z3c.RML Reference

Version 0.5

Introduction

RML is a XML dialect for generating PDF files. Like HTML produces a page within the browser, RML produces a PDF file. The RML processor uses the ReportLab library to convert the RML text into a full PDF template.

The original version of RML was developed by ReportLab, Inc. as a commercial extension to the free ReportLab library. This original version of RML is still available and supported by ReportLab, Inc. This version of RML, z3c.RML, is a free implementation of the XML dialect based on the available documentation. While it tries to keep some level of compatibility with the original version of RML, it is intended to provde a as clean and feature-rich API as possible.

The contents of this document is auto-generated from the code itself and should thus be very accurate and complete.

Attribute Types

This section list the types of attributes used for the attributes within the RML elements.

Boolean
A boolean value. For true the values "true", "yes", and "1" are allowed. For false, the values "false", "no", "1" are allowed.

BooleanWithDefault
This is a boolean field that can also receive the value "default".

Choice
A choice of several values. The values are always case-insensitive.

Color
Requires the input of a color. There are several supported formats. Three values in a row are interpreted as RGB value ranging from 0-255. A string is interpreted as a name to a pre-defined color. The 'CMYK()' wrapper around four values represents a CMYK color specification.

Combination
A combination of several other attribute types.

File
This field will return a file object. The value itself can eith be be a relative or absolute path. Additionally the following syntax is supported: [path.to.python.mpackage]/path/to/file

FirstLevelTextNode
Gets all the text content of an element without traversing into any child-elements.

Float
An flaoting point. A minimum and maximum value can be specified.

Image
Similar to the file File attribute, except that an image is internally expected.

Integer
An integer. A minimum and maximum value can be specified.

Measurement
This field represents a length value. The units "in" (inch), "cm", and "mm" are allowed. If no units are specified, the value is given in points/pixels.

PageSize
A simple measurement pair that specifies the page size. Optionally you can also specify a the name of a page size, such as A4, letter, or legal.

RawXMLContent
Retrieve the raw content of an element. Only some special element substitution will be made.

Sequence
A list of values of a specified type.

String
A simple Bytes string.

StringOrInt
A (bytes) string or an integer.

Style
Requires a valid style to be entered. Whether the style is a paragraph, table or box style is irrelevant, except that it has to fit the tag.

Symbol
This attribute should contain the text representation of a symbol to be used.

Text
A simple unicode string.

TextNode
Return the text content of an element.

TextNodeGrid
A grid/matrix of values retrieved from the element's content. The number of columns is specified for every case, but the number of rows is dynamic.

TextNodeSequence
A sequence of values retrieved from the element's content.

XMLContent
Same as 'RawXMLContent', except that the whitespace is normalized.

Directives

alias
Defines an alias for a given style.

Attributes

id *(required)* - String
Id: The id as which the style will be known.

value *(required)* - Style
Value: The style that is represented.

Examples

```
<alias id="h1" value="style.Heading1"/>
```
(Extracted from file *tag-alias.rml*, line 15)

bar
Define the look of a bar.

Attributes

strokeColor - Color
Stroke Color: The color in which the bar border is drawn.

strokeWidth - Measurement
Stroke Width: The width of the bar border line.

fillColor - Color
Fill Color: The color with which the bar is filled.

Examples

```
<bar fillColor="blue" strokeColor="red" strokeWidth="0.5"/>
```
(Extracted from file *tag-barChart.rml*, line 25)

barChart
Creates a two-dimensional bar chart.

Attributes

dx - Measurement
Drawing X-Position: The x-position of the entire drawing on the canvas.

dy - Measurement
Drawing Y-Position: The y-position of the entire drawing on the canvas.

dwidth - Measurement
Drawing Width: The width of the entire drawing

dheight - Measurement
Drawing Height: The height of the entire drawing

angle - Float
Angle: The orientation of the drawing as an angle in degrees.

x - Measurement
Chart X-Position: The x-position of the chart within the drawing.

y - Measurement
Chart Y-Position: The y-position of the chart within the drawing.

width - Measurement

Chart Width: The width of the chart.

height - Measurement
Chart Height: The height of the chart.

strokeColor - Color
Stroke Color: Color of the chart border.

strokeWidth - Measurement
Stroke Width: Width of the chart border.

fillColor - Color
Fill Color: Color of the chart interior.

debug - Boolean
Debugging: A flag that when set to True turns on debug messages.

direction - Choice of ('horizontal', 'vertical')
Direction: The direction of the bars within the chart.

useAbsolute - Boolean
Use Absolute Spacing: Flag to use absolute spacing values.

barWidth - Measurement
Bar Width: The width of an individual bar.

groupSpacing - Measurement
Group Spacing: Width between groups of bars.

barSpacing - Measurement
Bar Spacing: Width between individual bars.

Sub-Directives

data *(One)*
bars *(ZeroOrOne)*
categoryAxis *(ZeroOrOne)*
valueAxis *(ZeroOrOne)*
texts *(ZeroOrOne)*

Examples

```
<barChart dx="2in" dy="7in" dwidth="6in" dheight="4in" x="0" y="0" width="5in"
          height="3in" barSpacing="7" groupSpacing="15">
  <bars>
    <bar fillColor="blue" strokeColor="red" strokeWidth="0.5"/>
    <bar fillColor="yellow" strokeColor="green" strokeWidth="1"/>
  </bars>
  <categoryAxis strokeColor="black" strokeWidth="1">
    <labels fontName="Helvetica"/>
    <categoryNames>
      <name>Category 1</name>
      <name>Category 2</name>
      <name>Category 3</name>
      <name>Category 4</name>
    </categoryNames>
  </categoryAxis>
  <valueAxis valueMin="0" valueMax="150" valueStep="30" visibleTicks="true"
             visibleLabels="true">
    <labels fontName="Helvetica"/>
  </valueAxis>
  <data>
    <series>100 110 120 130</series>
    <series> 70  80  85  90</series>
  </data>
</barChart>
```

(Extracted from file *tag-barChart.rml*, line 22)

barChart3D

Creates a three-dimensional bar chart.

Attributes

dx - Measurement
Drawing X-Position: The x-position of the entire drawing on the canvas.

dy - Measurement
Drawing Y-Position: The y-position of the entire drawing on the canvas.

dwidth - Measurement
Drawing Width: The width of the entire drawing

dheight - Measurement
Drawing Height: The height of the entire drawing

angle - Float
Angle: The orientation of the drawing as an angle in degrees.

x - Measurement
Chart X-Position: The x-position of the chart within the drawing.

y - Measurement
Chart Y-Position: The y-position of the chart within the drawing.

width - Measurement
Chart Width: The width of the chart.

height - Measurement
Chart Height: The height of the chart.

strokeColor - Color
Stroke Color: Color of the chart border.

strokeWidth - Measurement
Stroke Width: Width of the chart border.

fillColor - Color
Fill Color: Color of the chart interior.

debug - Boolean
Debugging: A flag that when set to True turns on debug messages.

direction - Choice of ('horizontal', 'vertical')
Direction: The direction of the bars within the chart.

useAbsolute - Boolean
Use Absolute Spacing: Flag to use absolute spacing values.

barWidth - Measurement
Bar Width: The width of an individual bar.

groupSpacing - Measurement
Group Spacing: Width between groups of bars.

barSpacing - Measurement
Bar Spacing: Width between individual bars.

thetaX - Float
Theta-X: Fraction of dx/dz.

thetaY - Float
Theta-Y: Fraction of dy/dz.

zDepth - Measurement
Z-Depth: Depth of an individual series/bar.

zSpace - Measurement
Z-Space: Z-Gap around a series/bar.

Sub-Directives

data *(One)*

bars *(ZeroOrOne)*
categoryAxis *(ZeroOrOne)*
valueAxis *(ZeroOrOne)*
texts *(ZeroOrOne)*

Examples

```
<barChart3D dx="2in" dy="4in" dwidth="6in" dheight="4in" x="0" y="0"
            width="5in" height="2in" thetaX="0.3" thetaY="0.3" zDepth="5"
            zSpace="10" direction="vertical">
  <categoryAxis strokeColor="black" strokeWidth="1" visibleGrid="true">
    <labels fontName="Helvetica"/>
    <categoryNames>
      <name>Category 1</name>
      <name>Category 2</name>
      <name>Category 3</name>
      <name>Category 4</name>
    </categoryNames>
  </categoryAxis>
  <valueAxis valueMin="0" valueMax="150" valueStep="30" visibleTicks="true"
             visibleLabels="true" visibleGrid="true">
    <labels fontName="Helvetica"/>
  </valueAxis>
  <data>
    <series>100 110 120 130</series>
    <series> 70  80  85  90</series>
  </data>
</barChart3D>
```

(Extracted from file *tag-barChart3d.rml*, line 48)

barCode

A barcode graphic.

Attributes

code *(required)* - Choice of ('ean13', 'extended39', 'standard39', 'ean8', 'extended93', 'codabar', 'msi', 'postnet', 'fim', 'code11', 'standard93', 'i2of5', 'code128')
Code: The name of the type of code to use.

width - Measurement
Width: The width of the barcode.

height - Measurement
Height: The height of the barcode.

strokeColor - Color
Stroke Color: The color of the line strokes in the area.

strokeWidth - Measurement
Stroke Width: The width of the line strokes in the area.

fillColor - Color
Fill Color: The color of the filled shapes in the area.

barStrokeColor - Color
Bar Stroke Color: The color of the line strokes in the barcode.

barStrokeWidth - Measurement
Bar Stroke Width: The width of the line strokes in the barcode.

barFillColor - Color
Bar Fill Color: The color of the filled shapes in the barcode.

gap - Measurement
Gap: The width of the inter-character gaps.

barWidth - Measurement

Bar Width: The width of the smallest bar within the barcode

barHeight - Measurement
Bar Height: The height of the symbol.

ratio - Float
Ratio: The ratio of wide elements to narrow elements. Must be between 2.0 and 3.0 (or 2.2 and 3.0 if the barWidth is greater than 20 mils (.02 inch)).

checksum - Integer
Ratio: A flag that enables the computation and inclusion of the check digit.

bearers - Float
Bearers: Height of bearer bars (horizontal bars along the top and bottom of the barcode). Default is 3 x-dimensions. Set to zero for no bearer bars.(Bearer bars help detect misscans, so it is suggested to leave them on).

quiet - Boolean
Quiet Zone: A flag to include quiet zones in the symbol.

lquiet - Measurement
Left Quiet Zone: Quiet zone size to the left of code, if quiet is true. Default is the greater of .25 inch or .15 times the symbol's length.

rquiet - Measurement
Right Quiet Zone: Quiet zone size to the right of code, if quiet is true. Default is the greater of .25 inch or .15 times the symbol's length.

fontName - String
Font Name: The font used to print the value.

fontSize - Measurement
Font Size: The size of the value text.

humanReadable - Boolean
Human Readable: A flag when set causes the value to be printed below the bar code.

stop - Boolean
Show Start/Stop: A flag to specify whether the start/stop symbols are to be shown.

spaceWidth - Measurement
Space Width: The space of the inter-character gaps.

shortHeight - Measurement
Short Height: The height of the short bar.

textColor - Color
Text Color: The color of human readable text.

x - Measurement
X-Position: The x-position of the lower-left corner of the barcode.

y - Measurement
Y-Position: The y-position of the lower-left corner of the barcode.

Content

TextNode *(required)*
Value: The value represented by the code.

Examples

```
<barCode x="5cm" y="24cm" code="MSI">
  23465092892
</barCode>
```

(Extracted from file *tag-barcode.rml*, line 14)

```
<barCode x="5cm" y="17cm" height="2cm" width="5cm" code="Ean13"
         humanReadable="true" fontName="Helvetica" fontSize="7"
         barStrokeColor="blue" barFillColor="blue" textColor="blue"
         quiet="false" barHeight="0.4in" barWidth="0.009in">
  123456789012
```

```
</barCode>
```
(Extracted from file *tag-barcode.rml*, line 28)

barCodeFlowable

Creates a bar code as a flowable.

Attributes

code *(required)* - Choice of ('ean13', 'extended39', 'standard39', 'ean8', 'extended93', 'codabar', 'msi', 'postnet', 'fim', 'code11', 'standard93', 'i2of5', 'code128')
Code: The name of the type of code to use.

width - Measurement
Width: The width of the barcode.

height - Measurement
Height: The height of the barcode.

strokeColor - Color
Stroke Color: The color of the line strokes in the area.

strokeWidth - Measurement
Stroke Width: The width of the line strokes in the area.

fillColor - Color
Fill Color: The color of the filled shapes in the area.

barStrokeColor - Color
Bar Stroke Color: The color of the line strokes in the barcode.

barStrokeWidth - Measurement
Bar Stroke Width: The width of the line strokes in the barcode.

barFillColor - Color
Bar Fill Color: The color of the filled shapes in the barcode.

gap - Measurement
Gap: The width of the inter-character gaps.

barWidth - Measurement
Bar Width: The width of the smallest bar within the barcode

barHeight - Measurement
Bar Height: The height of the symbol.

ratio - Float
Ratio: The ratio of wide elements to narrow elements. Must be between 2.0 and 3.0 (or 2.2 and 3.0 if the barWidth is greater than 20 mils (.02 inch)).

checksum - Integer
Ratio: A flag that enables the computation and inclusion of the check digit.

bearers - Float
Bearers: Height of bearer bars (horizontal bars along the top and bottom of the barcode). Default is 3 x-dimensions. Set to zero for no bearer bars.(Bearer bars help detect misscans, so it is suggested to leave them on).

quiet - Boolean
Quiet Zone: A flag to include quiet zones in the symbol.

lquiet - Measurement
Left Quiet Zone: Quiet zone size to the left of code, if quiet is true. Default is the greater of .25 inch or .15 times the symbol's length.

rquiet - Measurement
Right Quiet Zone: Quiet zone size to the right of code, if quiet is true. Default is the greater of .25 inch or .15 times the symbol's length.

fontName - String
Font Name: The font used to print the value.

fontSize - Measurement

Font Size: The size of the value text.

humanReadable - Boolean
Human Readable: A flag when set causes the value to be printed below the bar code.

stop - Boolean
Show Start/Stop: A flag to specify whether the start/stop symbols are to be shown.

spaceWidth - Measurement
Space Width: The space of the inter-character gaps.

shortHeight - Measurement
Short Height: The height of the short bar.

textColor - Color
Text Color: The color of human readable text.

value *(required)* - String
Value: The value represented by the code.

Examples

```
<barCodeFlowable code="Code128" value="PFWZF"/>
```
(Extracted from file *tag-barCodeFlowable.rml*, line 26)

```
<barCodeFlowable code="Code128" value="PFWZF" humanReadable="true"
                 fontName="Helvetica" fontSize="10" barFillColor="red"
                 barStrokeColor="red" quiet="false" barHeight="0.4in"
                 barWidth="0.009in"/>
```
(Extracted from file *tag-barCodeFlowable.rml*, line 35)

bars
Collection of bar subscriptions.

Attributes

strokeColor - Color
Stroke Color: The color in which the bar border is drawn.

strokeWidth - Measurement
Stroke Width: The width of the bar border line.

fillColor - Color
Fill Color: The color with which the bar is filled.

Sub-Directives

bar *(ZeroOrMore)*

blockAlignment
Set the text alignment.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

value *(required)* - Choice of ('decimal', 'right', 'center', 'centre', 'left')
Text Alignment: The text alignment within the cell.

Examples

```
<blockAlignment start="0,0" stop="-1,1" value="center"/>
```
(Extracted from file *tag-blockTableStyle.rml*, line 16)

blockBackground
Define the background color of the cells. It also supports alternating colors.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

colorName - Color
Color Name: The color to use as the background for every cell.

colorsByRow - Sequence of Color
Colors By Row: A list of colors to be used circularly for rows.

colorsByCol - Sequence of Color
Colors By Column: A list of colors to be used circularly for columns.

Examples

```
<blockBackground start="1,1" stop="-2,-2" colorName="green"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 18)

blockBottomPadding

Set the bottom padding of the cells.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

length *(required)* - Measurement
Length: The size of the padding.

Examples

```
<blockBottomPadding start="0,0" stop="-1,1" length="5mm"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 24)

blockColBackground

Define the background colors for columns.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

colorNames *(required)* - Sequence of Color
Colors By Row: A list of colors to be used circularly for rows.

Examples

```
<blockColBackground start="0,0" stop="2,-1" colorNames="red green"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 37)

blockFont

Set the font properties for the texts.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

name - String
Font Name: The name of the font for the cell.

size - Measurement
Font Size: The font size for the text of the cell.

leading - Measurement
Leading: The height of a single text line. It includes character height.

Examples

```
<blockFont start="0,0" stop="1,-1" name="Courier" size="14" leading="18"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 29)

blockLeading

Set the text leading.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

length *(required)* - Measurement
Length: The height of a single text line. It includes character height.

Examples

```
<blockLeading start="0,0" stop="-1,1" length="18"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 43)

blockLeftPadding

Set the left padding of the cells.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

length *(required)* - Measurement
Length: The size of the padding.

Examples

```
<blockLeftPadding start="0,0" stop="-1,1" length="5mm"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 26)

blockRightPadding

Set the right padding of the cells.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

length *(required)* - Measurement
Length: The size of the padding.

Examples

```
<blockRightPadding start="0,0" stop="-1,1" length="5mm"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 22)

blockRowBackground

Define the background colors for rows.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

colorNames *(required)* - Sequence of Color
Colors By Row: A list of colors to be used circularly for rows.

Examples

```
<blockRowBackground start="3,0" stop="-1,-1" colorNames="blue yellow"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 40)

blockSpan

Define a span over multiple cells (rows and columns).

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

Examples

```
<blockSpan start="0,0" stop="2,2"/>
```

(Extracted from file *tag-blockTableStyle.rml*, line 48)

blockTable

A typical block table.

Attributes

style - Style
Style: The table style that is applied to the table.

rowHeights - Sequence of Measurement
Row Heights: A list of row heights in the table.

colWidths - Sequence of Measurement
Column Widths: A list of column widths in the table.

repeatRows - Integer
Repeat Rows: A flag to repeat rows upon table splits.

Sub-Directives

tr *(ZeroOrMore)*
bulkData *(ZeroOrOne)*
blockTableStyle *(ZeroOrMore)*

Examples

```
<blockTable colWidths="50% 50%" rowHeights="1cm 1cm">
  <tr>
   <td>This</td>
   <td>is</td>
  </tr>
  <tr>
```

```
      <td>a</td>
      <td>blockTable.</td>
    </tr>
  </blockTable>
```
(Extracted from file *tag-blockTable-1.rml*, line 17)

blockTableStyle

A style defining the look of a table.

Attributes

id *(required)* - String
Id: The name/id of the style.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

Sub-Directives

blockFont *(ZeroOrMore)*
blockLeading *(ZeroOrMore)*
blockTextColor *(ZeroOrMore)*
blockAlignment *(ZeroOrMore)*
blockLeftPadding *(ZeroOrMore)*
blockRightPadding *(ZeroOrMore)*
blockBottomPadding *(ZeroOrMore)*
blockTopPadding *(ZeroOrMore)*
blockBackground *(ZeroOrMore)*
blockRowBackground *(ZeroOrMore)*
blockColBackground *(ZeroOrMore)*
blockValign *(ZeroOrMore)*
blockSpan *(ZeroOrMore)*
lineStyle *(ZeroOrMore)*

Examples

```
<blockTableStyle id="custom-table">
  <blockFont start="0,0" stop="-1,-1" name="Courier-Bold" size="10"/>
</blockTableStyle>
```
(Extracted from file *tag-blockTableStyle.rml*, line 56)

blockTextColor

Set the text color.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

colorName *(required)* - Color
Color Name: The color in which the text will appear.

Examples

```
<blockTextColor start="0,0" stop="1,-1" colorName="red"/>
```
(Extracted from file *tag-blockTableStyle.rml*, line 31)

blockTopPadding

Set the top padding of the cells.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')

Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

length *(required)* - Measurement
Length: The size of the padding.

Examples

```
<blockTopPadding start="0,0" stop="-1,1" length="5mm"/>
```
(Extracted from file *tag-blockTableStyle.rml*, line 20)

blockValign

Define the vertical alignment of the cells.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

value *(required)* - Choice of ('middle', 'top', 'bottom')
Vertical Alignment: The vertical alignment of the text with the cells.

Examples

```
<blockValign start="0,0" stop="2,2" value="middle"/>
```
(Extracted from file *tag-blockTableStyle.rml*, line 53)

bookmark

This creates a bookmark to the current page which can be referred to with the given key elsewhere. PDF offers very fine grained control over how Acrobat reader is zoomed when people link to this. The default is to keep the user's current zoom settings. the last arguments may or may not be needed depending on the choice of 'fitType'.

Attributes

name *(required)* - Text
Name: The name of the bookmark.

fitType - Choice of ('fitr', 'fith', 'fitv', 'fit')
Fit Type: The Fit Type.

left - Measurement
Left: The left position.

top - Measurement
Top: The top position.

right - Measurement
Right: The right position.

zoom - Float
Zoom: The zoom level when clicking on the bookmark.

Examples

```
<bookmark name="TITLE"/>
```
(Extracted from file *tag-bookmark.rml*, line 20)

```
<bookmark name="PAGE_1" fitType="fitv" zoom="2" left="2cm" right="10cm"
          top="20cm"/>
```
(Extracted from file *tag-bookmark.rml*, line 31)

bulkData

Bulk Data allows one to quickly create a table.

Content

TextNodeSequence of Sequence of Text *(required)*
Content: The bulk data.

Examples

```
<bulkData>
Product,Profit
Sprockets,26
Widgets,34
Thingummies,217
Bits &amp;amp; Bobs,23
Total,277
</bulkData>
```

(Extracted from file *tag-blockTable-bulkData.rml*, line 16)

buttonField

A button field within the PDF

Attributes

title *(required)* - Text
Title: The title of the field.

x *(required)* - Measurement
X-Position: The x-position of the lower-left corner of the field.

y *(required)* - Measurement
Y-Position: The y-position of the lower-left corner of the field.

value *(required)* - Choice of ('yes', 'off')
Value: The value of the button.

categoryAxis

Attributes

visible - Boolean

visibleAxis - Boolean

visibleTicks - Boolean

visibleLabels - Boolean

visibleGrid - Boolean

strokeWidth - Measurement

strokeColor - Color

strokeDashArray - Sequence of Float

gridStrokeWidth - Measurement

gridStrokeColor - Color

gridStrokeDashArray - Sequence of Float

gridStart - Measurement

gridEnd - Measurement

style - Choice of ('stacked', 'parallel', 'parallel_3d')

categoryNames - Sequence of Text

joinAxis - Boolean

joinAxisPos - Measurement

reverseDirection - Boolean

labelAxisMode - Choice of ('high', 'low', 'axis')

tickShift - Boolean

Sub-Directives

categoryNames *(ZeroOrOne)*
labels *(ZeroOrMore)*

categoryNames

Attributes

visible - Boolean

visibleAxis - Boolean

visibleTicks - Boolean

visibleLabels - Boolean

visibleGrid - Boolean

strokeWidth - Measurement

strokeColor - Color

strokeDashArray - Sequence of Float

gridStrokeWidth - Measurement

gridStrokeColor - Color

gridStrokeDashArray - Sequence of Float

gridStart - Measurement

gridEnd - Measurement

style - Choice of ('stacked', 'parallel', 'parallel_3d')

Sub-Directives

name *(OneOrMore)*

circle

Draws a circle on the canvas.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the shape.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the shape.

fill - Boolean
Fill: A flag to specify whether the shape should be filled.

stroke - Boolean
Stroke: A flag to specify whether the shape's outline should be drawn.

radius *(required)* - Measurement
Radius: The radius of the circle.

Examples

```
<circle x="10cm" y="25cm" radius="2cm" fill="false" stroke="false"/>
```

(Extracted from file *tag-circle.rml*, line 10)

color

Define a new color and give it a name to be known under.

Attributes

id *(required)* - String

Id: The id/name the color will be available under.

RGB *(required)* - Color
Color: The color value that is represented.

condPageBreak

Switch to the next page if not enough vertical space is available.

Attributes

height *(required)* - Measurement
height: The minimal height that must be remaining on the page.

curves

A path of connected bezier curves drawn on the canvas.

Content

TextNodeGrid with 8 cols of Measurement *(required)*
Curve List: A list of curve coordinates to draw.

Examples

```
<curves>
  1in 1in 2in 2in 2in 3in 1in 3in
  1in 2in 2in 3in 2in 4in 1in 4in
  1in 3in 2in 4in 2in 5in 1in 5in
</curves>
```

(Extracted from file *tag-curves.rml*, line 9)

curvesto

Create a bezier curve from the current location to the specified one.

Content

TextNodeGrid with 6 cols of Measurement *(required)*
Curve Specification: Describes the end position and the curve properties.

data

A 2-D data set.

Sub-Directives

series *(OneOrMore)*

data

A 1-D data set.

Sub-Directives

series *(One)*

data

A 1-D data set.

Sub-Directives

series *(OneOrMore)*

docinit

Sub-Directives

registerType1Face *(ZeroOrMore)*
registerFont *(ZeroOrMore)*

registerTTFont *(ZeroOrMore)*
registerCidFont *(ZeroOrMore)*
color *(ZeroOrMore)*

document

Attributes

filename *(required)* - String
File Name: The default name of the output file, if no output file was provided.

debug - Boolean
Debug: A flag to activate the debug output.

compression - BooleanWithDefault
Compression: A flag determining whether page compression should be used.

invariant - BooleanWithDefault
Invariant: A flag that determines whether the produced PDF should be invariant with respect to the date and the exact contents.

Sub-Directives

docinit *(ZeroOrOne)*
stylesheet *(ZeroOrOne)*
template *(ZeroOrOne)*
story *(ZeroOrOne)*
pageInfo *(ZeroOrOne)*
pageDrawing *(ZeroOrMore)*

drawAlignedString

Draws a simple string (aligned to the pivot character) onto the canvas at the specified location.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the string.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the string.

pivotChar *(required)* - Text
Text: The string/text that is put onto the canvas.

Content

TextNode *(required)*
Text: The string/text that is put onto the canvas.

Examples

```
<drawAlignedString x="4.1in" y="9.8in">$ 13.63</drawAlignedString>
```
(Extracted from file *tag-drawAlignedString.rml*, line 11)

drawCenteredString

Draws a simple string (centered aligned) onto the canvas at the specified location.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the string.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the string.

Content

TextNode *(required)*

Text: The string/text that is put onto the canvas.

drawRightString

Draws a simple string (right aligned) onto the canvas at the specified location.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the string.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the string.

Content

TextNode *(required)*
Text: The string/text that is put onto the canvas.

Examples

```
<drawRightString x="4.1in" y="5.8in">Hello World.</drawRightString>
```

(Extracted from file *tag-drawRightString.rml*, line 12)

drawString

Draws a simple string (left aligned) onto the canvas at the specified location.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the string.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the string.

Content

TextNode *(required)*
Text: The string/text that is put onto the canvas.

Examples

```
<drawString x="4.1in" y="5.8in">Hello World.</drawString>
```

(Extracted from file *tag-drawString.rml*, line 9)

ellipse

Draws an ellipse on the canvas.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the shape.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the shape.

fill - Boolean
Fill: A flag to specify whether the shape should be filled.

stroke - Boolean
Stroke: A flag to specify whether the shape's outline should be drawn.

width *(required)* - Measurement
Width: The width of the ellipse.

height *(required)* - Measurement
Height: The height of the ellipse.

Examples

```
<ellipse x="10cm" y="25cm" width="5cm" height="3cm" fill="false"
         stroke="false"/>
```
(Extracted from file *tag-ellipse.rml*, line 10)

fill
Set the fill color.

Attributes

color *(required)* - Color
Color: The color value to be set.

fixedSize
Create a container flowable of a fixed size.

Attributes

width *(required)* - Measurement
Width: The width the flowables are allotted.

height *(required)* - Measurement
Height: The height the flowables are allotted.

frame
A frame on a page.

Attributes

x1 *(required)* - Measurement
X-Position: The X-Position of the lower-left corner of the frame.

y1 *(required)* - Measurement
Y-Position: The Y-Position of the lower-left corner of the frame.

width *(required)* - Measurement
Width: The width of the frame.

height *(required)* - Measurement
Height: The height of the frame.

id - Text
Id: The id of the frame.

leftPadding - Measurement
Left Padding: The left padding of the frame.

rightPadding - Measurement
Right Padding: The right padding of the frame.

topPadding - Measurement
Top Padding: The top padding of the frame.

bottomPadding - Measurement
Bottom Padding: The bottom padding of the frame.

showBoundary - Boolean
Show Boundary: A flag to show the boundary of the frame.

grid
A shape to be drawn on the canvas.

Attributes

xs *(required)* - Sequence of Measurement
X-Coordinates: A sequence x-coordinates that represent the vertical line positions.

ys *(required)* - Sequence of Measurement
Y-Coordinates: A sequence y-coordinates that represent the horizontal line positions.

Examples

```
<grid xs="1in 2in 3in 4in 5in 6in" ys="7in 7.2in 7.4in 7.6in 7.8in 8.0in"/>
```

(Extracted from file *tag-grid.rml*, line 13)

h1

Heading 1 is a simple paragraph with a special heading 1 style.

Attributes

fontName - String
Font Name: The name of the font for the paragraph.

fontSize - Measurement
Font Size: The font size for the text of the paragraph.

leading - Measurement
Leading: The height of a single paragraph line. It includes character height.

leftIndent - Measurement
Left Indentation: General indentation on the left side.

rightIndent - Measurement
Right Indentation: General indentation on the right side.

firstLineIndent - Measurement
First Line Indentation: The indentation of the first line in the paragraph.

spaceBefore - Measurement
Space Before: The vertical space before the paragraph.

spaceAfter - Measurement
Space After: The vertical space after the paragraph.

alignment - Choice of ('right', 'justify', 'center', 'centre', 'left')
Alignment: The text alignment.

bulletFontName - String
Bullet Font Name: The font in which the bullet character will be rendered.

bulletFontSize - Measurement
Bullet Font Size: The font size of the bullet character.

bulletIndent - Measurement
Bullet Indentation: The indentation that is kept for a bullet point.

textColor - Color
Text Color: The color in which the text will appear.

backColor - Color
Background Color: The background color of the paragraph.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

Content

XMLContent *(required)*
Text: The text that will be layed out.

h2

Heading 2 is a simple paragraph with a special heading 2 style.

Attributes

fontName - String
Font Name: The name of the font for the paragraph.

fontSize - Measurement
Font Size: The font size for the text of the paragraph.

leading - Measurement
Leading: The height of a single paragraph line. It includes character height.

leftIndent - Measurement
Left Indentation: General indentation on the left side.

rightIndent - Measurement
Right Indentation: General indentation on the right side.

firstLineIndent - Measurement
First Line Indentation: The indentation of the first line in the paragraph.

spaceBefore - Measurement
Space Before: The vertical space before the paragraph.

spaceAfter - Measurement
Space After: The vertical space after the paragraph.

alignment - Choice of ('right', 'justify', 'center', 'centre', 'left')
Alignment: The text alignment.

bulletFontName - String
Bullet Font Name: The font in which the bullet character will be rendered.

bulletFontSize - Measurement
Bullet Font Size: The font size of the bullet character.

bulletIndent - Measurement
Bullet Indentation: The indentation that is kept for a bullet point.

textColor - Color
Text Color: The color in which the text will appear.

backColor - Color
Background Color: The background color of the paragraph.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

Content

XMLContent *(required)*
Text: The text that will be layed out.

h3

Heading 3 is a simple paragraph with a special heading 3 style.

Attributes

fontName - String
Font Name: The name of the font for the paragraph.

fontSize - Measurement
Font Size: The font size for the text of the paragraph.

leading - Measurement

Leading: The height of a single paragraph line. It includes character height.

leftIndent - Measurement
Left Indentation: General indentation on the left side.

rightIndent - Measurement
Right Indentation: General indentation on the right side.

firstLineIndent - Measurement
First Line Indentation: The indentation of the first line in the paragraph.

spaceBefore - Measurement
Space Before: The vertical space before the paragraph.

spaceAfter - Measurement
Space After: The vertical space after the paragraph.

alignment - Choice of ('right', 'justify', 'center', 'centre', 'left')
Alignment: The text alignment.

bulletFontName - String
Bullet Font Name: The font in which the bullet character will be rendered.

bulletFontSize - Measurement
Bullet Font Size: The font size of the bullet character.

bulletIndent - Measurement
Bullet Indentation: The indentation that is kept for a bullet point.

textColor - Color
Text Color: The color in which the text will appear.

backColor - Color
Background Color: The background color of the paragraph.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

Content

XMLContent *(required)*
Text: The text that will be layed out.

hr

Create a horizontal line on the page.

Attributes

width - Measurement
Width: The width of the line on the page.

thickness - Measurement
Thickness: Line Thickness

color - Color
Color: The color of the line.

lineCap - Choice of ('default', 'square', 'round', 'butt')
Cap: The cap at the end of the line.

spaceBefore - Measurement
Space Before: The vertical space before the line.

spaceAfter - Measurement

Space After: The vertical space after the line.

align - Choice of ('decimal', 'right', 'center', 'centre', 'left')
Alignment: The alignment of the line within the frame.

valign - Choice of ('middle', 'top', 'bottom')
Vertical Alignment: The vertical alignment of the line.

dash - Sequence of Measurement
Dash-Pattern: The dash-pattern of a line.

illustration

Inserts an illustration with graphics elements.

Attributes

width *(required)* - Measurement
Width: The width of the illustration.

height *(required)* - Measurement
Height: The height of the illustration.

image

Draws an external image on the canvas.

Attributes

file *(required)* - Image
File: Reference to the external file of the iamge.

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the shape.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the shape.

width - Measurement
Width: The width of the image.

height - Measurement
Height: The height of the image.

showBoundary - Boolean
Show Boundary: A flag determining whether a border should be drawn around the image.

preserveAspectRatio - Boolean
Preserve Aspect Ratio: A flag determining whether the image's aspect ration should be conserved under any circumstances.

Examples

```
<image file="[z3c.rml.tests]/input/zope3logo.gif" x="2in" y="2in" width="0.5in"
       height="3in"/>
```

(Extracted from file *tag-image.rml*, line 20)

imageAndFlowables

An image with flowables around it.

Attributes

imageName *(required)* - Image
Image: The file that is used to extract the image data.

imageWidth - Measurement
Image Width: The width of the image.

imageHeight - Measurement
Image Height: The height the image.

imageMask - Color

Mask: The height the image.

imageLeftPadding - Measurement
Image Left Padding: The padding on the left side of the image.

imageRightPadding - Measurement
Image Right Padding: The padding on the right side of the image.

imageTopPadding - Measurement
Image Top Padding: The padding on the top of the image.

imageBottomPadding - Measurement
Image Bottom Padding: The padding on the bottom of the image.

imageSide - Choice of ('right', 'left')
Image Side: The side at which the image will be placed.

indent

Indent the contained flowables.

Attributes

left - Measurement
Left: The indentation to the left.

right - Measurement
Right: The indentation to the right.

initialize

Do some RML processing initialization.

Sub-Directives

name *(ZeroOrMore)*
alias *(ZeroOrMore)*

keepInFrame

Ask a flowable to stay within the frame.

Attributes

maxWidth - Measurement
Maximum Width: The maximum width the flowables are allotted.

maxHeight - Measurement
Maximum Height: The maximum height the flowables are allotted.

mergeSpace - Boolean
Merge Space: A flag to set whether the space should be merged.

onOverflow - Choice of ('overflow', 'shrink', 'truncate', 'error')
On Overflow: Defines what has to be done, if an overflow is detected.

id - Text
Name/Id: The name/id of the flowable.

frame - StringOrInt
Frame: The frame to which the flowable should be fitted.

label

A label for the chart.

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement
Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

x - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the label.

y - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the label.

Content

TextNode *(required)*
Text: The label text to be displayed.

label

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement
Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

Content

TextNode

label

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement
Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

row - Integer

col - Integer

format - String

dR - Float

Content

TextNode

label

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement
Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement

Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

Content

TextNode *(required)*
Text: The label text to be displayed.

labels

Sub-Directives

label *(OneOrMore)*

labels

A set of labels.

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement
Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

x - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the label.

y - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the label.

Sub-Directives

label *(ZeroOrMore)*

line

Attributes

strokeWidth - Measurement

strokeColor - Color

strokeDashArray - Sequence of Float

symbol - Symbol

name - Text

lineMode

Set the line mode for the following graphics elements.

Attributes

width - Measurement
Width: The line width.

dash - Sequence of Measurement
Dash-Pattern: The dash-pattern of a line.

miterLimit - Measurement
Miter Limit: The ???.

join - Choice of ('bevelled', 'mitered', 'round')
Join: The way lines are joined together.

cap - Choice of ('default', 'square', 'round', 'butt')
Cap: The cap is the desciption of how the line-endings look.

linePlot

Attributes

dx - Measurement
Drawing X-Position: The x-position of the entire drawing on the canvas.

dy - Measurement
Drawing Y-Position: The y-position of the entire drawing on the canvas.

dwidth - Measurement
Drawing Width: The width of the entire drawing

dheight - Measurement
Drawing Height: The height of the entire drawing

angle - Float
Angle: The orientation of the drawing as an angle in degrees.

x - Measurement
Chart X-Position: The x-position of the chart within the drawing.

y - Measurement
Chart Y-Position: The y-position of the chart within the drawing.

width - Measurement
Chart Width: The width of the chart.

height - Measurement
Chart Height: The height of the chart.

strokeColor - Color
Stroke Color: Color of the chart border.

strokeWidth - Measurement
Stroke Width: Width of the chart border.

fillColor - Color
Fill Color: Color of the chart interior.

debug - Boolean
Debugging: A flag that when set to True turns on debug messages.

reversePlotOrder - Boolean

lineLabelNudge - Measurement

lineLabelFormat - String

joinedLines - Boolean

Sub-Directives

data *(One)*
lines *(ZeroOrOne)*
xValueAxis *(ZeroOrOne)*
yValueAxis *(ZeroOrOne)*
lineLabels *(ZeroOrOne)*
texts *(ZeroOrOne)*

lineStyle

Define the border line style of each cell.

Attributes

start *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
Start Coordinates: The start table coordinates for the style instruction

stop *(required)* - Sequence of Combination of Integer, Choice of ('splitfirst', 'splitlast')
End Coordinates: The end table coordinates for the style instruction

kind *(required)* - Choice of ('box', 'outline', 'innergrid', 'linebefore', 'lineabove', 'linebelow', 'lineafter', 'grid')
Kind: The kind of line actions to be taken.

thickness *(required)* - Measurement
Thickness: Line Thickness

colorName *(required)* - Color
Color: The color of the border line.

cap *(required)* - Choice of ('default', 'square', 'round', 'butt')
Cap: The cap at the end of a border line.

dash - Sequence of Measurement
Dash-Pattern: The dash-pattern of a line.

join - Choice of ('bevelled', 'mitered', 'round')
Join: The way lines are joined together.

count - Integer
Count: Describes whether the line is a single (1) or double (2) line.

lines

Attributes

strokeWidth - Measurement

strokeColor - Color

strokeDashArray - Sequence of Float

symbol - Symbol

Sub-Directives

line *(OneOrMore)*

Examples

```
<lines>
  1in 7in 6in 7in
  1in 5in 6in 4in
  1in 3in 1in 1in
</lines>
```

(Extracted from file *tag-lines.rml*, line 9)

lines

A path of connected lines drawn on the canvas.

Content

TextNodeGrid with 4 cols of Measurement *(required)*
Line List: A list of lines coordinates to draw.

Examples

```
<lines>
  1in 7in 6in 7in
  1in 5in 6in 4in
  1in 3in 1in 1in
</lines>
```

(Extracted from file *tag-lines.rml*, line 9)

link

Place an internal link around a set of flowables.

Attributes

destination - Text
Destination: The name of the destination to link to.

url - Text
URL: The URL to link to.

boxStrokeWidth - Measurement

Box Stroke Width: The width of the box border line.

boxStrokeDashArray - Sequence of Float
Box Stroke Dash Array: The dash array of the box border line.

boxStrokeColor - Color
Box Stroke Color: The color in which the box border is drawn.

Examples

```
<link destination="PAGE_1" boxStrokeColor="red" boxStrokeWidth="1"
      boxStrokeDashArray="1 2">
  <para>Go to page 1 now!</para>
</link>
```

(Extracted from file *tag-bookmark.rml*, line 23)

```
<link destination="TITLE">
  <imageAndFlowables imageName="images/replogo.gif" imageSide="left">
    <para>Go to title.</para>
  </imageAndFlowables>
</link>
```

(Extracted from file *tag-bookmark.rml*, line 34)

```
<link url="http://www.reportlab.org">
  <para>Link to ReporLab Web Site.</para>
</link>
```

(Extracted from file *tag-bookmark.rml*, line 46)

moveto

Move the path cursor to the specified location.

Content

TextNodeSequence of Measurement *(required)*
Position: Position to which the path pointer is moved to.

name

Content

TextNode *(required)*
Text:

name

Defines a name for a string.

Attributes

id *(required)* - String
Id: The id under which the value will be known.

value *(required)* - Text
Value: The text that is displayed if the id is called.

nextFrame

Switch to the next frame.

Attributes

name - StringOrInt
Name: The name or index of the next frame.

nextPage

Switch to the next page.

option
An option in the select field.

Content

TextNode *(required)*
Value: The value of the option.

outlineAdd
Add a new entry to the outline of the PDF.

Attributes

key - String
Key: The unique key of the item.

level - Integer
Level: The level in the outline tree.

closed - Boolean
Closed: A flag to determine whether the sub-tree is closed by default.

Content

TextNode *(required)*
Title: The text displayed for this item.

pageDrawing
Draws directly on the content of one page's canvas. Every call of this directive creates a new page.

Sub-Directives

drawString *(ZeroOrMore)*
drawRightString *(ZeroOrMore)*
drawCenteredString *(ZeroOrMore)*
drawCentredString *(ZeroOrMore)*
drawAlignedString *(ZeroOrMore)*
ellipse *(ZeroOrMore)*
circle *(ZeroOrMore)*
rect *(ZeroOrMore)*
grid *(ZeroOrMore)*
lines *(ZeroOrMore)*
curves *(ZeroOrMore)*
image *(ZeroOrMore)*
place *(ZeroOrMore)*
textAnnotation *(ZeroOrMore)*
path *(ZeroOrMore)*
fill *(ZeroOrMore)*
stroke *(ZeroOrMore)*
setFont *(ZeroOrMore)*
scale *(ZeroOrMore)*
translate *(ZeroOrMore)*
rotate *(ZeroOrMore)*
skew *(ZeroOrMore)*
transform *(ZeroOrMore)*
lineMode *(ZeroOrMore)*
barCode *(ZeroOrMore)*
textField *(ZeroOrMore)*
buttonField *(ZeroOrMore)*
selectField *(ZeroOrMore)*
barChart *(ZeroOrMore)*
barChart3D *(ZeroOrMore)*
linePlot *(ZeroOrMore)*

pieChart *(ZeroOrMore)*
pieChart3D *(ZeroOrMore)*
spiderChart *(ZeroOrMore)*

pageGraphics
Define the page graphics for the page template.

pageInfo
Set's up page-global settings.

Attributes

pageSize *(required)* - PageSize
Page Size: The page size of all pages within this document.

pageTemplate
Define a page template.

Attributes

id *(required)* - Text
Id: The id of the template.

pagesize - PageSize
Page Size: The Page Size.

rotation - Integer
Rotation: The rotation of the page in multiples of 90 degrees.

Sub-Directives

frame *(OneOrMore)*
pageGraphics *(ZeroOrOne)*

para
Lays out an entire paragraph.

Attributes

fontName - String
Font Name: The name of the font for the paragraph.

fontSize - Measurement
Font Size: The font size for the text of the paragraph.

leading - Measurement
Leading: The height of a single paragraph line. It includes character height.

leftIndent - Measurement
Left Indentation: General indentation on the left side.

rightIndent - Measurement
Right Indentation: General indentation on the right side.

firstLineIndent - Measurement
First Line Indentation: The indentation of the first line in the paragraph.

spaceBefore - Measurement
Space Before: The vertical space before the paragraph.

spaceAfter - Measurement
Space After: The vertical space after the paragraph.

alignment - Choice of ('right', 'justify', 'center', 'centre', 'left')
Alignment: The text alignment.

bulletFontName - String
Bullet Font Name: The font in which the bullet character will be rendered.

bulletFontSize - Measurement
Bullet Font Size: The font size of the bullet character.

bulletIndent - Measurement
Bullet Indentation: The indentation that is kept for a bullet point.

textColor - Color
Text Color: The color in which the text will appear.

backColor - Color
Background Color: The background color of the paragraph.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

Content

XMLContent *(required)*
Text: The text that will be layed out.

paraStyle

Defines a paragraph style and gives it a name.

Attributes

fontName - String
Font Name: The name of the font for the paragraph.

fontSize - Measurement
Font Size: The font size for the text of the paragraph.

leading - Measurement
Leading: The height of a single paragraph line. It includes character height.

leftIndent - Measurement
Left Indentation: General indentation on the left side.

rightIndent - Measurement
Right Indentation: General indentation on the right side.

firstLineIndent - Measurement
First Line Indentation: The indentation of the first line in the paragraph.

spaceBefore - Measurement
Space Before: The vertical space before the paragraph.

spaceAfter - Measurement
Space After: The vertical space after the paragraph.

alignment - Choice of ('right', 'justify', 'center', 'centre', 'left')
Alignment: The text alignment.

bulletFontName - String
Bullet Font Name: The font in which the bullet character will be rendered.

bulletFontSize - Measurement
Bullet Font Size: The font size of the bullet character.

bulletIndent - Measurement
Bullet Indentation: The indentation that is kept for a bullet point.

textColor - Color
Text Color: The color in which the text will appear.

backColor - Color
Background Color: The background color of the paragraph.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

name *(required)* - String
Name: The name of the style.

alias - String
Alias: An alias under which the style will also be known as.

parent - Style
Parent: The apragraph style that will be used as a base for this one.

param

Sets one paramter for the text annotation.

Attributes

name *(required)* - String
Name: The name of the paramter.

Content

TextNode *(required)*
Value: The parameter value.

path

Create a line path.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the shape.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the shape.

fill - Boolean
Fill: A flag to specify whether the shape should be filled.

stroke - Boolean
Stroke: A flag to specify whether the shape's outline should be drawn.

close - Boolean
Close Path: A flag specifying whether the path should be closed.

Content

TextNodeGrid with 2 cols of Measurement *(required)*
Points: A list of coordinate points that define th path.

Sub-Directives

moveto *(ZeroOrMore)*
curvesto *(ZeroOrMore)*

Examples

```
<path x="2cm" y="15cm" fill="true">
  8cm 15cm
  <curvesto>
    10cm 15cm 10cm 12cm 8cm 12cm
  </curvesto>
  2cm 12cm
  <moveto>
    2cm 15cm
  </moveto>
```

`</path>`
(Extracted from file *tag-path.rml*, line 38)

pieChart

Attributes

dx - Measurement
Drawing X-Position: The x-position of the entire drawing on the canvas.

dy - Measurement
Drawing Y-Position: The y-position of the entire drawing on the canvas.

dwidth - Measurement
Drawing Width: The width of the entire drawing

dheight - Measurement
Drawing Height: The height of the entire drawing

angle - Float
Angle: The orientation of the drawing as an angle in degrees.

x - Measurement
Chart X-Position: The x-position of the chart within the drawing.

y - Measurement
Chart Y-Position: The y-position of the chart within the drawing.

width - Measurement
Chart Width: The width of the chart.

height - Measurement
Chart Height: The height of the chart.

strokeColor - Color
Stroke Color: Color of the chart border.

strokeWidth - Measurement
Stroke Width: Width of the chart border.

fillColor - Color
Fill Color: Color of the chart interior.

debug - Boolean
Debugging: A flag that when set to True turns on debug messages.

startAngle - Integer
Start Angle: The start angle in the chart of the first slice in degrees.

direction - Choice of ('clockwise', 'anticlockwise')

checkLabelOverlap - Boolean

pointerLabelMode - Choice of ('none', 'leftandright', 'leftright')

sameRadii - Boolean

orderMode - Choice of ('alternate', 'fixed')

xradius - Measurement

yradius - Measurement

Sub-Directives

data *(One)*
slices *(ZeroOrOne)*
labels *(ZeroOrOne)*
texts *(ZeroOrOne)*

pieChart3D

Attributes

dx - Measurement
Drawing X-Position: The x-position of the entire drawing on the canvas.

dy - Measurement
Drawing Y-Position: The y-position of the entire drawing on the canvas.

dwidth - Measurement
Drawing Width: The width of the entire drawing

dheight - Measurement
Drawing Height: The height of the entire drawing

angle - Float
Angle: The orientation of the drawing as an angle in degrees.

x - Measurement
Chart X-Position: The x-position of the chart within the drawing.

y - Measurement
Chart Y-Position: The y-position of the chart within the drawing.

width - Measurement
Chart Width: The width of the chart.

height - Measurement
Chart Height: The height of the chart.

strokeColor - Color
Stroke Color: Color of the chart border.

strokeWidth - Measurement
Stroke Width: Width of the chart border.

fillColor - Color
Fill Color: Color of the chart interior.

debug - Boolean
Debugging: A flag that when set to True turns on debug messages.

startAngle - Integer
Start Angle: The start angle in the chart of the first slice in degrees.

direction - Choice of ('clockwise', 'anticlockwise')

checkLabelOverlap - Boolean

pointerLabelMode - Choice of ('none', 'leftandright', 'leftright')

sameRadii - Boolean

orderMode - Choice of ('alternate', 'fixed')

xradius - Measurement

yradius - Measurement

perspective - Float

depth_3d - Measurement

angle_3d - Float

Sub-Directives

slices *(One)*
texts *(ZeroOrOne)*

place

Draws a set of flowables on the canvas within a given region.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the place.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the place.

width - Measurement
Width: The width of the place.

height - Measurement
Height: The height of the place.

Examples

```
<place x="1cm" y="10cm" width="13cm" height="4cm">
  <para>A paragraph within the place.</para>
</place>
```

(Extracted from file *tag-place.rml*, line 52)

plugInFlowable

Inserts a custom flowable developed in Python.

Attributes

module *(required)* - String
Module: The Python module in which the flowable is located.

function *(required)* - String
Function: The name of the factory function within the module that returns the custom flowable.

Content

TextNode
Parameters: A list of parameters encoded as a long string.

pointer

Attributes

strokeColor - Color

strokeWidth - Measurement

elbowLength - Measurement

edgePad - Measurement

piePad - Measurement

pre

A preformatted text, similar to the <pre> tag in HTML.

Attributes

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

Content

RawXMLContent *(required)*
Text: The text that will be layed out.

pto

A container for flowables decorated with trailer & header lists. If the split operation would be called then the trailer and header lists are injected before and after the split. This allows specialist "please turn over" and "continued from previous" like behaviours.

rect

Draws an ellipse on the canvas.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the shape.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the shape.

fill - Boolean
Fill: A flag to specify whether the shape should be filled.

stroke - Boolean
Stroke: A flag to specify whether the shape's outline should be drawn.

width *(required)* - Measurement
Width: The width of the rectangle.

height *(required)* - Measurement
Height: The height of the rectangle.

round - Measurement
Corner Radius: The radius of the rounded corners.

Examples

```
<rect x="8cm" y="20cm" width="5cm" height="3cm"/>
```

(Extracted from file *tag-rectange.rml*, line 9)

registerCidFont

Register a new CID font given the face name.

Attributes

faceName *(required)* - String
Face Name: The name of the face the font uses. The face has to be previously registered.

registerFont

Register a new font based on a face and encoding.

Attributes

name *(required)* - String
Name: The name under which the font can be used in style declarations or other parameters that lookup a font.

faceName *(required)* - String
Face Name: The name of the face the font uses. The face has to be previously registered.

encName *(required)* - String
Encoding Name: The name of the encdoing to be used.

registerTTFont

Register a new TrueType font given the TT file and face name.

Attributes

faceName *(required)* - String
Face Name: The name of the face the font uses. The face has to be previously registered.

fileName *(required)* - String
File Name: File path of the of the TrueType font.

registerType1Face

Register a new Type 1 font face.

Attributes

afmFile *(required)* - String
AFM File: Path to AFM file used to register the Type 1 face.

pfbFile *(required)* - String
PFB File: Path to PFB file used to register the Type 1 face.

rotate

Rotate the drawing counterclockwise.

Attributes

degrees *(required)* - Measurement
Angle: The angle in degrees.

scale

Scale the drawing using x and y scaling factors.

Attributes

sx *(required)* - Float
X-Scaling-Factor: The scaling factor applied on x-coordinates.

sy *(required)* - Float
Y-Scaling-Factor: The scaling factor applied on y-coordinates.

selectField

A selection field within the PDF

Attributes

title *(required)* - Text
Title: The title of the field.

x *(required)* - Measurement
X-Position: The x-position of the lower-left corner of the field.

y *(required)* - Measurement
Y-Position: The y-position of the lower-left corner of the field.

width *(required)* - Measurement
Width: The width of the select field.

height *(required)* - Measurement
Height: The height of the select field.

value - Text
Value: The default value of the field.

Sub-Directives

option *(ZeroOrMore)*

Examples

```
<selectField title="select1" value="Option 2" x="4cm" y="22.9cm" width="5cm"
             height="15">
  <option>Option 1</option>
  <option>Option 2</option>
  <option>Option 3</option>
</selectField>
```

(Extracted from file *tag-selectField.rml*, line 19)

series

A one-dimensional series.

Content

TextNodeSequence of Float *(required)*

Values: Numerical values representing the series' data.

series

A two-dimensional series.

Content

TextNodeGrid with 2 cols of Float *(required)*
Values: Numerical values representing the series' data.

setFont

Set the font name and/or size.

Attributes

name *(required)* - String
Font Name: The name of the font as it was registered.

size *(required)* - Measurement
Size: The font size.

leading - Measurement
Leading: The font leading.

setNextFrame

Define the next frame to switch to.

Attributes

name *(required)* - StringOrInt
Name: The name or index of the next frame.

setNextTemplate

Define the next page template to use.

Attributes

name *(required)* - StringOrInt
Name: The name or index of the next page template.

skew

Skew the drawing.

Attributes

alpha *(required)* - Measurement
Alpha: The amount to skew the drawing in the horizontal.

beta *(required)* - Measurement
Beta: The amount to skew the drawing in the vertical.

slice

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

popout - Measurement

fontName - String

fontSize - Measurement

labelRadius - Measurement

swatchMarker - Symbol

Sub-Directives

label *(ZeroOrOne)*
pointer *(ZeroOrOne)*

slice

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

popout - Measurement

fontName - String

fontSize - Measurement

labelRadius - Measurement

swatchMarker - Symbol

fillColorShaded - Color

slices

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

popout - Measurement

fontName - String

fontSize - Measurement

labelRadius - Measurement

fillColorShaded - Color

Sub-Directives

slice *(OneOrMore)*

slices

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

popout - Measurement

fontName - String

fontSize - Measurement

labelRadius - Measurement

Sub-Directives

slice *(OneOrMore)*

spacer

Creates a vertical space in the flow.

Attributes

width - Measurement
Width: The width of the spacer. Currently not implemented.

length *(required)* - Measurement
Length: The height of the spacer.

spiderChart

Attributes

dx - Measurement
Drawing X-Position: The x-position of the entire drawing on the canvas.

dy - Measurement
Drawing Y-Position: The y-position of the entire drawing on the canvas.

dwidth - Measurement
Drawing Width: The width of the entire drawing

dheight - Measurement
Drawing Height: The height of the entire drawing

angle - Float
Angle: The orientation of the drawing as an angle in degrees.

x - Measurement
Chart X-Position: The x-position of the chart within the drawing.

y - Measurement
Chart Y-Position: The y-position of the chart within the drawing.

width - Measurement
Chart Width: The width of the chart.

height - Measurement
Chart Height: The height of the chart.

strokeColor - Color
Stroke Color: Color of the chart border.

strokeWidth - Measurement
Stroke Width: Width of the chart border.

fillColor - Color
Fill Color: Color of the chart interior.

debug - Boolean
Debugging: A flag that when set to True turns on debug messages.

startAngle - Integer

direction - Choice of ('clockwise', 'anticlockwise')

Sub-Directives

data *(One)*
strands *(ZeroOrOne)*
strandLabels *(ZeroOrOne)*
spokes *(ZeroOrOne)*
spokeLabels *(ZeroOrOne)*
labels *(ZeroOrOne)*
texts *(ZeroOrOne)*

spoke

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

labelRadius - Measurement

visible - Measurement

spokeLabels

A set of spoke labels.

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement
Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

Sub-Directives

label *(OneOrMore)*

spokes

A collection of spokes.

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

labelRadius - Measurement

visible - Measurement

Sub-Directives

spoke *(OneOrMore)*

story

The story of the PDF file.

Attributes

firstPageTemplate - Text
First Page Template: The first page template to be used.

Sub-Directives

spacer *(ZeroOrMore)*
illustration *(ZeroOrMore)*
pre *(ZeroOrMore)*
xpre *(ZeroOrMore)*
plugInFlowable *(ZeroOrMore)*
barCodeFlowable *(ZeroOrMore)*
outlineAdd *(ZeroOrMore)*
title *(ZeroOrMore)*
h1 *(ZeroOrMore)*
h2 *(ZeroOrMore)*
h3 *(ZeroOrMore)*
para *(ZeroOrMore)*
blockTable *(ZeroOrMore)*
nextFrame *(ZeroOrMore)*
setNextFrame *(ZeroOrMore)*
nextPage *(ZeroOrMore)*
setNextTemplate *(ZeroOrMore)*
condPageBreak *(ZeroOrMore)*
keepInFrame *(ZeroOrMore)*

imageAndFlowables *(ZeroOrMore)*
pto *(ZeroOrMore)*
indent *(ZeroOrMore)*
fixedSize *(ZeroOrMore)*
bookmark *(ZeroOrMore)*
link *(ZeroOrMore)*
hr *(ZeroOrMore)*
name *(ZeroOrMore)*

strand

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

symbol - Symbol

symbolSize - Measurement

name - Text

strandLabels

A set of strand labels.

Attributes

dx - Measurement
Horizontal Extension: The width of the label.

dy - Measurement
Vertical Extension: The height of the label.

angle - Float
Angle: The angle to rotate the label.

boxAnchor - Choice of ('c', 'e', 'sw', 'ne', 'n', 's', 'w', 'autox', 'autoy', 'se', 'nw')
Box Anchor: The position relative to the label.

boxStrokeColor - Color
Box Stroke Color: The color of the box border line.

boxStrokeWidth - Measurement
Box Stroke Width: The width of the box border line.

boxFillColor - Color
Box Fill Color: The color in which the box is filled.

boxTarget - Text
Box Target: The box target.

fillColor - Color
Fill Color: The color in which the label is filled.

strokeColor - Color
Stroke Color: The color of the label.

strokeWidth - Measurement
Stroke Width: The width of the label line.

frontName - String
Font Name: The font used to print the value.

frontSize - Measurement
Font Size: The size of the value text.

leading - Measurement

Leading: The height of a single text line. It includes character height.

width - Measurement
Width: The width the label.

maxWidth - Measurement
Maximum Width: The maximum width the label.

height - Measurement
Height: The height the label.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

visible - Boolean
Visible: A flag making the label text visible.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

row - Integer

col - Integer

format - String

Content

TextNode

Sub-Directives

label *(OneOrMore)*

strands

A collection of strands.

Attributes

strokeWidth - Measurement

fillColor - Color

strokeColor - Color

strokeDashArray - Sequence of Float

symbol - Symbol

symbolSize - Measurement

name - Text

Sub-Directives

strand *(OneOrMore)*

stroke

Set the fill color.

Attributes

color *(required)* - Color
Color: The color value to be set.

stylesheet

A styleheet defines the styles that can be used in the document.

Sub-Directives

initialize *(ZeroOrOne)*
paraStyle *(ZeroOrMore)*
blockTableStyle *(ZeroOrMore)*

td

A table cell within a table.

Attributes

fontName - String
Font Name: The name of the font for the cell.

fontSize - Measurement
Font Size: The font size for the text of the cell.

leading - Measurement
Leading: The height of a single text line. It includes character height.

fontColor - Color
Font Color: The color in which the text will appear.

leftPadding - Measurement
Left Padding: The size of the padding on the left side.

rightPadding - Measurement
Right Padding: The size of the padding on the right side.

topPadding - Measurement
Top Padding: The size of the padding on the top.

bottomPadding - Measurement
Bottom Padding: The size of the padding on the bottom.

background - Color
Background Color: The color to use as the background for the cell.

align - Choice of ('decimal', 'right', 'center', 'centre', 'left')
Text Alignment: The text alignment within the cell.

vAlign - Choice of ('middle', 'top', 'bottom')
Vertical Alignment: The vertical alignment of the text within the cell.

lineBelowThickness - Measurement
Line Below Thickness: The thickness of the line below the cell.

lineBelowColor - Color
Line Below Color: The color of the line below the cell.

lineBelowCap - Choice of ('default', 'square', 'round', 'butt')
Line Below Cap: The cap at the end of the line below the cell.

lineBelowCount - Integer
Line Below Count: Describes whether the line below is a single (1) or double (2) line.

lineBelowSpace - Measurement
Line Below Space: The space of the line below the cell.

lineAboveThickness - Measurement
Line Above Thickness: The thickness of the line above the cell.

lineAboveColor - Color
Line Above Color: The color of the line above the cell.

lineAboveCap - Choice of ('default', 'square', 'round', 'butt')
Line Above Cap: The cap at the end of the line above the cell.

lineAboveCount - Integer

Line Above Count: Describes whether the line above is a single (1) or double (2) line.

lineAboveSpace - Measurement
Line Above Space: The space of the line above the cell.

lineLeftThickness - Measurement
Left Line Thickness: The thickness of the line left of the cell.

lineLeftColor - Color
Left Line Color: The color of the line left of the cell.

lineLeftCap - Choice of ('default', 'square', 'round', 'butt')
Line Left Cap: The cap at the end of the line left of the cell.

lineLeftCount - Integer
Line Left Count: Describes whether the left line is a single (1) or double (2) line.

lineLeftSpace - Measurement
Line Left Space: The space of the line left of the cell.

lineRightThickness - Measurement
Right Line Thickness: The thickness of the line right of the cell.

lineRightColor - Color
Right Line Color: The color of the line right of the cell.

lineRightCap - Choice of ('default', 'square', 'round', 'butt')
Line Right Cap: The cap at the end of the line right of the cell.

lineRightCount - Integer
Line Right Count: Describes whether the right line is a single (1) or double (2) line.

lineRightSpace - Measurement
Line Right Space: The space of the line right of the cell.

Content

RawXMLContent *(required)*
Content: The content of the cell; can be text or any flowable.

template

Define a page template.

Attributes

pagesize - PageSize
Page Size: The Page Size.

rotation - Integer
Rotation: The rotation of the page in multiples of 90 degrees.

leftMargin - Measurement
Left Margin: The left margin of the template.

rightMargin - Measurement
Right Margin: The right margin of the template.

topMargin - Measurement
Top Margin: The top margin of the template.

bottomMargin - Measurement
Bottom Margin: The bottom margin of the template.

showBoundary - Boolean
Show Boundary: A flag to show the boundary of the template.

allowSplitting - Boolean
Allow Splitting: A flag to allow splitting over multiple templates.

title - Text
Title: The title of the PDF document.

author - Text

Author: The author of the PDF document.

Sub-Directives

pageTemplate *(OneOrMore)*

text

Draw a text on the chart.

Attributes

x *(required)* - Measurement
X-Coordinate: The X-coordinate of the lower-left position of the text.

y *(required)* - Measurement
Y-Coordinate: The Y-coordinate of the lower-left position of the text.

angle - Float
Rotation Angle: The angle about which the text will be rotated.

fontName - String
Font Name: The name of the font.

fontSize - Measurement
Font Size: The font size for the text.

fillColor - Color
Fill Color: The color in which the text will appear.

textAnchor - Choice of ('start', 'boxauto', 'end', 'middle')
Text Anchor: The position in the text to which the coordinates refer.

Content

TextNode *(required)*
Text: The text to be printed.

textAnnotation

Writes a low-level text annotation into the PDF.

Attributes

contents *(required)* - FirstLevelTextNode
Contents: The PDF commands that are inserted as annotation.

Sub-Directives

param *(ZeroOrMore)*

Examples

```
<textAnnotation>
  <param name="Rect">0,0,1,1</param>
  <param name="F">3</param>
  <param name="escape">6</param>
X::PDF
PX(S)
MT(PINK)
</textAnnotation>
```

(Extracted from file *tag-textAnnotation.rml*, line 8)

textField

A text field within the PDF

Attributes

title *(required)* - Text
Title: The title of the field.

x *(required)* - Measurement
X-Position: The x-position of the lower-left corner of the field.

y *(required)* - Measurement
Y-Position: The y-position of the lower-left corner of the field.

width *(required)* - Measurement
Width: The width of the text field.

height *(required)* - Measurement
Height: The height of the text field.

value - Text
Value: The default text value of the field.

maxLength - Integer
Maximum Length: The maximum amount of characters allowed in the field.

multiline - Boolean
Multiline: A flag when set allows multiple lines within the field.

Examples

```
<textField title="input1" x="3.5cm" y="22.9cm" width="5cm" height="14"/>
```
(Extracted from file *tag-textField.rml*, line 19)

```
<textField title="input2" value="Default Value" x="3.5cm" y="18.4cm"
           width="5cm" height="3cm" multiline="yes" maxLength="30"/>
```
(Extracted from file *tag-textField.rml*, line 28)

texts
A set of texts drawn on the chart.

Sub-Directives

text *(ZeroOrMore)*

title
The title is a simple paragraph with a special title style.

Attributes

fontName - String
Font Name: The name of the font for the paragraph.

fontSize - Measurement
Font Size: The font size for the text of the paragraph.

leading - Measurement
Leading: The height of a single paragraph line. It includes character height.

leftIndent - Measurement
Left Indentation: General indentation on the left side.

rightIndent - Measurement
Right Indentation: General indentation on the right side.

firstLineIndent - Measurement
First Line Indentation: The indentation of the first line in the paragraph.

spaceBefore - Measurement
Space Before: The vertical space before the paragraph.

spaceAfter - Measurement
Space After: The vertical space after the paragraph.

alignment - Choice of ('right', 'justify', 'center', 'centre', 'left')
Alignment: The text alignment.

bulletFontName - String
Bullet Font Name: The font in which the bullet character will be rendered.

bulletFontSize - Measurement
Bullet Font Size: The font size of the bullet character.

bulletIndent - Measurement
Bullet Indentation: The indentation that is kept for a bullet point.

textColor - Color
Text Color: The color in which the text will appear.

backColor - Color
Background Color: The background color of the paragraph.

keepWithNext - Boolean
Keep with Next: When set, this paragraph will always be in the same frame as the following flowable.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

Content

XMLContent *(required)*
Text: The text that will be layed out.

tr

A table row in the block table.

Sub-Directives

td *(OneOrMore)*

transform

A full 2-D matrix transformation

Content

TextNodeSequence of Float *(required)*
Matrix: The transformation matrix.

translate

Translate the drawing coordinates by the specified x and y offset.

Attributes

dx *(required)* - Measurement
X-Offset: The amount to move the drawing to the right.

dy *(required)* - Measurement
Y-Offset: The amount to move the drawing upward.

valueAxis

Attributes

visible - Boolean

visibleAxis - Boolean

visibleTicks - Boolean

visibleLabels - Boolean

visibleGrid - Boolean

strokeWidth - Measurement

strokeColor - Color
strokeDashArray - Sequence of Float
gridStrokeWidth - Measurement
gridStrokeColor - Color
gridStrokeDashArray - Sequence of Float
gridStart - Measurement
gridEnd - Measurement
style - Choice of ('stacked', 'parallel', 'parallel_3d')
forceZero - Boolean
minimumTickSpacing - Measurement
maximumTicks - Integer
labelTextFormat - String
labelTextPostFormat - Text
labelTextScale - Float
valueMin - Float
valueMax - Float
valueStep - Float
valueSteps - Measurement
rangeRound - Text
zrangePref - Float

xValueAxis

Attributes

visible - Boolean
visibleAxis - Boolean
visibleTicks - Boolean
visibleLabels - Boolean
visibleGrid - Boolean
strokeWidth - Measurement
strokeColor - Color
strokeDashArray - Sequence of Float
gridStrokeWidth - Measurement
gridStrokeColor - Color
gridStrokeDashArray - Sequence of Float
gridStart - Measurement
gridEnd - Measurement
style - Choice of ('stacked', 'parallel', 'parallel_3d')
forceZero - Boolean
minimumTickSpacing - Measurement
maximumTicks - Integer
labelTextFormat - String
labelTextPostFormat - Text
labelTextScale - Float

valueMin - Float

valueMax - Float

valueStep - Float

valueSteps - Measurement

rangeRound - Text

zrangePref - Float

tickUp - Measurement

tickDown - Measurement

joinAxis - Boolean

joinAxisMode - Choice of ('top', 'points', 'none', 'value', 'bottom')

joinAxisPos - Measurement

xpre

A preformatted text that allows paragraph markup.

Attributes

style *(required)* - Style
Style: The paragraph style that is applied to the paragraph. See the ``paraStyle`` tag for creating a paragraph style.

bulletText - String
Bullet Character: The bullet character is the ASCII representation of the symbol making up the bullet in a listing.

dedent - Integer
Dedent: Number of characters to be removed in front of every line of the text.

Content

RawXMLContent *(required)*
Text: The text that will be layed out.

yValueAxis

Attributes

visible - Boolean

visibleAxis - Boolean

visibleTicks - Boolean

visibleLabels - Boolean

visibleGrid - Boolean

strokeWidth - Measurement

strokeColor - Color

strokeDashArray - Sequence of Float

gridStrokeWidth - Measurement

gridStrokeColor - Color

gridStrokeDashArray - Sequence of Float

gridStart - Measurement

gridEnd - Measurement

style - Choice of ('stacked', 'parallel', 'parallel_3d')

forceZero - Boolean

minimumTickSpacing - Measurement

maximumTicks - Integer

labelTextFormat - String
labelTextPostFormat - Text
labelTextScale - Float
valueMin - Float
valueMax - Float
valueStep - Float
valueSteps - Measurement
rangeRound - Text
zrangePref - Float
tickLeft - Measurement
tickRight - Measurement
joinAxis - Boolean
joinAxisMode - Choice of ('top', 'points', 'none', 'value', 'bottom')
joinAxisPos - Measurement