The

# *CSSToXSLFO*

User Guide

Version 2.2.0
Werner Donné
Pincette bv
14 March 2025

# TABLE OF CONTENTS

# 1 INTRODUCTION

CSSTOXSLFO is a tool which converts an XML document, combined with a CSS2 style sheet, into an XSL-FO file. It has some special provisions for XHTML, which is also an XML vocabulary. The tool implements a reasonable subset of CSS2. It also adds a few extensions for handling page-related issues properly. Note that the tool is not a general-purpose printing tool for any kind of HTML pages you can find on the Internet.

The goal of CSSTOXSLFO is to provide a rather easy interface to fine printing environments that use XSL-FO as their input. It is a compromise between the simplicity of style sheet expression and the quality of the result. XSL-FO is quite difficult. Writing style sheets that produce it are mostly written in XSLT, which is not straightforward to everyone either. CSS on the other hand is rather simple and yet it is powerful. In fact it combines element selection and formatting specification in one easy-to-learn syntax. The cost is that a lot of interesting XSL-FO features are not available.

An area where the tool can be a plus is the programmatic generation of reports within applications. The variety in style for reports is not that great. The offered feature set of CSSTOXSLFO can be sufficient. Having report programmers learn XSL-FO and XSLT is not always an option, while many know CSS and XHTML well enough to be productive with it.

Another use-case for CSSTOXSLFO is writing documents in XML. One can put work in a style sheet once and reuse that through the mark-up paradigm, in which content and formatting are separated. The formatting features should be sufficient to produce day-to-day documents in a typical business environment. Such documents don't tend to be very complicated, with respect to layout that is.

# IN PRACTICE 2

## 2.1 SPECIFYING STYLE SHEETS

The most general way of specifying a style sheet for a document with CSSTOXSLFO is the proposal in section 2.2 of [CSS2]. It consists of a processing instruction, which precedes the document, looking like this:

```
<?xml-stylesheet type="text/css" href="style.css"?>
```

For XHTML there are a few additional options. You can use the `link` element to link a style sheet (only persistent style sheets) to the document or you can embed it with the `style` element. The `style` attribute is also available as specified in [XHTML].

## 2.2 RUNNING IT

There are five packages you can run from the command-line: one that produces plain XSL-FO, one that returns the output of XEP, a product from RenderX (http://www.renderx.com), another that returns the output of XSLFormatter, a product from Antenna House (http://www.antennahouse.com), yet another that returns the output of Xinc, a product from Lunasil LTD (http://www.lunasil.com) and finally, one that runs FOP (http://xml.apache.org/fop/).

You need JDK1.8 or higher to run the packages. The command-lines look as follows for plain CSSTOXSLFO:

```
> java -jar css2xslfo.jar url_or_filename <options>
```

For XEP4:

```
> java -Dcom.renderx.xep.CONFIG=<XEP location>/xep.xml
    -jar <XEP location>/lib/css2xep.jar url_or_filename
    <options>
```

For XSLFormatter:

```
> set dynamic library path to <XSLFormatter location>/lib
> set environment variable AH_FONT_CONFIGFILE to
    <XSLFormatter location>/etc/font-config.xml
> java -jar <XSLFormatter location>/lib/css2xsl.jar
    url_or_filename <options>
```

For Xinc:

```
> java -jar css2xinc.jar url_or_filename <options>
```

For FOP:

```
> java -jar css2fop.jar url_or_filename <options>
```

Additional system properties and/or enviroment variables can be set. Please consult the product-specific documentation for this.

For css2xsl.jar to work, you should place it in <XSLFormatter location>/lib. The css2fop.jar file needs to be next to fop.jar, which should be next to the packages is uses. Therefore you should copy fop.jar from the FOP build directory to its lib directory. The css2xinc.jar should be in the XINC lib directory.

CSSTOXSLFO uses the XSLT-processor that comes with the JDK, which is Xalan from Apache. For better performance you can prepend Saxon 9+ to your boot classpath as follows (assuming /usr/local as the installation directory of Saxon):

```
> java -Xbootclasspath/p:/usr/local/saxon/saxon9.jar
    -jar css2xslfo.jar url_or_filename <options>
```

For XEP there is a special note. You have to specify another XSLT processor, because XEP uses Saxon 6.5.x, with which it doesn't work. You can either prepend another XSLT processor to the boot classpath or you can simply copy saxon9.jar in the XEP lib directory.

### *2.2.1 Common Options*

The following options are common to all six variants. The document to be processed can be specified with a URL or filename. If it is omitted, stdin will be read.

-baseurl <URL>
: Change the base URL of the input document. By default it is the URL of the document itself.

-c <URL or filename>
: Specify a catalog in the format defined by SGML Open Technical Resolution TR9401:1997. Only the "PUBLIC" and "SYSTEM" keywords are supported.

-h
: Display the command-line syntax.

-p <comma-separated list of URLs or filenames>
: A list of pre-processing XSLT style sheets that are executed on the input document, in the specified order, before anything else.

-uacss <URL or filename>
: Use another User Agent style sheet than the one built-in.

-v
: Turn on XML validation of the input document.

-screen
: Turn on screen mode. The @screen rules will be evaluated instead of the @print rules.

-data <URL or filename>
: An XML data file that will be merged into the document prior to conversion. See chapter  for more information.

parameter=value
: Specify User Agent parameters. Equivalent CSS constructs precede these.

### *2.2.2 Options Specific To css2xslfo.jar*

-debug
: Produces a number of intermediary files representing the different processing steps.

-debug-filters <comma-separated list of classnames>
: With this option the intermediary files of only one processing step can be obtained. The classname should correspond to the name of a filter class.

-fo <filename>
: The XSL-FO output file. If it is omitted stdout will be written instead.

### *2.2.3 Options Specific To css2xep.jar*

One the following options should be specified.

-pdf <filename>
: The PDF output file. Either this option or the "-ps" options should be present.

-ps <filename>
: The PostScript output file. Either this option or the "-pdf" options should be present.

-config <URL or filename>
: The XEP configuration file.

-q
: Makes XEP silent.

### *2.2.4 Options Specific To css2xsl.jar*

-pdf <filename>
: The PDF output file. This option is mandatory.

-config <URL or filename>
: The XSLFormatter configuration file.

### *2.2.5 Options Specific To css2fop.jar*

-fop <options>
: The rest of the command-line will be parsed by FOP. Specifying an input file here doesn't work. The "-q" option will only work if you configure the Apache logger environment. The FOP command-line logger can be set by assigning the value `org.apache.fop.util.CommandLineLogger` to the system property `org.apache.commons.logging.Log`. This option is mandatory.

### *2.2.6 Options Specific To css2xinc.jar*

One the following options should be specified.

-pdf <filename>
: The PDF output file. This option is mandatory.

-config <URL or filename>
: The XINC configuration file.

### *2.2.7 User Agent Parameters*

The User Agent parameters are common to all three packages. They have no effect if there are @page rules in the style sheet, except for the "rule-thickness" parameter. Furthermore, equivalent CSS constructs, when present in the style sheet, always precede.

column-count
: The number of columns on a page. The default is "1".

country
: The country code. The default is "GB".

font-size
: The point size of the font. The default for paper sizes "a5" and "b5" is "10pt". For all other paper sizes the default is "11pt". See also the "paper-size" parameter.

html-header-mark
: An HTML element can be passed here. Its contents will be used as the running header. By default there is no mark.

language
: The language code. The default is "en".

odd-even-hift
: The amount by which the page contents is shifted in the inline progression direction when the paper mode is "twosided". The default is "10mm". See also the "paper-mode" parameter.

orientation
: The allowed values are "portrait", which is the default, and "landscape".

paper-margin-bottom
: The bottom margin of a page. The default is "0mm".

paper-margin-left
: The left margin of a page. The default is "25mm".

paper-margin-right
: The right margin of a page. The default is "25mm".

paper-margin-top
: The top margin of a page. The default is "10mm".

paper-mode
: The allowed values are "onesided", which is the default, and "twosided".

paper-size
: The allowed values are "a4", which is the default, "a0", "a1", "a2", "a3", "a5", "b5", "executive", "letter" and "legal".

rule-thickness
: The default thickness for rules when there was no CSS specification for it. The default is "0.2pt".

writing-mode
: The XSL-FO writing mode. The default is "lr-tb". Other possible values are "rl-tb", "tb-rl", "lr", "rl" and "tb". See also [XSL-FO].

## 2.3 BUILDING CSSTOXSLFO

The tool comes with an ANT file. The default target only builds the css2xslfo.jar file. Then there are also the `xep`, `xsl`, `xinc` and `fop` targets, which produce css2xep.jar, css2xsl.jar, css2xinc.jar and css2fop.jar respectively, with a version number attached to it.

## 2.4 ANT TASKS

In each package there is a corresponding Ant task. The following table gives the respective class names.

| *Package* | *Ant task class name* |
|---|---|
| css2xslfo.jar | be.re.css.ant.CSSToXSLFO |
| css2xep.jar | be.re.css.ant.CSSToXEP |
| css2xsl.jar | be.re.css.ant.CSSToXSLFormatter |
| css2fop.jar | be.re.css.ant.CSSToFOP |
| css2xinc.jar | be.re.css.ant.CSSToXinc |

### *2.4.1 Common Options*

All tasks support the nested elements `parameter` and `preprocessor`. With the former the parameters described in "User Agent Parameter" can be specified through its `name` and `value` attributes. With the latter a sequence of pre-processing XSLT style sheets can be provided. Its `stylesheet` attribute should be set to a filename or URL. The following attributes are common to all tasks.

| *Attribute* | *Description* | *Required* |
|---|---|---|
| baseurl | Change the base URL of the input document. By default it is the URL of the document itself. | No |
| catalog | Specify a catalog in the format defined by SGML Open Technical Resolution TR9401:1997. Only the "PUBLIC" and "SYSTEM" keywords are supported. | No |

| *Attribute* | *Description* | *Required* |
|---|---|---|
| input | The input document as a URL or filename. | Yes |
| output | The output document as a filename. The format is derived from its extension. | Yes |
| useragentstylesheet | Use another User Agent style sheet than the one built-in. | No |
| validate | Turn on XML validation of the input document. Defaults to `false`. | No |

### *2.4.2 Options Specific To css2xep.jar*

| *Attribute* | *Description* | *Required* |
|---|---|---|
| config | The XEP configuration file. It may be a filename or a URL. | No |
| quiet | Makes XEP silent. | No |

### *2.4.3 Options Specific To css2fop.jar*

| *Attribute* | *Description* | *Required* |
|---|---|---|
| config | A user configuration file. | No |
| quiet | Makes FOP silent. | No |

### *2.4.4 Options Specific To css2fopnew.jar*

| *Attribute* | *Description* | *Required* |
|---|---|---|
| config | A user configuration file. | No |

# COMPLIANCE WITH CSS2 3

## 3.1 SPECIFICATIONS

| *Section* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| 4.1 Syntax | yes | Thanks to Flute. |
| 4.2 Rules for handling parsing errors | partial | Unknown properties will end up in the XSL-FO file and cause errors in a XSL-FO processor. |
| 4.3 Values | yes | Thanks to Flute. |
| 4.4 CSS document representation | yes | Thanks to Flute. |
| 5 Selectors | partial | All sections but 5.11.2 and 5.11.3. The `:first-letter` pseudo element is implemented with the restriction that letter combinations, which are considered as one letter, are not examined. As a workaround you can use the ligature Unicode characters instead. The `vertical-align` is also valid when `float` is `none`. In that case it applies to the inline material which is affected by the pseudo element. |
| 6 Assigning property values, Cascading and Inheritance | yes | |
| 7 Media types | yes | By design, only types `all` and `print` are supported. |
| 8 Box model | yes | |
| 9.1.1 The viewport | no | |
| 9.1.2 Containing blocks | yes | |
| 9.2.1 Block-level elements and block boxes | partial | Compact and run-in boxes are not supported. |

| *Section* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| 9.2.2 Inline-level elements and inline boxes | partial | Compact and run-in boxes and inline tables are not supported. |
| 9.2.3 Compact boxes | no | |
| 9.2.4 Run-in boxes | no | |
| 9.2.5 The 'display' property | partial | See property table. |
| 9.3 Positioning schemes | yes | |
| 9.4 Normal flow | yes | |
| 9.5 Floats | yes | |
| 9.6 Absolute positioning | yes | |
| 9.7 Relationships between 'display', 'position', and 'float' | yes | |
| 9.9 Layered presentation | yes | |
| 9.10 Text direction: the 'direction' and 'unicode-bibi' properties | yes | |
| 10 Visual formatting model details | partial | See the property table for the height property. |
| 11 Visual effects | yes | |
| 12.1 The :before and :after pseudo-elements | yes | |
| 12.2 The 'content' property | yes | |
| 12.3 Interaction of :before and :after with 'compact' and 'run-in' elements | no | |
| 12.4 Quotation marks | yes | |
| 12.5 Automatic counters and numbering | yes | |
| 12.6.1 Markers: the 'marker-offset' property | yes | |
| 12.6.2 Lists: the 'list-style-type', 'list-style-image', 'list-style-position', and 'list-style' properties | yes | |

| *Section* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| 13.2.1 Page margins | yes | |
| 13.2.2 Page size: the 'size' property | yes | |
| 13.2.3 Crop marks: the 'marks' property | no | |
| 13.2.4 Left, right, and first pages | yes | |
| 13.2.5 Content outside the page box | yes | |
| 13.3 Page breaks | partial | Named pages are only supported for block-level and table elements, which are not inside of a table and have an ancestor with the `region` property set to `body`. |
| 13.4 Cascading in the page context | yes | |
| 14 Colors and Backgrounds | yes | |
| 15 Fonts | partial | @font-face and descriptors are not supported. |
| 16 Text | yes | |
| 17 Tables | partial | Inline tables are not supported. Anonymous table objects are only supported for missing table groups and missing table cells in a row, on the condition that there are table column elements. Audio rendering is not supported. |
| 18 User interface | no | |
| 19 Aural style sheets | no | |

## 3.2 PROPERTIES

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| azimuth | no | |
| background | yes | |

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| background-attachment | yes | |
| background-color | yes | |
| background-image | yes | |
| background-position | yes | |
| background-repeat | yes | |
| border | yes | |
| border-bottom | yes | |
| border-bottom-color | yes | |
| border-bottom-style | yes | |
| border-bottom-width | yes | |
| border-collapse | partial | Not for `inline-table`. |
| border-color | yes | |
| border-left | yes | |
| border-left-color | yes | |
| border-left-style | yes | |
| border-left-width | yes | |
| border-right | yes | |
| border-right-color | yes | |
| border-right-style | yes | |
| border-right-width | yes | |
| border-spacing | partial | Not for `inline-table`. |
| border-style | yes | |
| border-top | yes | |
| border-top-color | yes | |
| border-top-style | yes | |
| border-top-width | yes | |

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| borded-width | yes | |
| bottom | yes | |
| caption-side | yes | |
| clear | yes | |
| clip | yes | |
| color | yes | |
| content | yes | |
| counter-increment | yes | |
| counter-reset | yes | |
| cue | no | |
| cue-after | no | |
| cue-before | no | |
| cursor | no | |
| direction | yes | |
| display | partial | The values `run-in`, `compact` and `inline-table` are not supported. The `marker` value is supported with the limiation that the value `auto` for the `width` property is not. Markers also don't work with floats. |
| elevation | no | |
| empty-cells | yes | |
| float | yes | |
| fonts | yes | |
| font-family | yes | |
| font-size | yes | |
| font-size-adjust | yes | |
| font-stretch | yes | |

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| font-style | yes | |
| font-variant | yes | |
| font-weight | yes | |
| height | partial | A percentage value for the height of a block, which is in another block with an explicit height, will be treated as `auto`. This stems from the fact that in this case a block has to be split in a fo:block-container and a nested fo:block, because there are properties that don't apply to both of them. The inner original block will therefore have a parent without an explicit height specification. The latter has moved to the surrounding fo:block-container. |
| left | yes | |
| letter-spacing | yes | |
| line-height | yes | |
| list-style | partial | See individual properties. |
| list-style-image | no | |
| list-style-position | partial | A list should be uniform. Specifying different values for different list items will produce undesired results. |
| list-style-type | partial | The styles in section 9.1 and 9.2 of [CSS3L] are supported, together with the values `lower-roman`, `upper-roman`, `lower-alpha`, `lower-latin`, `upper-alpha`, `upper-latin` and `none`. (The value `hyphen` that was defined in Working Draft 7 is also retained.)<br>A list should be uniform. Specifying different values for |

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| | | different list items will produce undesired results. |
| margin | yes | |
| margin-bottom | yes | |
| margin-left | yes | |
| margin-right | yes | |
| margin-top | yes | |
| marker-offset | yes | |
| marks | no | |
| max-height | yes | |
| max-width | yes | |
| min-height | yes | |
| min-width | yes | |
| orphans | yes | |
| outline | no | |
| outline-color | no | |
| outline-style | no | |
| outline-width | no | |
| overflow | yes | |
| padding | yes | |
| padding-bottom | yes | |
| padding-left | yes | |
| padding-right | yes | |
| padding-top | yes | |
| page | partial | Only for block-level and table elements, which are not inside of a table and have an ancestor with |

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| | | the `region` property set to `body`. |
| page-break-after | yes | |
| page-break-before | yes | |
| page-break-inside | yes | |
| pause | no | |
| pause-after | no | |
| pause-before | no | |
| pitch | no | |
| play-during | no | |
| play-range | no | |
| position | yes | |
| quotes | yes | |
| richness | no | |
| right | yes | |
| size | yes | |
| speak | no | |
| speak-header | no | |
| speak-numeral | no | |
| speak-punctuation | no | |
| speech-rate | no | |
| stress | no | |
| table-layout | yes | |
| text-align | yes | |
| text-decoration | yes | |
| text-indent | yes | |
| text-transform | yes | |

| *Property* | *Implemented* | *Remarks and restrictions* |
|---|---|---|
| top | yes | |
| unicode-bibi | yes | |
| vertical-align | yes | |
| visibility | yes | |
| voice-family | no | |
| volume | no | |
| white-space | yes | |
| widows | yes | |
| width | yes | |
| word-spacing | yes | |
| z-index | yes | |

# EXTENSIONS 4

The extension features of the tool mostly pertain to page-oriented aspects. Care has been taken to not introduce new syntax. There are, however, a number of new properties. Those are normally safely ignored by browsers. In the case where there would be an impact on the layout produced by browsers, the properties can be confined to the "print" medium through @media rules.

## 4.1 PAGE REGIONS

The functionality described in this section is an alternative for margin boxes as described in [CSS3P]. There is one exception, however. All property declarations in margin boxes should end with a semi-colon, because it was not possible to write an LL-grammar for the specified productions. The alternative functionality is useful for cases where the margin content has more structure than text.

This extension introduces XSL-FO-compatible page regions. Regions can be defined by placing a `region` property on an element. The allowed values are `bottom`, `left`, `right`, `top` and `body`. At least one element with the `region` property set to `body` should be present in the document.[1] Page sequences are only generated for the content of such an element. The regions other than the body region must be the first direct children of the body region. Otherwise they are ignored. In the case of XHTML, for example, this means that they should come at the beginning of the `body` element.

On top of that, either the `width` property, for left and right regions, or the `height` property, for top and bottom regions, should be defined. They will determine the dimensions of the page regions. The default value for `width` is "20mm". For `height` it is "10mm".

The extension property `precedence` is also available for the top and bottom regions. Its value can be `true` or `false`, the latter being the initial value. The property says whether the width of the top or bottom region is equal to that of the page reference area or if they give way to the left and right regions.

The regions work together with the @page rules, of which there should be at least one. It is possible to specify different regions, which correspond to the different page types in the style sheet. This can be achieved by also specifying the `page` property, which is a standard CSS2 property. Consider the following example:

```
div.bottom-left, div.bottom-right { display: none; }

@media print
{
  div.bottom-left
```

1 The XHTML User Agent style sheet sets this property to the `body` element.

```
    {
      height: 15mm;
      page: left;
      region: bottom;
      text-align: left;
    }

    div.bottom-right
    {
      height: 15mm;
      page: right;
      region: bottom;
      text-align: right;
    }

    span.page:before { content: counter(page); }
  }
```

This says that on left pages the bottom region is left-aligned, while on right pages it is right-aligned. The `span` element is used in the following region definitions:

```
<div class="bottom-left">
  <p> </p>
  <div><span class="page"/></div>
</div>

<div class="bottom-right">
  <p> </p>
  <div><span class="page"/></div>
</div>
```

The `page` property bears a kind of inheritance mechanism. For any page the regions with the most specific `page` property will be selected. A region without a `page` property is the least specific. A named page is more specific and the values `left` and `right` are yet more specific. After this comes the new pseudo page `blank`, which is for blank pages that are generated because of page positioning constraints such as left and right. The first page of a chapter, for example, is sometimes forced to be a right page. This can produce an extra blank page for the previous chapter. In fact, this maps directly to the XSL-FO blank pages. There are special values, which are even more specific, such as `first-right`, `blank-left`, `left-<page-name>`, etc. If, for example, there is no bottom region for `first-right`, but there is one for `first`, the latter will be selected if the first page happens to on the right. See section "*page*" for the precise precedence rules.

In order for the top, bottom, left and right region elements not to interfere with the normal flow it is best to set their display type to `none`.

## 4.2 PAGE NUMBERING

The two special counters `page` and `pages` have been added in this tool. The former is taken over from the CSS3 Paged Media Module (see also [CSS3P]). The `page` can be used just like any other counter, except that it is confined to the regions. The following example shows a document with a preface and a body. Each reset the page

count. The preface has a lower Roman numbering style, while the body uses the decimal style. If the body page didn't reset the counter, numbering would continue from the preface, but with a change of style.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
    <style type="text/css">
@page preface
{
  counter-reset: page;
  margin: 10%;

  @bottom-center
  {
    content: counter(page, lower-roman);
  }
}

@page body
{
  counter-reset: page;
  margin: 10%;

  @bottom-center
  {
    content: counter(page, decimal);
  }
}

div.preface { page: preface; }
div.body { page: body; }
    </style>
  </head>
  <body>
    <div class="preface">
      <p>Text.</p>
    </div>
    <div class="body">
      <p>Text.</p>
    </div>
  </body>
</html>
```

When switching between named pages you can control how the ending named page sequence should be terminated with the extension property `force-page-count`. For example, it some page sequence produces five pages, you can force the sequence to produce six pages by setting the property to `even`. An extra blank page will then be generated before starting the new page sequence. If you don't want such behaviour, you should set the property to `no-force`, since the initial value is `auto`.

## 4.3 PAGE REFERENCES

You sometimes want to write phrases like "The diagram on page 19 ...". The CSSTOXSLFO tool provides this functionality through the `page-ref` function, which can be used in the `content` property. Its only parameter is the name of an attribute that contains the ID of another element. The function call will be replaced with the number of the page that element is on.

In XHTML it is a bit more complicated to achieve the desired result, because there aren't many extension attributes available for it. The following fragment shows how it can be done:

```
<img id="img1" src="file:///t.png"/>
...
<span class="page-ref"><span class="img1"/></span>
```

The accompanying style sheet rule would then be:

```
span.page-ref > span:before { content: page-ref(class); }
```

## 4.4 LEADERS

It is possible to use XSL-FO leaders through the display type `leader`. The properties defined in section 7.21 of [XSL-FO] ("Leader and Rule Properties") can be used in a CSS style sheet, with the exception that the `leader-length` property cannot have a length range as a value. If you want to create table of contents lines or something similar, you also need the XSL-FO property `text-align-last`, described in section 7.15.10 of [XSL-FO]. The following example shows how a table of contents line could be made in XHTML.

```
<div class="toc">
  <a href="#chapter1">Title of Chapter 1</a>
  <span class="leader"/>
  <span class="page-ref"><span class="chapter1"/></span>
</div>
```

The piece of style sheet that goes with it is:

```
div.toc
{
  text-align-last: justify;
}

span.leader
{
  display: leader;
  leader-pattern: dots;
  leader-pattern-width: 5pt;
}

span.page-ref > span:before
{
```

```
  content: page-ref(class);
}
```

The `leader()` function, as described in [CSS3G], is also supported. It can be used in the `content` property. The optional second argument is not supported. This is an example:

```
div.toc > a:after
{
  content: leader(dotted);
}
```

## 4.5 NAMED STRINGS

Named strings, as described in [CSS3G], are supported in CSSTOXSLFO. This consists of the `string-set` property, with which contents can be captured, and the `string()` function. The latter can occur in the value of the `content` property. The `string-set` property accepts values which are similar to those of the `content` property. There are the additional keywords `contents`, `content-element`, `content-before`, `content-after` and `content-first-letter`, which are replaced with a variant of the string value of the element carrying the `string-set` property.

The following is a simple XHTML example of how you can create a running header that refers to the current chapter.

```
<body>
  ...
  <h1>Chapter Title</h1>
  ...
</body>
```

Here is the bit of style sheet that does it:

```
@page
{
  @top-center
  {
    content: string(mark);
  }
}

h1
{
  string-set: mark contents;
}
```

Note that the optional second argument to the `string()` function is not supported.

## 4.6 RUNNING ELEMENTS

CSSTOXSLFO supports running elements, as described in [CSS3G]. They work like named strings except that a complete element with styling included is saved instead

of only the string in it. Op top of that the element is taken out of the normal document flow. That is why the `position` property can now have a call to the `running()` function as its value. Here is an example:

```
<body>
  ...
  <div class="mytitle">Chapter Title</div>
  ...
</body>
```

Here is the bit of style sheet that does it:

```
@page
{
  @top-center
  {
    content: element(title);
  }
}

div.mytitle
{
  position: running(title);
}
```

Note that the optional second argument to the `running()` function is not supported.

## 4.7 HYPHENATION

Text can be hyphenated through the `hyphenate` property, which is inherited. The possible values are `true` and `false`. Hyphenation is turned off by default.

## 4.8 FOOTNOTES

It is possible to produce footnotes using the `footnote-reference` and `footnote-body` display types. The former is displayed in the flow, while the latter goes to the footnote area at the bottom of the page. When a footnote body occurs it must be either immediately preceded by a footnote reference or have a `:before` pseudo element with the display type `footnote-reference`. Otherwise it is treated as if the display were `none`. Whitespace between a footnote reference and body is gobbled. A footnote reference can also occur on its own.

The contents of both the footnote reference and body is free. Both display types exist to give you complete control over the contents and style. Usually some footnote counter is used, as shown in the example below. There is an extra counter style `footnote`, which produces symbols, such as an asterix, dagger, etc.

```
h1 { counter-reset: footnote; }

span.footnote-body
{
  display: footnote-body;
  font-size: 0.83em;
```

```
}

span.footnote-body:before
{
  content: counter(footnote);
  padding-right: 1em;
}

span.footnote-reference
{
  display: footnote-reference;
}

span.footnote-reference:before
{
  counter-increment: footnote;
  content: counter(footnote);
  font-size: 0.83em;
  vertical-align: super;
}
```

In the document a footnote would then look like this:

```
<p>Paragraph text.<span class="footnote-reference"/><span
class="footnote-body">Footnote text.</span></p>
```

You might find it cumbersome to have to place a footnote reference in front of every footnote body. It can be avoided, at the expense of formatting control however. You can define a `:before` pseudo element for the footnote body and give it the display type `footnote-reference`. Whatever contents it generates will then be used for the reference in the flow, as well as in the footnote body at the bottom of the page. As a consequence, the style is constrained by the fact that it must be decent for both contexts. The style sheet becomes a bit simpler:

```
h1 { counter-reset: footnote; }

span.footnote-body
{
  display: footnote-body;
  font-size: 0.83em;
}

span.footnote-body:before
{
  counter-increment: footnote;
  content: counter(footnote);
  display: footnote-reference;
  font-size: 0.83em;
  vertical-align: super;
}
```

If you want full control over the formatting in both contexts and at the same time want to omit the footnote reference elements in the document, the solution is to pre-process the document. The transformation is rather trivial.

## 4.9 ORIENTATION

You can rotate text with the `orientation` property. This works only for block, table and table cell elements. The possible values are 0, 90, 180, 270, -90, -180, -270. They represent the degrees in the counter-clockwise direction. The initial value is 0.

## 4.10 LIST STYLE TYPES

The glyphs for the `list-style-type` property, as defined in sections 9.1 and 9.2 of [CSS3L], are implemented.

## 4.11 MULTICOLUMN

With the properties `column-count`, which must be strictly positive, and `column-gap`, which is a length, a multi-column layout can be specified for a page. Both properties are allowed in an @page rule. As a consequence, if you want to switch between column modes, you have to switch pages as well.

With the `column-span` property a blocks and tables, that are not themselves inside of another table, can be made to span all the columns of a multi-column page. The allowed values for the property are `all` and `none`.

## 4.12 CHANGE BARS

The change bar properties introduced in [XSL-FO11] are available for `:before` and `:after` pseudo elements. For the latter, only the `change-bar-class` property is relevant. The following is a simple example:

```
p.changed:before
{
  change-bar-class: changed;
  change-bar-style: solid; /* initial value is none */
  change-bar-width: 0.2pt;
}

p.changed:after { change-bar-class: changed; }
```

## 4.13 LINKS

The `link` property can have the name of an attribute as its value. The value of that attribute will be used for the generated link, as the target URL or the internal target ID, if it is an IDREF attribute, which distinguishes it from a relative URL. Likewise, the value of the `anchor` property can be the name of an attribute, the value of which will become the target ID. This way an internal link destination can be created. For example:

```
a[href] { link: href; }
a[name] { anchor: name; }
```

## 4.14 GRAPHICS

An external graphic can be included in a document through the display type `graphic`, which is an inline level display type. The elements marked with it are "replaced elements". As a consequence, the properties `height` and `width` apply. The XSL-FO properties `content-height`, `content-width`, `content-type`, `scaling` and `scaling-method` are also supported. Their definition is in [XSL-FO]. The property `src` is interpreted differently. Its value should be the name of an attribute that has a URI for a value. For the XHTML element `img`, for example, the User Agent style sheet contains the following:

```
img
{
  content-height: scale-to-fit;
  content-width: scale-to-fit;
  display: graphic;
  scaling: uniform;
  src: src;
}
```

## 4.15 COLUMN AND ROW SPANNING

In XHTML one can specify column and row spanning with the `colspan` and `rowspan` attributes on the `td` and `th` elements. It is, however, also possible to apply CSS to other XML vocabularies. Hence, there should be an equivalent feature in CSS to express this. The extension properties `colspan` and `rowspan` serve that purpose. They can be used for elements with the display type `table-cell`.

## 4.16 PROPORTIONAL COLUMN WIDTHS

Again in XHTML it is possible to say that a column should occupy a relative portion of the total table width. It is done by setting the `width` attribute to a number, followed by an asterix. If we have, for example, three columns with the widths "1*", "2*" and "3*", they occupy 1, 2 and 3 sixth of the table width respectively. This is not part of the HTML specification, but it is a widely supported feature.

In order to provide it for other XML vocabularies then XHTML, the unit `pcw`, which stands for "proportional column width", is available for the `width` property of an element with the display type `table-column`.

## 4.17 REPEATING TABLE HEADERS AND FOOTERS

By default table headers and footers are repeated when a table spans several pages. You can suppress this by setting the `table-omit-header-at-break` and `table-omit-footer-at-break` properties to `true` respectively.

## 4.18 CSS3 NAMESPACES

Namespaces for selectors, as defined in [CSS3S], are implemented. This means you can use namespace prefixes in element selectors and attribute conditions. The prefixes are separated from the local name with a pipe sign ("|").

The namespaces are declared with the `@namespace` rule, which should always come right after the `@import` rules if there are any. In the following example the XHTML namespace has been declared as the default namespace. Next to that, the DeltaXML namespace is declared with the prefix "dx". You also see the use of the "attr" function with an attribute that has a prefix.

```
@namespace url(http://www.w3.org/1999/xhtml);
@namespace dx
  url(http://www.deltaxml.com/ns/well-formed-delta-v1);

*[dx|delta=add], dx|new
{
  text-decoration: underline;
}

*[dx|delta=delete], dx|old
{
  text-decoration: line-through;
}

p[dx|delta]:before
{
  content: attr(dx|delta);
  display: marker;
  marker-offset: 0.5em;
  text-align: right;
}
```

## 4.19 WRAPPERS

When processing XML in general you might encounter elements which represent pure structure, i.e. they are not directly related to layout. For such elements there shouldn't be any formatting objects in the output. Normally you would have to pre-process the document in order to get rid of them in the proper way.

The display type `wrapper` is introduced to cope with common cases. When an element has this display type, it will not contribute any formatting objects. However, its inherited properties will be passed on to its child elements, according to the property inheritance rules.

With respect to XML processing, a wrapper seems to be "transparant". Note however that, while a wrapper can occur anywhere, it influences CSS selector matching. For instance, it will interfere with "direct sibling" and "direct child" selectors.

## 4.20 FOREIGN ELEMENTS

With the display type `foreign` it is possible to transfer part of a document unmodified to an `fo:instream-foreign-object` element. This may be useful for

elements that are in another namespace than that of the document itself and which are supported by the XSL-FO processor. Typical examples are SVG and MathML.

## 4.21 CROSS-REFERENCES

Cross-references, as described in [CSS3G], are supported by CSSTOXSLFO. This introduces the counter functions `target-counter()` and `target-counters()`, which function in the same way as the `counter()` and `counters()` functions except that they take the value of a counter at the element they refer to instead of the element for which they are specified. That is why the first argument should resolve to a URL. This is an eample where the anchor element is empty:

```
a:after
{
  content: "(see page " target-counter(attr(href), page) ")";
}
```

It is also possible to take the text from a remote element through the `target-text()` function like this:

```
<h1 id="chapter1">Chapter 1</h1>
...
<p>See <a href="#chapter1"/> for ...
```

The style sheet:

```
a:after
{
  content:
    'chapter "' target-text(attr(href), content-element) '"';
}
```

The keywords that determine which part of the text is taken are `content`, `contents`, `content-element`, `content-before`, `content-after` and `content-first-letter`.

## 4.22 BOOKMARKS

It is possible to generate bookmarks, as described in [CSS3G]. You need the properties `bookmark-level` and `bookmark-label` to achieve this. The result will be a bookmark tree. Here is a simple example:

```
h1
  {
    bookmark-level: 1;
  }

  h2
  {
    bookmark-level: 2;
  }
```

```
h3
{
  bookmark-level: 3;
}

h1, h2, h3
{
  bookmark-label: content-before " " content-element;
}
```

The keywords that determine which part of the text is used for the label are `content`, `contents`, `content-element`, `content-before`, `content-after` and `content-first-letter`.

## 4.23 EXTRA VALUES FOR THE FLOAT PROPERTY

The values defined in section 12 of [CSS3G] can be used. However, they are all mapped to XSL-FO before floats.

## 4.24 PROPERTY SPECIFICATIONS

### *4.24.1 anchor*

*Value:* <identifier> | attr(X)
*Initial:* none
*Applies to:* block-level and inline-level elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

<identifier>
: The qualified name of an attribute, the value of which is the target ID. This type of value is *deprecated*, because it doesn't support namespace prefixes.

attr(X)
: This returns the value of the attribute of the subject with the qualified name X. The CSS3 namespace prefixes are supported. The value is the target ID.

### *4.24.2 change-bar-class*

*Value:* <name>
*Initial:* none, value required
*Applies to:* before and after pseudo elements
*Inherited:* no
*Percentages:* N/A

*Media:* print

<name>
: An NCName, as defined in [NAMES], to allow pairing of before and after elements, which don't have to belong to the same element. This way a change bar context is created.

### *4.24.3 change-bar-color*

*Value:* <color>
*Initial:* the value of the `color` property
*Applies to:* before pseudo elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

<color>
: Specifies the color of the change bar.

### *4.24.4 change-bar-offset*

*Value:* <length>
*Initial:* 6pt
*Applies to:* before pseudo elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

<length>
: Gives the distance from the edge of the column area containing the text that is marked as changed to the center of the generated change bar. A positive distance is directed away from the column region and into the margin regardless of the `change-bar-placement` property. Relative lengths (i.e., percentage values and lengths with units of "em") are not permitted for the value of this property.

### *4.24.5 change-bar-placement*

*Value:* left | right | inside | outside | alternate
*Initial:* start
*Applies to:* before pseudo elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

alternate
: When there are exactly two columns, the change bar will be offset from the left edge of all column one areas and the right edge of all column two areas; when there are any other number of columns, this value is equivalent to "outside".

inside
: If the page binding edge is on the left-edge, the change bar will be offset from the left edge of all column areas. If the binding is the right-edge, the change bar will be offset from the right edge of all column areas. If the page binding edge is on neither the left-edge nor right-edge, the change bar will be offset from the left edge of all column areas.

left
: The change bar will be offset from the left edge of all column areas.

outside
: If the page binding edge is on the left-edge, the change bar will be offset from the right edge of all column areas. If the binding is the right-edge, the change bar will be offset from the left edge of all column areas. If the page binding edge is on neither the left-edge nor right-edge, the change bar will be offset from the right edge of all column areas.

right
: The change bar will be offset from the right edge of all column areas.

#### *4.24.6 change-bar-style*

*Value:* <border-style>
*Initial:* none
*Applies to:* before pseudo elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

#### *4.24.7 change-bar-width*

*Value:* <border-width>
*Initial:* medium
*Applies to:* before pseudo elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

<border-width>
: Relative lengths (i.e., percentage values and lengths with units of "em") are not permitted for the value of this property.

#### *4.24.8 colspan*

*Value:* <integer>
*Initial:* 1
*Applies to:* table cells
*Inherited:* no
*Percentages:* N/A

*Media:* print

<integer>
Expresses the number of columns the table cell will span. The value must be larger than or equal to 1.

### *4.24.9 column-count*

*Value:* <integer> | inherit
*Initial:* 1
*Applies to:* the page context
*Inherited:* no
*Percentages:* N/A
*Media:* print

<integer>
The value must be larger than or equal to 1.

### *4.24.10 column-gap*

*Value:* <length> | <percentage> | inherit
*Initial:* 12.0pt
*Applies to:* the page context
*Inherited:* no
*Percentages:* refer to the width of the body region
*Media:* print

<length>
This is an unsigned length, If a negative value has been specified a value of opt will be used.
<percentage>
The value is a percentage of the width of the body region.

### *4.24.11 column-span*

*Value:* none | all | inherit
*Initial:* none
*Applies to:* block elements which are not in table elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

all
This element spans all columns of a multi-column region.
none
This element does not span multiple columns of a multi-column region.

### *4.24.12 content-height*

*Value:* auto | scale-to-fit | <length> | <percentage> | inherit
*Initial:* auto
*Applies to:* graphic elements
*Inherited:* no
*Percentages:* intrinsic height
*Media:* print

auto
: The content-height should be the intrinsic content-height.

scale-to-fit
: A size of the content-height equal to the height of the viewport. This implies a certain scaling factor to be applied onto the content.

<length>
: An absolute size for the content-height. This implies a certain scaling factor to be applied onto the content.

<percentage>
: A percentage representing a scaling factor applied to the intrinsic height.

### *4.24.13 content-type*

*Value:* auto | <string>
*Initial:* auto
*Applies to:* graphic elements
*Inherited:* no
*Percentages:* intrinsic height
*Media:* print

auto
: No identification of the content-type. The User Agent may determine it by "sniffing" or by other means.

<string>
: A specification of the content-type in terms of a mime-type, which has the form "content-type:" followed by a mime content-type, e.g., content-type="content-type:image/svg+xml".

### *4.24.14 content-width*

*Value:* auto | scale-to-fit | <length> | <percentage> | inherit
*Initial:* auto
*Applies to:* graphic elements
*Inherited:* no
*Percentages:* intrinsic width
*Media:* print

auto
: The content-width should be the intrinsic content-width.

scale-to-fit
: A size of the content-width equal to the width of the viewport. This implies a certain scaling factor to be applied onto the content.

<length>
: An absolute size for the content-width. This implies a certain scaling factor to be applied onto the content.

<percentage>
: A percentage representing a scaling factor applied to the intrinsic width.

#### *4.24.15 display*

This section specifies additional values for the property.

| | |
|---|---|
| *Value:* | footnote-body \| footnote-reference \| foreign \| graphic \| inline-block \| leader \| wrapper |
| *Initial:* | inline |
| *Applies to:* | all elements |
| *Inherited:* | no |
| *Percentages:* | N/A |
| *Media:* | print |

footnote-body
: The contents of the element goes to the footnote area. The element must be either immediately preceded by an element of type `footnote-reference` or have a `:before` pseudo element of that type. Otherwise it is treated as if its display type were `none`. Whitespace between a footnote reference and body is removed. In case a pseudo element is used, the contents it generates is displayed in the flow, as well as in the footnote body.

footnote-reference
: This is an inline variant. Its contents is displayed in the flow. It can occur without a following `footnote-body` element.

foreign
: If an element has this display type, it is placed unmodified in an `fo:in-stream-foreign-object` element.

graphic
: This display type is used to include external graphics. It is an inline level display type. Elements marked with it are replaced elements.

inline-block
: A block box, which itself is flowed as a single inline box, similar to a replaced element. The inside of an inline-block is formatted as a block box, and the box itself is formatted as an inline box. See also section 4.1 of [CSS3B].

leader
: This display type is used to produce XSL-FO leaders. It is an inline level display type.

wrapper
: An element with this display type doesn't contribute any formatting objects. Its inherited properties are nevertheless inherited by its subtree.

### *4.24.16 force-page-count*

*Value:* auto | even | odd | end-on-even | end-on-odd | no-force | inherit
*Initial:* auto
*Applies to:* the page context
*Inherited:* no
*Percentages:* N/A
*Media:* print

The property is used to impose a constraint on the number of pages in a page sequence. In the event that this constraint is not satisfied, an additional page will be added to the end of the sequence. This page becomes the "last" page of that sequence.

auto
: Force the last page in this page sequence to be an odd page if the initial page number of the next page sequence is even. Force it to be an even page if the initial page number of the next page sequence is odd. If there is no next page sequence or if the value of its initial page number is "auto" do not force any page.

even
: Force an even number of pages in this page sequence.

odd
: Force an odd number of pages in this page sequence.

end-on-even
: Force the last page in this page sequence to be an even page.

end-on-odd
: Force the last page in this page sequence to be an odd page.

no-force
: Do not force either an even or an odd number of pages in this page sequence.

### *4.24.17 hyphenate*

*Value:* false | true | inherit
*Initial:* false
*Applies to:* block-level and inline-level elements
*Inherited:* yes
*Percentages:* N/A
*Media:* print

false
: Hyphenation is not active for the text in this element.

true
: Hyphenation is active for the text in this element.

### *4.24.18 initial-page-number*

*Value:* auto | auto-odd | auto-even | <integer> | inherit
*Initial:* auto
*Applies to:* the page context

*Inherited:* no
*Percentages:* N/A
*Media:* print

auto
: The initial number shall be set to 1 if no previous page-sequence exists in the document. If a preceding page-sequence exists, the initial number will be one greater than the last number for that sequence.

auto-odd
: A value is determined in the same manner as for “auto”. If that value is an even number 1 is added.

auto-even
: A value is determined in the same manner as for “auto”. If that value is an odd number 1 is added.

<integer>
: A positive integer. If a negative or non-integer value is provided, the value will be rounded to the nearest integer value greater than or equal to 1.

#### *4.24.19 leader-alignment*

*Value:* none | reference-area | page | inherit
*Initial:* none
*Applies to:* leader elements
*Inherited:* yes
*Percentages:* N/A
*Media:* print

Specifies whether leader elements having identical content and property values shall have their patterns aligned with each other, with respect to their common reference-area or page. For leader elements where the `leader-pattern` property is specified as `dots` or as `use-content`, this property will be honored. If the leader elements is aligned, the left-edge of each cycle of the repeated pattern will be placed on the left-edge of the next cycle in the appropriate pattern-alignment grid.

none
: Leader-pattern has no special alignment.

page
: Leader-pattern is aligned as if it began on the current page's left-edge.

reference-area
: Leader-pattern is aligned as if it began on the current reference-area's content-rectangle left-edge.

#### *4.24.20 leader-length*

*Value:* <length> | <percentage> | inherit
*Initial:* 12.0pt
*Applies to:* leader elements
*Inherited:* yes

*Percentages:* refer to the width of the content-rectangle of the parent area.
*Media:* print

<length>
Sets the length of a leader element.
<percentage>
Sets the length of a leader element to a percentage of the width of the content-rectangle of the parent area.

### *4.24.21 leader-pattern*

*Value:* space | rule | dots | use-content | inherit
*Initial:* space
*Applies to:* leader elements
*Inherited:* yes
*Percentages:* N/A
*Media:* print

dots
Leader is to be filled with a repeating sequence of dots. The choice of dot character is dependent on the user agent.
rule
Leader is to be filled with a rule. If this choice is selected, the `rule-thickness` and `rule-style` properties are used to set the leader's style.
space
Leader is to be filled with blank space.
use-content
Leader is to be filled with a repeating pattern as specified by the children of the leader element.

### *4.24.22 leader-pattern-width*

*Value:* use-font-metrics | <length> | <percentage> | inherit
*Initial:* use-font-metrics
*Applies to:* leader elements
*Inherited:* yes
*Percentages:* refer to the width of the content-rectangle of the parent area.
*Media:* print

use-font-metrics
Use the width of the leader-pattern as determined from its font metrics.
<length>
Sets the length for leader-pattern-repeating. The leader will have an inline-space inserted after each pattern cycle to account for any difference between the width of the pattern as determined by the font metrics and the width specified in this property. If the length specified is less than the value that would be determined via the `use-font-metrics` choice, the value of this property is computed as if `use-font-metrics` choice had been specified.

<percentage>
: Sets the length for leader-pattern-repeating to a percentage of the width of the content-rectangle of the parent area.

For leader elements where the `leader-pattern` property is specified as `dots` or as `use-content`, this property will be honored.

#### *4.24.23 link*

*Value:* <identifier> | attr(X)
*Initial:* none
*Applies to:* block-level and inline-level elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

<identifier>
: The qualified name of an attribute, the value of which is either a target ID or a URI. It is considered as an ID if the attribute is of type IDREF. This way a distinction is made with a relative URL. Note that the attribute type information should be available. This requires a document type definition. This type of value is *deprecated*, because it doesn't support namespace prefixes.

attr(X)
: This returns the value of the attribute of the subject with the qualified name X. The CSS3 namespace prefixes are supported. The value is considered as an ID if the attribute is of type IDREF. This way a distinction is made with a relative URL. Note that the attribute type information should be available. This requires a document type definition.

#### *4.24.24 list-style-type*

This section specifies additional glyph values for the property.

*Value:* box | check | diamond | hyphen
*Initial:* disc
*Applies to:* elements with “display: list-item”
*Inherited:* yes
*Percentages:* N/A
*Media:* print

box
: A hollow square.

check
: A check mark.

diamond
: A filled diamond.

hyphen
: A hyphen bullet.

### *4.24.25 orientation*

*Value:* 0 | 90 | 180 | 270 | -90 | -180 | -270 | inherit
*Initial:* 0
*Applies to:* block, table and table cell elements
*Inherited:* yes
*Percentages:* N/A
*Media:* print

0
: The material in this element is not rotated.

90
: The material in this element is rotated 90 degrees counter-clockwise with respect to the containing block element.

180
: The material in this element is rotated 180 degrees counter-clockwise with respect to the containing block element.

270
: The material in this element is rotated 270 degrees counter-clockwise with respect to the containing block element.

-90
: The material in this element is rotated 90 degrees clockwise with respect to the containing block element.

-180
: The material in this element is rotated 180 degrees clockwise with respect to the containing block element.

-270
: The material in this element is rotated 270 degrees clockwise with respect to the containing block element.

### *4.24.26 page*

This section specifies the property in the context of static regions. It defines the pages to which the static region applies. If more than one static region of the same kind (left, right, top or bottom) applies to a page, the most specific is selected, i.e. the one for which the most conditions are fulfilled. Each property value expresses a number of conditions.

*Value:* auto | first | last | left | right | blank | first-left | first-right | last-left | last-right | blank-left | blank-right | first-<identifier> | last-<identifier> | left-<identifier> | right-<identifier> | blank-<identifier> | first-left-<identifier> | first-right-<identifier> | last-left-<identifier> | last-right-<identifier> | blank-left-<identifier> | blank-right-<identifier> | <identifier>
*Initial:* auto
*Applies to:* static regions
*Inherited:* no
*Percentages:* N/A

*Media:* print

auto
: Applies to any page.

blank
: Applies if the page is a blank page. Blank pages can be generated, for example, when page breaks are forced to left or right pages.

blank-left
: Applies if the page is a blank and a left page.

blank-right
: Applies if the page is a blank and a right page.

blank-<identifier>
: Applies if the page is a blank and a named page, with the name set to the specified identifier.

blank-left-<identifier>
: Applies if the page is a blank, left and named page, with the name set to the specified identifier.

blank-right-<identifier>
: Applies if the page is a blank, right and named page, with the name set to the specified identifier.

first
: Applies if the page is a first page.

first-left
: Applies if the page is a first and a left page.

first-right
: Applies if the page is a first and a right page.

first-<identifier>
: Applies if the page is a first and a named page, with the name set to the specified identifier. When the document switches to a named page sequence, using the `page` property in the regular way, the first page of that sequence is a first page.

first-left-<identifier>
: Applies if the page is a first, left and named page, with the name set to the specified identifier.

first-right-<identifier>
: Applies if the page is a first, right and named page, with the name set to the specified identifier.

last
: Applies if the page is a last page.

last-left
: Applies if the page is a last and a left page.

last-right
: Applies if the page is a last and a right page.

last-<identifier>
: Applies if the page is a last and a named page, with the name set to the specified identifier. When the document switches to a named page sequence, using the `page` property in the regular way, the last page of that sequence is a last page.

last-left-<identifier>
: Applies if the page is a last, left and named page, with the name set to the specified identifier.

last-right-<identifier>
: Applies if the page is a last, right and named page, with the name set to the specified identifier.

left
: Applies if the page is a left page.

left-<identifier>
: Applies if the page is a left and a named page, with the name set to the specified identifier.

right
: Applies if the page is a right page.

right-<identifier>
: Applies if the page is a right and a named page, with the name set to the specified identifier.

<identifier>
: Applies if the page is a named page, with the name set to the specified identifier.

#### *4.24.27 precedence*

| | |
|---|---|
| *Value:* | false \| true \| inherit |
| *Initial:* | false |
| *Applies to:* | static top and bottom regions |
| *Inherited:* | no |
| *Percentages:* | N/A |
| *Media:* | print |

false
: The width of the region is reduced by the incursions of the left and right regions.

true
: The height of the left and right regions is reduced by the incursions of this region.

#### *4.24.28 region*

| | |
|---|---|
| *Value:* | body \| left \| right \| top \| bottom \| none |
| *Initial:* | none |
| *Applies to:* | all elements, but see prose |
| *Inherited:* | no |
| *Percentages:* | N/A |
| *Media:* | print |

body
: There should be one element with this value for the property. For the contents of this element the page sequences will be generated.

bottom
: This element becomes the bottom static region. The pages for which this is the case can be limited through the `page` property.

left
: This element becomes the left static region.

none
: The element is not a region.

right
: This element becomes the right static region.

top
: This element becomes the top static region.

The static region elements should be the first child elements of the body region. In other words, they should precede all elements which are not static regions, otherwise their `region` property is ignored. The property is also ignored if there are no @page rules. In that case the default page set-up is generated.

#### *4.24.29 rowspan*

| | |
|---|---|
| *Value:* | <integer> |
| *Initial:* | 1 |
| *Applies to:* | table cells |
| *Inherited:* | no |
| *Percentages:* | N/A |
| *Media:* | print |

<integer>
: Expresses the number of rows the table cell will span. The value must be larger than or equal to 1.

#### *4.24.30 rule-style*

This property applies only if the `leader-pattern` property is specified as `rule`.

| | |
|---|---|
| *Value:* | none \| dotted \| dashed \| solid \| double \| groove \| ridge \| inherit |
| *Initial:* | solid |
| *Applies to:* | leader elements |
| *Inherited:* | yes |
| *Percentages:* | N/A |
| *Media:* | print |

dashed
: The rule is a series of short line segments.

dotted
: The rule is a series of dots.

double
: The rule is two solid lines. The sum of the two lines and the space between them equals the value of the `rule-thickness` property.

groove
: The rule looks as though it were carved into the canvas. (Top/left half of the rule's thickness is the color specified; the other half is white.)

none
: No rule, forces `rule-thickness` to 0.

ridge
: The opposite of “groove”, the rule looks as though it were coming out of the canvas. (Bottom/right half of the rule's thickness is the color specified; the other half is white.)

solid
: The rule is a single line segment.

#### *4.24.31 rule-thickness*

This property applies only if the `leader-pattern` property is specified as `rule`.

| | |
|---|---|
| *Value:* | <length> |
| *Initial:* | 1.0pt |
| *Applies to:* | leader elements |
| *Inherited:* | yes |
| *Percentages:* | N/A |
| *Media:* | print |

<length>
: The rule-thickness is always perpendicular to its length-axis. The rule is thickened equally above and below the line's alignment position.

#### *4.24.32 scaling*

| | |
|---|---|
| *Value:* | uniform \| non-uniform \| inherit |
| *Initial:* | uniform |
| *Applies to:* | graphic elements |
| *Inherited:* | no |
| *Percentages:* | intrinsic width |
| *Media:* | print |

non-uniform
: Scaling need not preserve the intrinsic aspect ratio.

uniform
: Scaling should preserve the intrinsic aspect ratio.

#### *4.24.33 scaling-method*

| | |
|---|---|
| *Value:* | auto \| integer-pixels \| resample-any-method \| inherit |
| *Initial:* | auto |
| *Applies to:* | graphic elements |
| *Inherited:* | no |
| *Percentages:* | intrinsic width |

*Media:* print

auto
: The User Agent is free to choose either resampling, integer scaling, or any other scaling method.

integer-pixels
: The User Agent should scale the image such that each pixel in the original image is scaled to the nearest integer number of device-pixels that yields an image less-then-or-equal-to the image size derived from the content-height, content-width, and scaling properties.

resample-any-method
: The User Agent should resample the supplied image to provide an image that fills the size derived from the `content-height`, `content-width`, and `scaling` properties. The user agent may use any sampling method.

#### *4.24.34 src*

*Value:* <identifier> | attr(X)
*Initial:* none, value required
*Applies to:* graphic elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

<identifier>
: The qualified name of an attribute, the value of which is a URI. This type of value is *deprecated*, because it doesn't support namespace prefixes.

attr(X)
: This returns the value of the attribute of the subject with the qualified name X. The CSS3 namespace prefixes are supported. The value is a URI.

#### *4.24.35 string-set*

*Value:* none | <identifier> contents | <identifier> <content>
*Initial:* none
*Applies to:* all elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

none
: No named string is set.

<identifier> contents
: The string named by the identifier is set to the textual contents of the element.

<identifier> <content>
: The string named by the identifier is set to the result of the evaluation of the expression in <content>. The syntax for the expression is the same as that for the `content` property.

### *4.24.36 text-align-last*

*Value:* center | inside | justify | left | outside | relative | right | inherit
*Initial:* relative
*Applies to:* block elements
*Inherited:* no
*Percentages:* N/A
*Media:* print

center
: Specifies that the contents is to be centered horizontally.

inside
: If the page binding edge is on the left-edge, the alignment will be left. If the binding is on the right-edge, the alignment will be right. If neither, use left alignment.

justify
: Specifies that the contents is to be expanded to fill the available width.

left
: Specifies that the contents is to be aligned on the left-edge.

outside
: If the page binding edge is on the left-edge, the alignment will be right. If the binding is on the right-edge, the alignment will be left. If neither, use right alignment.

relative
: If `text-align` is `justify`, then the alignment of the last line, and of any line ending in U+000A, will be left. If `text-align` is not `justify`, `text-align-last` will use the value of `text-align`.

right
: Specifies that the contents is to be aligned on the right-edge.

### *4.24.37 table-omit-footer-at-break*

*Value:* false | true
*Initial:* false
*Applies to:* tables
*Inherited:* no
*Percentages:* N/A
*Media:* print

false
: This property specifies that the footer should not be ommitted.

true
: This property specifies that the footer should be ommitted.

### *4.24.38 table-omit-header-at-break*

*Value:* false | true
*Initial:* false

*Applies to:* tables
*Inherited:* no
*Percentages:* N/A
*Media:* print

false
: This property specifies that the header should not be ommitted.

true
: This property specifies that the header should be ommitted.

## 4.25 MISCELLANEOUS SPECIFICATIONS

### *4.25.1 The :blank Pseudo-class*

The :blank pseudo-class is available to specify properties in the page context for blank pages. Those can be generated, for example, when pages are forced to start at the left or right.

### *4.25.2 The :last Pseudo-class*

The :last pseudo-class is available to specify properties in the page context for last pages. This is analogous to the :first pseudo-class.

### *4.25.3 The background, border and padding page properties*

The background, border and padding properties, as defined in [CSS3P], are implemented. They are, however, not entirely compatible with that specification. The implementation applies the properties to the `region-body`, because in XSL-FO they are not defined at the page master level.

### *4.25.4 The page and pages Counters*

The page counter represents the current page number, while the pages counter represents the total number of pages in the document. Both can be used in static regions only. The page counter may be reset in the page context.

### *4.25.5 The page-ref Function*

The page-ref function can be used in the `content` property. Its only parameter is either the qualified name of an attribute that contains the ID of another element, or the attr(X) function, where X is the qualified name of such an attribute. The former is deprecated, because it doesn't support CSS3 namespace prefixes, while the latter does. The function call will be replaced with the number of the page the target element is on.

### *4.25.6 The string Function*

The string function produces the string that was saved with a `string-set` property. Its argument is the name used in a `string-set` property. If a named string is set more than once on a page, the first occurrence will be returned by the string function.

### *4.25.7 The footnote Counter Style*

This counter style produces symbols in the following order: *, †, ‡, §, ||, ¶, #, **, ††, ‡‡, §§. If the counter value if larger than the number of symbols in the preceding list, the * symbol is generated.

### *4.25.8 The pcw Unit*

This unit is available for the `width` property of an element with display type `table-column`. It expresses the proportional width for a table column. The value should be divided by the sum of all the present proportional widths, which itself is equal to the width of the table minus all fixed column widths.

### *4.25.9 The @namespace Rule*

With the @namespace rule a namespace can be declared, with or without a prefix. In the latter case it is the default namespace. The scope of a declared namespace is limited to the style sheet entity in which it is declared. The @namespace rule should come right after the @import rules if there are any and before all other rules. An @namespace rule has an optional prefix argument, which is an identifier, followed by a mandatory URI specification. Consult [CSS3S] to learn how namespaces work with selectors.

### *4.25.10 The style Attribute*

The `style` attribute in the namespace `http://www.w3.org/1998/CSS` is treated in the same way as in the namespace `http://www.w3.org/1999/xhtml`.

### *4.25.11 The XHTML meta Element*

The XHTML `meta` element is converted to meta-data elements for XEP, Antenna House and FOP. For the former two the `name` and `content` attributes become the `name` and `value` attributes respectively. For the latter the `name` attribute becomes the name of a Dublin Core element, while the `content` attribute becomes its value.

# MERGING XML DATA 5

Generating an XML document, which is decorated with CSS, remains a difficult task, because it involves complex transformations. However, it can be simplified if those transformations don't have to take into account the layout of the result. This is why CSSTOXSLFO provides the possibility to merge XML data with a template, based on placeholders in the latter. Those are paths in the XML data. The mechanism supports mapping to individual data elements as well as master-detail structures.

The transformation of the XML data can be limited to a data-to-data transformation. This is necessary in case the original data is not fit for presentation purposes. In any case, if a transformation can't be avoided it will be much simpler than when it must also produce the material that would go in a template.

## 5.1 MAPPING INDIVIDUAL ELEMENTS

This paragraph contains **some bold text** that was merged from the data file below.

```
<data>
  <simple>
    <element>some bold text</element>
  </simple>
</data>
```

The XHTML fragment in the manual for this is as follows, where the prefix "d" is bound to the namespace `urn:com-renderx:xmlmerge`.

```
<p>This paragraph contains <b
d:binding="/data/simple/element" /> that was
merged from the data file below.</p>
```

## 5.2 MASTER-DETAIL STRUCTURES

A master-detail structure consists of an iteration of elements, which contain in turn iterations of elements. This can be several levels deep. Such a structure can be merged into a template. The template language should have some repeating structure to receive the data. In XHTML the obvious example is the `table` element, which can be nested. Note that bindings may have relative paths, which are resolved against the enclosing binding level. The next table is the result of merging the data file below it.

| *Level 1* | *Level 2* |
|---|---|
| Cell 1 | *Subtable* |
| | Cell 1 1 |
| | Cell 1 2 |
| | Cell 1 3 |
| | Cell 1 4 |
| Cell 2 | *Subtable* |
| | Cell 2 1 |
| | Cell 2 2 |
| | Cell 2 3 |
| | Cell 2 4 |
| Cell 3 | *Subtable* |
| | Cell 3 1 |
| | Cell 3 2 |
| | Cell 3 3 |
| | Cell 3 4 |

```
<data>
  <master-detail>
    <level1>
      <value>Cell 1</value>
      <level2>
        <value>Cell 1 1</value>
      </level2>
      <level2>
        <value>Cell 1 2</value>
      </level2>
      <level2>
        <value>Cell 1 3</value>
      </level2>
      <level2>
        <value>Cell 1 4</value>
      </level2>
    </level1>
    <level1>
      <value>Cell 2</value>
      <level2>
```

```
        <value>Cell 2 1</value>
      </level2>
      <level2>
        <value>Cell 2 2</value>
      </level2>
      <level2>
        <value>Cell 2 3</value>
      </level2>
      <level2>
        <value>Cell 2 4</value>
      </level2>
    </level1>
    <level1>
      <value>Cell 3</value>
      <level2>
        <value>Cell 3 1</value>
      </level2>
      <level2>
        <value>Cell 3 2</value>
      </level2>
      <level2>
        <value>Cell 3 3</value>
      </level2>
      <level2>
        <value>Cell 3 4</value>
      </level2>
    </level1>
  </master-detail>
</data>
```

This is the template fragment:

```
<table rules="all">
  <thead>
    <th>Level 1</th>
    <th>Level 2</th>
  </thead>
  <tbody>
    <tr d:binding="/data/master-detail/level1">
      <td d:binding="value"/>
      <td>
        <table rules="all">
          <thead>
            <th>Subtable</th>
          </thead>
          <tbody>
            <tr d:binding="level2">
              <td d:binding="value"/>
            </tr>
          </tbody>
        </table>
      </td>
    </tr>
  </tbody>
</table>
```

The bindings on the `tr` elements point to the repeating elements in the data file, while the paths that go deeper point to individual elements within a repreating element. The prefix "d" is bound to the namespace `urn:com-renderx:xmlmerge`.

## 5.3 PRESENCE OF ELEMENTS

An alternative to the `binding` attribute is the `present` attribute. It should also have a path as its value. When it is placed on an element in the template then that element is only considered for processing when the path occurs in the data and has non-empty values in it, directly or indirectly. Otherwise the element is removed with everything below it.

## 5.4 ATTRIBUTE SUBSTITUTION

In the value of any attribute in the template you can use an expression like `binding(<path>)`. If the data contains a value with that path then the expression is substituted with that value. Otherwise it is removed from the attribute.

# 6 EMBEDDING CHARTS

Charts can be embedded in a document with an XML vocabulary in the `urn:com-renderx:charts` namespace. The charts are produced with the great library JFreeChart, which is embedded in CSSTOXSLFO.

The top element always denotes the type of chart. For example, the element `xy-bar` will produce an XY chart with bars, while the `pie` element is for a pie chart.

The two first elements within the top element are `properties` for general properties and `theme` for JFreeChart theme properties. Inside them a property is an element with a text value.

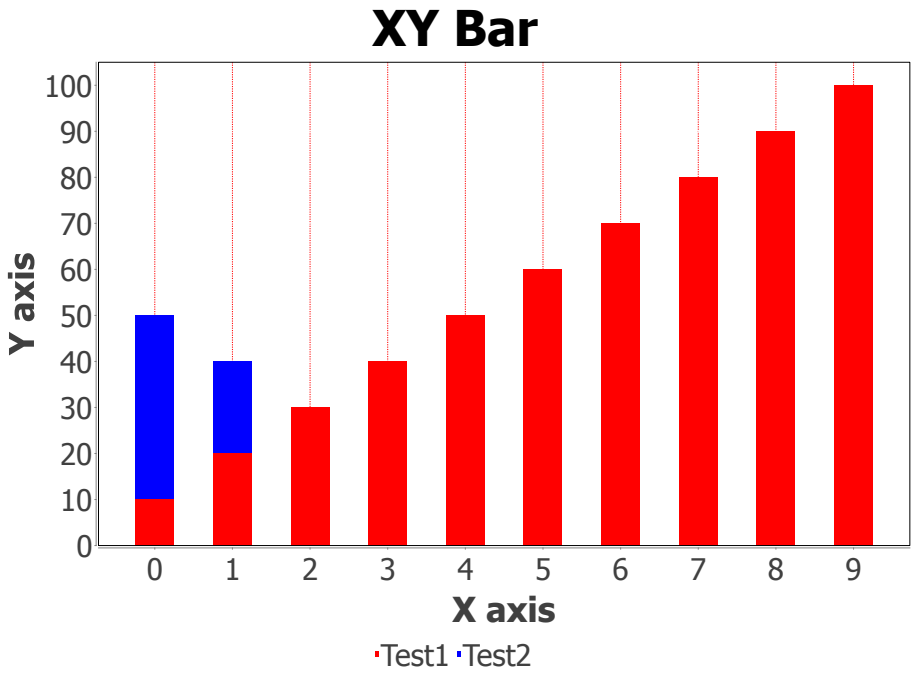
The rest of the XML fragment depends on the kind of chart. An XY chart will contain a `series` element, while a pie chart will have a `section` element.

The data for the charts don't have to be embedded in the chart description. With the merge function of chapter  it can be taken from another source.

## 6.1 EXAMPLES

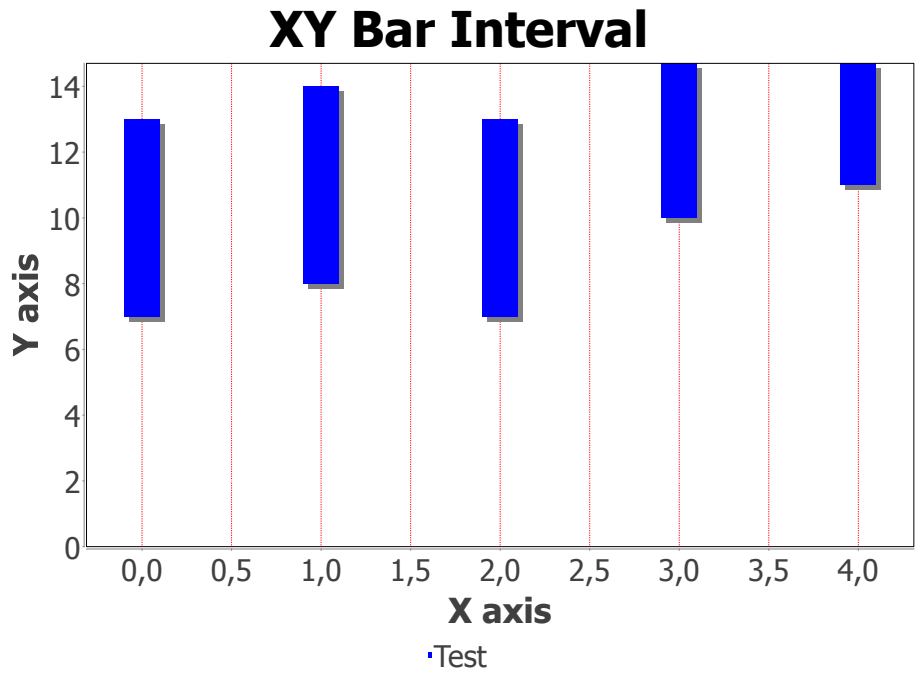
The examples show a chart, followed by its XML description.

### *6.1.1 XY Bar*



```
<xy-bar xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Bar</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <bar-width>0.5</bar-width>
    <legend>true</legend>
  </properties>
  <theme>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>10.0</y></value>
        <value><x>1.0</x><y>20.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>40.0</y></value>
        <value><x>4.0</x><y>50.0</y></value>
        <value><x>5.0</x><y>60.0</y></value>
        <value><x>6.0</x><y>70.0</y></value>
        <value><x>7.0</x><y>80.0</y></value>
        <value><x>8.0</x><y>90.0</y></value>
        <value><x>9.0</x><y>100.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>50.0</y></value>
        <value><x>1.0</x><y>40.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>20.0</y></value>
        <value><x>4.0</x><y>10.0</y></value>
      </values>
    </serie>
  </series>
</xy-bar>
```

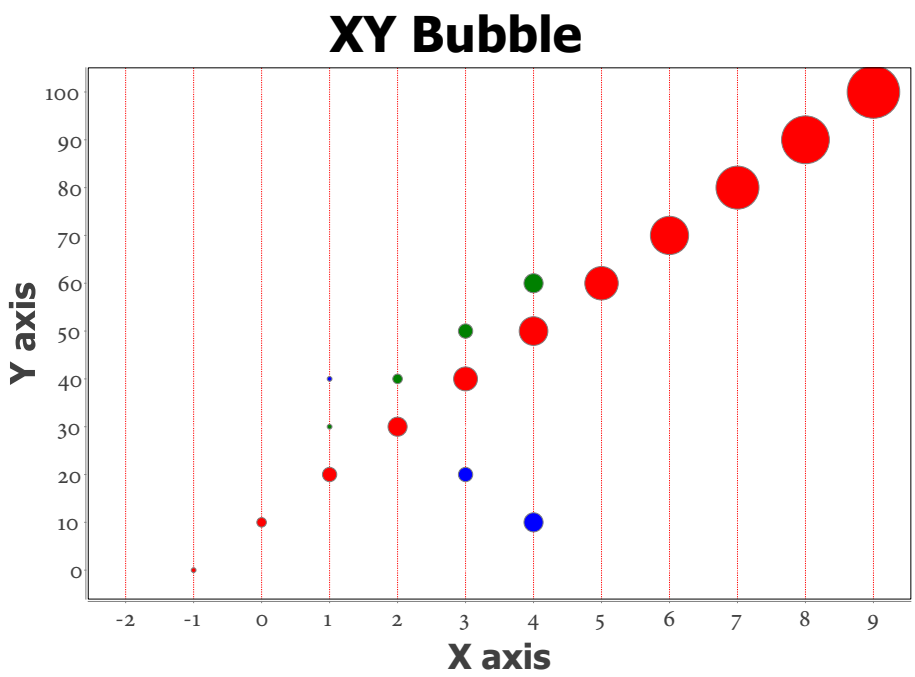
### 6.1.2 XY Bar Interval




```
<xy-bar xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Bar Interval</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <bar-width>0.5</bar-width>
    <legend>true</legend>
    <renderer-base-fill-paint>black</renderer-base-fill-paint>
    <renderer-bar-alignment-factor>0.5</renderer-bar-alignment-factor>
    <renderer-base>50</renderer-base>
    <renderer-shadow-visible>true</renderer-shadow-visible>
    <renderer-shadow-x-offset>2pt</renderer-shadow-x-offset>
    <renderer-shadow-y-offset>2pt</renderer-shadow-y-offset>
    <renderer-use-y-interval>true</renderer-use-y-interval>
  </properties>
  <theme>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
```

```
      <values>
        <value>
          <x>0.0</x><x-low>-0.1</x-low><x-high>0.1</x-high>
          <y>10.0</y><y-low>7.0</y-low><y-high>13.0</y-high>
        </value>
        <value>
          <x>1.0</x><x-low>0.9</x-low><x-high>1.1</x-high>
          <y>11.0</y><y-low>8.0</y-low><y-high>14.0</y-high>
        </value>
        <value>
          <x>2.0</x><x-low>1.9</x-low><x-high>2.1</x-high>
          <y>12.0</y><y-low>7.0</y-low><y-high>13.0</y-high>
        </value>
        <value>
          <x>3.0</x><x-low>2.9</x-low><x-high>3.1</x-high>
          <y>13.0</y><y-low>10.0</y-low><y-high>15.0</y-high>
        </value>
        <value>
          <x>4.0</x><x-low>3.9</x-low><x-high>4.1</x-high>
          <y>14.0</y><y-low>11.0</y-low><y-high>18.0</y-high>
        </value>
      </values>
    </serie>
  </series>
</xy-bar>
```

### 6.1.3 XY Bubble

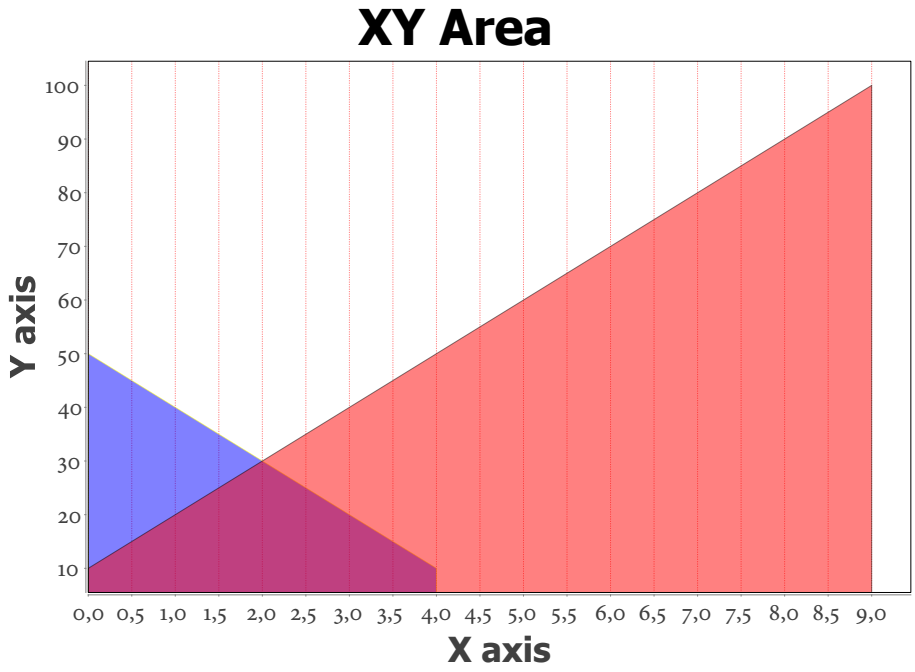



```
<xy-bubble xmlns="urn:com-renderx:charts">
  <properties>
```

```
  <title>XY Bubble</title>
  <format>SVG</format>
  <width>400pt</width>
  <height>300pt</height>
  <domain-axis-label>X axis</domain-axis-label>
  <range-axis-label>Y axis</range-axis-label>
</properties>
<theme>
  <regular-font-family>serif</regular-font-family>
  <regular-font-size>10</regular-font-size>
  <plot-background-paint>white</plot-background-paint>
  <domain-gridline-paint>red</domain-gridline-paint>
</theme>
<series>
  <serie>
    <properties>
      <label>Test1</label>
      <renderer-paint>red</renderer-paint>
    </properties>
    <values>
      <value><x>-2.0</x><y>-1.0</y><z>0.0</z></value>
      <value><x>-1.0</x><y>0.0</y><z>1.0</z></value>
      <value><x>0.0</x><y>10.0</y><z>2.0</z></value>
      <value><x>1.0</x><y>20.0</y><z>3.0</z></value>
      <value><x>2.0</x><y>30.0</y><z>4.0</z></value>
      <value><x>3.0</x><y>40.0</y><z>5.0</z></value>
      <value><x>4.0</x><y>50.0</y><z>6.0</z></value>
      <value><x>5.0</x><y>60.0</y><z>7.0</z></value>
      <value><x>6.0</x><y>70.0</y><z>8.0</z></value>
      <value><x>7.0</x><y>80.0</y><z>9.0</z></value>
      <value><x>8.0</x><y>90.0</y><z>10.0</z></value>
      <value><x>9.0</x><y>100.0</y><z>11.0</z></value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test2</label>
      <renderer-paint>blue</renderer-paint>
    </properties>
    <values>
      <value><x>0.0</x><y>50.0</y><z>0.0</z></value>
      <value><x>1.0</x><y>40.0</y><z>1.0</z></value>
      <value><x>2.0</x><y>30.0</y><z>2.0</z></value>
      <value><x>3.0</x><y>20.0</y><z>3.0</z></value>
      <value><x>4.0</x><y>10.0</y><z>4.0</z></value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test3</label>
      <renderer-paint>green</renderer-paint>
    </properties>
    <values>
      <value><x>0.0</x><y>20.0</y><z>0.0</z></value>
      <value><x>1.0</x><y>30.0</y><z>1.0</z></value>
      <value><x>2.0</x><y>40.0</y><z>2.0</z></value>
      <value><x>3.0</x><y>50.0</y><z>3.0</z></value>
```

```
        <value><x>4.0</x><y>60.0</y><z>4.0</z></value>
      </values>
    </serie>
  </series>
</xy-bubble>
```

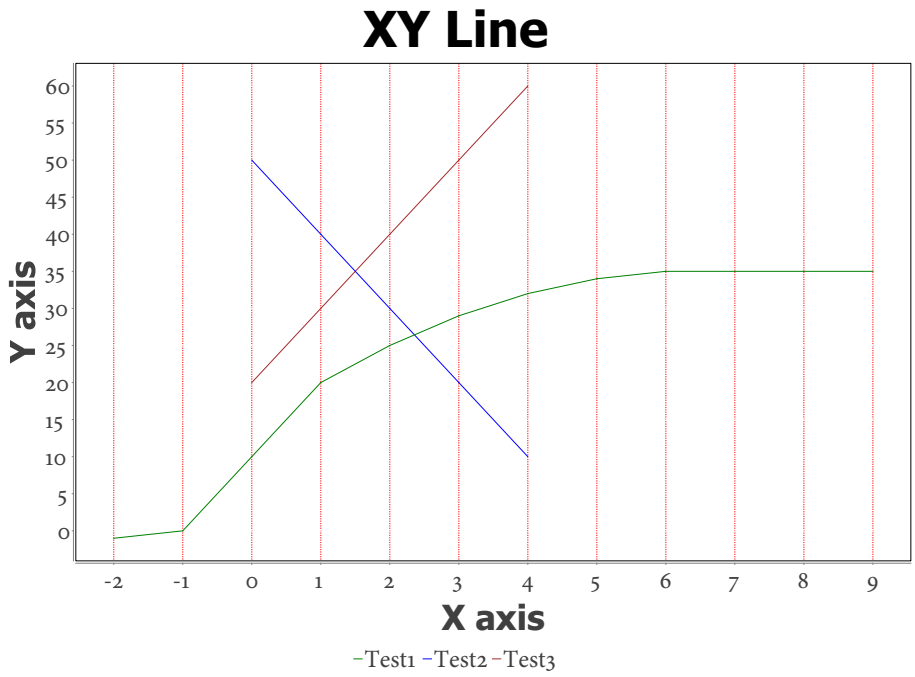
### 6.1.4 XY Area




```
<xy-area xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Area</title>
    <format>SVG</format>
    <width>400pt</width>
    <height>300pt</height>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <axis-auto-range-includes-zero>false</axis-auto-range-includes-zero>
    <renderer-outline>true</renderer-outline>
    <renderer-base-outline-paint>black</renderer-base-outline-paint>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
```

```
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>10.0</y></value>
        <value><x>1.0</x><y>20.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>40.0</y></value>
        <value><x>4.0</x><y>50.0</y></value>
        <value><x>5.0</x><y>60.0</y></value>
        <value><x>6.0</x><y>70.0</y></value>
        <value><x>7.0</x><y>80.0</y></value>
        <value><x>8.0</x><y>90.0</y></value>
        <value><x>9.0</x><y>100.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
        <renderer-outline-paint>yellow</renderer-outline-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>50.0</y></value>
        <value><x>1.0</x><y>40.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>20.0</y></value>
        <value><x>4.0</x><y>10.0</y></value>
      </values>
    </serie>
  </series>
</xy-area>
```

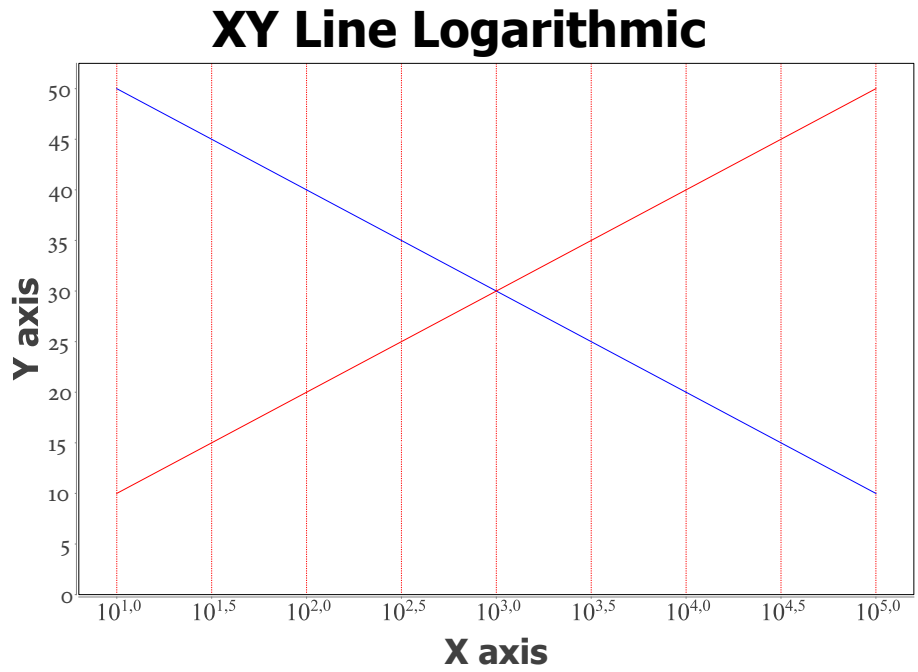
### 6.1.5 XY Line




```
<xy-line xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Line</title>
    <format>SVG</format>
    <width>400pt</width>
    <height>300pt</height>
    <legend>true</legend>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <use-spline>false</use-spline>
    <use-difference>false</use-difference>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>green</renderer-paint>
      </properties>
      <values>
        <value><x>-2.0</x><y>-1.0</y></value>
        <value><x>-1.0</x><y>0.0</y></value>
        <value><x>0.0</x><y>10.0</y></value>
```

```
        <value><x>1.0</x><y>20.0</y></value>
        <value><x>2.0</x><y>25.0</y></value>
        <value><x>3.0</x><y>29.0</y></value>
        <value><x>4.0</x><y>32.0</y></value>
        <value><x>5.0</x><y>34.0</y></value>
        <value><x>6.0</x><y>35.0</y></value>
        <value><x>7.0</x><y>35.0</y></value>
        <value><x>8.0</x><y>35.0</y></value>
        <value><x>9.0</x><y>35.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>50.0</y></value>
        <value><x>1.0</x><y>40.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>20.0</y></value>
        <value><x>4.0</x><y>10.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test3</label>
        <renderer-paint>brown</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>20.0</y></value>
        <value><x>1.0</x><y>30.0</y></value>
        <value><x>2.0</x><y>40.0</y></value>
        <value><x>3.0</x><y>50.0</y></value>
        <value><x>4.0</x><y>60.0</y></value>
      </values>
    </serie>
  </series>
</xy-line>
```

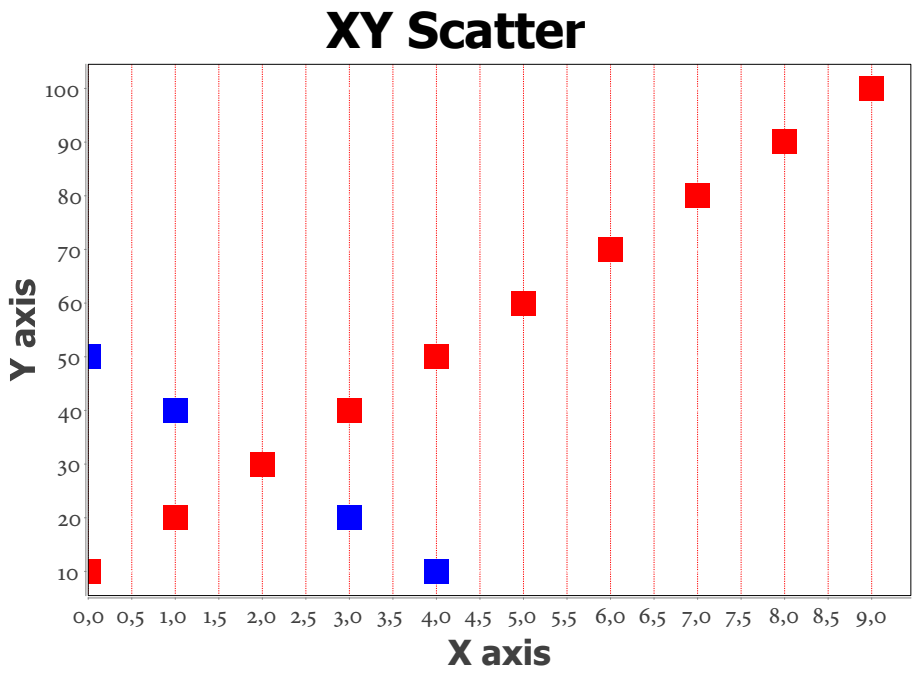
### 6.1.6 XY Line Logarithmic




```
<xy-line xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Line Logarithmic</title>
    <format>SVG</format>
    <width>400pt</width>
    <height>300pt</height>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <use-log>true</use-log>
    <domain-axis-base>10</domain-axis-base>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value><x>10.0</x><y>10.0</y></value>
        <value><x>100.0</x><y>20.0</y></value>
        <value><x>1000.0</x><y>30.0</y></value>
        <value><x>10000.0</x><y>40.0</y></value>
```

```
        <value><x>100000.0</x><y>50.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value><x>10.0</x><y>50.0</y></value>
        <value><x>100.0</x><y>40.0</y></value>
        <value><x>1000.0</x><y>30.0</y></value>
        <value><x>10000.0</x><y>20.0</y></value>
        <value><x>100000.0</x><y>10.0</y></value>
      </values>
    </serie>
  </series>
</xy-line>
```

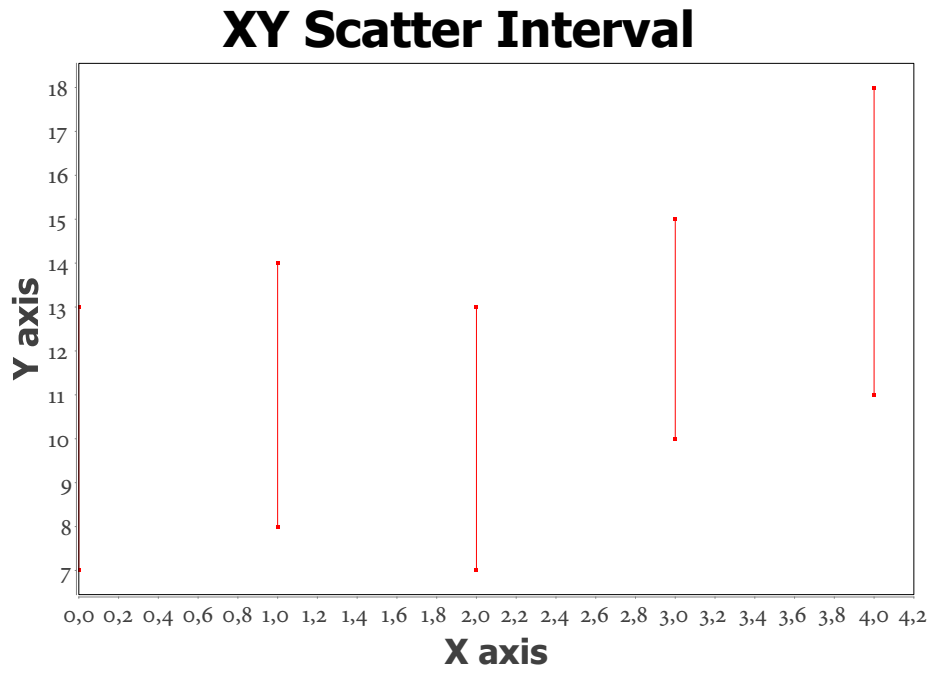
### 6.1.7 XY Scatter




```
<xy-scatter xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Scatter</title>
    <format>SVG</format>
    <width>400pt</width>
    <height>300pt</height>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <use-dot>true</use-dot>
```

```
    <axis-auto-range-includes-zero>false</axis-auto-range-includes-zero>
    <renderer-dot-width>10pt</renderer-dot-width>
    <renderer-dot-height>10pt</renderer-dot-height>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>10.0</y></value>
        <value><x>1.0</x><y>20.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>40.0</y></value>
        <value><x>4.0</x><y>50.0</y></value>
        <value><x>5.0</x><y>60.0</y></value>
        <value><x>6.0</x><y>70.0</y></value>
        <value><x>7.0</x><y>80.0</y></value>
        <value><x>8.0</x><y>90.0</y></value>
        <value><x>9.0</x><y>100.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>50.0</y></value>
        <value><x>1.0</x><y>40.0</y></value>
        <value><x>2.0</x><y>30.0</y></value>
        <value><x>3.0</x><y>20.0</y></value>
        <value><x>4.0</x><y>10.0</y></value>
      </values>
    </serie>
  </series>
</xy-scatter>
```

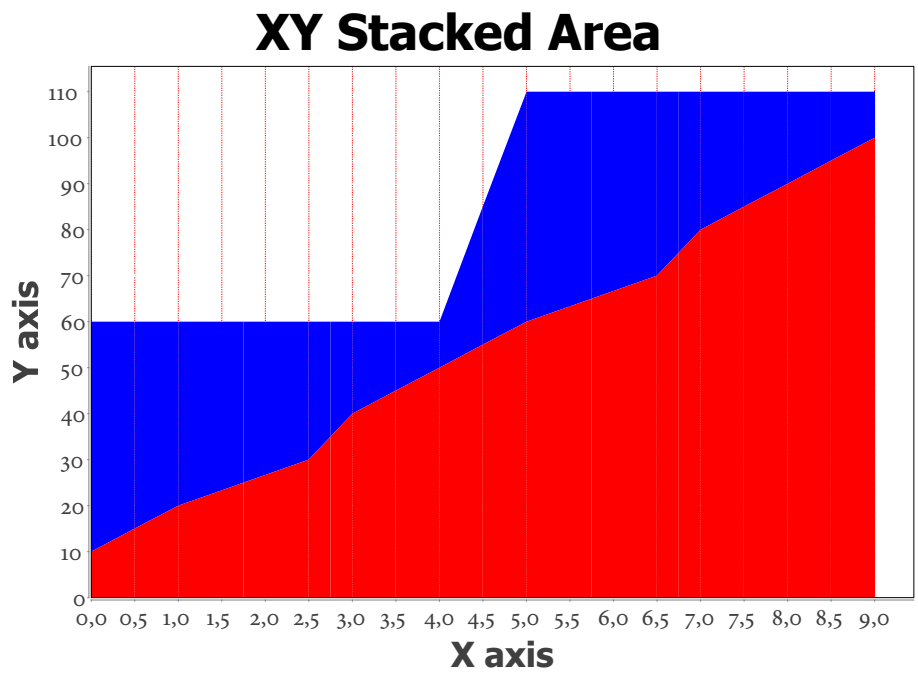
### 6.1.8 XY Scatter Interval




```
<xy-scatter xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Scatter Interval</title>
    <format>SVG</format>
    <width>400pt</width>
    <height>300pt</height>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value>
          <x>0.0</x>
          <y>10.0</y><y-low>7.0</y-low><y-high>13.0</y-high>
        </value>
        <value>
          <x>1.0</x>
          <y>11.0</y><y-low>8.0</y-low><y-high>14.0</y-high>
```

```
        </value>
        <value>
          <x>2.0</x>
          <y>12.0</y><y-low>7.0</y-low><y-high>13.0</y-high>
        </value>
        <value>
          <x>3.0</x>
          <y>13.0</y><y-low>10.0</y-low><y-high>15.0</y-high>
        </value>
        <value>
          <x>4.0</x>
          <y>14.0</y><y-low>11.0</y-low><y-high>18.0</y-high>
        </value>
      </values>
    </serie>
  </series>
</xy-scatter>
```

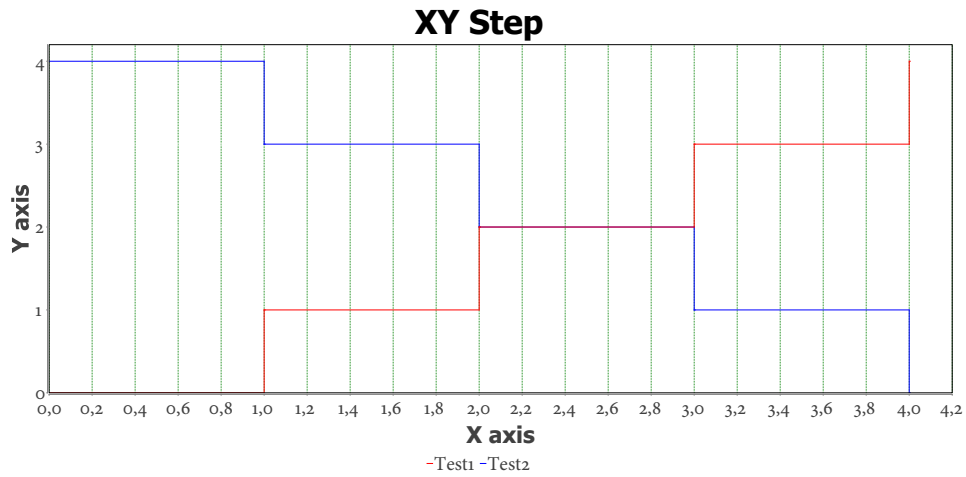
### 6.1.9 XY Stacked Area




```
<xy-stacked-area xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Stacked Area</title>
    <format>SVG</format>
    <width>400pt</width>
    <height>300pt</height>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <axis-auto-range-includes-zero>false</axis-auto-range-includes-zero>
    <renderer-round-x-coordinates>true</renderer-round-x-coordinates>
```

```
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
        <renderer-outline-paint>black</renderer-outline-paint>
        <renderer-fill-paint>yellow</renderer-fill-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>10.0</y></value>
        <value><x>1.0</x><y>20.0</y></value>
        <value><x>2.5</x><y>30.0</y></value>
        <value><x>3.0</x><y>40.0</y></value>
        <value><x>4.0</x><y>50.0</y></value>
        <value><x>5.0</x><y>60.0</y></value>
        <value><x>6.5</x><y>70.0</y></value>
        <value><x>7.0</x><y>80.0</y></value>
        <value><x>8.0</x><y>90.0</y></value>
        <value><x>9.0</x><y>100.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
        <renderer-outline-paint>black</renderer-outline-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>50.0</y></value>
        <value><x>1.0</x><y>40.0</y></value>
        <value><x>2.5</x><y>30.0</y></value>
        <value><x>3.0</x><y>20.0</y></value>
        <value><x>4.0</x><y>10.0</y></value>
        <value><x>5.0</x><y>50.0</y></value>
        <value><x>6.5</x><y>40.0</y></value>
        <value><x>7.0</x><y>30.0</y></value>
        <value><x>8.0</x><y>20.0</y></value>
        <value><x>9.0</x><y>10.0</y></value>
      </values>
    </serie>
  </series>
</xy-stacked-area>
```

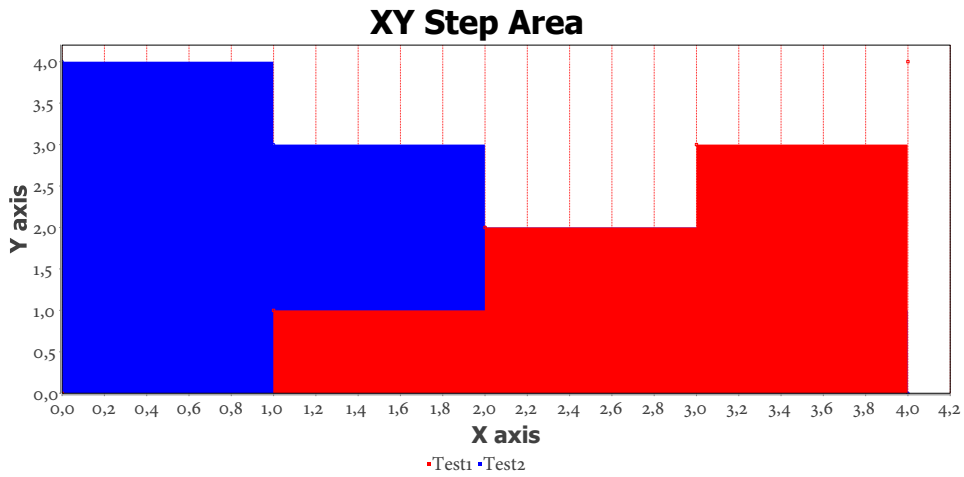
### 6.1.10 XY Step




```
<xy-step xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Step</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>600pt</width>
    <height>300pt</height>
    <padding>20pt</padding>
    <padding-right>50pt</padding-right>
    <format>SVG</format>
    <legend>true</legend>
    <renderer-step-point>1.0</renderer-step-point>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>12</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>green</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>0.0</y></value>
        <value><x>1.0</x><y>1.0</y></value>
        <value><x>2.0</x><y>2.0</y></value>
        <value><x>3.0</x><y>3.0</y></value>
        <value><x>4.0</x><y>4.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
```

```
      </properties>
      <values>
        <value><x>0.0</x><y>4.0</y></value>
        <value><x>1.0</x><y>3.0</y></value>
        <value><x>2.0</x><y>2.0</y></value>
        <value><x>3.0</x><y>1.0</y></value>
        <value><x>4.0</x><y>0.0</y></value>
      </values>
    </serie>
  </series>
</xy-step>
```

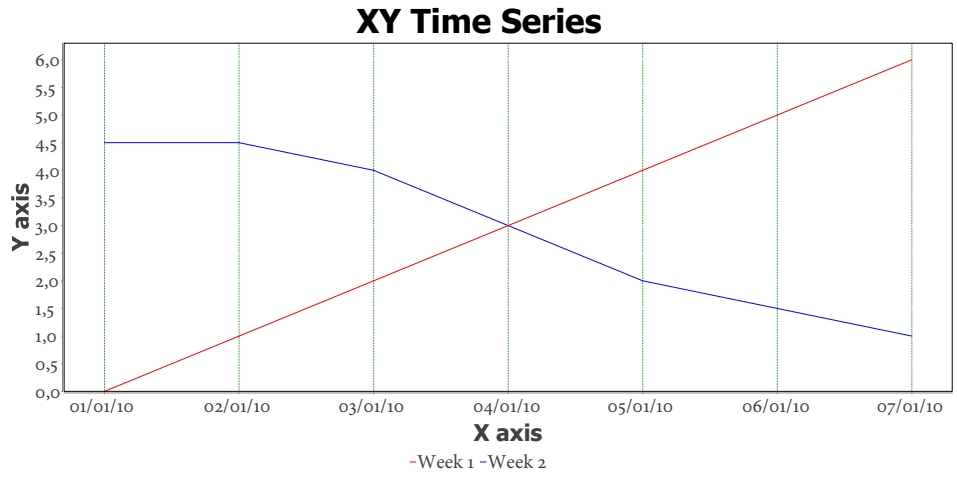
### 6.1.11 XY Step Area




```
<xy-step-area xmlns="urn:com-renderx:charts">
  <properties>
    <title>XY Step Area</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>600pt</width>
    <height>300pt</height>
    <padding>20pt</padding>
    <padding-right>50pt</padding-right>
    <format>SVG</format>
    <legend>true</legend>
    <renderer-step-point>1.0</renderer-step-point>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>12</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
```

```
      </properties>
      <values>
        <value><x>0.0</x><y>0.0</y></value>
        <value><x>1.0</x><y>1.0</y></value>
        <value><x>2.0</x><y>2.0</y></value>
        <value><x>3.0</x><y>3.0</y></value>
        <value><x>4.0</x><y>4.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value><x>0.0</x><y>4.0</y></value>
        <value><x>1.0</x><y>3.0</y></value>
        <value><x>2.0</x><y>2.0</y></value>
        <value><x>3.0</x><y>1.0</y></value>
        <value><x>4.0</x><y>0.0</y></value>
      </values>
    </serie>
  </series>
</xy-step-area>
```

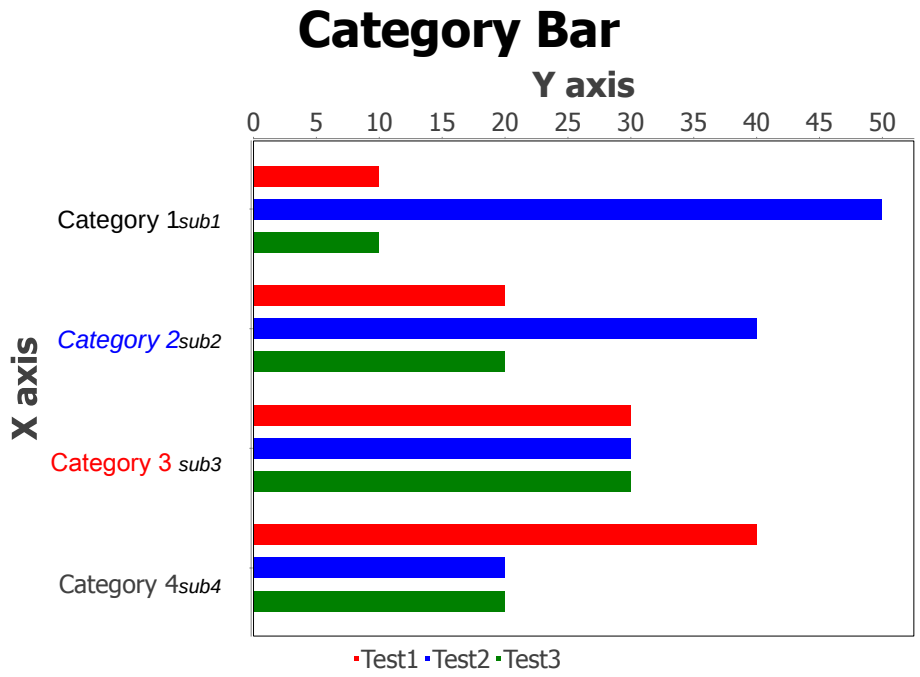
### 6.1.12 XY Time Series




```
<xy-time-series xmlns="urn:com-renderx:charts" xml:lang="en">
                                          <!-- For the dates -->
  <properties>
    <title>XY Time Series</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>600pt</width>
    <height>300pt</height>
    <padding>20pt</padding>
    <padding-right>50pt</padding-right>
    <format>SVG</format>
```

```
    <legend>true</legend>
    <domain-axis-tick-mark-position>start</domain-axis-tick-mark-position>
    <domain-axis-tick-unit>day</domain-axis-tick-unit>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>12</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>green</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Week 1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value><x>2010-01-01</x><y>0.0</y></value>
        <value><x>2010-01-02</x><y>1.0</y></value>
        <value><x>2010-01-03</x><y>2.0</y></value>
        <value><x>2010-01-04</x><y>3.0</y></value>
        <value><x>2010-01-05</x><y>4.0</y></value>
        <value><x>2010-01-06</x><y>5.0</y></value>
        <value><x>2010-01-07</x><y>6.0</y></value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Week 2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value><x>2010-01-01</x><y>4.5</y></value>
        <value><x>2010-01-02</x><y>4.5</y></value>
        <value><x>2010-01-03</x><y>4.0</y></value>
        <value><x>2010-01-04</x><y>3.0</y></value>
        <value><x>2010-01-05</x><y>2.0</y></value>
        <value><x>2010-01-06</x><y>1.5</y></value>
        <value><x>2010-01-07</x><y>1.0</y></value>
      </values>
    </serie>
  </series>
</xy-time-series>
```

### 6.1.13 Category Bar

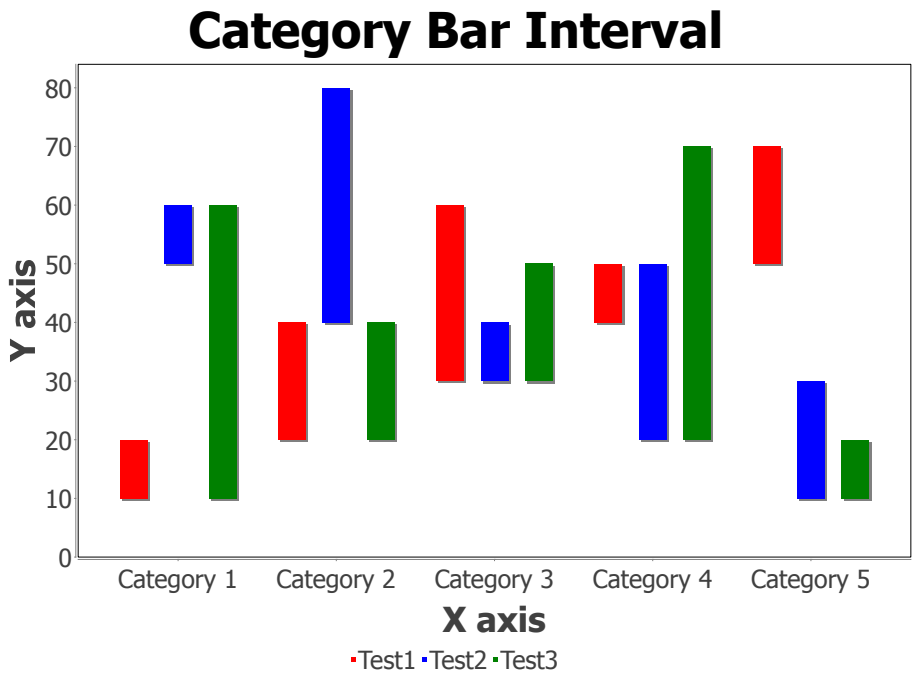



```
<category-bar xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Bar</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <orientation>horizontal</orientation>
    <domain-axis-sublabels>Category 1: sub1,Category 2: sub2,
 Category 3: sub3,Category 4: sub4</domain-axis-sublabels>
    <domain-axis-sublabel-font-family>Helvetica
</domain-axis-sublabel-font-family>
    <domain-axis-sublabel-font-size>8
</domain-axis-sublabel-font-size>
    <domain-axis-sublabel-font-style>italic
</domain-axis-sublabel-font-style>
    <domain-axis-category-label-position-offset>10
</domain-axis-category-label-position-offset>
    <domain-axis-category-label-positions>0
</domain-axis-category-label-positions>
    <domain-axis-maximum-category-label-width-ratio>5
</domain-axis-maximum-category-label-width-ratio>
    <domain-axis-tick-label-font-family>Category 1:Helvetica,
Category 2:Helvetica,Category 3:Arial
</domain-axis-tick-label-font-family>
    <domain-axis-tick-label-font-style>Category 1:plain,
```

```
Category 2:italic,Category 3:bold
</domain-axis-tick-label-font-style>
    <domain-axis-tick-label-font-size>Category 1:10,Category 2:10,
Category 3:10</domain-axis-tick-label-font-size>
    <domain-axis-tick-label-paint>Category 1:black,
Category 2:blue,Category 3:red</domain-axis-tick-label-paint>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>10.0</val>
        </value>
        <value>
          <category>Category 2</category><val>20.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>40.0</val>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>50.0</val>
        </value>
        <value>
          <category>Category 2</category><val>40.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>20.0</val>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test3</label>
        <renderer-paint>green</renderer-paint>
```

```
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>10.0</val>
        </value>
        <value>
          <category>Category 2</category><val>20.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>20.0</val>
        </value>
      </values>
    </serie>
  </series>
</category-bar>
```

### 6.1.14 Category Bar Interval

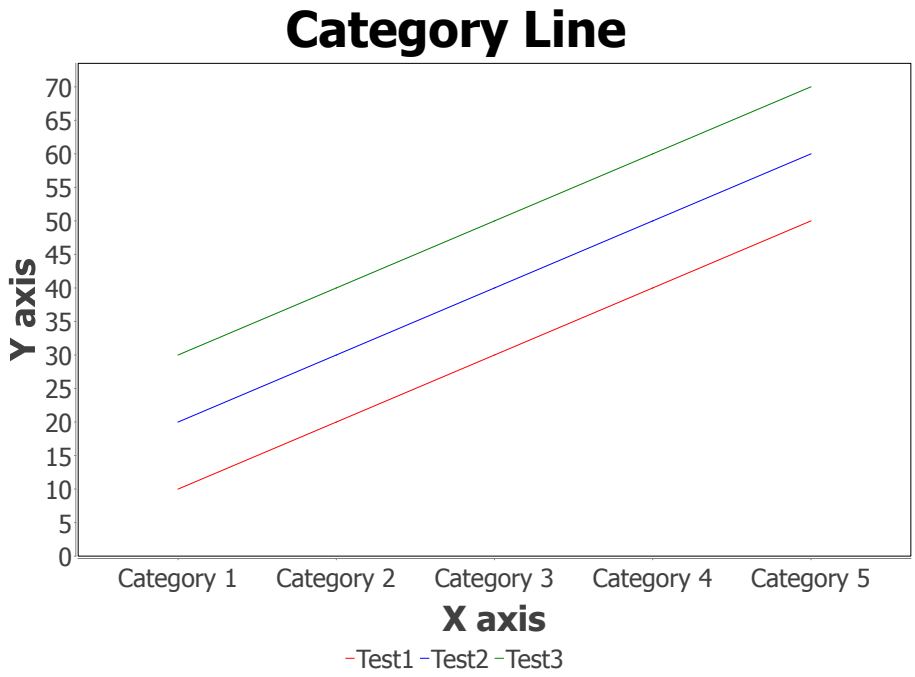



```
<category-bar xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Bar Interval</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
```

```
  <orientation>vertical</orientation>
</properties>
<theme>
  <regular-font-size>10</regular-font-size>
  <plot-background-paint>white</plot-background-paint>
  <domain-gridline-paint>red</domain-gridline-paint>
</theme>
<series>
  <serie>
    <properties>
      <label>Test1</label>
      <renderer-paint>red</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category>
        <start>10.0</start>
        <end>20.0</end>
      </value>
      <value>
        <category>Category 2</category>
        <start>20.0</start>
        <end>40.0</end>
      </value>
      <value>
        <category>Category 3</category>
        <start>30.0</start>
        <end>60.0</end>
      </value>
      <value>
        <category>Category 4</category>
        <start>40.0</start>
        <end>50.0</end>
      </value>
      <value>
        <category>Category 5</category>
        <start>50.0</start>
        <end>70.0</end>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test2</label>
      <renderer-paint>blue</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category>
        <start>50.0</start>
        <end>60.0</end>
      </value>
      <value>
        <category>Category 2</category>
        <start>40.0</start>
        <end>80.0</end>
      </value>
```

```
        <value>
          <category>Category 3</category>
          <start>30.0</start>
          <end>40.0</end>
        </value>
        <value>
          <category>Category 4</category>
          <start>20.0</start>
          <end>50.0</end>
        </value>
        <value>
          <category>Category 5</category>
          <start>10.0</start>
          <end>30.0</end>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test3</label>
        <renderer-paint>green</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category>
          <start>10.0</start>
          <end>60.0</end>
        </value>
        <value>
          <category>Category 2</category>
          <start>20.0</start>
          <end>40.0</end>
        </value>
        <value>
          <category>Category 3</category>
          <start>30.0</start>
          <end>50.0</end>
        </value>
        <value>
          <category>Category 4</category>
          <start>20.0</start>
          <end>70.0</end>
        </value>
        <value>
          <category>Category 5</category>
          <start>10.0</start>
          <end>20.0</end>
        </value>
      </values>
    </serie>
  </series>
</category-bar>
```

### 6.1.15 Category Line

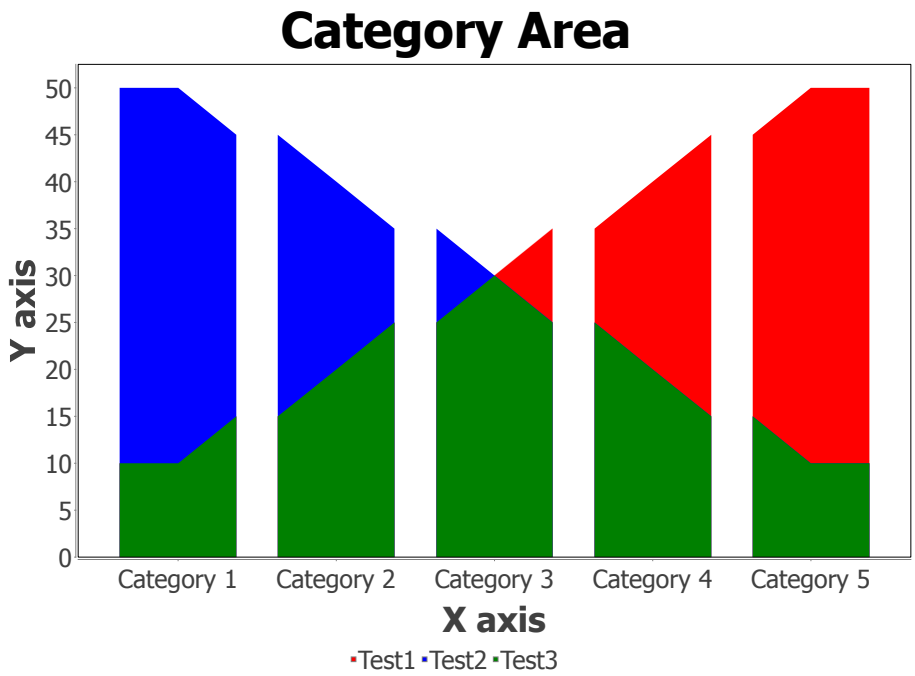



```
<category-line xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Line</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <orientation>vertical</orientation>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>10.0</val>
        </value>
        <value>
          <category>Category 2</category><val>20.0</val>
```

```
      </value>
      <value>
        <category>Category 3</category><val>30.0</val>
      </value>
      <value>
        <category>Category 4</category><val>40.0</val>
      </value>
      <value>
        <category>Category 5</category><val>50.0</val>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test2</label>
      <renderer-paint>blue</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>20.0</val>
      </value>
      <value>
        <category>Category 2</category><val>30.0</val>
      </value>
      <value>
        <category>Category 3</category><val>40.0</val>
      </value>
      <value>
        <category>Category 4</category><val>50.0</val>
      </value>
      <value>
        <category>Category 5</category><val>60.0</val>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test3</label>
      <renderer-paint>green</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>30.0</val>
      </value>
      <value>
        <category>Category 2</category><val>40.0</val>
      </value>
      <value>
        <category>Category 3</category><val>50.0</val>
      </value>
      <value>
        <category>Category 4</category><val>60.0</val>
      </value>
      <value>
        <category>Category 5</category><val>70.0</val>
      </value>
    </values>
```

```
    </serie>
  </series>
</category-line>
```

### 6.1.16 Category Area

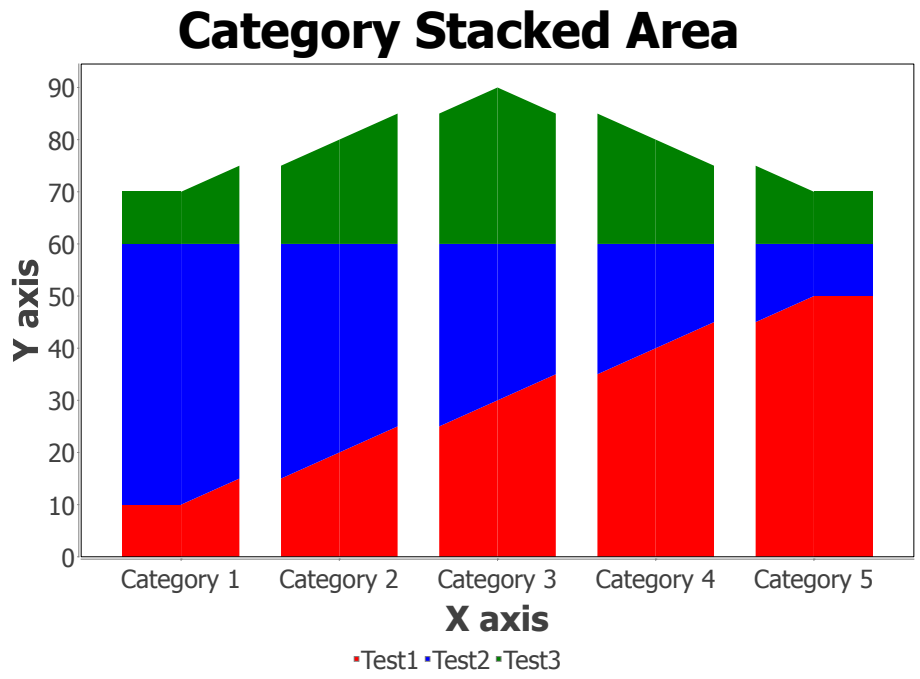



```
<category-area xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Area</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <orientation>vertical</orientation>
    <renderer-end-type>level</renderer-end-type>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
```

```
      <value>
        <category>Category 1</category><val>10.0</val>
      </value>
      <value>
        <category>Category 2</category><val>20.0</val>
      </value>
      <value>
        <category>Category 3</category><val>30.0</val>
      </value>
      <value>
        <category>Category 4</category><val>40.0</val>
      </value>
      <value>
        <category>Category 5</category><val>50.0</val>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test2</label>
      <renderer-paint>blue</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>50.0</val>
      </value>
      <value>
        <category>Category 2</category><val>40.0</val>
      </value>
      <value>
        <category>Category 3</category><val>30.0</val>
      </value>
      <value>
        <category>Category 4</category><val>20.0</val>
      </value>
      <value>
        <category>Category 5</category><val>10.0</val>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test3</label>
      <renderer-paint>green</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>10.0</val>
      </value>
      <value>
        <category>Category 2</category><val>20.0</val>
      </value>
      <value>
        <category>Category 3</category><val>30.0</val>
      </value>
      <value>
        <category>Category 4</category><val>20.0</val>
```

```
        </value>
        <value>
          <category>Category 5</category><val>10.0</val>
        </value>
      </values>
    </serie>
  </series>
</category-area>
```

### *6.1.17 Category Stacked Area*

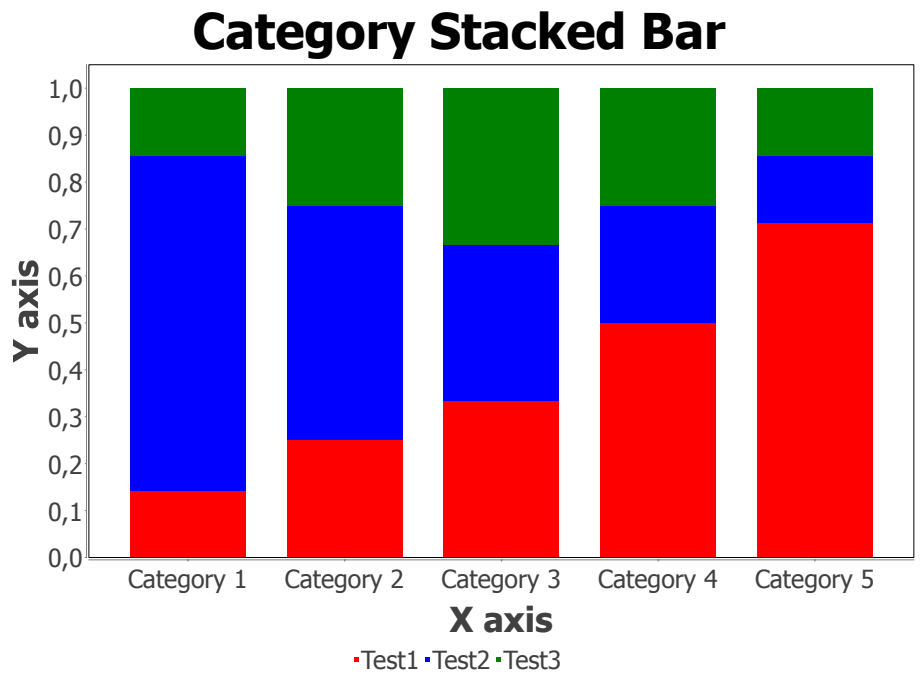



```
<category-stacked-area xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Stacked Area</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <orientation>vertical</orientation>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
```

```
      <label>Test1</label>
      <renderer-paint>red</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>10.0</val>
      </value>
      <value>
        <category>Category 2</category><val>20.0</val>
      </value>
      <value>
        <category>Category 3</category><val>30.0</val>
      </value>
      <value>
        <category>Category 4</category><val>40.0</val>
      </value>
      <value>
        <category>Category 5</category><val>50.0</val>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test2</label>
      <renderer-paint>blue</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>50.0</val>
      </value>
      <value>
        <category>Category 2</category><val>40.0</val>
      </value>
      <value>
        <category>Category 3</category><val>30.0</val>
      </value>
      <value>
        <category>Category 4</category><val>20.0</val>
      </value>
      <value>
        <category>Category 5</category><val>10.0</val>
      </value>
    </values>
  </serie>
  <serie>
    <properties>
      <label>Test3</label>
      <renderer-paint>green</renderer-paint>
    </properties>
    <values>
      <value>
        <category>Category 1</category><val>10.0</val>
      </value>
      <value>
        <category>Category 2</category><val>20.0</val>
      </value>
      <value>
```

```
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>20.0</val>
        </value>
        <value>
          <category>Category 5</category><val>10.0</val>
        </value>
      </values>
    </serie>
  </series>
</category-stacked-area>
```

### 6.1.18 Category Stacked Bar

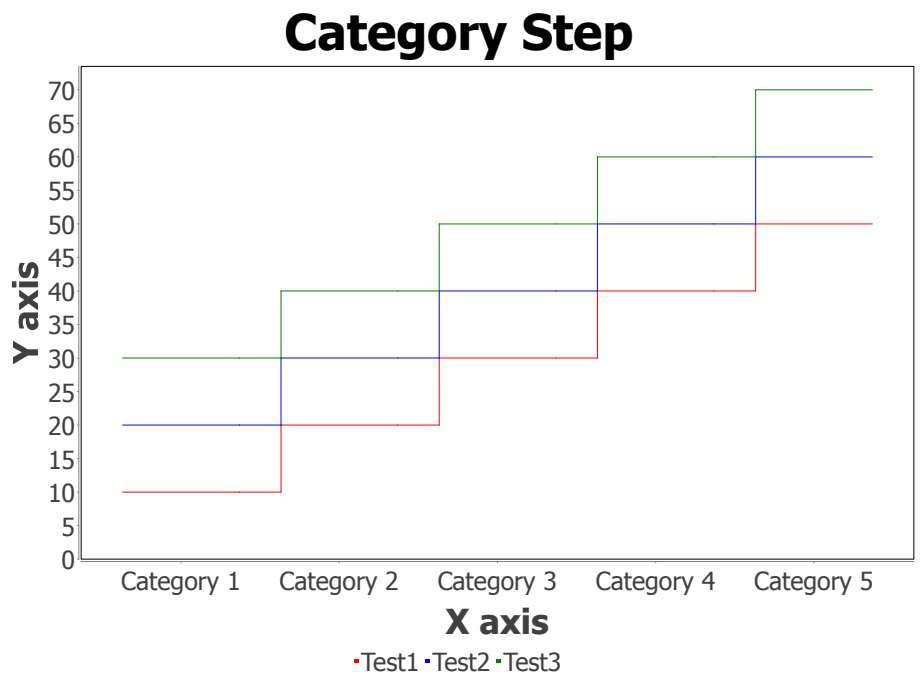



```
<category-stacked-bar xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Stacked Bar</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <orientation>vertical</orientation>
    <renderer-render-as-percentages>true</renderer-render-as-percentages>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
```

```
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>10.0</val>
        </value>
        <value>
          <category>Category 2</category><val>20.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>40.0</val>
        </value>
        <value>
          <category>Category 5</category><val>50.0</val>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>50.0</val>
        </value>
        <value>
          <category>Category 2</category><val>40.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>20.0</val>
        </value>
        <value>
          <category>Category 5</category><val>10.0</val>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test3</label>
        <renderer-paint>green</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>10.0</val>
```

```
        </value>
        <value>
          <category>Category 2</category><val>20.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>20.0</val>
        </value>
        <value>
          <category>Category 5</category><val>10.0</val>
        </value>
      </values>
    </serie>
  </series>
</category-stacked-bar>
```

### 6.1.19 Category Step

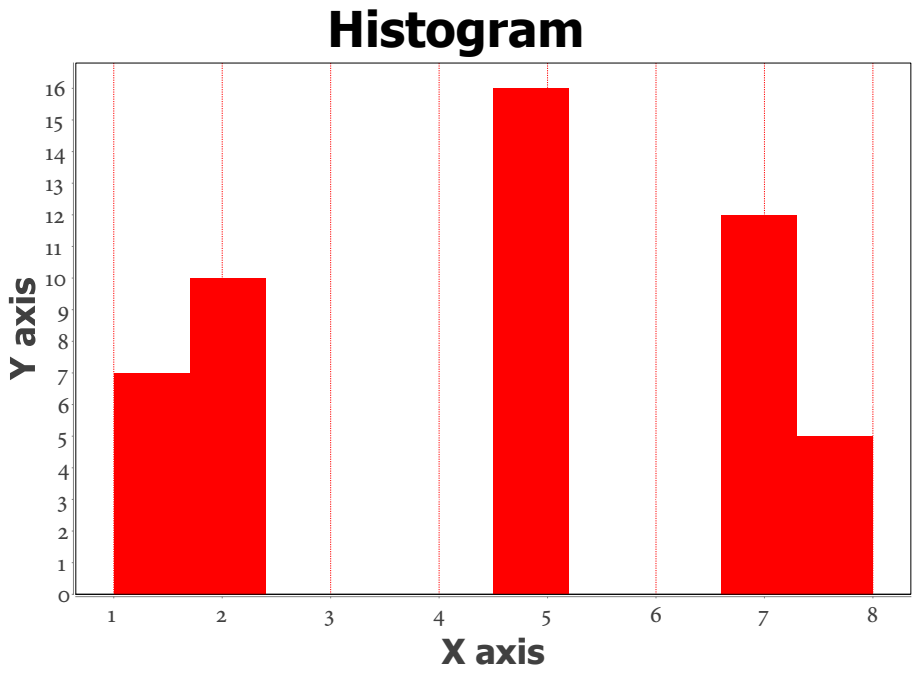



```
<category-step xmlns="urn:com-renderx:charts">
  <properties>
    <title>Category Step</title>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <orientation>vertical</orientation>
```

```
    <renderer-stagger>false</renderer-stagger>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>10.0</val>
        </value>
        <value>
          <category>Category 2</category><val>20.0</val>
        </value>
        <value>
          <category>Category 3</category><val>30.0</val>
        </value>
        <value>
          <category>Category 4</category><val>40.0</val>
        </value>
        <value>
          <category>Category 5</category><val>50.0</val>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test2</label>
        <renderer-paint>blue</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>20.0</val>
        </value>
        <value>
          <category>Category 2</category><val>30.0</val>
        </value>
        <value>
          <category>Category 3</category><val>40.0</val>
        </value>
        <value>
          <category>Category 4</category><val>50.0</val>
        </value>
        <value>
          <category>Category 5</category><val>60.0</val>
        </value>
      </values>
    </serie>
    <serie>
      <properties>
        <label>Test3</label>
```

```
        <renderer-paint>green</renderer-paint>
      </properties>
      <values>
        <value>
          <category>Category 1</category><val>30.0</val>
        </value>
        <value>
          <category>Category 2</category><val>40.0</val>
        </value>
        <value>
          <category>Category 3</category><val>50.0</val>
        </value>
        <value>
          <category>Category 4</category><val>60.0</val>
        </value>
        <value>
          <category>Category 5</category><val>70.0</val>
        </value>
      </values>
    </serie>
  </series>
</category-step>
```

### 6.1.20 Histogram

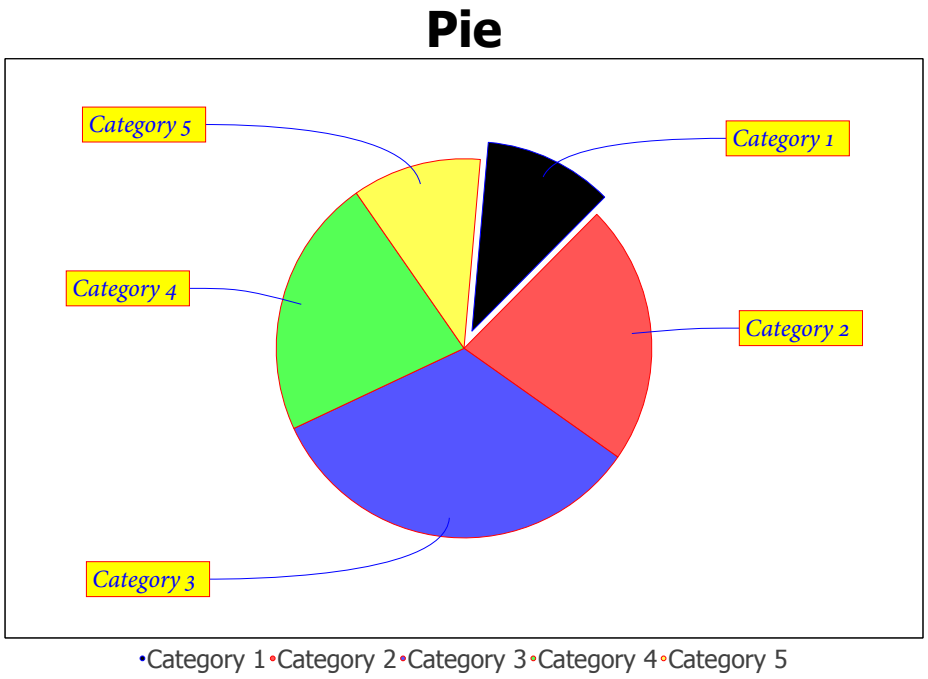



```
<histogram xmlns="urn:com-renderx:charts">
  <properties>
    <title>Histogram</title>
    <format>SVG</format>
    <width>400pt</width>
```

```
    <height>300pt</height>
    <domain-axis-label>X axis</domain-axis-label>
    <range-axis-label>Y axis</range-axis-label>
    <domain-axis-tick-unit>1.0</domain-axis-tick-unit>
  </properties>
  <theme>
    <regular-font-family>serif</regular-font-family>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
    <domain-gridline-paint>red</domain-gridline-paint>
  </theme>
  <series>
    <serie>
      <properties>
        <label>Test1</label>
        <renderer-paint>red</renderer-paint>
        <bins>10</bins>
      </properties>
      <values>
        <value>1.0</value>
        <value>1.0</value>
        <value>1.0</value>
        <value>1.0</value>
        <value>1.0</value>
        <value>1.0</value>
        <value>1.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>2.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>5.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
```

```
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>7.0</value>
        <value>8.0</value>
        <value>8.0</value>
        <value>8.0</value>
        <value>8.0</value>
        <value>8.0</value>
      </values>
    </serie>
  </series>
</histogram>
```

### 6.1.21 Pie




```
<pie xmlns="urn:com-renderx:charts">
  <properties>
    <title>Pie</title>
    <width>400pt</width>
    <height>300pt</height>
    <format>SVG</format>
    <legend>true</legend>
    <plot-base-section-outline-paint>red
</plot-base-section-outline-paint>
    <plot-base-section-paint>green</plot-base-section-paint>
    <plot-circular>false</plot-circular>
```

```
    <plot-direction>clockwise</plot-direction>
    <plot-interior-gap>0.1</plot-interior-gap>
    <plot-label-background-paint>yellow
</plot-label-background-paint>
    <plot-label-font-family>serif</plot-label-font-family>
    <plot-label-font-size>10</plot-label-font-size>
    <plot-label-font-style>italic</plot-label-font-style>
    <plot-label-gap>0.05</plot-label-gap>
    <plot-label-link-margin>0.05</plot-label-link-margin>
    <plot-label-link-paint>blue</plot-label-link-paint>
    <plot-label-link-style>cubic</plot-label-link-style>
    <plot-label-links-visible>true</plot-label-links-visible>
    <plot-label-outline-paint>red</plot-label-outline-paint>
    <plot-label-padding-top>5pt</plot-label-padding-top>
    <plot-label-padding-bottom>5pt</plot-label-padding-bottom>
    <plot-label-padding-left>10pt</plot-label-padding-left>
    <plot-label-padding-right>10pt</plot-label-padding-right>
    <plot-label-paint>blue</plot-label-paint>
    <plot-label-shadow-paint>none</plot-label-shadow-paint>
    <plot-maximum-label-width>0.2</plot-maximum-label-width>
    <plot-minimum-arc-angle-to-draw>1
</plot-minimum-arc-angle-to-draw>
    <plot-section-outlines-visible>true
</plot-section-outlines-visible>
    <plot-shadow-paint>none</plot-shadow-paint>
    <plot-shadow-x-offset>1pt</plot-shadow-x-offset>
    <plot-shadow-y-offset>1pt</plot-shadow-y-offset>
    <plot-start-angle>85</plot-start-angle>
    <plot-ignore-zero-values>true</plot-ignore-zero-values>
  </properties>
  <theme>
    <regular-font-size>10</regular-font-size>
    <plot-background-paint>white</plot-background-paint>
  </theme>
  <sections>
    <section>
      <properties>
        <explode-percent>0.1</explode-percent>
        <section-outline-paint>blue</section-outline-paint>
        <section-paint>black</section-paint>
      </properties>
      <category>Category 1</category><value>10.0</value>
    </section>
    <section>
      <category>Category 2</category><value>20.0</value>
    </section>
    <section>
      <category>Category 3</category><value>30.0</value>
    </section>
    <section>
      <category>Category 4</category><value>20.0</value>
    </section>
    <section>
      <category>Category 5</category><value>10.0</value>
    </section>
    <section>
      <category>Category 6</category><value>0.0</value>
```

```
    </section>
  </sections>
</pie>
```

## 6.2 SPECIFICATION

### 6.2.1 Definitions

*dimension*
: A number optionally followed by a unit, which can be “mm”, “cm”, “in”, “pc”, “pt” or “px”. When no unit is given “px” is assumed.

*paint*
: A colour in CSS syntax.

*percentage*
: A value between 0.0 and 1.0.

### 6.2.2 General Properties

These are the properties that can be used in the `properties` subelement of the chart description.

#### 6.2.2.1 All

| *Property* | *Default value* | *Description* |
|---|---|---|
| format | SVG | The output format, which can be SVG, PNG or JPEG. |
| height | 400pt | The height *dimension* of the resulting chart. |
| padding | 50pt | The padding *dimension* at all sides of the resulting chart. |
| padding-bottom | 50pt | The padding *dimension* at the bottom side of the resulting chart. |
| padding-left | 50pt | The padding *dimension* at the left side of the resulting chart. |
| padding-right | 50pt | The padding *dimension* at the right side of the resulting chart. |
| padding-top | 50pt | The padding *dimension* at the top side of the resulting chart. |
| plot-background-alpha | 1.0 | Sets the alpha transparency of the plot area background. Values are between 0.0 and 1.0. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| plot-background-paint | white | Sets the background *paint* of the plot area. |
| plot-foreground-alpha | 1.0 | Sets the alpha-transparency for the plot. Values are between 0.0 and 1.0. |
| plot-outline-paint | gray | Sets the *paint* used to draw the outline of the plot area. |
| plot-outline-visible | true | Sets the flag that controls whether or not the plot's outline is drawn. |
| plot-padding-bottom | 4 | Sets the bottom padding for the plot. |
| plot-padding-left | 8 | Sets the left padding for the plot. |
| plot-padding-right | 8 | Sets the right padding for the plot. |
| plot-padding-top | 4 | Sets the top padding for the plot. |
| ppi | 300 | The number of pixels per inch. This is relevant when the output format is a bitmap. |
| renderer-base-fill-paint | white | Sets the base fill *paint*. |
| renderer-base-item-label-font-family | sansserif | Sets the base item label font family. |
| renderer-base-item-label-font-size | 10 | Sets the base item label font size (*dimension*). |
| renderer-base-item-label-font-style | plain | Sets the base item label font style. |
| renderer-base-item-label-paint | plain | Sets the base item label *paint*. |
| renderer-base-item-labels-visible | false | Sets the base flag that controls whether or not item labels are visible. |
| renderer-base-legend-text-font-family | Nothing | Sets the default legend text font family. |
| renderer-base-legend-text-font-size | Nothing | Sets the default legend text font size (*dimension*). |
| renderer-base-legend-text-font-style | Nothing | Sets the default legend text font style. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| renderer-base-legend-text-paint | Nothing | Sets the default legend text *paint*. |
| use-difference | false | Uses a renderer that highlights the differences between two series. |
| use-dot | false | Uses a renderer that draws a small dot at each data point. |
| use-spline | false | Uses a renderer that connects data points with natural cubic splines and/or draws shapes at each data point. |
| width | 400pt | The width *dimension* of the resulting chart. |

#### *6.2.2.2 xy-*, category-*, histogram-**

| *Property* | *Default value* | *Description* |
|---|---|---|
| axis-offset-bottom | 0 | Sets the axis offset at the bottom of the chart. |
| axis-offset-left | 0 | Sets the axis offset at the left side of the chart. |
| axis-offset-right | 0 | Sets the axis offset at the right side of the chart. |
| axis-offset-top | 0 | Sets the axis offset at the top of the chart. |
| [domain- or range-]axis-auto-range | false | Sets a flag that determines whether or not the axis range is automatically adjusted to fit the data for the axis. |
| [domain- or range-]axis-auto-range-includes-zero | false | Sets the flag that indicates whether or not the axis range, if automatically calculated, is forced to include zero for the axis. |
| [domain- or range-]axis-auto-range-minimum-size | 0 | Sets the auto range minimum *dimension* for the axis. |
| [domain- or range-]axis-auto-range-sticky-zero | false | Sets a flag that affects the auto-range when zero falls outside the data |

| *Property* | *Default value* | *Description* |
|---|---|---|
| | | range but inside the margins defined for the axis for the axis. |
| [domain- or range-]axis-auto-tick-unit-selection | false | Sets a flag indicating whether or not the tick unit is automatically selected from a range of standard tick units for the axis. |
| [domain- or range-]axis-base | 10.0 | Sets the base for the logarithm calculation (must be > 1.0) for the domain axis. |
| [domain- or range-]axis-base-format | "0" | Sets the base format pattern for the logarithmic scale when it is displayed numerically for the axis. The possible patterns are described in `java.text.DecimalFormat`. |
| [domain- or range-]axis-base-symbol | Uses the base formatter | Sets the symbol used to represent the base value of the logarithmic scale. |
| [domain- or range-]axis-default-auto-range-lower | 0.0 | Sets the default auto lower range. |
| [domain- or range-]axis-default-auto-range-upper | 1.0 | Sets the default auto upper range. |
| [domain- or range-]axis-fixed-auto-range | 0.0 | Sets the fixed auto range for the axis. |
| [domain- or range-]axis-inverted | false | Sets a flag that controls the direction of values on the axis. |
| [domain- or range-]axis-label | Nothing | Sets the label for the axis. |
| [domain- or range-]axis-label-angle | 0.0 | Sets the angle for the label. |
| [domain- or range-]axis-label-font-family | sansserif | Sets the font family for the axis label. |
| [domain- or range-]axis-label-font-size | sansserif | Sets the font family for the axis label. |
| [domain- or range-]axis-label-font-style | plain | Sets the font style for the axis label. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| [domain- or range-]axis-label-location | middle | Sets the axis label location. Possible values are “bottom”, “middle” and “top”. |
| [domain- or range-]axis-label-padding-bottom | 3 | Sets the bottom padding for the axis label. |
| [domain- or range-]axis-label-padding-left | 3 | Sets the left padding for the axis label. |
| [domain- or range-]axis-label-padding-right | 3 | Sets the right padding for the axis label. |
| [domain- or range-]axis-label-padding-bottom | 3 | Sets the bottom padding for the axis label. |
| [domain- or range-]axis-label-paint | black | Sets the *paint* used to draw the axis label. |
| [domain- or range-]axis-line-paint | gray | Sets the *paint* used to draw the axis line. |
| [domain- or range-]axis-line-visible | true | Sets a flag that controls whether or not the axis line is visible. |
| [domain- or range-]axis-lower | 0.0 | Sets the lower bound of the range for the axis. |
| [domain- or range-]axis-lower-margin | 0.05 | Sets the lower margin for the axis (as a *percentage* of the axis range). |
| [domain- or range-]axis-lower-with-margins | 0.0 | Sets the lower bound of the range for the axis (after first adding the current margins to the specified range). |
| [domain- or range-]axis-maximum-date | Nothing | Sets the maximum date visible on the axis. If maximum date is on or before the current minimum date for the axis, the minimum date will be shifted to preserve the current length of the axis. |
| [domain- or range-]axis-minimum-date | Nothing | Sets the minimum date visible on the axis. If date is on or after the current maximum date for the axis, the maximum date will be shifted to preserve the current length of the axis. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| [domain- or range-]axis-minor-tick-count | 0 | Sets the number of minor tick marks to display. |
| [domain- or range-]axis-minor-tick-mark-inside-length | 0.0 | Sets the inside *dimension* of the minor tick marks. |
| [domain- or range-]axis-minor-tick-mark-outside-length | 2.0 | Sets the outside *dimension* of the minor tick marks. |
| [domain- or range-]axis-minor-tick-marks-visible | false | Sets the flag that indicates whether or not the minor tick marks. |
| [domain- or range-]axis-negative-arrow-visible | false | Sets a flag that controls whether or not the axis lines has an arrow drawn that points in the negative direction for the axis. |
| [domain- or range-]axis-number-format-override | Nothing | Sets the number format override. |
| [domain- or range-]axis-period(1-3)-divider-paint | gray | Sets the *paint* used to draw the period dividers. |
| [domain- or range-]axis-period(1-3)-draw-divider | true | Sets the flag that controls whether or not dividers are drawn. |
| [domain- or range-]axis-period(1-3)-font-family | sansserif | Sets the font family for the period label. |
| [domain- or range-]axis-period(1-3)-font-size | 10 | Sets the font size for the period label. |
| [domain- or range-]axis-period(1-3)-font-style | plain | Sets the font style for the period label. |
| [domain- or range-]axis-period(1-3)-format | MMM | Sets the date formatter for the period. The possible patterns are described in `java.text.SimpleDateFormat`. |
| [domain- or range-]axis-period(1-3)-padding-bottom | 2 | Sets the bottom padding for the period band. |
| [domain- or range-]axis-period(1-3)-padding-left | 2 | Sets the left padding for the period band. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| [domain- or range-]axis-period(1-3)-padding-right | 2 | Sets the right padding for the period band. |
| [domain- or range-]axis-period(1-3)-padding-top | 2 | Sets the top padding for the period band. |
| [domain- or range-]axis-period(1-3)-paint | black | Sets the *paint* of the period label. |
| [domain- or range-]axis-positive-arrow-visible | false | Sets a flag that controls whether or not the axis lines has an arrow drawn that points in the positive direction for the axis. |
| [domain- or range-]axis-range-about-value | Nothing | Sets the axis range. The value must be a comma-separated pair of values. The first is used to center the axis. The second sets the size of the axis. |
| [domain- or range-]axis-range-type | full | Sets the axis range type. Possible values are “full”, “negative” and “positive”. |
| [domain- or range-]axis-smallest-value | 1E-100 | Sets the smallest value represented by the axis. |
| [domain- or range-]axis-tick-label-font-family | sansserif | Sets the font family for the tick labels. |
| [domain- or range-]axis-tick-label-font-size | 10 | Sets the font size for the tick labels. |
| [domain- or range-]axis-tick-label-font-style | plain | Sets the font style for the tick labels. |
| [domain- or range-]axis-tick-label-padding-bottom | 2 | Sets the bottom padding for the tick labels. |
| [domain- or range-]axis-tick-label-padding-left | 4 | Sets the left padding for the tick labels. |
| [domain- or range-]axis-tick-label-padding-right | 4 | Sets the right padding for the tick labels. |
| [domain- or range-]axis-tick-label-padding-top | 2 | Sets the top padding for the tick labels. |
| [domain- or range-]axis-tick-label-paint | black | Sets the *paint* used to draw tick labels (if they are showing). |

| *Property* | *Default value* | *Description* |
|---|---|---|
| [domain- or range-]axis-tick-labels-visible | true | Sets the flag that determines whether or not the tick labels are visible. |
| [domain- or range-]axis-tick-mark-inside-length | 0 | Sets the inside *dimension* of the tick marks |
| [domain- or range-]axis-tick-mark-outside-length | 2 | Sets the outside *dimension* of the tick marks |
| [domain- or range-]axis-tick-mark-paint | gray | Sets the *paint* used to draw tick marks. |
| [domain- or range-]axis-tick-mark-position | start | Sets the tick mark position ("start", "middle" or "end" of the time period) |
| [domain- or range-]axis-tick-marks-visible | true | Sets the flag that indicates whether or not the tick marks are showing. |
| [domain- or range-]axis-tick-unit | 1.0 or "day" | Sets the tick unit for the axis. The value can be a number or one of the tokens "day", "hour", "millisecond", "minute", "month", "second" and "year". |
| [domain- or range-]axis-upper | 1.0 | Sets the upper bound of the range for the axis. |
| [domain- or range-]axis-upper-margin | 0.05 | Sets the upper margin for the axis (as a *percentage* of the axis range). |
| [domain- or range-]axis-upper-with-margins | 1.0 | Sets the upper bound of the range for the axis (after first adding the current margins to the specified range). |
| [domain- or range-]axis-use-fill-paint | false | Sets the flag that controls whether the fill paint is used to fill shapes. |
| [domain- or range-]axis-use-outline-paint | false | Sets the flag that controls whether the outline paint is used to draw shape outlines. |
| [domain- or range-]axis-use-y-interval | false | Sets the flag that determines whether the y-interval from the dataset is used to calculate the length of each bar. |
| [domain- or range-]axis-vertical-tick-labels | false | Sets the flag that controls whether the tick labels are displayed vertically |

| *Property* | *Default value* | *Description* |
|---|---|---|
| | | (that is, rotated 90 degrees from horizontal). |
| [domain- or range-]axis-visible | true | Sets a flag that controls whether or not the axis is visible. |
| plot-domain-axis-location | bottom | Sets the location of the primary domain axis. Possible values are "bottom", "left", "top" and "right". |
| plot-domain-grid-line-paint | lightgray | Sets the *paint* for the grid lines plotted against the domain axis. |
| plot-domain-grid-lines-visible | true | Sets the flag that controls whether or not the domain grid-lines are visible. |
| plot-domain-minor-grid-line-paint | white | Sets the *paint* for the minor grid lines plotted against the domain axis. |
| plot-domain-minor-grid-lines-visible | false | Sets the flag that controls whether or not the minor domain grid-lines are visible. |
| plot-domain-tick-band-paint | none | Sets the *paint* for the domain tick bands. |
| plot-domain-zero-baseline-paint | black | Sets the *paint* for the zero baseline plotted against the domain axis. |
| plot-domain-zero-baseline-visible | false | Sets the flag that controls whether or not the zero baseline is displayed for the domain axis. |
| plot-dot-height | 1 | Sets the dot height. |
| plot-dot-width | 1 | Sets the dot width. |
| plot-draw-shared-domain-axis | false | Sets the flag that controls whether the shared domain axis is drawn when this plot is being used as a subplot. |
| plot-quadrant-origin | 0.0,0.0 | Sets the quadrant origin. The value must be two comma-separated coordinates. |
| plot-quadrant(0-3)-paint | Nothing | Sets the *paint* used for the specified quadrant. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| plot-range-axis-location | left | Sets the location of the range axis. Possible values are “bottom”, “left”, “top” and “right”. |
| plot-range-grid-line-paint | lightgray | Sets the *paint* for the grid lines plotted against the range axis. |
| plot-range-grid-lines-visible | true | Sets the flag that controls whether or not the range grid-lines are visible. |
| plot-range-minor-grid-line-paint | white | Sets the *paint* for the minor grid lines plotted against the range axis. |
| plot-range-minor-grid-lines-visible | false | Sets the flag that controls whether or not the minor range grid-lines are visible. |
| plot-range-tick-band-paint | none | Sets the *paint* for the range tick bands. |
| plot-range-zero-baseline-paint | black | Sets the *paint* for the zero baseline plotted against the range axis. |
| plot-range-zero-baseline-visible | false | Sets the flag that controls whether or not the zero baseline is displayed for the range axis. |
| renderer-bar-alignment-factor | -1.0 | Sets the bar alignment factor. If the alignment factor is outside the range 0.0 to 1.0, no alignment will be performed by the renderer. |
| renderer-base | 0.0 | Sets the base value for the bars. The base value is not used if the dataset's y-interval is being used to determine the bar length. |
| renderer-base-lines-visible | true | Sets the base 'lines visible' flag. |
| renderer-base-outline-paint | gray | Sets the base outline *paint*. |
| renderer-base-paint | blue | Sets the base *paint*. |
| renderer-base-series-visible | true | Sets the base series visibility. |
| renderer-base-series-visible-in-legend | true | Sets the base visibility in the legend. |
| renderer-base-shapes-filled | true | Sets the base 'shapes filled' flag. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| renderer-base-shaped-visible | true | Sets the base 'shapes visible' flag. |
| renderer-draw-bar-outline | false | Sets the flag that controls whether or not bar outlines are drawn. |
| renderer-draw-outlines | true | Sets the flag that controls whether outlines are drawn for shapes. |
| renderer-fill-type (use-spline=true) | none | Sets the fill type. Possible values are “none”, “upper”, “lower” and “zero”. |
| renderer-item-label-anchor-offset | 2.0 | Sets the item label anchor offset. |
| renderer-negative-paint | red | Sets the *paint* used to highlight negative differences. Set `use-difference` to `true` to use this. |
| renderer-positive-paint | green | Sets the *paint* used to highlight posotive differences. Set `use-difference` to `true` to use this. |
| renderer-precision | 5 | Set the resolution of splines. The value must be greater than 0. Set `use-spline` to `true` to user this. |
| renderer-round-X-coordinates | false | Sets the flag that controls whether or not the x-coordinates (in Java2D space) are rounded to integer values. Set `use-difference` to `true` to use this. |
| renderer-series-fill-paint | white | Sets the fill *paint* for the series element in which this property is placed. |
| renderer-series-item-label-font-family | sansserif | Sets the label font family for the series element in which this property is placed. |
| renderer-series-item-label-font-size | 10 | Sets the label font size for the series element in which this property is placed. |
| renderer-series-item-label-font-style | plain | Sets the label font style for the series element in which this property is placed. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| renderer-series-item-label-paint | black | Sets the item label *paint* for the series element in which this property is placed. |
| renderer-series-item-labels-visible | false | Sets a flag that controls the visibility of the item labels for the series element in which this property is placed. |
| renderer-series-lines-visible | true | Sets a flag that controls the visibility of the lines for the series element in which this property is placed. |
| renderer-series-outline-paint | gray | Sets the outline *paint* for the series element in which this property is placed. |
| renderer-series-paint | blue | Sets the *paint* for the series element in which this property is placed. |
| renderer-series-shapes-filled | true | Sets a flag that controls the filling of the shapes for the series element in which this property is placed. |
| renderer-series-shapes-visible | true | Sets a flag that controls the visibility of the shapes for the series element in which this property is placed. |
| renderer-series-visible | true | Sets a flag that controls the visibility of the series for the series element in which this property is placed. |
| renderer-series-visible-in-legend | true | Sets a flag that controls the visibility of the series in the legend for the series element in which this property is placed. |
| renderer-shapes-filled | true | Sets the 'shapes filled' flag. |
| renderer-shaped-visible | true | Sets the 'shapes visible' flag. |

#### *6.2.2.3 category-**

| *Property* | *Default value* | *Description* |
|---|---|---|
| [domain- or range-]axis-category-label-position-offset | 4 | Sets the offset between the axis and the category labels (before label positioning is taken into account). |

| *Property* | *Default value* | *Description* |
|---|---|---|
| [domain- or range-]axis-category-label-positions | 0.0 | Sets the category label position specification for the axis. The value is the angle in radians. |
| [domain- or range-]axis-category-margin | 0.2 | Sets the category margin. The overall category margin is distributed over N-1 gaps, where N is the number of categories on the axis. Values are between 0.0 and 1.0. |
| [domain- or range-]axis-maximum-category-label-lines | 1 | Sets the maximum number of lines to use for each category label. |
| [domain- or range-]axis-maximum-category-label-width-ratio | 0.0 | Sets the maximum category label width ratio. |
| [domain- or range-]axis-sublabel-font-family | sansserif | Sets the font family for the sublabels. |
| [domain- or range-]axis-sublabel-font-size | 10 | Sets the font size for the sublabels. |
| [domain- or range-]axis-sublabel-font-style | plain | Sets the font style for the sublabels. |
| [domain- or range-]axis-sublabel-paint | black | Sets the *paint* for the sublabels. |
| [domain- or range-]axis-sublabels | Nothing | Sets the sublabels. The value must be a comma-separated list of colon-separated pairs. The first item of a pair is the category and the second is the sublabel. |
| [domain- or range-]axis-use-series-offset | false | Sets the flag that controls whether or not the x-position for each data item is offset within its category according to the series. |
| renderer-item-margin | 0.0 | Sets the item margin, which is the gap between items within a category (expressed as a *percentage* of the overall category width). |

#### 6.2.2.4 xy-area, xy-step-area

| Property | Default value | Description |
|---|---|---|
| renderer-outline | false | Sets a flag that controls whether or not outlines of the areas are drawn. |

#### 6.2.2.5 xy-step-area

| Property | Default value | Description |
|---|---|---|
| renderer-plot-area | true | Sets a flag that controls whether or not areas are drawn for each data item. |
| renderer-range-base | Nothing | Sets the value on the range axis which defines the default border of the area. |

#### 6.2.2.6 xy-scatter

| Property | Default value | Description |
|---|---|---|
| plot-paint | red | Sets the *paint* for the data points. |

#### 6.2.2.7 category-area

| Property | Default value | Description |
|---|---|---|
| renderer-end-type | taper | Sets a token that controls how the renderer draws the end points. Possible values are “level”, “taper” and “truncate”. |

#### 6.2.2.8 category-step

| Property | Default value | Description |
|---|---|---|
| renderer-stagger | false | Sets the flag that controls whether or not the series steps are staggered. |

#### 6.2.2.9 xy-bar, category-bar. category-stacked-bar

| *Property* | *Default value* | *Description* |
|---|---|---|
| renderer-item-margin | 0.2 | Sets the item margin. The value is expressed as a *percentage* of the available width for plotting all the bars, with the resulting amount to be distributed between all the bars evenly. |
| renderer-maximum-bar-width | 1.0 | Sets the maximum bar width, which is specified as a *percentage* of the available space for all bars. |
| renderer-minimum-bar-length | 2 | Sets the minimum bar *dimension*. It can be used to prevent bars that represent very small data values from disappearing when drawn on the screen. Typically you would set this to (say) 0.5 or 1.0. Use this attribute with caution, however, because setting it to a non-zero value will artificially increase the length of bars representing small values, which may misrepresent your data. |
| renderer-shadow-paint | gray | Sets the shadow *paint*. |
| renderer-shadow-visible | true | Sets the shadow visibility flag. |
| renderer-shadow-x-offset | 4.0 | Sets the x-offset for the shadow effect. |
| renderer-shadow-y-offset | 4.0 | Sets the y-offset for the shadow effect. |

#### 6.2.2.10 xy-bar

| *Property* | *Default value* | *Description* |
|---|---|---|
| renderer-margin | 0.0 | Sets the *percentage* amount by which the bars are trimmed. |

#### 6.2.2.11 xy-step, xy-step-area

| Property | Default value | Description |
|---|---|---|
| renderer-step-point | 1.0 | Sets the step point. Values are between 0.0 and 1.0. |

#### 6.2.2.12 category-stacked-bar

| Property | Default value | Description |
|---|---|---|
| renderer-render-as-percentages | false | Sets the flag that controls whether the renderer displays each item value as a percentage (so that the stacked bars add to 100%). |

#### 6.2.2.13 pie

| Property | Default value | Description |
|---|---|---|
| plot-base-section-outline-paint | gray | Sets the base section outline *paint*. |
| plot-base-section-paint | gray | Sets the base section *paint*. |
| plot-circular | true | A flag indicating whether the pie chart is circular, or stretched into an elliptical shape. |
| plot-direction | clockwise | Sets the direction in which the pie sections are drawn. Possible values are "clockwise" and "anticlockwise". |
| plot-ignore-zero-values | false | Sets a flag that controls whether zero values are ignored. This only affects whether or not a label appears for the non-visible pie section. |
| plot-interior-gap | 0.08 | Sets the interior gap. This controls the space between the edges of the pie plot and the plot area itself (the region where the section labels appear). Values are between 0.0 and 1.0. |
| plot-label-background-paint | rgb(255, 255, 255, 192) | Sets the section label background *paint*. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| plot-label-font-family | sansserif | Sets the section label font family. |
| plot-label-font-size | 10 | Sets the section label font size. |
| plot-label-font-style | plain | Sets the section label font style. |
| plot-label-gap | 0.025 | Sets the gap between the edge of the pie and the labels (expressed as a *percentage* of the plot width). |
| plot-label-link-margin | 0.025 | Sets the link margin. Values are between 0.0 and 1.0. |
| plot-label-link-paint | black | Sets the *paint* used for the lines that connect pie sections to their corresponding labels. |
| plot-label-link-style | standard | Sets the label link style. Possible values are "standard", "quad" and "cubic". |
| plot-label-links-visible | true | Sets the flag that controls whether or not label linking lines are visible. Please take care when hiding the linking lines - depending on the data values, the labels can be displayed some distance away from the corresponding pie section. |
| plot-label-outline-paint | black | Sets the section label outline *paint*. |
| plot-label-padding-bottom | 2 | Sets the bottom padding between each label and its outline. |
| plot-label-padding-left | 2 | Sets the left padding between each label and its outline. |
| plot-label-padding-right | 2 | Sets the right padding between each label and its outline. |
| plot-label-padding-top | 2 | Sets the top padding between each label and its outline. |
| plot-label-paint | black | Sets the section label *paint*. |
| plot-label-shadow-paint | rgb(151, 151, 151, 128) | Sets the section label shadow *paint*. The value "none" is allowed. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| plot-maximum-label-width | 0.14 | Sets the maximum label width as a *percentage* of the plot. |
| plot-minimum-acr-angle-to-draw | 0.00001 | Sets the minimum arc angle that will be drawn. The angle is in radians. |
| plot-shadow-paint | gray | Sets the shadow *paint*. The value can be "none". |
| plot-shadow-x-offset | 4.0 | Sets the x-offset for the shadow effect. |
| plot-shadow-y-offset | 4.0 | Sets the y-offset for the shadow effect. |
| plot-simple-label-offset-bottom | 0.18 | Sets the offset for the simple bottom label. |
| plot-simple-label-offset-left | 0.18 | Sets the offset for the simple left label. |
| plot-simple-label-offset-right | 0.18 | Sets the offset for the simple right label. |
| plot-simple-label-offset-top | 0.18 | Sets the offset for the simple top label. |
| plot-simple-labels | false | Sets the flag that controls whether simple or extended labels are displayed on the plot. |
| plot-start-angle | 90 | Sets the starting angle in degrees. |
| section-outline-paint | gray | Sets the outline *paint* associated with the section element in which this property is placed. |
| section-outlines-visible | true | Sets the flag that controls whether or not the outline is drawn for each pie section. |
| section-paint | gray | Sets the *paint* associated with the section element in which this property is placed. |

### *6.2.3 Theme Properties*

These are the properties that can be used in the `theme` subelement of the chart description. Themes are collections of properties that are not directly related to a chart.

You can use the same theme for several charts. In practice you include them from a separate entity.

#### *6.2.3.1 All*

| *Property* | *Default value* | *Description* |
|---|---|---|
| baseline-paint | white | Sets the baseline *paint*. |
| chart-background-paint | black | Sets the chart background *paint*. |
| crosshair-paint | red | Sets the crosshair *paint*. |
| error-indicator-paint | lightgray | Sets the error indicator *paint*. |
| extra-large-font-family | Tahoma | Sets the largest font family for this theme. |
| extra-large-font-size | 20 | Sets the largest font size for this theme. |
| extra-large-font-style | bold | Sets the largest font bold for this theme. |
| item-label-paint | white | Sets the item label *paint*. |
| large-font-family | Tahoma | Sets the large font family for this theme. |
| large-font-size | 14 | Sets the large font size for this theme. |
| large-font-style | bold | Sets the large font bold for this theme. |
| legend-background-paint | black | Sets the legend background *paint*. |
| legend-item-paint | white | Sets the legend item *paint*. |
| plot-background-paint | black | Sets the plot background *paint*. |
| plot-outline-paint | yellow | Sets the plot outline *paint*. |
| range-gridline-paint | white | Sets the range grid line *paint*. |
| regular-font-family | Tahoma | Sets the regular font family for this theme. |
| regular-font-size | 12 | Sets the regular font size for this theme. |
| regular-font-style | bold | Sets the regular font bold for this theme. |

| *Property* | *Default value* | *Description* |
|---|---|---|
| shadow-paint | darkgray | Sets the shadow *paint*. |
| shadow-visible | false | Sets the shadow visibility flag. |
| small-font-family | Tahoma | Sets the small font family for this theme. |
| small-font-size | 10 | Sets the small font size for this theme. |
| small-font-style | bold | Sets the small font bold for this theme. |
| subtitle-paint | white | Sets the subtitle *paint*. |
| thermometer-paint | white | Sets the thermometer *paint*. |
| tick-label-paint | white | Sets the tick label *paint*. |
| title-paint | white | Sets the title *paint*. |

#### *6.2.3.2 xy-*, category-*, histogram-**

| *Property* | *Default value* | *Description* |
|---|---|---|
| axis-label-paint | darkgray | Sets the axis label *paint*. |
| domain-grid-line-paint | white | Sets the domain grid line *paint*. |
| grid-band-alternate-paint | rgb(0, 0, 0, 0) (transparant) | Sets the grid band alternate *paint*. |
| grid-band-paint | rgb(232, 234, 232, 128) | Sets the grid band *paint*. |
| range-grid-line-paint | white | Sets the range grid line *paint*. |

#### *6.2.3.3 pie*

| *Property* | *Default value* | *Description* |
|---|---|---|
| label-link-paint | lightgray | Sets the label link *paint* for pie charts. |
| label-link-style | standard | Sets the label link style for pie charts. Possible values are “standard”, “cubic” and “quad”. |

### 6.2.4 XML Structure

A chart is an XML element. The top element indicates which chart will be generated. The following elements are currently available: `category-area`, `category-bar`, `category-line`, `category-stacked-area`, `category-stacked-bar`, `category-step`, `histogram`, `pie`, `xy-area`, `xy-bar`, `xy-bubble`, `xy-scatter`, `xy-stacked-area`, `xy-step`, `xy-step-area` and `xy-time-series`.

The subelement `properties` contains all the general properties that will be applied to the chart. The subelement `theme` contains theme properties for the chart. The other elements constitute the data. For certain charts the form of the data will have an influence on how the chart will be rendered. The remainder of this section will discuss the structure of the data for all the currently supported charts.

#### 6.2.4.1 Common

```
<!ELEMENT properties ANY>
<!ELEMENT theme ANY>
<!ELEMENT series (serie*)>
<!ELEMENT serie (properties?, values)>
<!ELEMENT values (value*)>
```

#### 6.2.4.2 category-area

```
<!ELEMENT category-area (properties?, theme?, series)>
<!ELEMENT value (category, val)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT val (#PCDATA)>
```

#### 6.2.4.3 category-bar

```
<!ELEMENT category-bar (properties?, theme?, series)>
<!ELEMENT value (category, start, end)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT start (#PCDATA)>
<!ELEMENT end (#PCDATA)>
```

#### 6.2.4.4 category-line

```
<!ELEMENT category-line (properties?, theme?, series)>
<!ELEMENT value (category, val)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT val (#PCDATA)>
```

#### 6.2.4.5 category-stacked-area

```
<!ELEMENT category-stacked-area (properties?, theme?, series)>
<!ELEMENT value (category, val)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT val (#PCDATA)>
```

#### *6.2.4.6 category-stacked-bar*

```
<!ELEMENT category-stacked-bar (properties?, theme?, series)>
<!ELEMENT value (category, val)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT val (#PCDATA)>
```

#### *6.2.4.7 category-step*

```
<!ELEMENT category-step (properties?, theme?, series)>
<!ELEMENT value (category, val)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT val (#PCDATA)>
```

#### *6.2.4.8 category-step*

```
<!ELEMENT category-step (properties?, theme?, series)>
<!ELEMENT value (category, val)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT val (#PCDATA)>
```

#### *6.2.4.9 histogram*

```
<!ELEMENT histogram (properties?, theme?, series)>
<!ELEMENT value (#PCDATA)>
```

#### *6.2.4.10 pie*

```
<!ELEMENT pie (properties?, theme?, sections)>
<!ELEMENT sections (section*)>
<!ELEMENT section (properties?, category, value)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT value (#PCDATA)>
```

#### *6.2.4.11 xy-area*

```
<!ELEMENT xy-area (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.12 xy-bar*

```
<!ELEMENT xy-bar (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.13 xy-bubble*

```
<!ELEMENT xy-bubble (properties?, theme?, series)>
<!ELEMENT value (x, y, z)>
<!ELEMENT x (#PCDATA)>
```

```
<!ELEMENT y (#PCDATA)>
<!ELEMENT z (#PCDATA)>
```

#### *6.2.4.14 xy-line*

```
<!ELEMENT xy-line (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.15 xy-scatter*

```
<!ELEMENT xy-scatter (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.16 xy-stacked-area*

```
<!ELEMENT xy-stacked-area (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.17 xy-step*

```
<!ELEMENT xy-step (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.18 xy-step-area*

```
<!ELEMENT xy-step-area (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

#### *6.2.4.19 xy-time-series*

```
<!ELEMENT xy-time-series (properties?, theme?, series)>
<!ELEMENT value (x, y)>
<!ELEMENT x (#PCDATA)>
<!ELEMENT y (#PCDATA)>
```

# EMBEDDING IN AN APPLICATION 7

## 7.1 API SPECIFICATION

The API is specified at http://www.pincette.biz/css2xslfo/api/.

## 7.2 EXAMPLES

Since CSSToXSLFOFilter is derived from org.xml.sax.helpers.XMLFilterImpl., it implements all SAX event interfaces, as well as org.xml.sax.XMLFilter. As a consequence, the filter can occur in input and output filter chains.

### *7.2.1 Example 1*

The most straight-forward scenario is an application that reads the input document from a file and that writes an XSL-FO document into another file. For this we need an XML parser that can produce SAX events. The parser implements the org.xml.sax.XMLReader interface, so we can make it the parent of CSSToXSLFOFilter.

In order to create a parser, we first have to set up the parser factory and make it namespace-aware. This happens at the lines 6 through 8. The filter can now be created with the input document as the base URL (in case any relative URLs need to be resolved) and an XML parser as its parent. This is done at the lines 9 through 14.

We now have to prepare the output part. We use an XSLT transformer without a style sheet to copy the SAX events to the output. The transformer must be in a form that accepts SAX events. This is why a javax.xml.transform.sax.TransformerHandler is created at lines 15 through 19. It implements the org.xml.sax.ContentHandler interface. By giving it the output file as a result (lines 20 through 26), the SAX events are transformed in the XML syntax.

The input and output parts can now be connected by setting the content handler of the filter to the transformer handler (line 27). The whole chain is then activated by calling the `parse` method, passing it the input document in the form of a file. The filter will pass this call onto the parser, which is its parent. The parser starts producing SAX events that go through the filter and into the transformer handler.

```
1 public class Example1
2 {
3   public static void
4   main(final String[] args) throws Exception
5   {
6     final javax.xml.parsers.SAXParserFactory factory =
7       javax.xml.parsers.SAXParserFactory.newInstance();

8     factory.setNamespaceAware(true);
```

```
 9     final net.pincette.css.CSSToXSLFOFilter filter =
10       new net.pincette.css.CSSToXSLFOFilter
11       (
12         new java.io.File(args[0]).toURL(), // base URL.
13         factory.newSAXParser().getXMLReader()
14       );

15     final javax.xml.transform.sax.TransformerHandler handler =
16       (
17         (javax.xml.transform.sax.SAXTransformerFactory)
18           javax.xml.transform.TransformerFactory.newInstance()
19       ).newTransformerHandler();

20     handler.setResult
21     (
22       new javax.xml.transform.stream.StreamResult
23       (
24         new java.io.File(args[1])
25       )
26     );

27     filter.setContentHandler(handler);

28     filter.parse
29     (
30       new org.xml.sax.InputSource
31       (
32         new java.io.FileInputStream(args[0])
33       )
34     );
35   }
36 }
```

### *7.2.2 Example 2*

A variation of the previous example is to perform the transformation of the SAX events coming out of the filter to XML syntax in another way. In the previous example the parser had the control flow and the transformer acted as a handler of SAX events. We can also give the control flow to a transformer that reads the input and copies it to the output, because we don't give it any style sheet. We need to create a javax.xml.transform.Transformer. It is done at lines 15 through 17. The actual transformation is launched at lines 18 through 32. For this to work, we have to wrap our filter in a javax.xml.transform.sax.SAXSource. For the transformer it is as if it is going to call an XML parser.

```
 1 public class Example2
 2 {
 3   public static void
 4   main(final String[] args) throws Exception
 5   {
 6     final javax.xml.parsers.SAXParserFactory factory =
 7       javax.xml.parsers.SAXParserFactory.newInstance();

 8     factory.setNamespaceAware(true);
```

```
 9     final net.pincette.css.CSSToXSLFOFilter filter =
10       new net.pincette.css.CSSToXSLFOFilter
11       (
12         new java.io.File(args[0]).toURL(), // base URL.
13         factory.newSAXParser().getXMLReader()
14       );

15     final javax.xml.transform.Transformer transformer =
16       javax.xml.transform.TransformerFactory.newInstance().
17         newTransformer();

18     transformer.transform
19     (
20       new javax.xml.transform.sax.SAXSource
21       (
22         filter, // Acts as the XMLReader.
23         new org.xml.sax.InputSource
24         (
25           new java.io.FileInputStream(args[0])
26         )
27       ),
28       new javax.xml.transform.stream.StreamResult
29       (
30         new java.io.File(args[1])
31       )
32     );
33   }
34 }
```

### *7.2.3 Example 3*

This example shows how a pre-processing step can be added to the filter chain. The input document is transformed by the pre-processor and the resulting SAX events go through the conversion filter. The pre-processor is created at lines 9 through 13. This one does nothing, i.e. it lets the events go through unmodified. In reality you would replace it with a class of your own.

The pre-processor instead of the filter is now initialised with the XML parser as its parent. The pre-processor will become the parent of the filter, as shown at line 18. When the `parse` method is called, the filter passes the call onto the pre-processor, which in turn passes it onto the parser. The SAX events produced by the parser will then flow through the pre-processor, which in turn forwards them, possibly modified, to the filter.

```
 1 public class Example3
 2 {
 3   public static void
 4   main(final String[] args) throws Exception
 5   {
 6     final javax.xml.parsers.SAXParserFactory factory =
 7       javax.xml.parsers.SAXParserFactory.newInstance();

 8     factory.setNamespaceAware(true);
```

```
 9     final org.xml.sax.helpers.XMLFilterImpl myPreprocessor =
10       new org.xml.sax.helpers.XMLFilterImpl
11       (
12         factory.newSAXParser().getXMLReader()
13       );

14     final net.pincette.css.CSSToXSLFOFilter filter =
15       new net.pincette.css.CSSToXSLFOFilter
16       (
17         new java.io.File(args[0]).toURL(), // base URL.
18         myPreprocessor
19       );

20     final javax.xml.transform.sax.TransformerHandler handler =
21       (
22         (javax.xml.transform.sax.SAXTransformerFactory)
23           javax.xml.transform.TransformerFactory.newInstance()
24       ).newTransformerHandler();

25     handler.setResult
26     (
27       new javax.xml.transform.stream.StreamResult
28       (
29         new java.io.File(args[1])
30       )
31     );

32     filter.setContentHandler(handler);

33     filter.parse
34     (
35       new org.xml.sax.InputSource
36       (
37         new java.io.FileInputStream(args[0])
38       )
39     );
40   }
41 }
```

### *7.2.4 Example 4*

The previous example can be modified in such a way that the pre-processor is an XSLT style sheet. From this style sheet a org.xml.sax.XMLFilter must be made, because it will sit between the XML parser and the filter. This is shown at lines 9 through 19. The transformer factory is re-used afterwards to create also the output handler.

```
 1 public class Example4
 2 {
 3   public static void
 4   main(final String[] args) throws Exception
 5   {
 6     final javax.xml.parsers.SAXParserFactory factory =
 7       javax.xml.parsers.SAXParserFactory.newInstance();

 8     factory.setNamespaceAware(true);
```

```
 9     final javax.xml.transform.sax.SAXTransformerFactory trFactory =
10       (javax.xml.transform.sax.SAXTransformerFactory)
11         javax.xml.transform.TransformerFactory.newInstance();

12     final org.xml.sax.XMLFilter myPreprocessor =
13       trFactory.newXMLFilter
14       (
15         new javax.xml.transform.stream.StreamSource
16         (
17           new java.io.File(args[2])
18         )
19       );

20     myPreprocessor.setParent
21     (
22       factory.newSAXParser().getXMLReader()
23     );

24     final net.pincette.css.CSSToXSLFOFilter filter =
25       new net.pincette.css.CSSToXSLFOFilter
26       (
27         new java.io.File(args[0]).toURL(), // base URL.
28         myPreprocessor
29       );

30     final javax.xml.transform.sax.TransformerHandler handler =
31       trFactory.newTransformerHandler();

32     handler.setResult
33     (
34       new javax.xml.transform.stream.StreamResult
35       (
36         new java.io.File(args[1])
37       )
38     );

39     filter.setContentHandler(handler);

40     filter.parse
41     (
42       new org.xml.sax.InputSource
43       (
44         new java.io.FileInputStream(args[0])
45       )
46     );
47   }
48 }
```

### *7.2.5 Example 5*

In all previous examples we have been parsing an input document. In some applications, however, the data might come from somewhere else. It is possible, for example, to synthesize the XML from data that resides in the database. In such a scenario our filter no longer has a parent but becomes the SAX event handler of some system

method, `generateReport` in this example.[1] This system method has the control flow. It fetches the data and generates the SAX events. In the case the generated XML stream is not suitable for CSS conversion, a pre-processor may be specified as the parent of the filter.

```
 1 public class Example5
 2 {
 3   public static void
 4   main(final String[] args) throws Exception
 5   {
 6     final net.pincette.css.CSSToXSLFOFilter filter =
 7       new net.pincette.css.CSSToXSLFOFilter();

 8     final javax.xml.transform.sax.TransformerHandler handler =
 9       (
10         (javax.xml.transform.sax.SAXTransformerFactory)
11           javax.xml.transform.TransformerFactory.newInstance()
12       ).newTransformerHandler();

13     handler.setResult
14     (
15       new javax.xml.transform.stream.StreamResult
16       (
17         new java.io.File(args[0])
18       )
19     );

20     filter.setContentHandler(handler);
21     generateReport(filter);
22   }

23   private static void
24   generateReport(final org.xml.sax.ContentHandler handler)
25   {
26   }
27 }
```

### *7.2.6 Example 6*

It may be the case that you want to synthesize the XML stream in a system method, which needs the control flow, but that the interface of your XSL-FO formatter is such that it also needs the control flow. In other words, the formatter is not available in the form of a SAX event handler, but has some method that must be called to perform the actual formatting. At lines 21 through 31 there a hypothetical example of such a formatter.

To solve this control flow conflict you can create an adapter that implements the org.xml.sax.XMLReader interface. Instead of actually parsing some XML you let both `parse` methods call your system method. The parameters the latter needs are passed through the constructor of the adapter. When the formatter now calls the `parse` method it really ends up calling the system method, which synthesizes the SAX events.

1 Note that a real system method would probably need more than just the filter to do its work. It would therefore have more parameters.

```
 1 public class Example6
 2 {
 3   public static void
 4   main(final String[] args) throws Exception
 5   {
 6     final net.pincette.css.CSSToXSLFOFilter filter =
 7       new net.pincette.css.CSSToXSLFOFilter
 8       (
 9         new MyReportGenerator(new Object())
10       );
11     final MyXSLFOFormatter myFormatter = new MyXSLFOFormatter();

12     myFormatter.format
13     (
14       new javax.xml.transform.sax.SAXSource(filter, null),
15       new java.io.FileOutputStream(args[0]));
16   }

17   private static void
18   generateReport(final Object context)
19   {
20   }

21   public static class MyXSLFOFormatter
22   {
23     public void
24     format
25     (
26       javax.xml.transform.Source source,
27       java.io.OutputStream out
28     )
29     {
30     }
31   }

32   public static class MyReportGenerator
33     extends org.xml.sax.helpers.XMLFilterImpl
34   {
35     private Object context;

36     public
37     MyReportGenerator(final Object context)
38     {
39       this.context = context;
40     }

41     public void
42     parse(final org.xml.sax.InputSource input)
43       throws org.xml.sax.SAXException, java.io.IOException
44     {
45       generateReport(context);
46     }

47     public void
48     parse(final String systemId)
49       throws org.xml.sax.SAXException, java.io.IOException
50     {
```

```
51        generateReport(context);
52      }
53    }
54 }
```

# SOME TECHNIQUES 8

A few practical cases of formatting contructs, which are either more advanced or not yet very common, are described in this chapter. Gradually, new cases will be added. The chapter is some sort of "how to" section in the user guide. The examples use XHTML as the input document language.

## 8.1 CUSTOMISING LIST LABELS WITH MARKERS

The genaration of the labels of an itemised list is somewhat fixed. It depends on the value of the `list-style-type` property.[1] Sometimes, however, more control is required over how the labels look like. This can be achieved through markers.

Basically, you have to specify a `:before` pseudo element with the display type `marker` in your style sheet for those elements you have given the display type `list-item`. Strictly speaking that display type is not needed, but if you are about to convert your existing lists, those elements would have that display type.

In the pseudo element you have control over the formatting of the label. The only exception is that the `width` property must be fixed. The tool doesn't support the automatic calculation of the required width. If your style sheet doesn't specify a width, a default value will be used. In order to not depend on this value, it is best to specify one.

The following example is an ordered list with a nested ordered list in the second item. We are going to change the numbering as well as the alignment of the labels.

```
<ol>
  <li>Item 1</li>
  <li>Item 2
    <ol>
      <li>Subitem 1</li>
      <li>Subitem 2</li>
      <li>Subitem 3</li>
    </ol>
  </li>
  <li>Item 3</li>
</ol>
```

In the style sheet we say that the `:before` pseudo element of any `li` under a `ol`, no matter the level, is a marker. In there, we increment the counter that is reset for each level of `ol`. We also display it with the `lower-roman` counter style instead of the default style (`decimal`). This style will show the effect of the right alignment of

1 The `list-style-image` and `list-style-position` properties are not supported by this tool.

the text inside the label. The `marker-offset` property provides for a bit of space between the label and the list item body.

The `width` property deserves special attention. First of all it defines the width of the labels. Since markers shouldn't influence the positioning of the element they are attached to, the labels would stick out to the left by the amount of the value of the `width` property. In order to compensate this, we have to add a `margin-left` with the same value to the list item itself.

```
ol { counter-reset: list-counter; }

ol li { margin-left: 2em; }

ol li:before
{
  content: counter(list-counter, lower-roman) ".";
  counter-increment: list-counter;
  display: marker;
  marker-offset: 0.5em;
  text-align: right;
  width: 2em;
}
```

The rendered result would like this:

i. Item 1

ii. Item 2

    ii. Subitem 1

    iv. Subitem 2

    vi. Subitem 3

iii. Item 3

## 8.2 MAKING SECTION NUMBERS "STICK OUT"

Sometimes the text of the section titles must be aligned with the rest of the material, at the left side for example. As consequence, if the titles also have section numbers, those will stick out at the left side of the title, into the margin, just like the title of the current section. This can be obtained by specifying a `:before` pseudo element for the section titles with the display type `marker`. Because markers shouldn't influence the positioning of their associated element, the marker content is prepended. This is the piece of style sheet you would need:

```
h2:before
{
  display: marker;
  marker-offset: 0.5em;
  padding-right: 0pt;
  text-align: right;
  width: 3em;
}
```

## 8.3 THIS GUIDE'S PAGE SET-UP

The page set-up of this guide is rather advanced and is therefore an interesting practical case. The difficulty lies in specifying the margin boxes if there are many kinds of pages and if for each of those the margin boxes are different, i.e. very specific.

In order to avoid an explosion of CSS property specifications for all those regions, we can work in a sort of multidimensional way. This is possible through the CSS cascading mechanism in the page context and the differentiation of pages in this context. These are the general page rules:

```
@page
{
  margin-bottom: 35mm;
  margin-top: 35mm;
}

@page :left
{
  margin-left: 40mm;
  margin-right: 40mm;

  @bottom-left
  {
    padding-top: 2em;
    vertical-align: top;
  }

  @top-left
  {
    content: string(chapter);
    font-family: serif-title;
    font-style: italic;
    padding-bottom: 2em;
    text-align: left;
    vertical-align: bottom;
  }
}

@page :right
{
  margin-left: 40mm;
  margin-right: 40mm;

  @bottom-right
  {
    padding-top: 2em;
    vertical-align: top;
  }

  @top-right
  {
    content: string(chapter);
    font-family: serif-title;
    font-style: italic;
    padding-bottom: 2em;
```

```
      text-align: right;
      vertical-align: bottom;
    }
  }
```

For each page there is a chapter title at the top and an area that will receive the page number at the bottom. The page number will be different for the front matter and the rest. This is defined as follows:

```
  @page front:left
  {
    @bottom-left
    {
      content: counter(page, lower-roman);
    }
  }

  @page front:right
  {
    @bottom-right
    {
      content: counter(page, lower-roman);
    }
  }

  @page main:left
  {
    @bottom-left
    {
      content: counter(page);
    }
  }

  @page main:right
  {
    @bottom-right
    {
      content: counter(page);
    }
  }
```

Now we want to make sure that chapters always begin on a right page, which we achieve by forcing a chapter to have an even number of pages. Moreover, the front and main matter should restart page numbering from 1. For the main matter we need a special page, called `main-first` here, because this should only happen for the first chapter. This special page also needs its page number and we know it will appear on the right side. Here is how it is done:

```
  @page main
  {
    force-page-count: even;
  }

  @page front
  {
```

```
  counter-reset: page 1;
  force-page-count: even;
}

@page main-first
{
  counter-reset: page 1;
  force-page-count: even;

  @bottom-right
  {
    content: counter(page);
  }
}
```

Because of the even pages a blank page will be inserted each time a chapter occupies an odd number of pages. A blank page shouldn't have header and footer material so we have to overwrite this. The general rule for blank pages only eclipses the header, because the general rules for left and right pages only provides content for the header. For the named pages we have to repeat the eclipsing margin boxes for the footer, because a named page rule is stronger than a general rule. Here is an example:

```
@page :blank
{
  @top-left
  {
    content: none;
  }

  @top-right
  {
    content: none;
  }
}

@page main:blank
{
  @bottom-left
  {
    content: none;
  }

  @bottom-right
  {
    content: none;
  }
}
```

There is one more special construct left to discuss: the absence of top margin boxes on the first page of a chapter. As with blank pages they are not really absent. They are merely made empty. This empty margin box is assigned to the `first` pseudo page. All chapters are however in the named page sequence `main`. If we do nothing only the first page of the first chapter will have an empty top margin box. We therefore should toggle the `page` property without adding extra pages. This can be achieved by inserting an empty `div` element between the chapters with the class `separator`.

The page assignment for that class is `separator`. This named page is not used for anything else. Since the element is empty no page sequence is generated. The next `main` element, however, will start a new page sequence. This is the separator page:

```
@page separator
{
}
```

## 8.4 A TWO-COLUMN ARTICLE

Many articles and papers are formatted in two column mode. The title, abstract, authors, etc. are usually displayed across the two columns. With two extension properties it is possible to do this. The `column-count` should be set to "2" in the page context. The title material can be wrapped in a block element for which the `column-span` property is set to "all".

## 8.5 INITIAL CAPITALS

A typographical effect that is often used are initial capitals. It consists of making the first letter of an article or chapter stand out by rendering it bigger and perhaps in another font and/or colour. In CSS this is supported through the `:first-letter` pseudo element, which is described in section 5.12.2 of [CSS2]. In CSSTOXSLFO it is implemented with the restriction that letter combinations, which are considered as one letter, are not examined. In case you need that, you can always use the Unicode ligature characters instead.

The technique was applied to the previous paragraph using the piece of style sheet below. Note the second deviation from the specification being the usage of the property `vertical-align` while the `float` property has the value `none`. It is allowed in CSSTOXSLFO because otherwise we have no control over the alignment of the first letter with the lines next to it. This depends on the font and will always require some trial and error in order to get it right. The values for the other properties are obtained in the same way. In fact, for this special case, we work around the normal way a glyph is layed out in a line.

```
p:first-letter
{
  font-family: serif-swash;
  font-size: 46pt;
  font-style: italic;
  float: left;
  line-height: 46pt;
  padding-right: 6pt;
  margin-bottom: -12pt;
  vertical-align: 9pt;
}
```

# SPECIAL PROVISIONS FOR XHTML

# A

While the tool works for any XML vocabulary it does a number of things for XHTML specifically. Other vocabularies may be supported in the same way at some later stage. The items are the following:

- Non-CSS presentational hints are translated to the corresponding CSS rules, as prescribed in section 6.4.4 of [CSS2];
- The `lang` attribute is honored;
- Hyperlinks are recognized and translated in XSL-FO links;
- The `link` element can be used to specify external style sheets;
- Style sheets can be embedded with the `style` element;
- The `style` attribute is honored;
- The `img` element is interpreted and processed;
- The `link` attribute of the `body` element is supported;
- The `meta` elements are converted to meta-data for XEP, Antenna House and FOP (XML);
- The `html-header-mark` user agent parameter is available;
- There is a user agent style sheet for XHTML that cascades against the one in appendix A of [CSS2].

# THE USER AGENT STYLE SHEET

# B

## B.1 XHTML

```
@import "xhtml.css";
@namespace url(http://www.w3.org/1999/xhtml);

@media print
{
  a[href]
  {
    color: blue;
    link: attr(href);
    text-decoration: none;
  }

  a[name]
  {
    anchor: name;
  }

  blockquote, dl, ol, p, ul
  {
    margin: 0.83em 0pt;
  }

  blockquote
  {
    margin-left: 3em;
    margin-right: 3em;
  }

  body
  {
    font-family: serif;
    padding: 0pt;
    region: body;
  }

  body:lang(da)
  {
    quotes: "\00BB" "\00AB";
  }

  body:lang(de-DE), body:lang(de-AT)
  {
    quotes: "\201E" "\201C" "\201A" "\2018"
  }
```

```
body, body:lang(en), body:lang(es)
{
  quotes: "\201C" "\201D" "\2018" "\2019";
}

body:lang(fr)
{
  quotes: "\00AB " " \00BB" "\2039 " " \203A";
}

body:lang(it)
{
  quotes: "\00AB " " \00BB";
}

body:lang(nl)
{
  quotes: "\201D" "\201D" "\2019" "\2019";
}

body:lang(no), bodylang:(pt), body:lang(de-CH)
{
  quotes: "\00AB" "\00BB" "\2039" "\203A"
}

body:lang(sv)
{
  quotes: "\00BB" "\00BB";
}

caption
{
  margin: 0.5em 0pt;
}

dt
{
  page-break-after: avoid;
}

h1
{
  font-size: 1.6em;
  margin-bottom: 0.7em;
  margin-top: 1.4em;
}

h2
{
  font-size: 1.3em;
  margin-bottom: 0.6em;
  margin-top: 1.2em;
}

h3
{
  font-size: 1.1em;
```

```
}

h3, h4
{
  margin-bottom: 0.5em;
  margin-top: 1em;
}

h1, h2, h3, h4, h5, h6
{
  hyphenate: false;
}

hr
{
  border: 0.1pt solid;
}

img
{
  content-height: scale-to-fit;
  content-width: scale-to-fit;
  display: graphic;
  scaling: uniform;
  src: attr(src);
}

li
{
  margin-bottom: 0.8em;
  margin-top: 0.8em;
}

li p, li blockquote, li dl, li ol, li ul
{
  margin-bottom: 0.5em;
  margin-top: 0.5em;
}

li li
{
  margin-bottom: 0.5em;
  margin-top: 0.5em;
}

li li p, li li blockquote, li li dl, li li ol, li li ul
{
  margin-bottom: 0.3em;
  margin-top: 0.3em;
}

li li li
{
  margin-bottom: 0.4em;
  margin-top: 0.4em;
}
```

```
  li li li p, li li li blockquote, li li li dl, li li li ol,
    li li li ul
  {
    margin-bottom: 0.3em;
    margin-top: 0.3em;
  }

  li, p
  {
    text-align: justify;
  }

  pre
  {
    font-size: 0.85em;
  }

  ul
  {
    list-style-type: disc;
  }

  ol li ul, ul li ul
  {
    list-style-type: circle;
  }

  ol li ol li ul, ol li ul li ul, ul li ol li ul, ul li ul li ul
  {
    list-style-type: square;
  }

  q:after
  {
    content: close-quote;
  }

  q:before
  {
    content: open-quote;
  }

  script
  {
    display: none;
  }

  span.section-number
  {
    padding-right: 1em;
  }
}
```

## B.2 DELTAXML

```
@namespace deltaxml
  url(http://www.deltaxml.com/ns/well-formed-delta-v1);

@media print
{
  deltaxml|PCDATAnew, deltaxml|PCDATAold
  {
    display: inline;
  }

  deltaxml|exchange, deltaxml|new, deltaxml|old
  {
    display: wrapper;
  }

  *[deltaxml|delta="add"], deltaxml|PCDATAnew, deltaxml|new
  {
    text-decoration: underline;
  }

  *[deltaxml|delta="delete"], deltaxml|PCDATAold, deltaxml|old
  {
    text-decoration: line-through;
  }

  *[deltaxml|delta="add"]:before, deltaxml|PCDATAnew:before,
    deltaxml|new, *[deltaxml|delta="delete"]:before,
    deltaxml|PCDATAold:before, deltaxml|old
  {
    change-bar-class: changed;
    change-bar-placement: alternate;
    change-bar-style: solid;
    change-bar-width: 0.2pt;
  }

  *[deltaxml|delta="add"]:after, deltaxml|PCDATAnew:after,
    *[deltaxml|delta="delete"]:after, deltaxml|PCDATAold:after
  {
    change-bar-class: changed;
  }
}
```

## B.3 XLINK

```
@namespace xlink url(http://www.w3.org/1999/xlink);

@media print
{
  *[xlink|href]
  {
    link: attr(xlink|href);
  }
}
```

# REFERENCES

# C


[CSS2]
"Cascading Style Sheets, level 2, CSS2 Specification", W3C Recommendation 12 May 1998, Bert Bos, Håkon Wium Lie, Chris Lilley, Ian Jacobs, http://www.w3.org/TR/1998/REC-CSS2-19980512/.

[CSS3B]
"CSS basic box model", W3C Working Draft 9 August 2007, Bert Bos, http://www.w3.org/TR/2007/WD-css3-box-20070809/.

[CSS3G]
"CSS Generated Content for Paged Media Module", W3C Working Draft 13 May 2014, Dave Cramer, Håkon Wium Lie, http://www.w3.org/TR/2014/WD-css-gcpm-3-20140513/.

[CSS3L]
"CSS3 module: Lists", W3C Working Draft 24 November 2011, Simon Montagu, Daniel Yacob, Christopher Hoess, Daniel Glazman, http://www.w3.org/TR/2011/WD-css3-lists-20110524/.

[CSS3P]
"CSS3 Module: Paged Media", W3C Working Draft 10 October 2006, Håkon Wium Lie, Melinda Grant, http://www.w3.org/TR/2006/WD-css3-page-20061010/.

[CSS3S]
"Selectors Level 3", W3C Recommendation 29 September 2011, Tantek Çelik, Elika J. Etemad, Daniel Glazman, Ian Hickson, Peter Linss, John Williams, http://www.w3.org/TR/2011/REC-css3-selectors-20110929/.

[DELTA]
"How DeltaXML Represents Changes to XML Files", DeltaXML Ltd., http://www.deltaxml.com/library/how-deltaxml-represents-changes.html.

[NAMES]
"Namespaces in XML", W3C Recommendation 14 January 1999, Tim Bray, Dave Hollander, Andrew Layman, http://www.w3.org/TR/REC-xml-names/.

[XHTML]
"XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition)", W3C Recommendation 26 January 2000, revised 1 August 2002, http://www.w3.org/TR/2002/REC-xhtml1-20020801/.

[XLINK]
"XML Linking Language (XLink) Version 1.0", W3C Recommendation 27 June 2001, Steve DeRose, Eve Maler, David Orchard, http://www.w3.org/TR/2001/REC-xlink-20010627/.

[XSL-FO]
"Extensible Stylesheet Language (XSL), Version 1.0", W3C Recommendation 15 October 2001, Sharon Adler, Anders Berglund, Jeff Caruso, Stephen Deach,

Tony Graham, Paul Grosso, Eduardo Gutentag, Alex Milowski, Scott Parnell, Jeremy Richman, Steve Zilles, http://www.w3.org/TR/2001/REC-xsl-20011015/. [XSL-FO11]

"Extensible Stylesheet Language (XSL), Version 1.1", W3C Recommendation 05 December 2006, Anders Berglund, http://www.w3.org/TR/2006/REC-xsl11-20061205/.