes the Time between two date fields

Details

TODO
Document Function/Expression

TokenizeStringExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Replaces all occurrences of a certain substring with an other string.

TODO
Document function/expression

ToLowerCaseStringExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Converts the string to all lower case letters

TODO
Document function/expression

ToUpperCaseStringExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Converts the string to all upper case letters

TODO
Document function/expression

VariableDateExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Constructs a date using the specified parameters and using values read from various fields.

Details

TODO
Document Function/Expression

Pentaho Charting Extensions

This page last changed on Nov 28, 2006 by mbatchelor.

Description

JFreeReport and JFreeChart aren't tightly integrated - that is, neither project "knows" that the other project exists. Pentaho has provided some objects that can be used within the Pentaho platform for integrating these two projects at a fairly high level. However, to make them work together, you need to have two cooperative pieces - A Data Collector Function, and a Chart Expression. The data collector function is responsible for creating a JFreeChart dataset type that JFreeChart understands and can use as input data to plot the chart. The Chart Expression is used to actually draw the chart based on the collected data.

Pentaho currently provides two types of collectors:

Collect Chart Data

Function Class	Description
PieSetCollectorFunction	Data collector for Pie Charts
CategorySetCollectorFunction	Data collector for Bar, Line, and Area, and Multi-pie

Collector XML Examples

Pentaho 1.0 Collector XML Examples
Pentaho 1.1 and Later Collector XML Examples

Chart Properties and Hierarchy

Chart Properties And Hierarchy

Chart Expression XML Examples

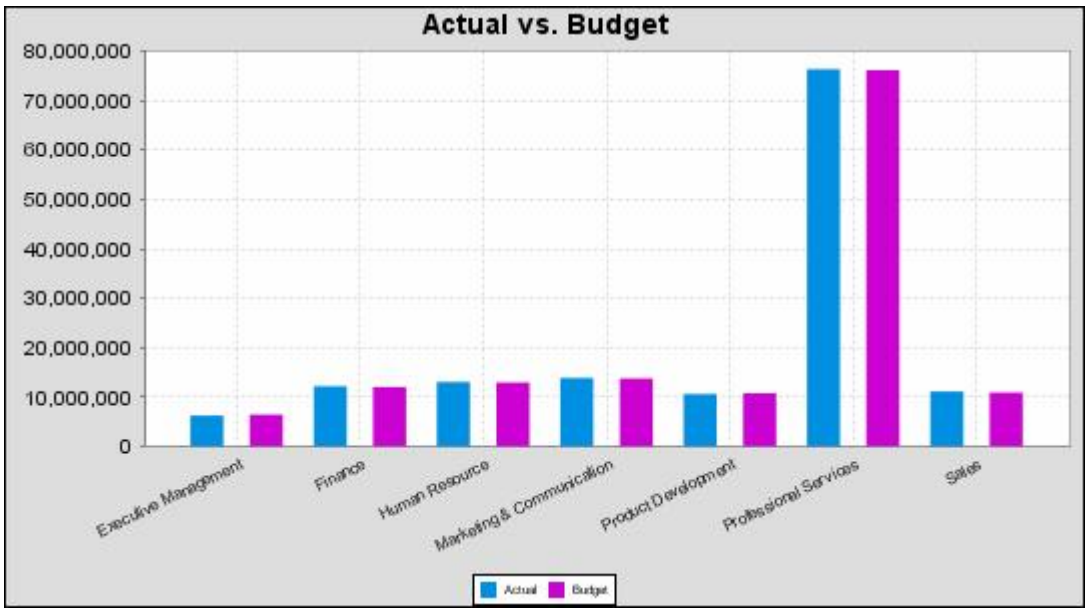
Pentaho 1.0 Chart Expression XML Examples
Pentaho 1.1 and Later Chart Expression XML Examples

CategorySetCollectorFunction

This page last changed on Nov 28, 2006 by mbatchelor.

Description

To generate categorical charts using JFreeChart, we need to create a JFreeChart dataset of type DefaultCategoryDataset. A categorical dataset is a three-dimensional (three column) dataset. The first dimension is the category column which defines the columns that contain the category names. The category can be considered a grouping of the bars (for example). The next dimension is the series. In bar charts for example, you'll have one bar for each series. Finally, the numerical value being plotted is the third dimension.



So, on the above bar chart, Executive Management, Finance, etc. are in the supplied category column (department). The first series name is Actual, and the second series name is Budget.

Chart types that work with CategorySetCollectorFunction

There are five chart types that work with Categorical datasets. The Area Chart (AreaChartExpression), the Bar Chart (BarChartExpression), the Line Chart (LineChartExpression), the Multi-pie chart (MultiPieChartExpression), and the Waterfall Chart (WaterfallChartExpression). For details on what each of these charts look like, please see the JFreeChart web site.

CategorySetCollectorFunction Properties

Property Name	Data Type	Description
categoryColumn	String	The name of the column containing category names. See

		above for explanation of category columns. Example: <property name="categoryColumn">DEPARTMENT</prop
seriesName	Array of Strings	This is a zero-based array containing the series name that appears on the chart. By zero-based, we mean that the first element in the array is element zero, not element 1. Example: <property name="seriesName[0]">Actual</property> <property name="seriesName[1]">Budget</property>
valueColumn	Array of Strings	This is a zero-based array contain the data column names that correspond with the seriesName array above. For each valueColumn array element, you need to have a corresponding seriesName (see above). Example: <property name="valueColumn[0]">ACTUAL</property> <property name="valueColumn[1]">BUDGET</property>
group (optional)	String	This provides the name of the group that the chart is appearing in. If the chart is for detail information, then this is not necessary. This is used in only in conjunction with summaryOnly (see below). When group is specified, then the data will only be gathered when the group is finished in the report (groupFinished event). When the specified group is finished, and when summaryOnly is true, then the data values are gathered from the seriesColumn and valueColumn. So, this assumes that the grouping is on the seriesColumn.
summaryOnly	Boolean (true or false only)	If summaryOnly is true, then when the items advance, no action is taken to accumulate data. This should only be used with a group specified so that

		when the groups advance, the data can be properly accumulated.
resetGroup (optional)	String	The resetGroup specifies a group that will be used to create charts within a sub-group. For example, if charting one chart per region, and the chart is displaying data for departments within a region, the resetGroup would be set to the region group, and the chart group would be set to department group. For an example, see the JFreeReport_Chart_Pie.xml.

Chart Properties And Hierarchy

This page last changed on Nov 28, 2006 by mbatchelor.

The chart expressions are arranged hierarchically, with children inheriting properties from their parent. All chart expressions descend from a common parent, and get more specific as the hierarchy gets deeper. Understanding the hierarchy is key to understanding what properties are available on each chart type. Some properties are common across all chart types - these are at the highest level. Some properties are specific to categorical charts while others are specific to Pie-type charts. Here is the hierarchy:

So, looking at the above organization, BarChartExpression has properties from AbstractChartExpression, CategoricalChartExpression, and StackedCategoricalChartExpression, and it may have it's own properties unique to Bar Charts as well. RingChartExpression has properties from AbstractChartExpression, and PieChartExpression. And, like the Bar Chart, may have additional properties that are unique to ring charts.

To avoid duplicating information, each of the expressions will be documented, and a reference to all of the ancestors will be included so it's possible to reference any additional properties.

Class Descendants

- AbstractChartExpression
  - CategoricalChartExpression
    - LineChartExpression
    - StackedCategoricalChartExpression
      - AreaChartExpression
      - BarChartExpression
    - WaterfallChartExpression
  - MultiPieChartExpression
  - PieChartExpression
    - RingChartExpression

AbstractChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

The abstract chart expression is at the top of the hierarchy. Therefore, it follows that it contains properties that can be applied to all of the different chart types.

AbstractChartExpression Properties

Property Name	Data Type	Description
antiAlias	Boolean (true/false)	Anti-aliasing is a technique of minimizing aliasing (jagged or blocky patterns) when plotting images. Default is true.
backgroundColor	String	Sets the color of the background of the chart. It is a color in either HTML notation (like #80f000), or one of the w3c defined color names (like "cadet blue").
backgroundImage	String	Defines an image that's used in rendering the chart background. The image specified needs to be relative to the solution root. Example that loads an image from the test solution in the charts folder within that solution. <property name="backgroundImage">test/charts/ChartB
borderColor	String	Sets the color of the border, if showBorder is set to true. It is a color in either HTML notation (like #80f000), or one of the w3c defined color names (like "cadet blue").
chartDirectory	String	Specifies the folder into which the charts will be generated. Defaults to the system/tmp folder within the root solution.
chartFile	String	Provides the ability to tell the engine the name of the file to generate. By default, the engine generates a temporary file.
chartHeight	Integer	Height of the chart graphic in pixels.
chartSectionOutline	Boolean (true/false)	If true, draws each chart section with an outline stroke. Default: false
chartUrlMask	String	Specifies a URL containing a

		replaceable marker ({0}) for the file name. This is useful when the chart image is generated into a folder, and a different webserver is used to serve up the chart images.
chartWidth	Integer	Width of the chart graphic in pixels.
dataSource	A xxxSetCollector function - see above	Before the chart can be rendered, it must have data collected by either a PieSetCollectorFunction or a CategoricalSetCollectorFunction. This needs to be the name of the xxxSetCollectorFunction in the report document.
drawLegendBorder	Boolean (true/false)	If showLegend is true, draws a border around the legend. Default: true
labelFont	String	Font for the labels. The format for the font is as follows: fontfamilyname-style-pointsize Style can be one of: BOLD, BOLDITALIC, or ITALIC Example: Arial-BOLD-14 Default: SansSerif—8
legendFont	String	The font used on the legend. The format for the font is as follows: fontfamilyname-style-pointsize Style can be one of: BOLD, BOLDITALIC, or ITALIC Example: Arial-BOLD-14 Default: SansSerif—8
legendLocation	String	Used for placing the legend, if showLegend is true. Valid values: left, west, right, east, top, north, bottom, south. Note that left and west are the same as are the other pairs. Default depends on the JFreeChart default settings.
returnFileNameOnly	Boolean	If true, the function will only return the name of the file that was generated. Default: false
showBorder	Boolean (true/false)	If true, the chart will have a border around it.
showLegend	Boolean (true/false)	If true, shows the legend on the chart.
threeD	Boolean (true/false)	If true, the chart is rendered in

		3D (with shading and an apparent thickness). Default: false
title	String	The chart title.
titleFont	String	The font for the chart title. The format for the font is as follows: fontfamilyname-style-pointsize Style can be one of: BOLD, BOLDITALIC, or ITALIC Example: Arial-BOLD-14 Default: SansSerif-BOLD-14
useDrawable	Boolean (true/false)	Determines how the chart gets returned to the JFreeReport engine. If false, then the chart image is rendered in a file for use with the image-url function in the chart. Note that with HTML reports, using drawable is not useful unless the HTML File System in place is something other than the StreamHTMLFileSystem (the default in Pentaho). Default: true

Class Descendants

- CategoricalChartExpression
  - LineChartExpression
  - StackedCategoricalChartExpression
    - AreaChartExpression
    - BarChartExpression
  - WaterfallChartExpression
- MultiPieChartExpression
- PieChartExpression
  - RingChartExpression

CategoricalChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

Description

The CategoricalChartExpression is the ancestor for all categorical chart types. It inherits from AbstractChartExpression, and adds several additional properties specific to categorical chart types.

CategoricalChartExpression Properties

Property Name	Data Type	Description
categoricalItemLabelRotation	Decimal	Rotation of the categorical item label, between 0 and 1.
categoricalLabelDateFormat	String	Only used if categoricalLabelFormat is specified. Should contain a valid Java SimpleDateFormat mask for formatting the date value. Use this to format Java Date types. For formatting numbers, see categoricalLabelDecimalFormat.
categoricalLabelDecimalFormat	String	Only used if categoricalLabelFormat is specified. Should contain a valid Java DecimalFormat mask for formatting the decimal number. Use this to format numbers. For formatting dates, see categoricalLabelDateFormat.
categoricalLabelFormat	String	When the categorical chart is generated, by default, the bars (for example) don't have any visible value. Sometimes, it's desirable to see the actual value associated with each bar, or each line. Can be a combination of text, and any of the following replaceable parameters: {0} - The series name. {1} - The (pre-formatted) x-value {2} - The (pre-formatted) y-value Default (from JFreeChart): Empty - no label
categoryAxisLabel	String	The label for the categorical axis (commonly called the X-Axis).

horizontal	Boolean (true/false)	If true, the x-axis and y-axis will be swapped. Commonly used for horizontal bar charts, but could also be used for
labelRotation	Decimal	Rotation percentage for the category labels. Example: <property name="labelRotation">7.0</property>
maxCategoryLabelWidthRatio	Decimal	Sets the maximum category label width. See the JFreeChart reference manual setMaximumCategoryLabelWidthRatio for more details.
seriesColors	Array of Strings	Zero-based array of strings, one color for each series (bar for example). Each element is a color in either HTML notation (like #80f000), or one of the w3c defined color names (like "cadet blue"). Example: <property name="seriesColor[0]">#018FE1</property> <property name="seriesColor[1]">#CB02D2</property>
showGridLines	Boolean (true/false)	If true, will show Domain grid lines (see the JFreeChart reference manual setDomainGridLinesVisible for additional details).
valueAxisLabel	String	The label for the value axis (commonly called the Y-Axis).

Class Descendants

- LineChartExpression
- StackedCategoricalChartExpression
  - AreaChartExpression
  - BarChartExpression
- WaterfallChartExpression

LineChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

The LineChartExpression inherits from CategoricalChartExpression and AbstractChartExpression. It adds no additional properties.

StackedCategoricalChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

Description

The StackedCategoricalChartExpression is the ancestor for all categorical charts that support stacking (like stacked area and stacked bar charts). It inherits from CategoricalChartExpression and AbstractChartExpression. It only adds one property to CategoricalChartExpression and is as follows:

StackedCategoricalChartExpression Properties

Property Name	Data Type	Description
stacked	Boolean (true/false)	If true, then the chart will be rendered as stacked. Default: false

Class Descendants

- AreaChartExpression
- BarChartExpression

AreaChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

The AreaChartExpression inherits from StackedCategoricalChartExpression, CategoricalChartExpression and AbstractChartExpression. It adds no additional properties.

BarChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

Description

The BarChartExpression inherits from StackedCategoricalChartExpression, CategoricalChartExpression and AbstractChartExpression. It adds some additional properties as follows:

BarChartExpression Properties

Property Name	Data Type	*Description*
drawBarOutline	Boolean (true/false)	If true, bars will be drawn with an outline. Default: false
maxBarWidth	Decimal	Sets the maximum width for the bars.
stackedBarRenderPercentages	Boolean (true/false)	Used only when stacked is true. If true, will render the bars as a percentage relative to the others in the category. Otherwise, it will render the stacked bars with their values.

WaterfallChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

The WaterfallChartExpression inherits from CategoricalChartExpression and AbstractChartExpression. It adds no additional properties.

MultiPieChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

Description

The MultiPieChartExpression is used to create a group of pie-charts (much like a dashboard). It is a categorical chart because you end up getting one pie chart per category. It inherits directly from AbstractChartExpression, and adds a few additional properties.

MultiPieChartExpression Properties

Property Name	Data Type	Description
multipieByRow	Boolean (true/false)	If true, then the data for the chart is organized by row. If false, then the data is organized by column. Default: true
multipieLabelFormat	String	When the pie charts are generated, the pie pieces of each pie are labeled using this label format mask. Can be a combination of text, and any of the following replaceable parameters: {0} - The item key {1} - The item value {2} - The item value as a percentage Default (from JFreeChart): {0} = {1}
seriesColors	Array of Strings	Zero-based array of strings, one color for each series (pie slice). Each element is a color in either HTML notation (like #80f000), or one of the w3c defined color names (like "cadet blue"). Example: <property name="seriesColor[0]">#018FE1</property> <property name="seriesColor[1]">#CB02D2</property>

PieChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

Definition

The PieChartExpression inherits all the properties from the AbstractChartExpression, and adds some pie-chart specific properties. These are listed below:

Properties

Property Name	Data Type	Description
circular	Boolean (true/false)	If true, the pie chart will only be generated in a circle. If false, depending upon the chart width and height, the chart may result in an ellipse. Default: true
explodePct	Decimal	The percent amount that the piece will be exploded expressed as a number between 0 and 1. For example, .20 explodes the piece 20%.
explodeSegment	String	The provided functionality allows for the explosion of the pie segment having the minimum value, the pie segment having the maximum value, or the actual pie segment as an integer. Valid values for this can be minValue, maxValue, or an integer representing the segment number to explode.
ignoreNulls	Boolean (true/false)	If true, then the chart plot will ignore values that are null, and they won't be factored into any percentages. Default: true
ignoreZeros	Boolean (true/false)	If true, then the chart plot will ignore any values that are zero, and they won't be factored into any percentages. Default: true
pieLabelFormat	String	When the pie chart is generated, the pie pieces are labeled using this label format mask. Can be a combination of text, and any of the following replaceable parameters: {0} - The item key

		{1} - The item value {2} - The item value as a percentage Default (from JFreeChart): {0} = {1}
pieLegendLabelFormat	String	When the legend is plotted, the labels are created using this format mask. Can be a combination of text, and any of the following replaceable parameters: {0} - The item key {1} - The item value {2} - The item value as a percentage Default: (from JFreeChart): {0} = {1}
rotationClockwise	Boolean (true/false)	If true, the pie chart is rendered with each successive pie piece going clockwise. If false, then counter-clockwise. Default: true

RingChartExpression

This page last changed on Nov 28, 2006 by mbatchelor.

Description

The RingChartExpression inherits from PieChartExpression and AbstractChartExpression, but it doesn't add any additional properties.

Pentaho 1.0 Chart Expression XML Examples

This page last changed on Nov 28, 2006 by mbatchelor.

Pie Chart Example

```
<expression class="org.pentaho.jfree.reportcharts.PieChartExpression" name="PieChart">
 <properties>
  <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
  <property name="title">RegionDepartmentChartTitle</property>
  <property name="titleFont">Arial-BOLD-14</property>><!--fontfamilyname-style-pointsize-->
    <property name="labelFont">Arial--8</property>
    <property name="legendFont">Arial--6</property>
    <property name="backgroundColor">#DDDDDD</property>
    <property name="showBorder">true</property>
    <property name="legendLocation">BOTTOM</property>
    <property name="showLegend">true</property>
    <property name="chartWidth">500</property>
    <property name="chartHeight">400</property>
    <property name="useDrawable">false</property>
    <property name="ignoreNulls">true</property>
    <property name="ignoreZeros">true</property>
    <property name="pieLabelFormat">{2}</property>
    <property name="pieLegendLabelFormat">{0}</property>
    <property name="explodeSegment">maxValue</property>
    <property name="explodePct">.20</property>
  </properties>
</expression>
```

Bar Chart Example

```
<expression class="org.pentaho.jfree.reportcharts.BarChartExpression" name="BarChart">
 <properties>
  <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
  <property name="title">Sales By Customer</property>
  <property name="titleFont">Arial-BOLD-9</property><!--fontfamilyname-style-pointsize -->
  <property name="labelFont">Arial--9</property>
  <property name="legendFont">Arial--6</property>
  <property name="backgroundColor">white</property>
  <property name="showBorder">false</property>
  <property name="horizontal">true</property>
  <property name="drawBarOutline">true</property>
  <property name="seriesColor[0]">#2954A9</property>
  <property name="legendLocation">BOTTOM</property>
  <property name="showLegend">false</property>
  <property name="maxBarWidth">0.10</property>
  <property name="maxCategoryLabelWidthRatio">7.0</property>
  <property name="chartHeight">600</property>
  <property name="chartWidth">480</property>
  <property name="useDrawable">true</property>
 </properties>
</expression>
```

Pentaho 1.0 Collector XML Examples

This page last changed on Nov 28, 2006 by mbatchelor.

These samples apply to the 1.0 version of Pentaho.

Line, Bar, and Area Chart Data Collector Example*

```
<expression class="org.pentaho.jfree.reportcharts.CategorySetCollectorFunction"
name="chartDataSource">
 <properties>
  <property name="categoryColumn">CUSTOMERNAME</property>
  <property name="seriesName[0]">Customer Sales</property>
  <property name="seriesName[1]">Customer Margin</property>
  <!-- <property name="seriesName[n]">Series Name</property> -->
  <property name="valueColumn[0]">Total Sales</property>
  <property name="valueColumn[1]">Margin</property>
  <!-- <property name="valueColumn[n]">Column Name</property>  -->
  <property name="summaryOnly">false</property>
  <!-- <property name="group">Customer_Type</property> --><!--  Only use if summaryOnly=true
-->
 </properties>
</expression>
```

Pie Chart Data Collector

```
<expression class="org.pentaho.jfree.reportcharts.PieSetCollectorFunction"
name="ChartDataSource">
  <properties>
     <property name="group">RegionGroup</property><!\-\-  Only use if summaryOnly=true \-\->
     <property name="seriesColumn">REGION</property>
     <property name="valueColumn">RegionBudget</property>
     <property name="summaryOnly">true</property>
  </properties>
</expression>
```

Pentaho 1.1 and Later Chart Expression XML Examples

This page last changed on Dec 01, 2006 by mbatchelor.

Pie Chart

```
<expression class="org.pentaho.plugin.jfreereport.reportcharts.PieChartExpression"
name="PieChart">
<properties>
  <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
  <property name="title">RegionDepartmentChartTitle</property>
  <property name="titleFont">Arial-BOLD-14</property>><!--fontfamilyname-style-pointsize-->
  <property name="labelFont">Arial--8</property>
  <property name="legendFont">Arial--6</property>
  <property name="backgroundColor">#DDDDDD</property>
  <property name="showBorder">true</property>
  <property name="legendLocation">BOTTOM</property>
  <property name="showLegend">true</property>
  <property name="chartWidth">500</property>
  <property name="chartHeight">400</property>
  <property name="useDrawable">false</property>
  <property name="ignoreNulls">true</property>
  <property name="ignoreZeros">true</property>
  <property name="pieLabelFormat">{2}</property>
  <property name="pieLegendLabelFormat">{0}</property>
  <property name="explodeSegment">maxValue</property>
  <property name="explodePct">.20</property>
 </properties>
</expression>
```

Bar Chart

```
<expression class="org.pentaho.plugin.jfreereport.reportcharts.BarChartExpression"
name="BarChart">
 <properties>
    <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
  <property name="title">Sales By Customer</property>
  <property name="titleFont">Arial-BOLD-9</property><!--fontfamilyname-style-pointsize -->
  <property name="labelFont">Arial--9</property>
  <property name="seriesColor[0]">#018FE1</property><!—HTML Color Code -->
  <property name="seriesColor[1]">#CB02D2</property><!—HTML Color Code -->
  <!-- <property name="seriesColor[n]">#CCCCCC</property><!—HTML Color Code --> -->
  <property name="legendFont">Arial--6</property>
  <property name="backgroundColor">white</property>
  <property name="showBorder">false</property>
  <property name="horizontal">true</property>
  <property name="drawBarOutline">true</property>
  <property name="legendLocation">BOTTOM</property>
  <property name="showLegend">false</property>
  <property name="maxBarWidth">0.10</property>
  <property name="maxCategoryLabelWidthRatio">7.0</property>
  <property name="chartHeight">600</property>
  <property name="chartWidth">480</property>
  <property name="useDrawable">true</property>
  <property name="stacked">false</property>
 </properties>
</expression>
```

Line Chart

```
"LineChart">
  <properties>
    <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
    <property name="title">Actual vs. Budget</property>
    <property name="titleFont">Arial-BOLD-14</property><!--fontfamilyname-style-pointsize -->
    <property name="labelFont">Arial--8</property>
    <property name="legendFont">Arial--6</property>
    <property name="backgroundColor">#DDDDDD</property>
    <property name="showBorder">true</property>
    <property name="horizontal">false</property>
    <property name="seriesColor[0]">#018FE1</property><!—HTML Color Code -->
    <property name="seriesColor[1]">#CB02D2</property><!—HTML Color Code -->
    <!-- <property name="seriesColor[n]">#CCCCCC</property><!—HTML Color Code --> -->
    <property name="legendLocation">BOTTOM</property>
    <property name="showLegend">true</property>
    <property name="chartHeight">300</property>
    <property name="chartWidth">540</property>
    <property name="useDrawable">false</property>
    <property name="labelRotation">7.0</property>
  </properties>
</expression>
```

Area Chart

```
<expression class="org.pentaho.plugin.jfreereport.reportcharts.AreaChartExpression"
name="AreaChart">
  <properties>
    <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
    <property name="title">Actual vs. Budget</property>
    <property name="titleFont">Arial-BOLD-14</property><!--fontfamilyname-style-pointsize -->
    <property name="labelFont">Arial--8</property><!--fontfamilyname-style-pointsize -->
    <property name="legendFont">Arial--6</property><!--fontfamilyname-style-pointsize -->
    <property name="backgroundColor">#DDDDDD</property>
    <property name="showBorder">true</property>
    <property name="horizontal">false</property>
    <property name="dataSource">chartDataSource</property>
    <property name="seriesColor[0]">#018FE1</property><!—HTML Color Code -->
    <property name="seriesColor[1]">#CB02D2</property><!—HTML Color Code -->
    <!-- <property name="seriesColor[n]">#CCCCCC</property><!—HTML Color Code --> -->
    <property name="legendLocation">BOTTOM</property>
    <property name="showLegend">true</property>
    <property name="chartHeight">300</property>
    <property name="chartWidth">540</property>
    <property name="useDrawable">false</property>
    <property name="labelRotation">7.0</property>
   </properties>
</expression>
```

Multi Pie Chart

```
<expression class="org.pentaho.plugin.jfreereport.reportcharts.MultiPieChartExpression"
name="MulitPieChart">
  <properties>
    <property name="dataSource">chartDataSource</property><!—Set Data Collector -->
    <property name="title">Actual vs. Budget</property>
    <property name="titleFont">Arial-BOLD-14</property><!--fontfamilyname-style-pointsize -->
    <property name="labelFont">Arial--8</property><!--fontfamilyname-style-pointsize -->
    <property name="legendFont">Arial--6</property><!--fontfamilyname-style-pointsize -->
    <property name="backgroundColor">#DDDDDD</property>
    <property name="showBorder">true</property>
    <property name="seriesColor[0]">#018FE1</property><!—HTML Color Code -->
    <property name="seriesColor[1]">#CB02D2</property><!—HTML Color Code -->
    <!-- <property name="seriesColor[n]">#CCCCCC</property><!—HTML Color Code --> -->
    <property name="legendLocation">BOTTOM</property>
    <property name="showLegend">true</property>
    <property name="chartHeight">300</property>
```

```
      <property name="chartWidth">540</property>
      <property name="useDrawable">false</property>
      <property name="multipieByRow">false</property>
      <property name="multipieLabelFormat">{2}</property>
    </properties>
  </expression>
```

Pentaho 1.1 and Later Collector XML Examples

This page last changed on Nov 28, 2006 by mbatchelor.

These examples apply to 1.1 and later of Pentaho.

Line, Bar, and Area Chart Data Collector Example

```
<expression class="org.pentaho.plugin.jfreereport.reportcharts.CategorySetCollectorFunction"
name="chartDataSource">
 <properties>
  <property name="categoryColumn">CUSTOMERNAME</property>
  <property name="seriesName[0]">Customer Sales</property>
  <property name="seriesName[1]">Customer Margin</property>
  <!-- <property name="seriesName[n]">Series Name</property> -->
  <property name="valueColumn[0]">Total Sales</property>
  <property name="valueColumn[1]">Margin</property>
  <!-- <property name="valueColumn[n]">Column Name</property>  -->
  <property name="summaryOnly">false</property>
  <!-- <property name="group">Customer_Type</property> --><!--  Only use if summaryOnly=true
-->
 </properties>
</expression>
```

Pie Chart Data Collector

```
<expression class="org.pentaho.plugin.jfreereport.reportcharts.PieSetCollectorFunction"
name="ChartDataSource">
  <properties>
     <property name="group">RegionGroup</property><!--  Only use if summaryOnly=true -->
     <property name="seriesColumn">REGION</property>
     <property name="valueColumn">RegionBudget</property>
     <property name="summaryOnly">true</property>
  </properties>
</expression>
```

PieSetCollectorFunction

This page last changed on Nov 28, 2006 by mbatchelor.

Description

To generate pie charts in JFreeChart, we need to create a JFreeChart dataset of type DefaultPieDataset. The PieSetCollectorFunction is designed so that you can tell it a little about the data columns, and it will collect the data necessary to build a DefaultPieDataset which can then be handed to JFreeChart for plotting. Here are the properties for this collector. Pie datasets require two and only two dimensions (columns) - one for the pie piece names (the series) and one for the values that are used to calculate size of each pie piece.

Chart types that work with PieSetCollectorFunction

There are two chart types that work with Pie datasets. The Pie Chart (PieChartExpression) and the RingChartExpression. For details on what each of these charts look like, please see the JFreeChart web site.

Property Details

Property Name	Data Type	Description
seriesColumn	String	This is the name of the column in the data set that contains all the series names. For example, if creating a pie chart containing one pie slice for each region, then the seriesColumn would be the column containing region names.
valueColumn	String	This is the name of the column in the data set that contains the numbers that will be accumulated for each series. In other words, this is the column containing the numbers for the pie chart.
group (optional)	String	This provides the name of the group that the chart is appearing in. If the chart is for detail information, then this is not necessary. This is used in only in conjunction with summaryOnly (see below). When group is specified, then the data will only be gathered when the group is finished in the report

		(groupFinished event). When the specified group is finished, and when summaryOnly is true, then the data values are gathered from the seriesColumn and valueColumn. So, this assumes that the grouping is on the seriesColumn.
summaryOnly	Boolean (true or false only)	If summaryOnly is true, then when the items advance, no action is taken to accumulate data. This should only be used with a group specified so that when the groups advance, the data can be properly accumulated.
resetGroup (optional)	String	The resetGroup specifies a group that will be used to create charts within a sub-group. For example, if charting one chart per region, and the chart is displaying data for departments within a region, the resetGroup would be set to the region group, and the chart group would be set to department group. For an example, see the JFreeReport_Chart_Pie.xml.

This page last changed on Dec 13, 2006 by mbatchelor.

Summary Calculations

- CountDistinctFunction — Counts the distinct occurrences of an certain value of an column. This functionality is similar to the SQL distinct() function.
- GroupCountFunction — A report function that counts groups in a report. If a null-group name is given, all groups are counted.
- ItemAvgFunction — A report function that calculates the average of one field (column) from the data source
- ItemColumnQuotientExpression — A report function that calculates the quotient of two fields (columns) from the current row.
- ItemCountFunction — A report function that counts items in a report.
- ItemHideFunction — Hides equal values in a group where only the first changed value is printed.
- ItemMaxFunction — A report function that calculates the maximum value of one field (column) from the data source.
- ItemMinFunction — A report function that calculates the minimum value of one field (column) from the data source
- ItemPercentageFunction — Calculates the percentage value of a numeric field.
- ItemSumFunction — A report function that calculates the sum of one field (column) from the data source.
- TotalCalculationFunction — A report function that stores the result of a calculation for a group or the complete report.
- TotalGroupCountFunction — A report function that counts the total of groups in a report.
- TotalGroupSumFunction — A report function that calculates the sum of one field (column) from the data source.
- TotalGroupSumQuotientFunction — A report function that calculates the quotient of two summed fields (columns) from the data source.
- TotalGroupSumQuotientPercentFunction — A report function that calculates the quotient of two summed fields (columns) from the data source.
- TotalItemCountFunction — A report function that counts the total number of items contained in groups in a report.

CountDistinctFunction

This page last changed on Dec 13, 2006 by mbatchelor.

Description

Counts the distinct occurrences of an certain value of an column. This functionality is similar to the SQL distinct() function.

Example

```
<function class="org.jfree.report.function.CountDistinctFunction" name="Cnt_Dstnct_USA">
  <properties>
    <property name="field">Country</property>
    <property name="group">Product</property>
    <property name="value">USA</property>
  </properties>
</function>
```

GroupCountFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that counts groups in a report. If a null-group name is given, all groups are counted.

The group to be counted is defined using the property "group". An optional container group can be defined using the property "parent-group". When the group start event of that group is encountered, the counter will be reset to '0'. If the group property is not set, all group starts get counted.

Example

```
<function class="org.jfree.report.function.GroupCountFunction" deplevel="" name="Orders Count">
  <properties>
    <property name="group">ORDER NUMBER GROUP</property>
    <property name="parentGroup">CUSTOMER GROUP</property>
  </properties>
</function>
```

ItemAvgFunction

This page last changed on Dec 01, 2006 by mbatchelor.

A report function that calculates the average of one field (column) from the data source. This function produces a running average, no global average. The function can be used in two ways:

- to calculate an average value for the entire report
- to calculate an average value within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands two parameters:

- field parameter is required and denotes the name of an ItemBand-field which gets summed up.
- group parameter denotes the name of a group. When this group is started, the counter gets reset to null.

```
<function class="org.jfree.report.function.ItemAvgFunction" name="Actual_Avg">
  <properties>
     <property name="field">ACTUAL</property>
     <property name="group">RegionGroup</property>
  </properties>
</function>
```

ItemColumnQuotientExpression

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that calculates the quotient of two fields (columns) from the current row.

This function expects its input values to be java.lang.Number instances.

The function understands two parameters.

- *dividend* parameter is required and denotes the name of an Item Section field which is used as dividend.
- *divisor* parameter is required and denotes the name of an Item Section field which is used as divisor.

Example

```
<expression name="saved" class="org.jfree.report.function.ItemColumnQuotientExpression">
  <properties>
    <property name="dividend">article.price</property>
    <property name="divisor">article.reducedPrice</property>
  </properties>
</expression>
```

ItemCountFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that counts items in a report. If the "group" property is set, the item count is reset to zero whenever the group changes.

Example

```
<function class="org.jfree.report.function.ItemCountFunction" name="Product_Count">
  <properties>
    <property name="field">Product</property>
    <property name="group">RegionGroup</property>
  </properties>
</function>
```

ItemHideFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

The ItemHideFunction hides equal values in a group. Only the first changed value is printed. This function uses the property tag to define the name of the object in the Item Section that should be made visible or invisible by this function. The property tag, field, defines the field in the datasource or the expression which should be used to determine the visibility.

Example

```
<function class="org.jfree.report.function.ItemHideFunction" name="HideCategory">
  <properties>
    <property name="field">Category</property>
  </properties>
</function>
```

ItemMaxFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that calculates the maximum value of one field (column) from the data source.

The function can be used in two ways:

- to calculate a maximum value for the entire report
- to calculate a maximum value within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands two parameters:

- *field* parameter is required and denotes the name of an ItemBand-field which gets summed up.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

Example

```
<function class="org.jfree.report.function.ItemMaxFunction" name="Actual_Sales">
  <properties>
    <property name="field">ACTUAL</property>
    <property name="group">RegionGroup</property>
  </properties>
</function>
```

ItemMinFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Definition

A report function that calculates the minimum value of one field (column) from the data source. The function can be used in two ways:

- to calculate a minimum value for the entire report
- to calculate a minimum value within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands two parameters:

- *field* parameter is required and denotes the name of an ItemBand-field which gets summed up.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

Example

```
<function class="org.jfree.report.function.ItemMinFunction" name="Actual_Sales">
  <properties>
    <property name="field">ACTUAL</property>
    <property name="group">RegionGroup</property>
  </properties>
</function>
```

ItemPercentageFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

Calculates the percentage value of a numeric field. The total sum is taken and divided by the number of items counted.

This function expects its input values to be java.lang.Number instances.

Example

```
<function class="org.jfree.report.function.ItemPercentageFunction" name="Item_Percentage">
  <properties>
    <property name="field">ACTUAL</property>
  </properties>
</function>
```

ItemSumFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that calculates the sum of one field (column) from the data source. This function produces a running total, no global total. For a global sum, use the TotalGroupSumFunction function. The function can be used in two ways:

- to calculate a sum for the entire report
- to calculate a sum within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands two parameters:

- *field* parameter is required and denotes the name of an ItemBand-field which gets summed up.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

Example

```
<expression class="org.jfree.report.function.ItemSumFunction" name="RegionActual">
  <properties>
    <property name="field">ACTUAL</property>
    <property name="group">RegionGroup</property>
  </properties>
</expression>
```

TotalCalculationFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that stores the result of a calculation for a group or the complete report. The field value, that was read when the group finished, is stored and returned when the group gets active again in a higher processing level.

This function can be used to calculate total values for a group, e.g. a TotalMax for the group, which returns the maximum value encountered for that field in the current group. The computed values are available to all bands of the group.

The function understands two parameters:

- *field* parameter is required and denotes the name of an ItemBand-field which gets summed up.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

TotalGroupCountFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that counts the total of groups in a report. If a null group name is given, all groups are counted.

A group can be defined using the property *group*. If the group property is not set, all group starts get counted.

Example

```
<function class="org.jfree.report.function.TotalGroupCountFunction" deplevel="" name="Orders
Count">
  <properties>
    <property name="group">ORDER NUMBER GROUP</property>
  </properties>
</function>
```

TotalGroupSumFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that calculates the sum of one field (column) from the data source.

This function produces a global total. The total sum of the group is known when the group processing starts and the report is not performing a prepare-run. The sum is calculated in the prepare run and recalled in the printing run.

The function can be used in two ways:

- to calculate a sum for the entire report
- to calculate a sum within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands two parameters:

- *field* parameter is required and denotes the name of an ItemBand-field which gets summed up.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

Example

```
<function class="org.jfree.report.function.TotalGroupSumFunction" deplevel="" name="Orders
Count">
  <properties>
    <property name="group">ORDER NUMBER GROUP</property>
    <property name="field">CUSTOMER</property>
  </properties>
</function>
```

TotalGroupSumQuotientFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that calculates the quotient of two summed fields (columns) from the data source. This function produces a global total. The total sum of the group is known when the group processing starts and the report is not performing a prepare-run. The sum is calculated in the prepare run and recalled in the printing run.

The function can be used in two ways:

- to calculate a quotient for the entire report
- to calculate a quotient within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands three parameters.

- *dividend* parameter is required and denotes the name of an Item Section field which is used as dividend.
- *divisor* parameter is required and denotes the name of an Item Section field which is used as divisor.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

Example

```
<function class="org.jfree.report.function.TotalGroupSumQuotientFunction" deplevel=""
name="Year_Over_Ratio">
  <properties>
    <property name="group">REGION GROUP</property>
    <property name="dividend">Y2000</property>
    <property name="divisor">Y2001</property>
  </properties>
</function>
```

TotalGroupSumQuotientPercentFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that calculates the quotient of two summed fields (columns) from the data source. This function produces a global total. The total sum of the group is known when the group processing starts and the report is not performing a prepare-run. The sum is calculated in the prepare run and recalled in the printing run.

The function can be used in two ways:

- to calculate a quotient for the entire report
- to calculate a quotient within a particular group

This function expects its input values to be java.lang.Number instances.

The function understands three parameters.

- *dividend* parameter is required and denotes the name of an Item Section field which is used as dividend.
- *divisor* parameter is required and denotes the name of an Item Section field which is used as divisor.
- *group* parameter denotes the name of a group. When this group is started, the counter gets reset to null.

```
<function class="org.jfree.report.function.TotalGroupSumQuotientPercentFunction" deplevel=""
name="Year_Over_Percentage">
  <properties>
    <property name="group">REGION GROUP</property>
    <property name="dividend">Y2000</property>
    <property name="divisor">Y2001</property>
  </properties>
</function>
```

TotalItemCountFunction

This page last changed on Dec 01, 2006 by mbatchelor.

Description

A report function that counts the total number of items contained in groups in a report. If a 'null' group name is given, all groups are counted.

A group can be defined using the property *group*. If it is not set, all group starts get counted.

Example

```
<function class="org.jfree.report.function.TotalItemCountFunction" deplevel=""
name="Total_Items_Counted">
  <properties>
    <property name="group">REGION GROUP</property>
  </properties>
</function>
```

09. Displaying the Chart

Use the imageurl-field to display the chart in the report.

```
<imageurl\-field dynamic="true" x="5%" y="35" fieldname="BarChart"/>
```

```
<imageurl\-field dynamic="true" x="5%" y="35" fieldname="LineChart"/>
```

```
<imageurl\-field dynamic="true" x="5%" y="35" fieldname="PieChart"/>
```

10. Configuration

Description

The following describes the property settings available.

CSV default separator

Defines the default separator, that should be used for the output.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.csv.Separator">,</property>
</configuration>
```

CSV write column name into first row

Defines whether to write the column names of the data row into the first row.

```
<configuration>
 <property name="org.jfree.report.modules.output.csv.WriteDatarowNames">true</property>
</configuration>
```

Excel output set paper size

Defines the Paper that should be used in the excel print setup. If no paper is defined, a suitable paper will be selected automatically.

```
<configuration>
 <property name=" org.jfree.report.modules.output.table.xls.Paper">LETTER</property>
</configuration>
```

Excel output set paper orientation

Defines the Paper that should be used in the excel print setup. If no paper is defined, an suitable paper will be selected automatically. Valid values are "Landscape" or "Portrait"

```
<configuration>
 <property name="org.jfree.report.modules.output.table.xls.PaperOrientation">LETTER</property>
</configuration>
```

HTML encoding

Defines the encoding for the HTML.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.html.Encoding">UTF-8</property>
</configuration>
```

HTML body fragment only

Defines whether to define a body fragment only. This removes the header and body tags and makes it easy to include the generated content in own pages.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.html.BodyFragment">false</property>
</configuration>
```

HTML, XHTML 1.0 or HTML 4.0 format

Defines whether to generate XHTML1.0 or HTML4.0 content.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.html.GenerateXHTML">false</property>
</configuration>
```

HTML default title

Defines the default title for generated HTML documents.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.html.Title">My First Report</property>
</configuration>
```

HTML set author

Defines a author for generated HTML documents.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.html.Author">Shakespeare</property>
</configuration>
```

HTML set empty cells to use CSS

Defines, whether empty cells will be controlled using CSS. If this is set to true, the Table gets the style property "empty-cells: show" assigned and the otherwise required no-break-space entities will no longer be generated.

The Internet Explorer is not able to use correct CSS2 yet. Thus, the default is set to "false".
This is a local configuration setting an can be enabled in the report.

```
<configuration>
 <property name="org.jfree.report.modules.output.table.html.EmptyCellsUseCSS">false</property>
</configuration>
```

HTML image size in Points or Pixels

Defines, whether image sizes should be given in device independent Point unit (1/72 inch) or in pixels (which is device dependent and makes it hard to predict the final size in the HTML-Page.

The default is pixels, false.

```
<configuration>
 <property
name="org.jfree.report.modules.output.table.html.UseDeviceIndependentImageSize">false</property>
</configuration>
```

HTML table carrier a border definition

Defines, whether table rows should carry a border definition (in addition to the cell definitions). This is a workaround for a layout bug in the Mozilla browser family.

```
<configuration>
 <property
name="org.jfree.report.modules.output.table.html.TableRowBorderDefinition">false</property>
</configuration>
```

HTML Table, use proportional column widths

Enables the use of proportional column widths instead of static ones. The resulting table will have a width of 100% and the columns will have the proportional equivalent of their static widths as width.

```
<configuration>
 <property
name="org.jfree.report.modules.output.table.html.ProportionalColumnWidths">false</property>
</configuration>
```

PDF set author

The author string of the document.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Author">Larry Augustin</property>
</configuration>
```

PDF set title

The title of the document.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Title">Sales Report</property>
</configuration>
```

PDF encoding

Defines PDF encoding for report.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Encoding">Cp1252</property>
</configuration>
```

PDF embed fonts

Defines whether to embed fonts in the generated PDF. This will result in larger PDF files, but makes sure that all characters are displayed properly.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.EmbedFonts">Cp1252</property>
</configuration>
```

PDF version

The PDF Version that should be created. Valid values are "1.2", "1.3","1.4" or "1.5"

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Version">1.4</property>
</configuration>
```

PDF Security, Allow Copy

Defines, whether the PDF Security setting allows copying of the document's contents by default. PDF Security flags are only used with security enabled and a password set.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.AllowCopy">false</property>
</configuration>
```

PDF Security, Allow Printing

Defines, whether the PDF Security setting allows printing of the document's contents by default

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.AllowPrinting">false</property>
</configuration>
```

PDF Security, Allow Modifying Contents

Defines, whether the PDF Security setting allows modifying of the document by default.

```
<configuration>
 <property
name="org.jfree.report.modules.output.pageable.pdf.AllowModifyContents">false</property>
</configuration>
```

PDF Security, Allow Modifying Annotations

Defines, whether the PDF Security setting allows document annotations by default.

```
<configuration>
 <property
name="org.jfree.report.modules.output.pageable.pdf.AllowModifyAnnotations">false</property>
</configuration>
```

PDF Security, Allow Fill-in of Document Forms

Defines, whether the PDF Security setting allows the fill-in of document forms (input fields etc).

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.AllowFillIn">false</property>
</configuration>
```

PDF Security, Allow access for screenreaders

Defines, whether the PDF Security setting allows access for screen readers by default. Although this could make it easier to grab the contents your PDF document, enabling this property will allow blind people to access your content. It is usually a good idea to always enable this property.

```
<configuration>
 <property
name="org.jfree.report.modules.output.pageable.pdf.AllowScreenReaders">false</property>
</configuration>
```

PDF Security, Allows reassembly of document

Defines, whether the PDF Security setting allows reassembly of the document by default.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.AllowAssembly">false</property>
</configuration>
```

PDF Security, Allow low quality printing

Defines, whether the PDF Security setting allows low quality printing of the document's contents by default.

```
<configuration>
 <property
name="org.jfree.report.modules.output.pageable.pdf.AllowDegradedPrinting">false</property>
</configuration>
```

PDF Security, Encrypt

Defines whether the PDF file should be encrypted by default. Set this to "none" "40bit" or "128bit". Access restrictions have no effect if the document is not encrypted, and not all security settings are active on 40-Bit encryption.

No encryption

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Encryption">none</property>
</configuration>
```

40 bit encryption

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Encryption">40bit</property>
</configuration>
```

128 bit encryption

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.Encryption">128bit</property>
</configuration>
```

Specifying passwords in the property file can introduce a security risk. Do not use these settings on the client side or in unprotected (world-readable) server environments. In these cases use other means of defining the password.

PDF Security, Password

The default user password for the document. Users are able to access the document within the defined security restrictions.

```
<configuration>
 <property name="org.jfree.report.modules.output.pageable.pdf.UserPassword">password</property>
</configuration>
```

PDF Security, Owner

The default owner password for the document. The owner has all permissions on the document and will be able to change the security settings.

```
<configuration>
 <property
name="org.jfree.report.modules.output.pageable.pdf.OwnerPassword">password</property>
</configuration>
```