



SAP CHART ENGINE

XML FORMAT DESCRIPTION

Author	SAP
Version	2.0
Last changed	July 2006
Status	Released

SAP AG
Dietmar-Hopp-Allee 16
69190 Walldorf
Germany
T +49/6227/747474
F +49/6227/757575
www.sap.com



Table of content

1. XML Formats of the Chart Engine.....	3
1.1. SimpleChartData.....	3
1.2. ChartData.....	3
2. ChartData XML: Chart Types.....	5
2.1. Category Based Charts.....	5
2.2. Bars, Columns as “floating blocks”.....	6
2.3. Category axis with variable width.....	7
2.4. Category axis with hierarchical categories.....	8
2.5. Scatter.....	9
2.6. TimeScatter.....	10
2.7. Gantt.....	10
2.8. Portfolio.....	13
2.9. Milestone Trend Analysis (MTA).....	16
2.10. SmartMap.....	17
2.11. Histogram.....	18
2.12. “Blocks” with Time Axis.....	18
2.13. DeltaChart.....	20
3. ChartData XML attributes.....	22
3.1. Attribute label.....	22
3.2. Attribute customizing.....	22
3.3. Attribute extension.....	23

1. XML Formats of the Chart Engine

The Chart Engine accepts data in two formats, the SimpleChartData and the ChartData. Compared to the format SimpleChartData the new ChartData is not restricted to category-based chart types. It's also possible to describe data for scatter, portfolio and other new chart types with this more flexible format.

1.1. SimpleChartData

The SimpleChartData format is used for category-based chart types. The XML contains the following:

- A list of category names
- A number of data series
- Data points in each data series, one for each category name

Example of XML data used to display a category-based chart with four categories and two data series:

```
<?xml version="1.0" encoding="utf-8"?>
<SimpleChartData>
  <Categories>
    <C>1st quarter</C>
    <C>2nd quarter</C>
    <C>3rd quarter</C>
    <C>4th quarter</C>
  </Categories>
  <Series label="Coffee">
    <S>94</S>
    <S>110</S>
    <S>139</S>
    <S>192</S>
  </Series>
  <Series label="Espresso">
    <S>35</S>
    <S>39</S>
    <S>38</S>
    <S>59</S>
  </Series>
</SimpleChartData>
```

1.2. ChartData

The ChartData format can be used for every chart type. The main differences compared to the SimpleChartData format are partly different tag names and the ability to construct points with several values of several value types.

Please read the next chapter to see examples of this data format.

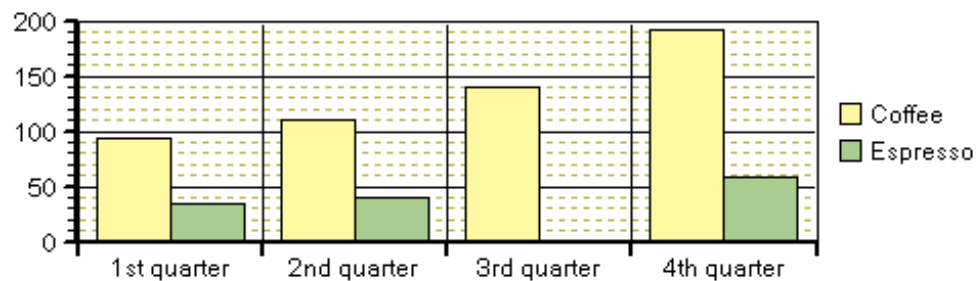
2. ChartData XML: Chart Types

In this chapter you can see how to structure the XML data depending which chart type you want to use (using the format ChartData).

2.1. Category Based Charts

Category base chart types are: **Lines, StackedLines, Profiles, StackedProfiles, Bars, StackedBars, Columns, StackedColumns, Area, StackedArea, ProfileArea, StackedProfileArea, Pie, Doughnut, SplitPie, Polar, Radar, StackedRadar, Speedometer.**

The category-based chart types contain a list of categories <Categories>. Each data series <Series> contains one data point <Point> for each category.



Example (4 categories, 2 data series, the third point is missing in the second data series)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>1st quarter</Category>
    <Category>2nd quarter</Category>
    <Category>3rd quarter</Category>
    <Category>4th quarter</Category>
  </Categories>
  <Series label="Coffee">
    <Point>
      <Value type="y">94</Value>
    </Point>
    <Point>
      <Value type="y">110</Value>
    </Point>
    <Point>
      <Value type="y">139</Value>
    </Point>
    <Point>
      <Value type="y">192</Value>
    </Point>
  </Series>
```

```

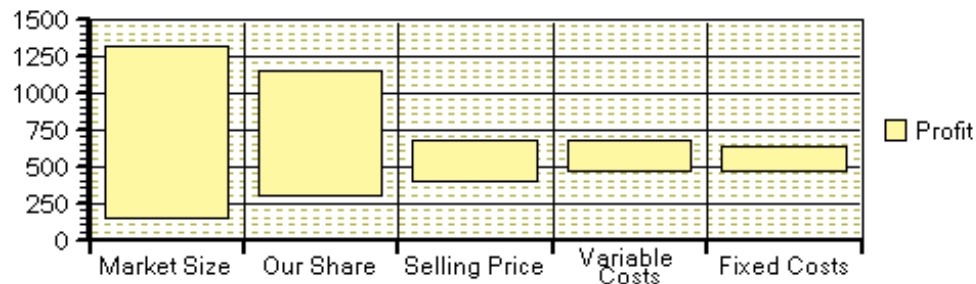
<Series label="Espresso">
  <Point>
    <Value type="y">35</Value>
  </Point>
  <Point>
    <Value type="y">39</Value>
  </Point>
  <Point>
    <Value type="y">59</Value>
  </Point>
</Series>
</ChartData>

```

2.2. Bars, Columns as “floating blocks”

The category-based chart types Bars and Columns usually draw the blocks starting from the zero line to the relevant value.

If the data XML for a point contains two <Value> tags with the type y then one value is interpreted as the lower limit and the other as the upper limit (left and right in the case of bars).



Example (5 categories, 1 data series each with two y-values):

```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>Market Size</Category>
    <Category>Our Share</Category>
    <Category>Selling Price</Category>
    <Category>Variable Costs</Category>
    <Category>Fixed Costs</Category>
  </Categories>
  <Series label="Profit" customizing="Series">
    <Point>
      <Value type="y">140</Value>
      <Value type="y">1310</Value>
    </Point>
    <Point>
      <Value type="y">300</Value>
      <Value type="y">1150</Value>
    </Point>
  </Series>
</ChartData>

```

```

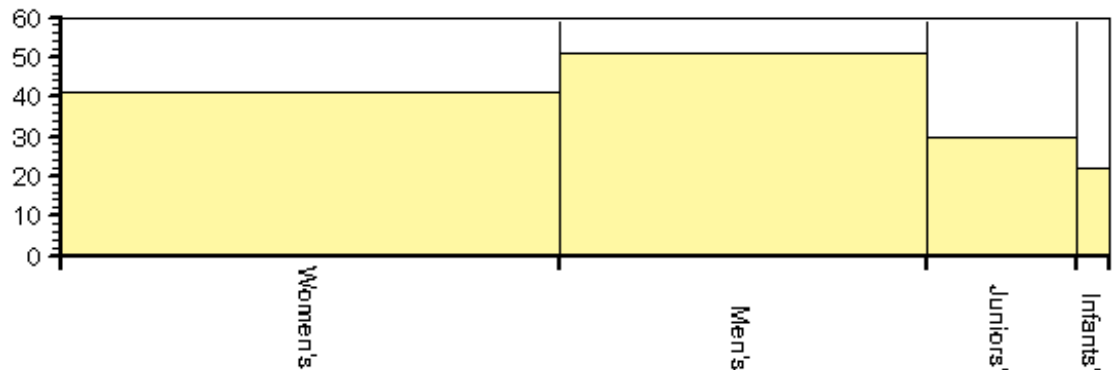
</Point>
<Point>
  <Value type="y">390</Value>
  <Value type="y">670</Value>
</Point>
<Point>
  <Value type="y">470</Value>
  <Value type="y">670</Value>
</Point>
<Point>
  <Value type="y">470</Value>
  <Value type="y">630</Value>
</Point>
</Series>
</ChartData>

```

2.3. Category axis with variable width

Every chart type using a category axis can activate the variable width of the categories (see property `VariableCategories` of the category axis).

In this case the width of a category is determined by the values of type `x`. If there are different `x`-values within a category the maximum of these values is taken for calculating the width of a category.



Example (4 categories, 1 data series each with x- and y-values)

```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>Women's</Category>
    <Category>Men's</Category>
    <Category>Juniors'</Category>
    <Category>Infants'</Category>
  </Categories>
  <Series label="Series1" customizing="Series1">
    <Point label="Label1/1">
      <Value type="x">162.6</Value>

```

```

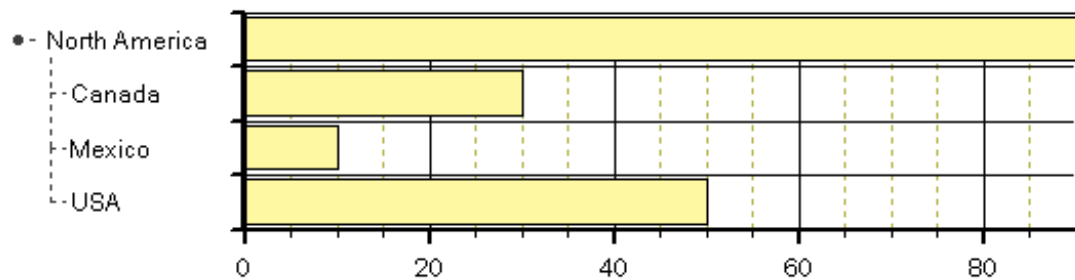
    <Value type="y">41</Value>
  </Point>
  <Point label="Label1/2">
    <Value type="x">119</Value>
    <Value type="y">51</Value>
  </Point>
  <Point label="Label1/3">
    <Value type="x">48.7</Value>
    <Value type="y">30</Value>
  </Point>
  <Point label="Label1/4">
    <Value type="x">10.5</Value>
    <Value type="y">22</Value>
  </Point>
</Series>
</ChartData>

```

2.4. Category axis with hierarchical categories

Every chart type using a category axis can activate the hierarchical view of the categories (see property `AxisType` of the category axis).

The hierarchical view requires that each category has a depth within the hierarchy. Using the `<Category>` tag of the XML format the name of a category must start with the prefix `"\D"` (with D as a depth from 1 to 9).



Example (4 categories, the first one as the root of the hierarchy)

```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>North America</Category>
    <Category>\1Canada</Category>
    <Category>\1Mexico</Category>
    <Category>\1USA</Category>
  </Categories>
  <Series>
    <Point>
      <Value type="y">90</Value>
    </Point>
  </Series>
</ChartData>

```



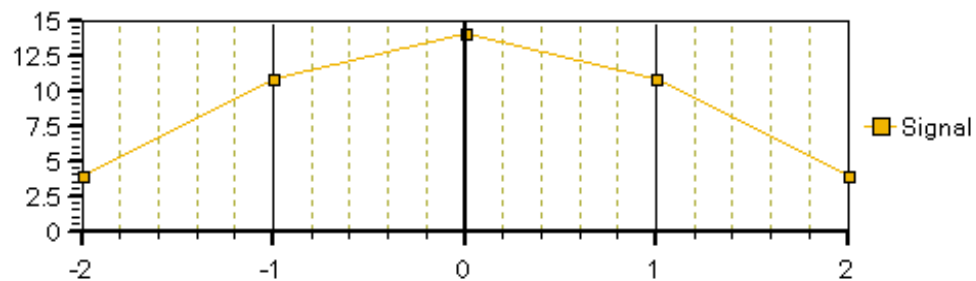
```

<Point>
  <Value type="y">30</Value>
</Point>
<Point>
  <Value type="y">10</Value>
</Point>
<Point>
  <Value type="y">50</Value>
</Point>
</Series>
</ChartData>

```

2.5. Scatter

There are no categories in the chart type Scatter. Instead every point in a data series is defined by its x value and y value.



Example (no categories, 1 data series each with x- and y-values)

```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series label="Signal">
    <Point>
      <Value type="x">-2</Value>
      <Value type="y">3.92222</Value>
    </Point>
    <Point>
      <Value type="x">-1</Value>
      <Value type="y">10.7763</Value>
    </Point>
    <Point>
      <Value type="x">0</Value>
      <Value type="y">14</Value>
    </Point>
    <Point>
      <Value type="x">1</Value>
      <Value type="y">10.7763</Value>
    </Point>
    <Point>
      <Value type="x">2</Value>
      <Value type="y">3.92222</Value>
    </Point>
  </Series>
</ChartData>

```

```

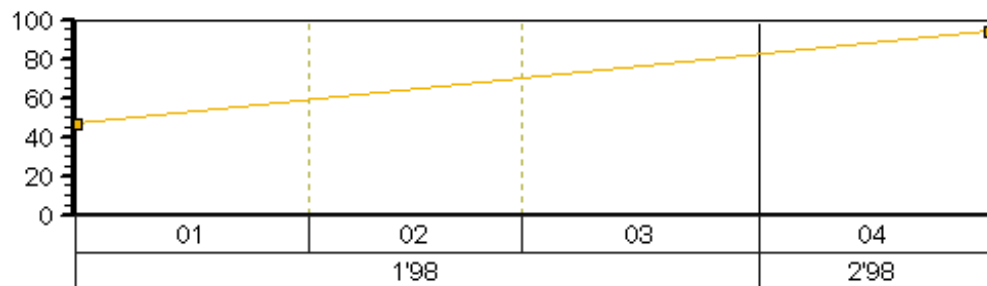
</Series>
</ChartData>

```

2.6. TimeScatter

With the chart type TimeScatter, as with the Scatter display, every point is defined in terms of two values. However, since the x axis is a time axis a point consists of a y value and a time-value. The format of a time value is as follows:

[yyyymmdd][;hhmmss[ms...]]



Example (no categories, 1 data series with time- and y-values)

```

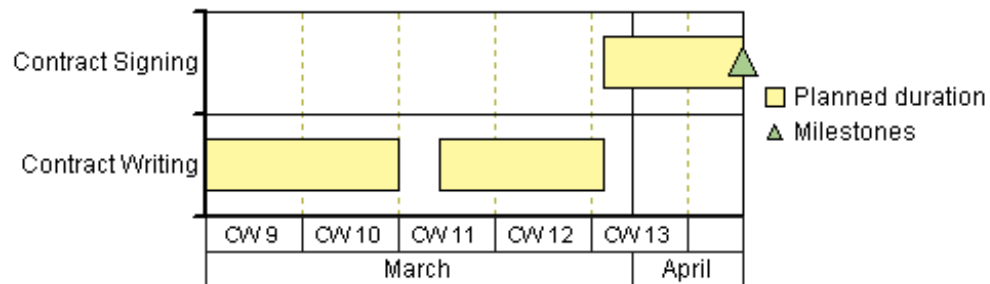
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series>
    <Point>
      <Value type="y">47</Value>
      <Value type="time">19980101;100000</Value>
    </Point>
    <Point>
      <Value type="y">94</Value>
      <Value type="time">19980501</Value>
    </Point>
  </Series>
</ChartData>

```

2.7. Gantt

The data for the Gantt chart resembles that for Bars/Columns as „floating blocks“. There is a <Point> tag for each category. Each point has time values with the type time. Each data point can have an even number of values. A bar/marker is drawn in the relevant category for each pair of values.

With this chart type series/point customizing determines whether a pair of values is displayed as a bar (MarkerShape None) or as a marker (MarkerShape <> None). The latter is used to display milestones. If the time values are to be displayed as a marker and are not identical then the marker symbol is placed between the time points.

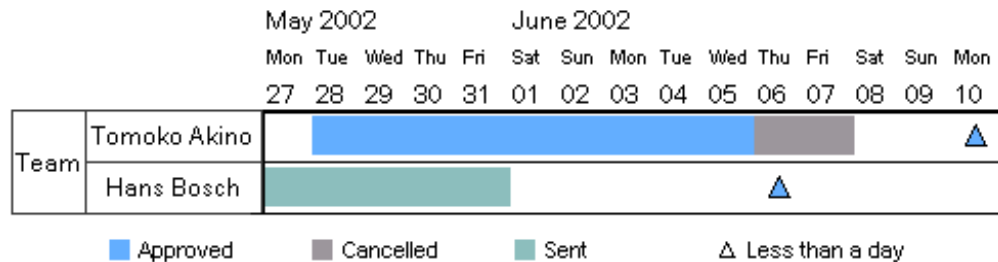


Example (two categories, 2 data series; the first data series has two bars in the first category, one bar in the second category; the second data series displays a milestone in the second category (assuming the appropriate Customizing)):

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>Contract Writing</Category>
    <Category>Contract Signing</Category>
  </Categories>
  <Series label="Planned duration" customizing="PlannedDuration">
    <Point>
      <Value type="time">20000301</Value>
      <Value type="time">20000315</Value>
      <Value type="time">20000318</Value>
      <Value type="time">20000330</Value>
    </Point>
    <Point>
      <Value type="time">20000330</Value>
      <Value type="time">20000409</Value>
    </Point>
  </Series>
  <Series label="Milestones" customizing="Milestone">
    <Point/>
    <Point>
      <Value type="time">20000409</Value>
      <Value type="time">20000409</Value>
    </Point>
  </Series>
</ChartData>
```

The chart type Gantt also allows an alternative way of evaluating the XML data. By disabling the Gantts property UseCategories you can easily switch the internal orientation of the data.

I.e. the data of a series is displayed within one category. With this solution you can easily display several entries using different customizings within a category.

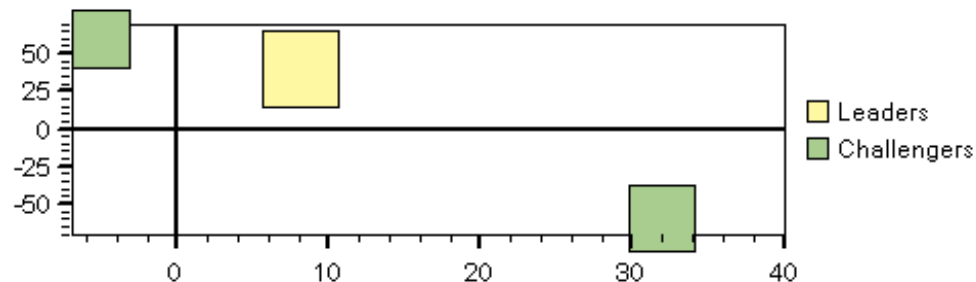


Example (Gantt with disabled "UseCategories"; hierarchical categories, 2 data series; each series is displayed horizontally)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>Team</Category>
    <Category>\1Tomoko Akino</Category>
    <Category>\1Hans Bosch</Category>
  </Categories>
  <Series label="Tomoko Akino">
    <Point customizing="approved">
      <Value type="time">20020528</Value>
      <Value type="time">20020606</Value>
    </Point>
    <Point customizing="cancelled">
      <Value type="time">20020606</Value>
      <Value type="time">20020608</Value>
    </Point>
    <Point customizing="approvedPartTime">
      <Value type="time">20020610</Value>
      <Value type="time">20020611</Value>
    </Point>
  </Series>
  <Series label="Hans Bosch">
    <Point customizing="sent">
      <Value type="time">20020527</Value>
      <Value type="time">20020601</Value>
    </Point>
    <Point customizing="approvedPartTime">
      <Value type="time">20020606</Value>
      <Value type="time">20020607</Value>
    </Point>
  </Series>
</ChartData>
```

2.8. Portfolio

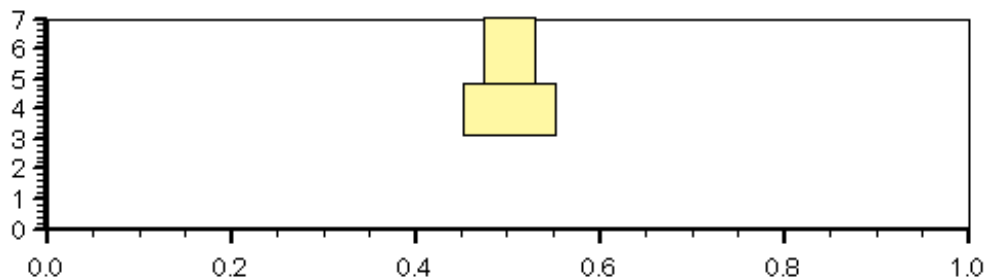
In the simplest type of portfolio chart each point is defined as in the scatter chart, that is, with an x value, a y value and also a size value. The latter defines the size of a data point. Categories are not required for this purpose.



Example (points with x, y and size)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series label="Leaders" customizing="LeadersCu">
    <Point label="SAP">
      <Value type="x">8</Value>
      <Value type="y">40</Value>
      <Value type="size">37.9174</Value>
    </Point>
  </Series>
  <Series label="Challengers" customizing="ChallengersCu">
    <Point label="Company 2">
      <Value type="x">-5</Value>
      <Value type="y">60</Value>
      <Value type="size">27</Value>
    </Point>
    <Point label="Company 3">
      <Value type="x">32</Value>
      <Value type="y">-60</Value>
      <Value type="size">32</Value>
    </Point>
  </Series>
</ChartData>
```

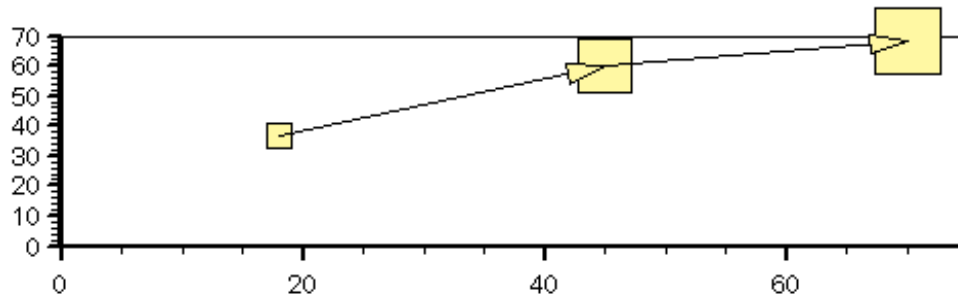
If the height and width of a point's size are not the same then values should be specified for width and height instead of size.



Example (points with width and height instead size)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series>
    <Point>
      <Value type="x">0.5</Value>
      <Value type="y">5.5</Value>
      <Value type="width">5</Value>
      <Value type="height">10</Value>
    </Point>
    <Point>
      <Value type="x">0.5</Value>
      <Value type="y">4</Value>
      <Value type="width">10</Value>
      <Value type="height">5</Value>
    </Point>
  </Series>
</ChartData>
```

To show trends in a portfolio chart you can display trend arrows in data points. To do this every data point from which an arrow is to be drawn must have a trendX- or trendY value. You can see three data points in the following example. They are linked by trend arrows.



Example (points linked by trend arrows)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series>
    <Point>
      <Value type="x">70</Value>
      <Value type="y">68</Value>
      <Value type="size">9800</Value>
    </Point>
    <Point>
      <Value type="x">45</Value>
      <Value type="y">60</Value>
      <Value type="size">7600</Value>
      <Value type="trendX">70</Value>
      <Value type="trendY">68</Value>
    </Point>
  </Series>
</ChartData>
```

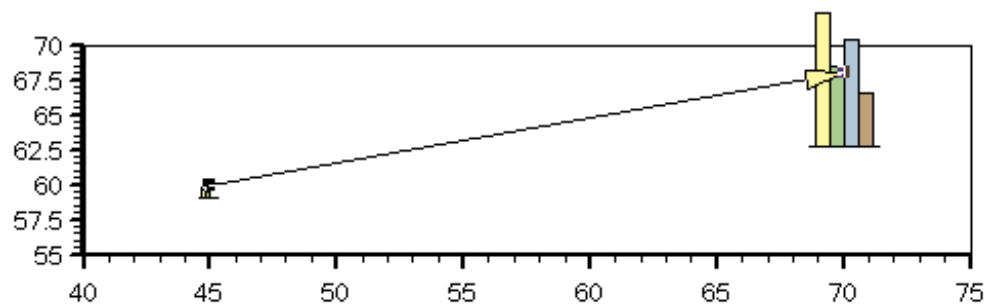
```

</Point>
<Point>
  <Value type="x">18</Value>
  <Value type="y">37</Value>
  <Value type="size">2600</Value>
  <Value type="trendX">45</Value>
  <Value type="trendY">60</Value>
</Point>
</Series>
</ChartData>

```

Every data point in a portfolio chart can contain its own chart. To do this the data point must contain the individual chart values with the type chart. The data point chart is drawn on the area specified by size or width and height. The chart type depends on the MarkerShape in customizing.

The example below shows a portfolio chart with two data points each of which displays one chart with four values.



Example (Charts in portfolio)

```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series>
    <Point>
      <Value type="x">70</Value>
      <Value type="y">68</Value>
      <Value type="width">120</Value>
      <Value type="height">30</Value>
      <Value type="chart">100</Value>
      <Value type="chart">60</Value>
      <Value type="chart">80</Value>
      <Value type="chart">40</Value>
    </Point>
    <Point>
      <Value type="x">45</Value>
      <Value type="y">60</Value>
      <Value type="size">100</Value>
      <Value type="trendX">70</Value>
      <Value type="trendY">68</Value>
      <Value type="chart">100</Value>
    </Point>
  </Series>
</ChartData>

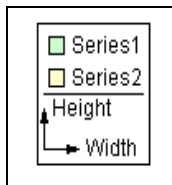
```

```

<Value type="chart">60</Value>
<Value type="chart">12</Value>
<Value type="chart">8</Value>
</Point>
</Series>
</ChartData>

```

Enabling the property `MoreInfo` of the legend allows you to display additional labels for width and height of a point. To display such labels add a `<Categories>` section to your XML data and fill the third and the fourth category:



Example (legend texts for width and height):

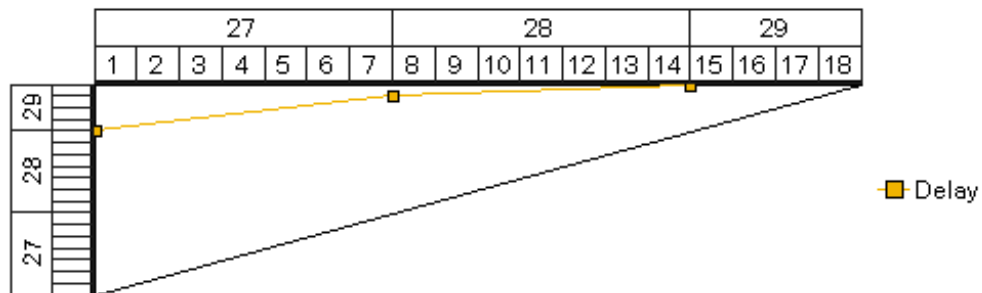
```

<Categories>
  <Category></Category>
  <Category></Category>
  <Category>Width</Category>
  <Category>Height</Category>
</Categories>

```

2.9. Milestone Trend Analysis (MTA)

In a MTA chart every data point is defined by two time values. The first value has the type `time` and relates to the x-axis (reporting point in time), the second – which also has the type `time` – relates to the y-axis (milestone).



Example (3 milestones in MTA)

```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series label="Delay">
    <Point>

```



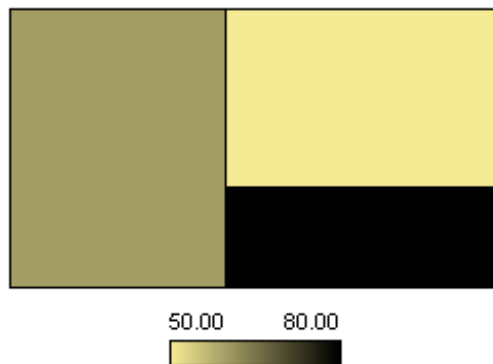
```

    <Value type="time">20020701</Value>
    <Value type="time">20020715</Value>
  </Point>
  <Point>
    <Value type="time">20020708</Value>
    <Value type="time">20020718</Value>
  </Point>
  <Point>
    <Value type="time">20020715</Value>
    <Value type="time">20020719</Value>
  </Point>
</Series>
</ChartData>

```

2.10. SmartMap

In a SmartMap chart every data point is defined by a size and a color value. The size value determines the size of the points rectangle. The coloring is done by evaluating the y value of a point.



Example (1 data series with 3 points)

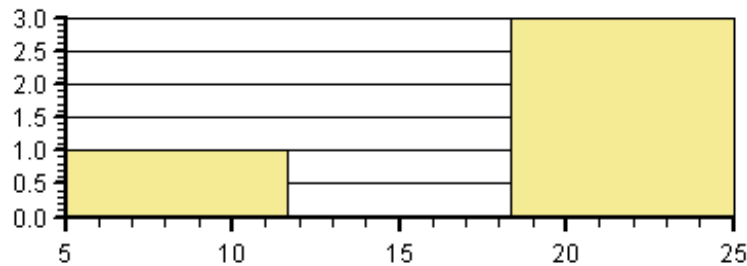
```

<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series customizing="Series1">
    <Point>
      <Value type="y">50</Value>
      <Value type="size">90</Value>
    </Point>
    <Point>
      <Value type="y">60</Value>
      <Value type="size">110</Value>
    </Point>
    <Point>
      <Value type="y">80</Value>
      <Value type="size">50</Value>
    </Point>
  </Series>
</ChartData>

```

2.11. Histogram

In a Histogram chart the range from the minimum value to the maximum value of all points (value type y) is divided in n classes (see customizing properties of this chart type). A class is displayed as a rectangle. Its height is determined by the number of y values within the corresponding part of this range.



Example (1 data series with 4 points and 3 classes)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Series customizing="Series1">
    <Point>
      <Value type="y">20</Value>
    </Point>
    <Point>
      <Value type="y">5</Value>
    </Point>
    <Point>
      <Value type="y">22</Value>
    </Point>
    <Point>
      <Value type="y">25</Value>
    </Point>
  </Series>
</ChartData>
```

2.12. "Blocks" with Time Axis*

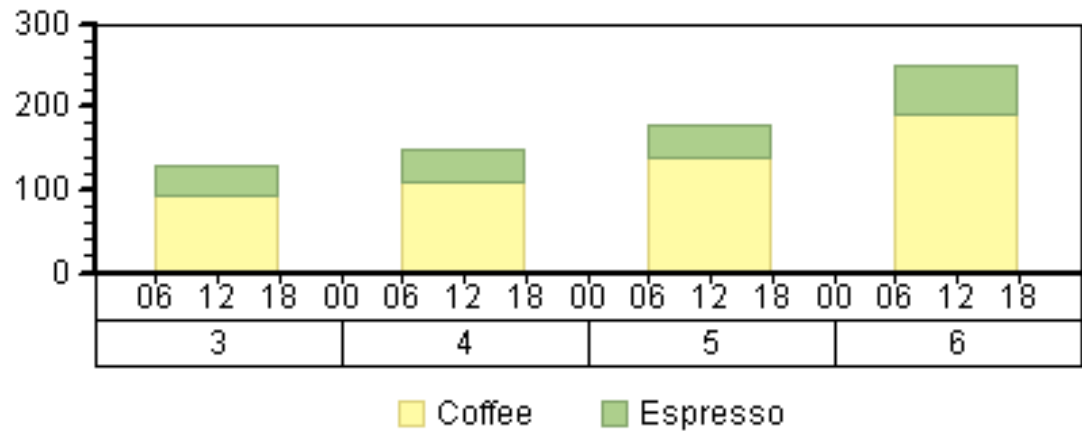
There are four additional chart types using a time axis:

TimeColumns, StackedTimeColumns, TimeBars and StackedTimeBars.

These chart types work similar to their category-based counterparts (see chapter 2.1 about Category Based Charts). A list of categories is followed by a list of series. But each category must contain a time value. The time format is described in chapter 2.6 (TimeScatter).

* Requires >= IGS 7.00

The position of the blocks is determined by the time values (i.e. the list of categories).



Example (StackedTimeColumns with 4 categories having time values, 2 data series)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>20060703;120000</Category>
    <Category>20060704;120000</Category>
    <Category>20060705;120000</Category>
    <Category>20060706;120000</Category>
  </Categories>
  <Series label="Coffee">
    <Point>
      <Value type="y">94</Value>
    </Point>
    <Point>
      <Value type="y">110</Value>
    </Point>
    <Point>
      <Value type="y">139</Value>
    </Point>
    <Point>
      <Value type="y">192</Value>
    </Point>
  </Series>
  <Series label="Espresso">
    <Point>
      <Value type="y">35</Value>
    </Point>
    <Point>
      <Value type="y">39</Value>
    </Point>
    <Point>
      <Value type="y">38</Value>
    </Point>
    <Point>
      <Value type="y">59</Value>
    </Point>
  </Series>
</ChartData>
```

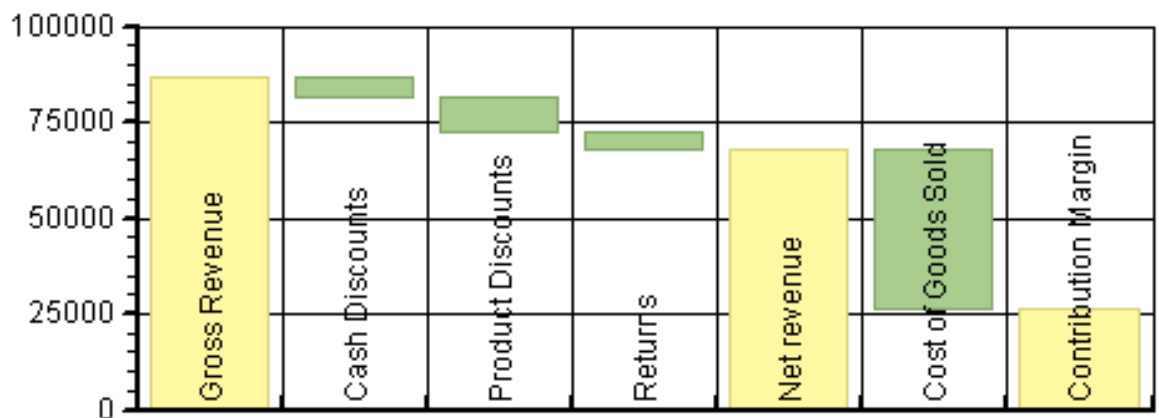
```
</Series>
</ChartData>
```

2.13. DeltaChart*

The DeltaChart displays columns with a set of delta rectangles in between. The ChartData has to consist of exactly one series.

The first and the last category display columns, all other categories display deltas.

If you need additional columns you have to define hierarchical categories (see chapter 2.4). In this case the first and the last category are displayed as columns. Additionally the last node of each level is also displayed as a column. The remaining categories are displayed as deltas (leaves only) or skipped (nodes as the parent of delta leaves).



Example (Hierarchical categories with three columns)

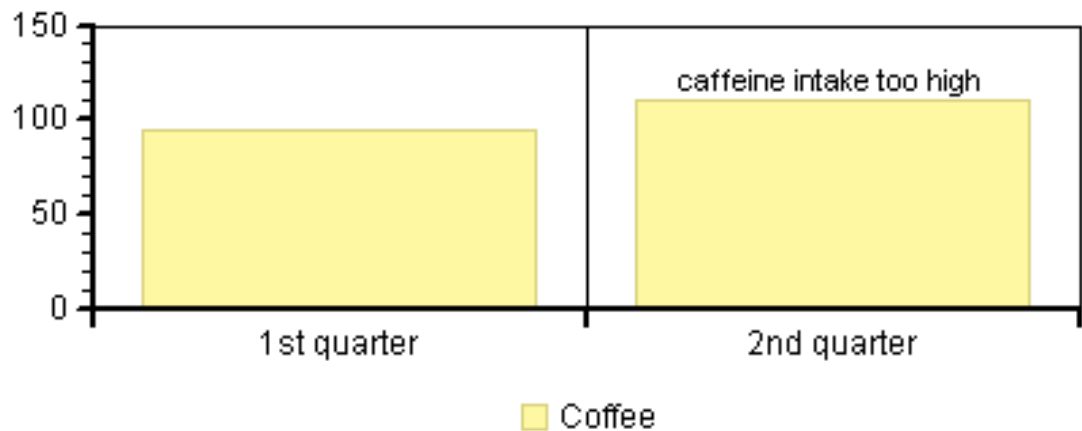
```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>Gross Revenue</Category>
    <Category>\1Sales Discounts</Category>
    <Category>\2Cash Discounts</Category>
    <Category>\2Product Discounts</Category>
    <Category>\1Sales Returns and AI</Category>
    <Category>\2Returns</Category>
    <Category>\1Net revenue</Category>
    <Category>\2Cost of Sales</Category>
    <Category>\3Cost of Goods Sold</Category>
    <Category>\2Contribution Margin</Category>
  </Categories>
  <Series>
    <Point>
      <Value type="y">87000</Value>
```

```
</Point>
<Point>
  <Value type="y">-14750</Value>
</Point>
<Point>
  <Value type="y">-5700</Value>
</Point>
<Point>
  <Value type="y">-9050</Value>
</Point>
<Point>
  <Value type="y">-4325</Value>
</Point>
<Point>
  <Value type="y">-4325</Value>
</Point>
<Point>
  <Value type="y">67925</Value>
</Point>
<Point>
  <Value type="y">-41500</Value>
</Point>
<Point>
  <Value type="y">-41500</Value>
</Point>
<Point>
  <Value type="y">26425</Value>
</Point>
</Series>
</ChartData>
```

3. ChartData XML attributes

3.1. Attribute label

The attribute label can be used to provide labels for series and points. It also depends on the customizing settings if a text is displayed in the chart (and how it looks like), e.g. the properties ShowLabels and Format.



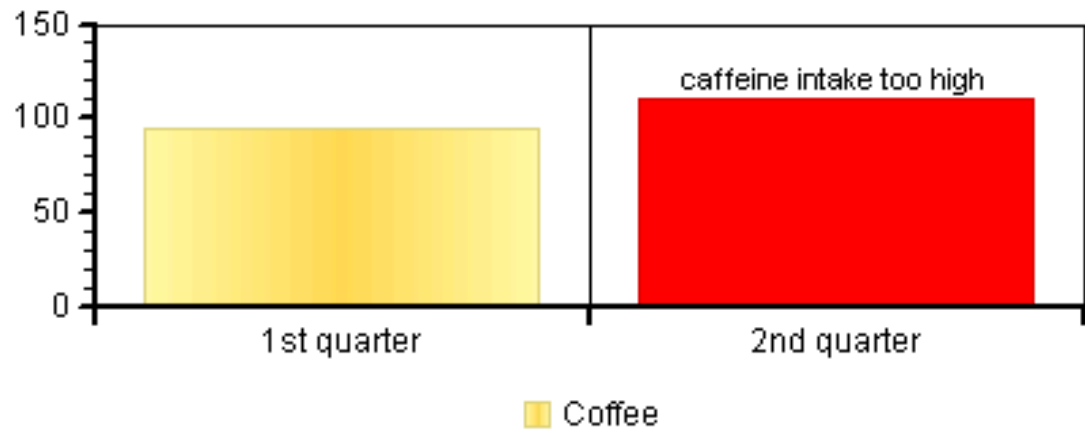
Example (Series and Point with label)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>1st quarter</Category>
    <Category>2nd quarter</Category>
  </Categories>
  <Series label="Coffee">
    <Point>
      <Value type="y">94</Value>
    </Point>
    <Point label="caffeine intake too high">
      <Value type="y">110</Value>
    </Point>
  </Series>
</ChartData>
```

3.2. Attribute customizing

The attribute customizing is the connection to the customizing XML. If you created series or point customizing (e.g. using the SAP Chart Designer) that's the way to tell the Chart Engine which data should use which customizing settings.

Usually series customizing created by the Chart Designer has IDs like Series1, Series2 etc. But as renaming is possible you can also use arbitrary IDs. Point customizing is used for data points that should have a different look compared to the rest of their series (e.g. for indicating an alert).

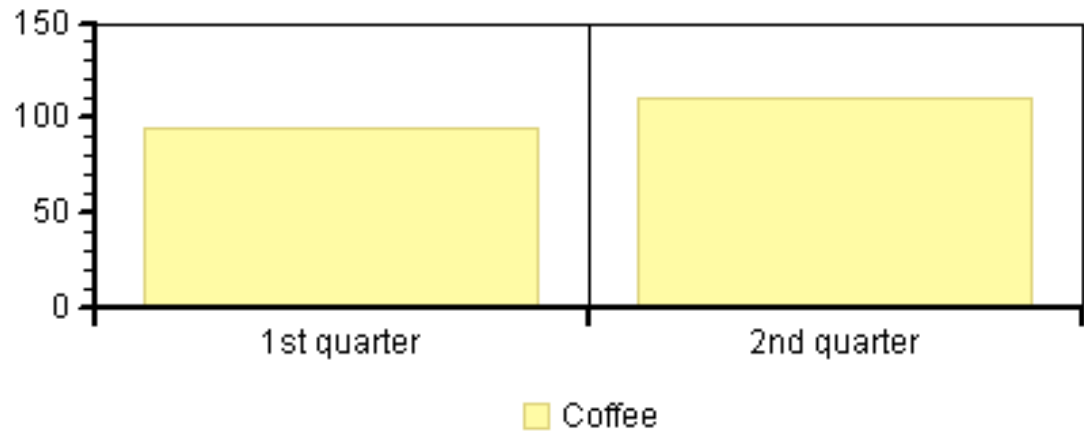


Example (Series uses series customizing, its second point uses point customizing)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>1st quarter</Category>
    <Category>2nd quarter</Category>
  </Categories>
  <Series customizing="Series1" label="Coffee">
    <Point>
      <Value type="y">94</Value>
    </Point>
    <Point customizing="caffeineAlert" label="caffeine intake too high">
      <Value type="y">110</Value>
    </Point>
  </Series>
</ChartData>
```

3.3. Attribute extension

Using the attribute extension it's possible to create an image map for the chart. Within the data you can define this attribute for Categories, Series and Points.



Corresponding image map:

<area shape=rect coords="55, 48,201, 112" alt='normal caffeine intake'>

Example (point using an extension)

```
<?xml version="1.0" encoding="utf-8"?>
<ChartData>
  <Categories>
    <Category>1st quarter</Category>
    <Category>2nd quarter</Category>
  </Categories>
  <Series label="Coffee">
    <Point extension="alt='normal caffeine intake'">
      <Value type="y">94</Value>
    </Point>
    <Point>
      <Value type="y">110</Value>
    </Point>
  </Series>
</ChartData>
```