Eclipse Chart Engine Scale/Axis/Grouping Enhancements

Function Specification
Draft 10: Oct. 25th, 2006

Abstract

This document describes the enhancements for scale/axis/grouping support in the Eclipse Chart Engine.

Document Revisions

Draft	Date	Primary Author(s)	Description of Changes
1	2006-9-4	Zhiqiang Qian	Initial Draft
2	2006-9-6	Zhiqiang Qian	Added more details and mockup, reorganized doc structure
3	2006-9-8	Yulin Wang	Updated Scale mockup
4	2006-9-12	Zhiqiang Qian	Added multiple orthogonal grouping section
5	2006-9-13	Zhiqiang Qian	Added multiple series aggregation section
6	2006-9-18	Zhiqiang Qian	Added custom axis label location section
7	2006-9-20	Zhiqiang Qian	Updated custom axis label location section to custom series datapoint location section
8	2006-9-25	Yulin Wang	Updated Scale mockup
9	2006-9-27	David Michonneau	Added missing screenshots, Moved 2 level grouping section. Updated bugzilla entries. Updated and refined general content.
20	2006-10-25	David Michonneau	Added multi-level support for Category Series grouping

Contents

1. Introduction

1.1.1 Goal

This project aims at providing a more powerful and flexible Grouping, Scale and Axis support in the Eclipse Chart Engine. Most features originated as bugzilla entries from end-users, feedback in the BIRT newsgroups and known limitations of the chart engine.

1.1.2 Scope

This document describes the features by detailing their functionality, Model API changes and proposes new mockups for the Chart Builder. Further Chart Engine API changes might be necessary at implementation stage and will be detailed in another specification document.

1.1.3 Backward compatibility
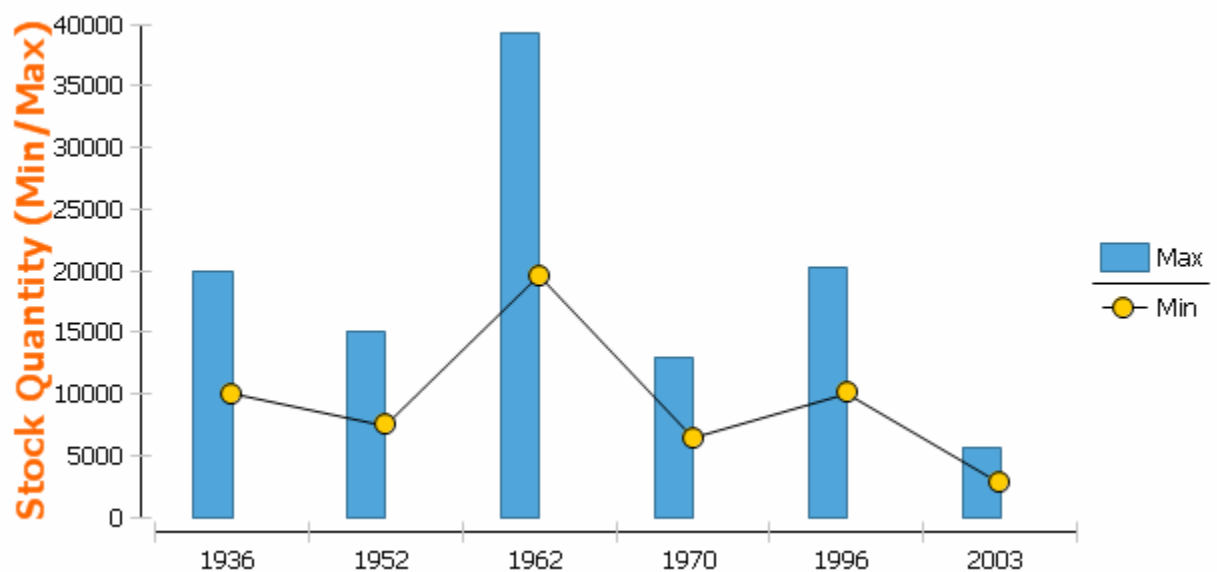
All model changes described in this document will ensure backward compatibility of the engine with old chart models.

2. Grouping

2.1   Multiple series aggregation support

2.1.1 Definition

When doing base series aggregation, each orthogonal series definition can specify its own aggregate function. Currently, all orthogonal series definitions must share the same function.

The example shows the first series represents the maximum stock quantity in a period of years (using 'max' as the aggregate function) and the second series represents the minimum stock quantity in the same period (using 'min' as the aggregate function).

The aggregate function defined in the base series definition will be treated as the default function. If no function set explicitly in orthogonal series definition, it will take this as default value. This also ensures the compatibility to old models.

2.1.2 Model Change

None. The SeriesDefinition Grouping element will be used to retrieve the aggregation function for each Value Series.

2.1.3 UI Mockup

A new "sigma" button will appear at the left of the Y series definition. It will be disabled until any grouping is defined on the X Series. Clicking on the button will show a dropdown menu with a list of possible aggregate functions and a check mark to indicate the one currently selected.
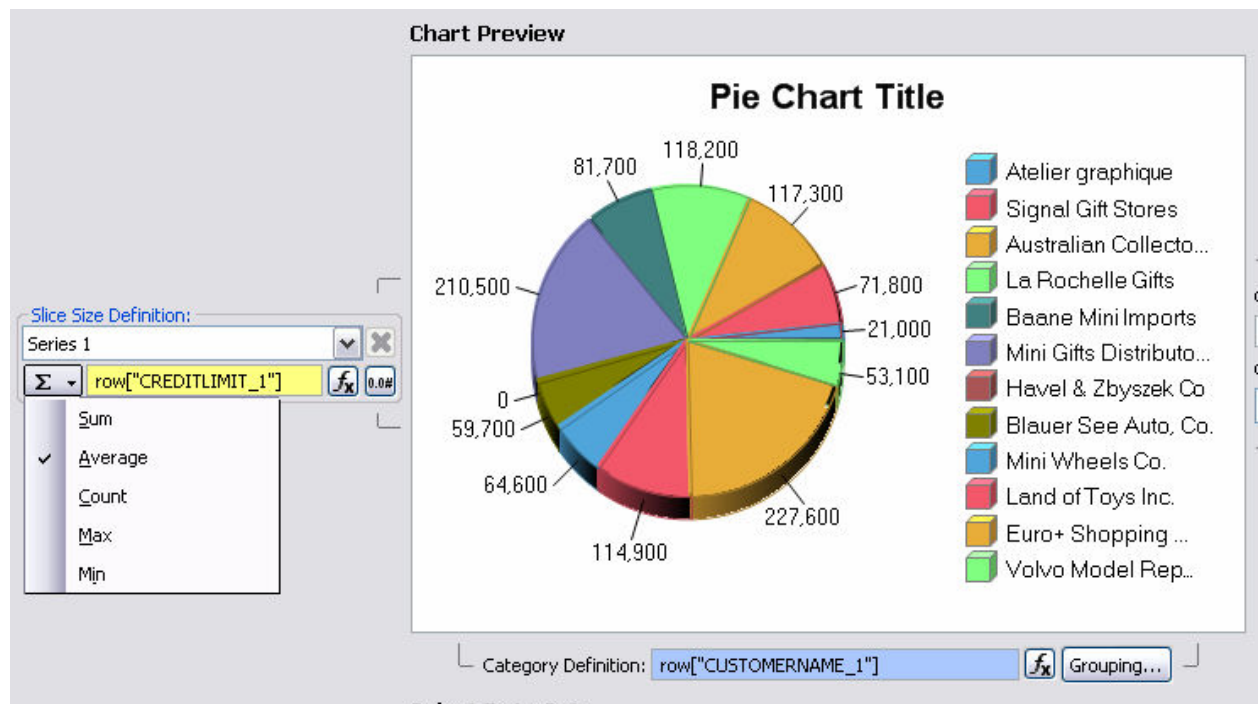
2.2 Grouping by range

2.2.1 Definition

This feature improves the datetime grouping to be by value instead of by range (i.e. for month, the groups will start at the beginning of the month and end at the end of the month). Currently the engine groups by 30 days interval based on the first value, which is not desirable and makes it difficult to read the chart.

For example, the labels on the Axis should be able to show the month names, not the data values:

Bar Chart Title
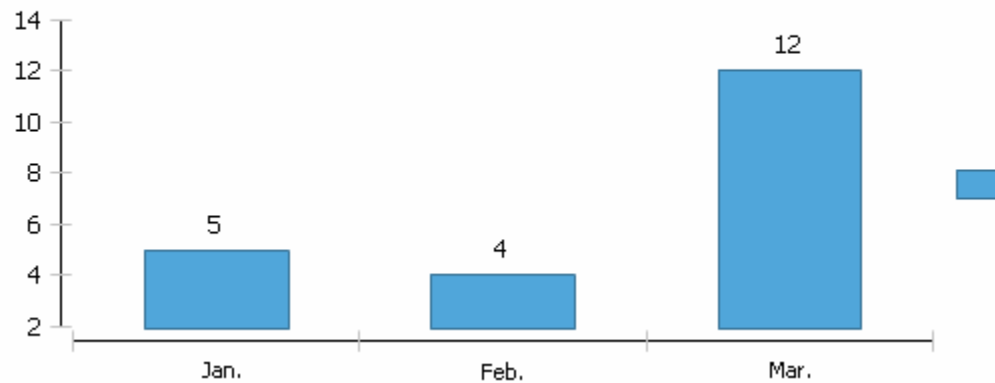
As a similar improvement for numeric range grouping, the axis labels should be more indicative of the ranges.

Bar Chart Title
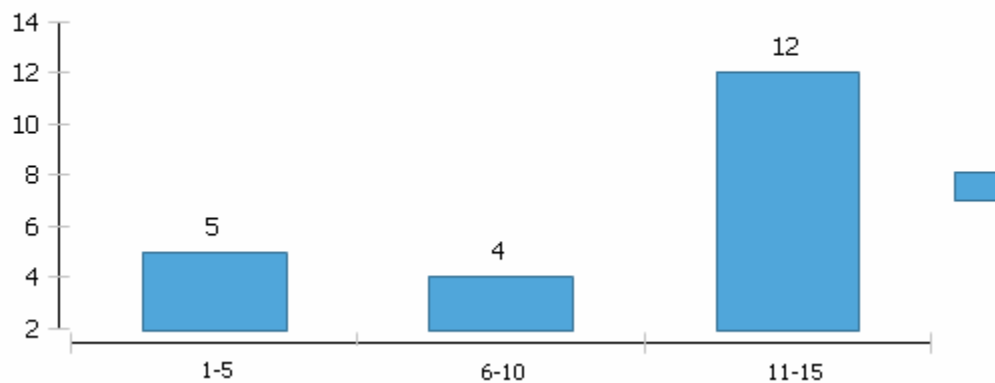
A new "Quarter" grouping category will also be supported:

Bar Chart Title

2.2.2 Model Change
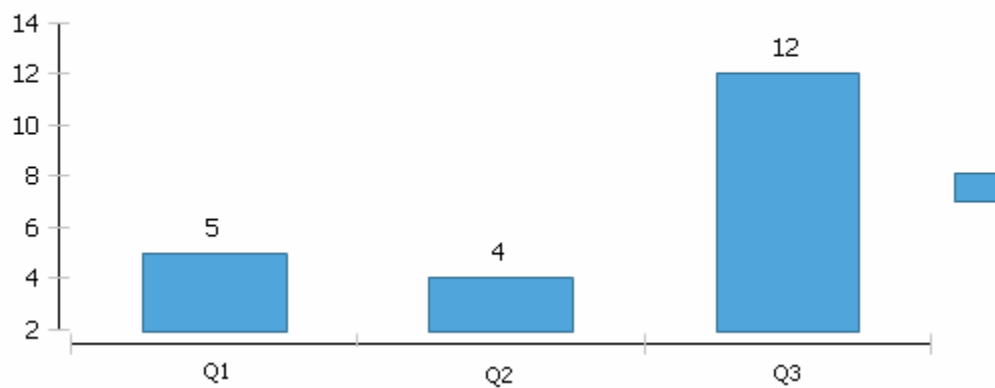
A new scale unit type is needed:

```xml
<xsd:simpleType name="ScaleUnitType">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        This type defines predefined date unit types for scale.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Seconds"/>
        <xsd:enumeration value="Minutes"/>
        <xsd:enumeration value="Hours"/>
        <xsd:enumeration value="Days"/>
        <xsd:enumeration value="Weeks"/>
        <xsd:enumeration value="Months"/>
        <xsd:enumeration value="Quarter"/>
        <xsd:enumeration value="Years"/>
    </xsd:restriction>
</xsd:simpleType>
```
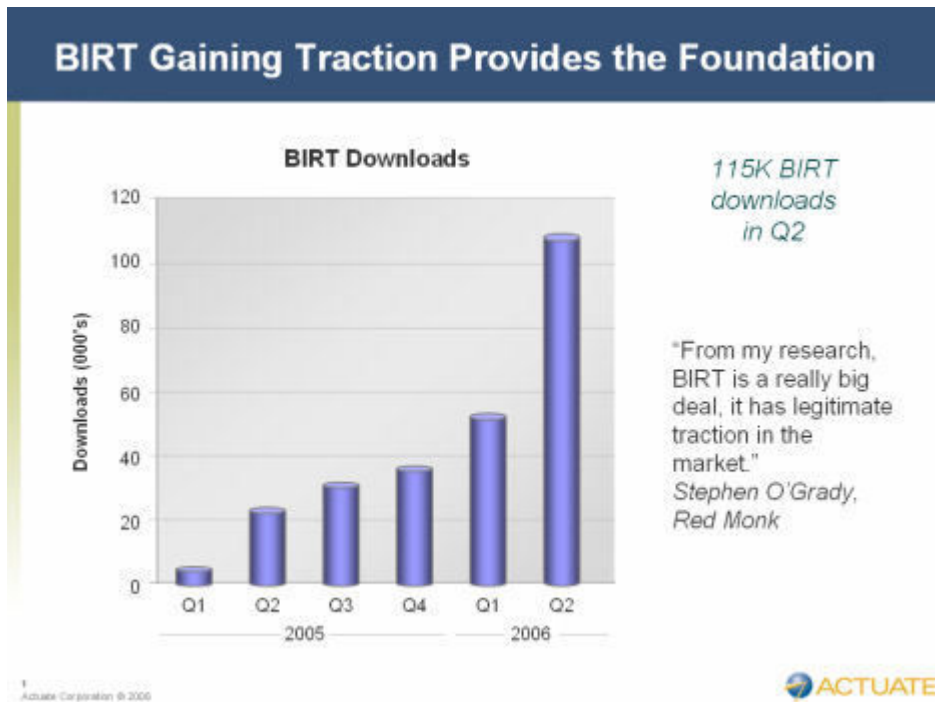
New group unit type as well:

```xml
<xsd:simpleType name="GroupingUnitType">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        This type represents the possible units for grouping data.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Seconds"/>
        <xsd:enumeration value="Minutes"/>
        <xsd:enumeration value="Hours"/>
        <xsd:enumeration value="Days"/>
        <xsd:enumeration value="Weeks"/>
        <xsd:enumeration value="Months"/>
        <xsd:enumeration value="Quarter"/>
        <xsd:enumeration value="Years"/>
    </xsd:restriction>
</xsd:simpleType>
```

2.3   Categories Grouping Sub-Levels

2.3.1   Definition

This screenshot shows that it is sometimes desirable to display the Axis labels on two or more levels, corresponding to a group and subgroup, to avoid any redundancy or ambiguity in the labels content (for instance Q1, Q2, Q3, Q4, Q1, Q2 … is ambiguous as the year is not shown, and Q1, 2005 – Q2, 2005 – Q3, 2005 is redundant as the year is repeated in each label.

To remedy that problem, it is possible to define the X Series grouping on several levels that can define a unit type, an interval, and with a customizable format.

Note that when sublevels are defined, the main level will not follow the stagger or rotation positioning of the X Axis labels. It will however use the font specified, that applies to all levels.

2.3.2   Model Change

```
<xsd:complexType name="SeriesGrouping">
        <xsd:sequence>
                <xsd:element name="Enabled" type="xsd:boolean"/>
                <xsd:element name="GroupingUnit"
type="attribute:GroupingUnitType"/>
                <xsd:element name="GroupingOrigin" type="DataElement"/>
                <xsd:element name="GroupingInterval" type="xsd:int"/>
                <xsd:element name="GroupType" type="attribute:DataType"/>
                <xsd:element name="AggregateExpression" type="xsd:string"/>
                <xsd:element name="SubLevels" type="GroupingLevel"
maxOccurs="unbounded"/>


        </xsd:sequence>
```
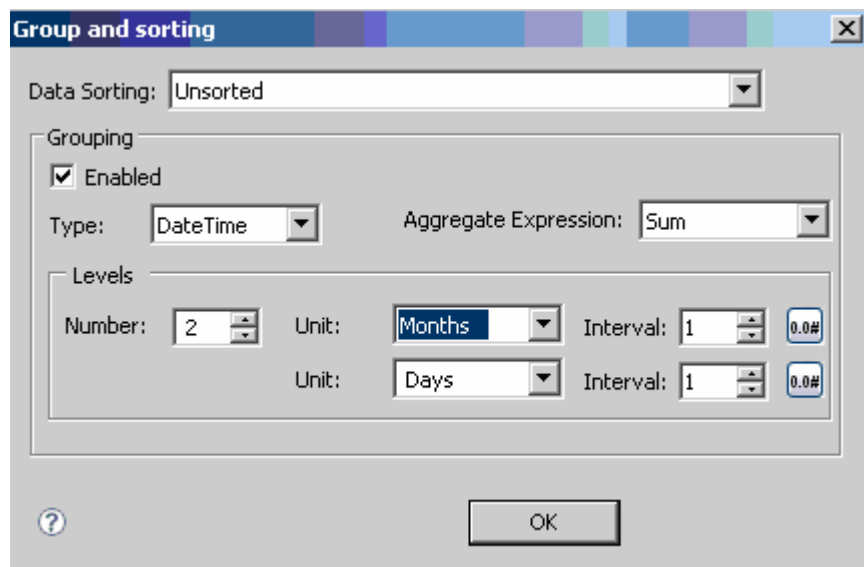
```
</xsd:complexType>

<xsd:complexType name="GroupingLevel">
      <xsd:sequence>
            <xsd:element name="GroupingUnit"
type="attribute:GroupingUnitType"/>
            <xsd:element name="GroupingInterval" type="Xsd:int"/>
            <xsd:element name="GroupingFormat"
type="attribute:GroupingFormatSpecifier"/>
      </xsd:sequence>
</xsd:complexType>
```
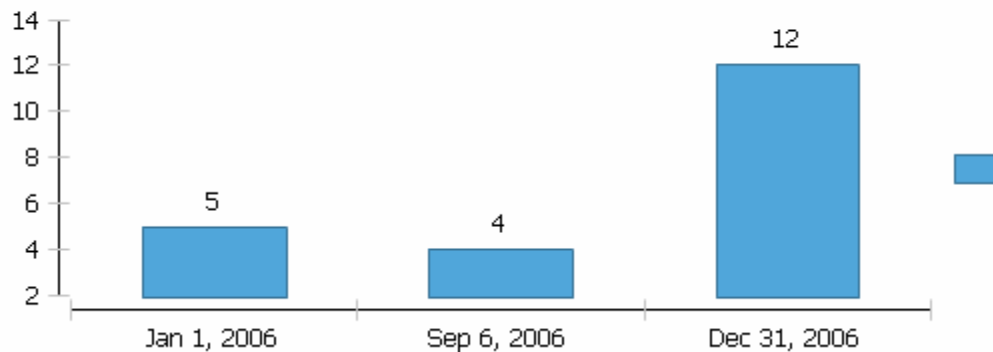
2.3.3   Mockup

3.   Axis

## 3.1   Linear Date/Time axis support for Line/Bar chart

3.1.1 Definition

Currently, only categoried-datetime axis is supported for Line/Bar chart. That means there is one data point per axis major unit, and without axis scale support.

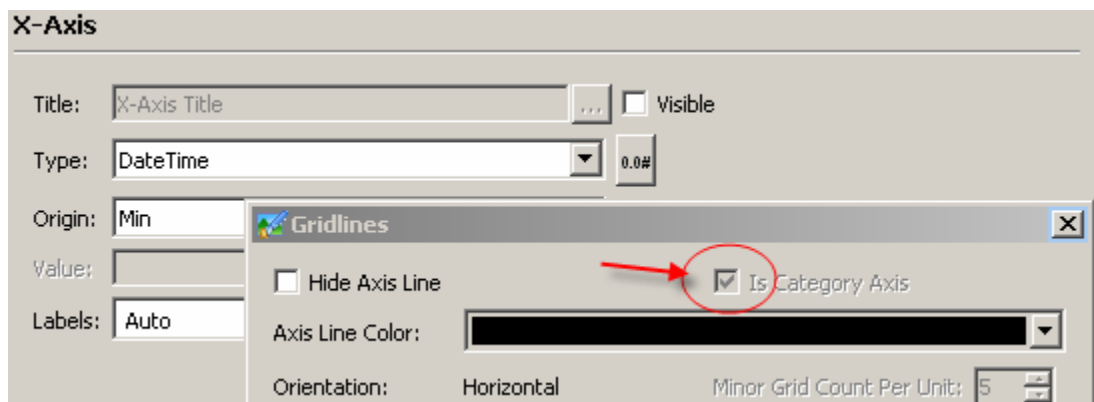A linear (non-categoried) datetime axis can support custom scale setting, and compute intermediate units automatically. The data point is placed in the axis by value of each point. If no data point relevant to particular major unit, it will be left as blank.
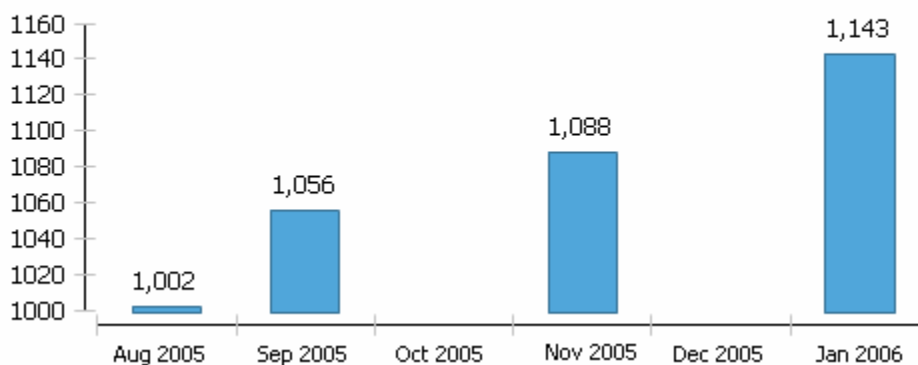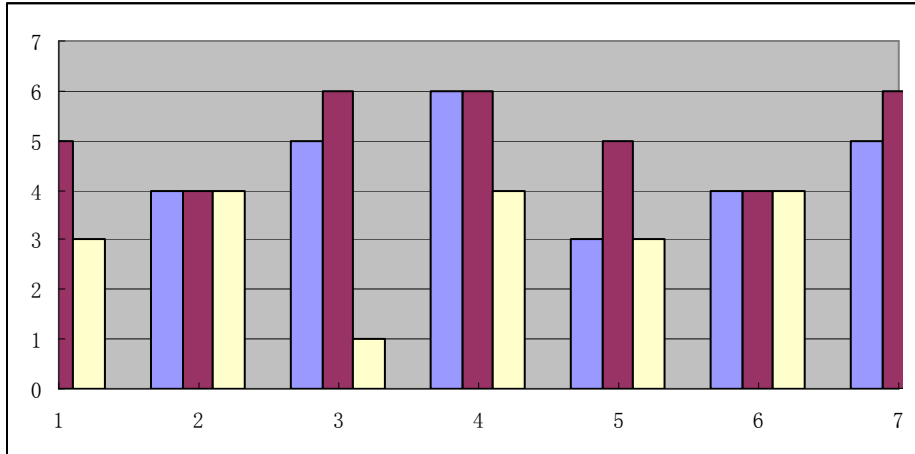

3.1.2 UI Mockup

It should allow user to uncheck this option when axis type is datetime. The default will be unchecked (linear).

3.2 Tick position between or across categories

3.2.1 Definition

The tick position on the X Axis should be configurable to be positioned across or between categories. Currently, this is automatically decided by whether the axis is categoried or text style. A new setting will control that behavior.

3.2.2 Bugzilla Entry

Display X-axis labels at tick marks rather than in between

https://bugs.eclipse.org/bugs/show_bug.cgi?id=146114

3.2.3 Model Change

```
<xsd:complexType name="Scale">
    <xsd:annotation>
        <xsd:documentation xml:lang="en">
        This type defines the scale associated with an axis.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
        <xsd:element name="Min" type="data:DataElement">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                Minimum value that should appear on the axis.
                </xsd:documentation>
            </xsd:annotation>
        </xsd:element>
        <xsd:element name="Max" type="data:DataElement">
            <xsd:annotation>
                <xsd:documentation xml:lang="en">
                Maximum value that should appear on the axis.
                </xsd:documentation>
            </xsd:annotation>
```
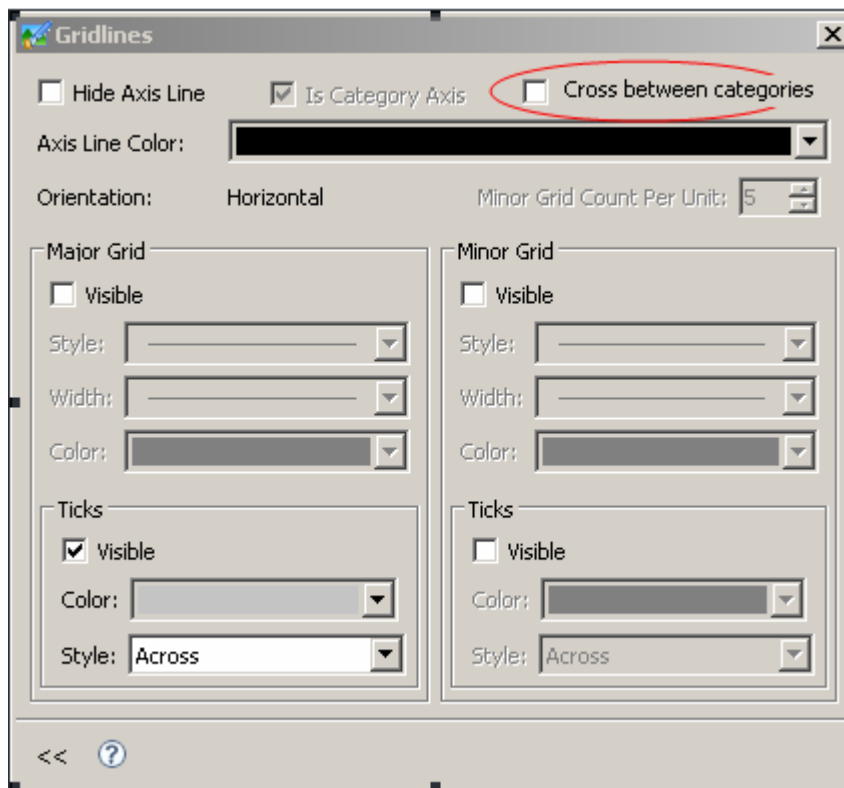
```xml
                        </xsd:element>
                        <xsd:element name="Step" type="xsd:double">
                                <xsd:annotation>
                                        <xsd:documentation xml:lang="en">
                                        Stepping in the values shown on the axis.
                                        </xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                        <xsd:element name="Unit" type="attribute:ScaleUnitType">
                                <xsd:annotation>
                                        <xsd:documentation xml:lang="en">
                                        Stepping in the values shown on the axis.
                                        </xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                        <xsd:element name="MinorGridsPerUnit" type="xsd:int">
                                <xsd:annotation>
                                        <xsd:documentation xml:lang="en">
                                        Specifies the number of minor grids per unit of the scale.
                                        </xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                        <xsd:element name="TickBetweenCategories" type="xsd:boolean" minOccurs="0">
                                <xsd:annotation>
                                        <xsd:documentation>Specifes if the major tick is rendered at the category value or between two categories. This only affects the category and text style axis.</xsd:documentation>
                                </xsd:annotation>
                        </xsd:element>
                </xsd:sequence>
        </xsd:complexType>
```
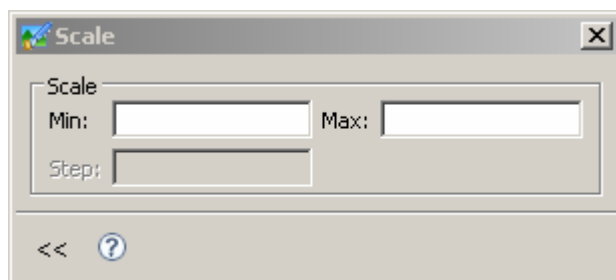
3.2.4 UI Mockup

The new option should only be enabled when the axis is category style or of text type.

4. Scale

4.1 Date/Time type Scaling support

4.1.1 Definition

Currently, for Datetime type axis, only minimum and maximum value can be specified for scale setting. Step and unit is not supported.

Chart engine should be enhanced to support fixed datetime step and unit. If no fixed value specified, auto scaling will still be used.
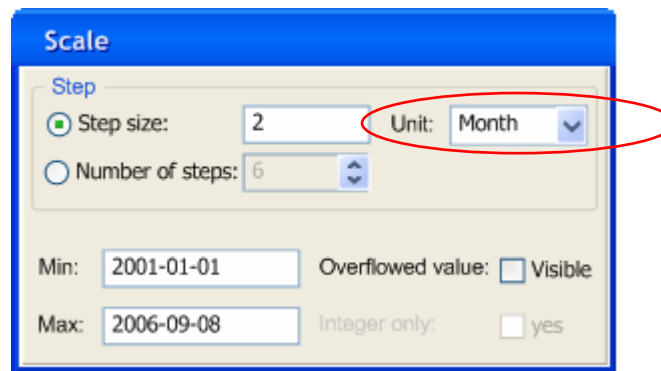
Currently, the minimum and maximum datetime value are constrained to be the "MM-dd-yyyy HH:mm:ss" format for input in UI. This is not very convenient, for usually, the time part is not used. So the UI should also be enhanced to be able to recognize several commonly used datetime format, for instance, current locale format, standard format, date format without time part, etc.

4.1.2 Bugzilla Entry

Setting min and max scale values causes improper data point plotting

https://bugs.eclipse.org/bugs/show_bug.cgi?id=128739

4.1.3 UI Mockup

4.2 Adding an option to hide overflowed data

4.2.1 Definition

When the value of datapoint is out-of-scale, i.e. the value is higher than Max or lower than Min in Scale, the overflowed value will display in the boundaries. This solution may confuse some users.

New solution is to hide overflowed data in chart preview, and only display the value within the scale boundaries. Users can switch these two choices by a checkbox in Scale UI to decide whether to display overflowed value or not.

4.2.2 Bugzilla Entry

Setting min and max scale values causes improper data point plotting

Bugzilla#128739.

4.2.3   UI Mockup

4.3   Adding an option to specify number of steps

4.3.1   Definition

Besides using step size to partition the axis, users may want to specify the number of steps instead.

In UI, add a spinner to adjust the number of steps as an alternative choice to define steps of Scale. Once users choose to specify the number of steps, step size option is disabled, vice versa.

4.3.2   Bugzilla Entry

Specify number of steps instead of step size

Bugzilla#152976.

4.3.3   UI Mockup

4.4   Adding an option to display Integer numbers only

4.4.1   Definition

When displaying line or bar chart with a linear Y axis, users may want to step by integer values only. That would mean only displaying integer values on the axis.

In UI, add a checkbox to determine whether only integer numbers display on the axis or not. If the Axis type is non-linear, this choice will be disabled.

4.4.2 Bugzilla Entry

Decimal values displayed on linear axis

Bugzilla#142866.

4.4.3 UI Mockup

4.5 Adding Date picker to input DateTime value

4.5.1 Definition

Provide a Data picker next to the input box for convenient of inputting DataTime value. The appearance of Date picker just likes the one in Windows system.

4.5.2 UI Mockup

4.6 Adding indications for empty value

If the input box is empty, it's better to display an indication rather than blank. The indication could be "Automatic" in grey. If users click the input box, the grey indication will become blank. When the input box loses the focus and the input box is still blank, the grey indication will display again.