

	Net Sales	Billed Quantity	abs	perc
South East	59.1m	50.8m	+8.3m	+16.4%
West	14.9m	15.4m	-455.2k	-3.0%
Central	38.2m	41.1m	-2.9m	-7.1%
North East	6.0m	5.6m	+391.3k	+6.9%
<b>- Internet</b>	<b>118.2m</b>	<b>112.9m</b>	<b>+5.3m</b>	<b>+4.7%</b>
Central	47.8m	54.5m	-6.7m	-12.4%
West	13.9m	13.2m	+702.9k	+5.3%
South East	70.1m	53.9m	+16.2m	+30.1%
North East	4.1m	4.1m	-56.3k	-1.4%
<b>- EDI</b>	<b>135.8m</b>	<b>125.7m</b>	<b>+10.1m</b>	<b>+8.1%</b>
Central	61.9m	47.0m	+14.9m	+31.6%
West	17.9m	15.8m	+2.1m	+13.4%
North East	11.9m	26.7m	-14.7m	-55.3%
South East	70.5m	64.1m	+6.5m	+10.1%
<b>- Phone</b>	<b>162.2m</b>	<b>153.5m</b>	<b>+8.7m</b>	<b>+5.7%</b>
<b>- Summe</b>	<b>416.2m</b>	<b>392.1m</b>	<b>+24.2m</b>	<b>+6.2%</b>

Version 2020.4 – as of October 2020

<https://www.graphomate.com>



## Table of contents

- Introduction
- Installation
- Quick Start
- Properties
- Standard Properties
- Interactivity at Runtime
- Known Issues
- Number Formatting
- CSS Selectors

## Introduction

The topic of visualization is becoming increasingly important for a fast and safe communication of information. Simple but meaningful representations of information support decision makers in the finding correlations, patterns or outliers. Good information design gives you a fast and effective overview of your business.

The International Business Communication Standards (IBCS) are a set of rules by Rolf Hichert that serves as a guideline how to use meaningful visualizations in business reporting correctly. In addition to our *graphomate charts* and *graphomate bullet graphs* we provide the *graphomate tables*, a further component for SAP Lumira Designer which is also based on the IBCS rules for displaying informative tables.

In the sense of a uniform language regulation we speak in the further manual of Lumira Designer or briefly Designer.

Charts, as they are realizable with our graphomate charts extension, can be interpreted quickly and efficiently by our eye-brain vision system. They do not have to be "read" like tables. But tables have the big advantage that they represent lots of information in very limited space. They are perfectly suited for presentation of detailed data and therefore are very important in the usage of BI systems.

Dashboards and BI applications should follow the visualization mantra of Ben Shneiderman: *"overview first, zoom and filter, then details-on-demand"*!

Charts provide a quick overview, whereas tables show detailed data from the end of the navigation path of a dashboard application. Our table component focuses on easy to read information design according to the rules of HICHERT@IBCS.

Multiple dimensions in rows of the Initial View are represented as (implicit) hierarchy. If an (external) BW hierarchy is used on a dimension, only this one dimension can be used with the *graphomate tables*. Please use measures and no further dimensions in columns and only select complete columns for each data series. Furthermore make sure that all cells to be displayed are delivered by the underlying Data Source: e. g. all hierarchy levels must be visible – otherwise the *graphomate tables* show wrong results.

The *graphomate tables* are not an analytic component. We highly recommend not to display more than 1000 cells with *graphomate tables*.

For multidimensional analyses please use the Lumira Designer Crosstab standard component.

For our *graphomate tables* extension Lumira Designer >= 2.0 SP03 and IE 11+ are required. For further information, for example about the deployment on the Business Intelligence Platform (BIP) or SAP Netweaver, please consult the Product Availability Matrix.

Please note that we will no longer support Internet Explorer 9 and 10 from 2017.

## Installation

- Before opening your old dashboards/BI apps with Designer 2.0, you need to migrate them to m mode with Design Studio 1.6.
- Our extensions for Design Studio 1.6 will also run in Designer 2.0. When starting Lumira Designer 2.0, please do not import our graphomate extension from SAP Design Studio 1.6. Instead we suggest using the versions specifically built for Designer 2.0.

### Local installation of the extension to Designer

You have installed Designer 2.0 SP03+ on your computer.

1. Save the zipfile *graphomate\_tables\_2020.4.x\_LumiraDesigner.zip* to a folder of your choice.
2. In Designer, choose *Tools Install Extension to Lumira Designer*, click on *Archive* and select the zipfile which has been saved before.
3. Choose *Finish* to start the installation.
4. Choose *Next* and again *Next* to confirm the installation.
5. Accept the terms of the license agreement and choose *Finish*.
6. Choose *Yes* to allow a restart of Designer.
7. After the restart, the graphomate extension appears in the Component View of Designer.

### Removing the extension from Designer

In Designer choose *Help > About...*

1. Click the *Installation Details* button.
2. Select the component *graphomate tables 2020.4.x*
3. Choose *Uninstall...*
4. In the *UninstallWizard* choose *Finish*.
5. Choose *Yes* to allow Designer to restart.

### Server installation of the extension

Users need to deploy the locally installed extension to the BI platform before launching the *graphomate tables* from the BI platform.

1. Choose in the BI Platform Mode *Tools > Platform Extensions*.
2. Select the extension *graphomate tables* that is installed on your local computer.
3. Choose *Install on Platform*.
4. Manually restart the BI Platform Adaptive Processing Servers that host the Analysis Application Service and accept the warnings in Designer.
5. The graphomate extension appears under *Extensions Installed on Platform*.
6. Choose *Close*.
7. Again restart the BI Platform Adaptive Processing Servers that host the Analysis Application manually.

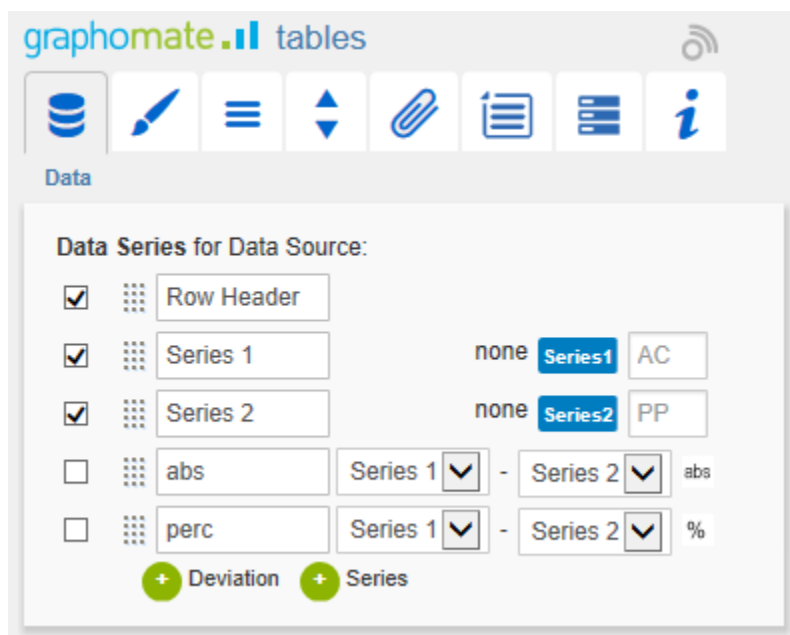
### Uninstalling the extension from the Server

1. Choose in the BI Platform Mode *Tools > Platform Extensions*.
2. Choose the *graphomate tables*.
3. Choose *Uninstall from Platform*.
4. Confirm the uninstalling by choosing *Yes*.
5. In order to complete the uninstalling process, manually restart the BI Platform Adaptive Processing Servers that host the Analysis Application Service and accept the warnings in Designer.
6. The *graphomate tables* is then removed from the *Extensions Installed on Platform*.
7. Choose *Close*.
8. Manually restart the BI Platform Adaptive Processing Servers that host the Analysis Application Service.

## Quick Start

You have defined a Data Source on basis of a SAP BW Query and now want to depict Data Series from this query with a *graphomate table*.

1. Drag the component *graphomate tables* to the canvas and bind the Data Source to this component via drag&drop.
2. Define the Data Series (column) in the Data tab by clicking the corresponding button (e. g. **Series 1**) in the Initial View. To choose a Data Series just click on the header of the column. The selected Data Series is highlighted in blue and shown in the table after confirmation (click **OK**).
3. The row titles in the first column of the table are *automatically* read from the metadata of the selected Data Series.
4. If you have defined Data Types and want to show them in the column headers of the table, enter the corresponding abbreviations for each column into the section *Data Types*.



## Properties

Basically, there are two possibilities to maintain the features of *graphomate tables*: with the *Standard Properties Sheet*, which lists all parameters in a structured list or with the *Additional Properties Sheet*. The latter provides a user-friendly interface. There are some special features, which can only be maintained on the *Standard Properties Sheet*.

The Checkboxes  next to the features in the *Additional Properties Sheet* are used to control the visibility of elements.

The chosen Data Source can only be attributed to the component via drag&drop or via the *Standard Properties Sheet* under *Data Binding*.

Clicking the **F5** key reinitializes the canvas as well as the *Additional Properties Sheet*. This is sometimes necessary when changes in the Properties are not directly accepted in the canvas.

### Data Tab

#### Data Series

Here you can link up to 32 Data Series from the Data Source that you want to display by clicking the corresponding button and then choosing a column or row in the following window.

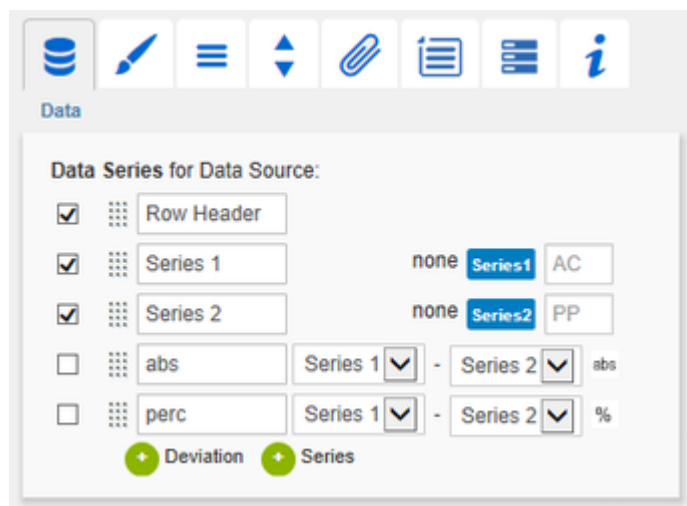
The Checkbox next to the *Deviation/Series* determines its visibility in the table. Change the name of the *Deviation/Series* by clicking on it. The given names are used as column headers in the table.

Use the input field following the series selection button to enter a *Data Type* for the Data Series. The selected Data Type will be displayed as a bar beneath the corresponding column header (styled as configured on the tab Data Types). Specified data types are internally converted to capital letters to eliminate problems with faulty inputs.

It's possible to visualize deviations of two Data Series within the *graphomate tables* with so called micro charts. The deviation configuration is integrated in this tab: Use the dropdown boxes to choose Data Series for the calculation of deviations according to the formulas on the right. Click the button *abs/%* to select absolute – visualized by bars – and percentage deviation – visualized by pins.

If display attributes are added in the initial view these are automatically inserted as additional columns in the table (between the row titles and the actual data). The heading of the attribute column is then the attribute dimension. Showing and hiding the attributes is only possible in the initial view.

The micro charts in deviation columns are formatted automatically according to the chosen Data Type: bar and pin elements use the Data Type of *DataSeries1/ACT*, the axis – if *Semantic Axis* is activated on the *Helper* tab – uses the Data Type of *DataSeries2/BUD*.

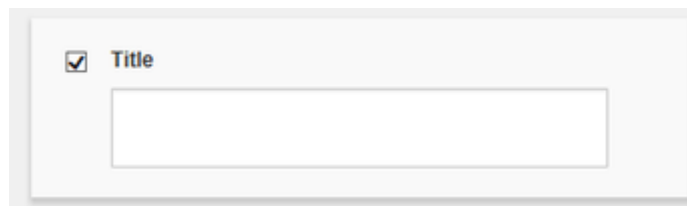


Formula for calculating absolute deviation:

$$(Series1 - Series2)$$

Formula for calculating percentage deviation:

$$\left( \left( \frac{Series1 - Series2}{abs(Series2)} \right) * 100 \right)$$





By clicking the green buttons you are able to add a new Deviation Series or a new Data Series. By clicking the red button which appears behind each series while hovering over the line, the according series is deleted. Series 1 and 2 are not removed by clicking the delete button, but only cleared in order to assign a new Data Series.

With the dotted area in front of each line you can drag and drop the according data assignment to a new position. The resulting order is used as a column layout for the table.

The row headers can be moved to any position within the table in this way and enable so-called *butterfly tables*. The visibility of the row titles can be set in the same way as the series and deviations via the checkbox.

The position of the attribute columns can currently only be set via the *Column Order* property in the *standard properties*. However, as soon as a setting is made in the default properties, it is overwritten by loading or changing the *Additional Properties Sheet*.

### Title

Enter a title for your table at this point. If the Checkbox is activated, the title will be shown. Line breaks are adopted.

## Appearance Tab

### Labels

Select the appearance of the font of the table.

### Font Size

Sets the font size. The font size influences implicitly the dimensioning of the table. Therefore parameters are specified in the unit "fs" (font size) in order to be dependent on the font size. Some of these parameter values can be maintained via the *Standard Properties Sheet*.

### Font Family

Sets the font family. Choose between *Arial*, *Lucida Console*, *Tahoma* and *Verdana* via the drop-down box.

### Text Color

Sets the color of the text (including the *title* text). If a hierarchy is used within the table, the color defined in the section *Hierarchy Nodes* (or by the *Property Node Style Color*) is set for those elements. If a Data Type is assigned to a column and the option *Apply Data Types to Cells* (in the section *Columns*) activated, the specified color of the corresponding Data Type is applied for the entire column and thus overrules the text color that is specified here.

### Value Format

Select the format of the table values:

The screenshot displays the 'Appearance' configuration panel for a table. At the top, there is a toolbar with icons for database, edit, list, zoom, link, table, and help. The main panel is titled 'Appearance' and contains several sections:

- Labels:**
  - Font Size (in px): 14
  - Font Family: Arial
  - Text Color: Black
- Value Format (Basic):**
  - Percentage: 0.0%
  - Absolute: 0.0a
  - Locale: en
- Value Format (Extended):**
  - Decimal Sep.: ,
  - Thousands Sep.: .
  - Negative Sign: -
  - Scaling: 1
  - Decimal Places: 1
  - Prefix:
  - Suffix: %
- Hierarchy Nodes:**
  - Color: Black
  - Bold
  - Italic

- **Basic**  
Select the format of the absolute and percentage values according the guidelines of numeral.js.  
*Locale* sets the label country format in the Basic mode. DE, FR, EN and AUTO are available options. Choosing AUTO the *Locale* is retrieved from the Data Source.
- **Extended**  
If a fixed format is preferred, put the format for absolute and percentage values here.  
Choosing a “(“ in the box *Negative Sign* will show negative values in brackets: (1234). The parameter in the box *Scaling* is used as a divisor for scaling the data label. The resulting “format string” can also be set via scripting language.

Set the *Use Formatted Data* property in the *Standard Properties Sheet* to "true" if you want to show the data according to format of the Data Source.

## Hierarchy Nodes

Here you can control Color and font style of the Hierarchy Nodes in the table. The color overrules the font color set in the section *Text Color*.

## Columns

*Data Types in Header* activates bars in the column headers showing the Data Type. *Apply Data Types to Cells* sets the color of the corresponding Data Type for the whole column and overrules all other color settings.

## Rows

Row headers of the first column can be aligned to the right by activating the Checkbox *Title Alignment Right*.

## Deviations

Set the colors for deviations. If positive deviations between values are not regarded as positive, for example for comparisons of actual vs. budget costs, activate the Checkbox *Invert*.

With the option *Standard Pin Heads*, the pin heads of the percentage deviations are displayed the legacy way or according to the IBCS.

**Columns**

Data Types in Header

Apply Data Types to Cells

**Rows**

Title Alignment Right

**Deviations**

Good ■

Bad ■

Invert

Standard Pin Heads

**Outlier Mode**

Short Long

**Outlier Mode**

Choose the appearance of the outliers here. In the mode *Short* the outliers are shown as small triangles on the axis. In the mode *Long* they are drawn over the entire available area – according to the rules by IBCS.



## Behavior Tab

### Scaling

The default *Auto* scales the table according to the *Min* and *Max*-values of all selected data.

The deviation charts of several tables can be scaled identically by using the same Comparison Group, which is activated by the Checkbox. The Comparison Group will then be shown during design time on the top right corner of the table. That way the *Min* and *Max* of the data basis in this group is used. Additionally, enabling the Comparison Group of a single table scales deviation charts of that table identically.

Comparison Groups are used **globally** within the BIApp, i. e. also charts in the *graphomate charts* are scaled identically by using the same Comparison Group abbreviation.

Alternatively it is possible to set the *Min* and *Max* scaling values under Manual. Chart Min values > "0" cut the value axis. Retrieve values proposes valid values!

### Column Width

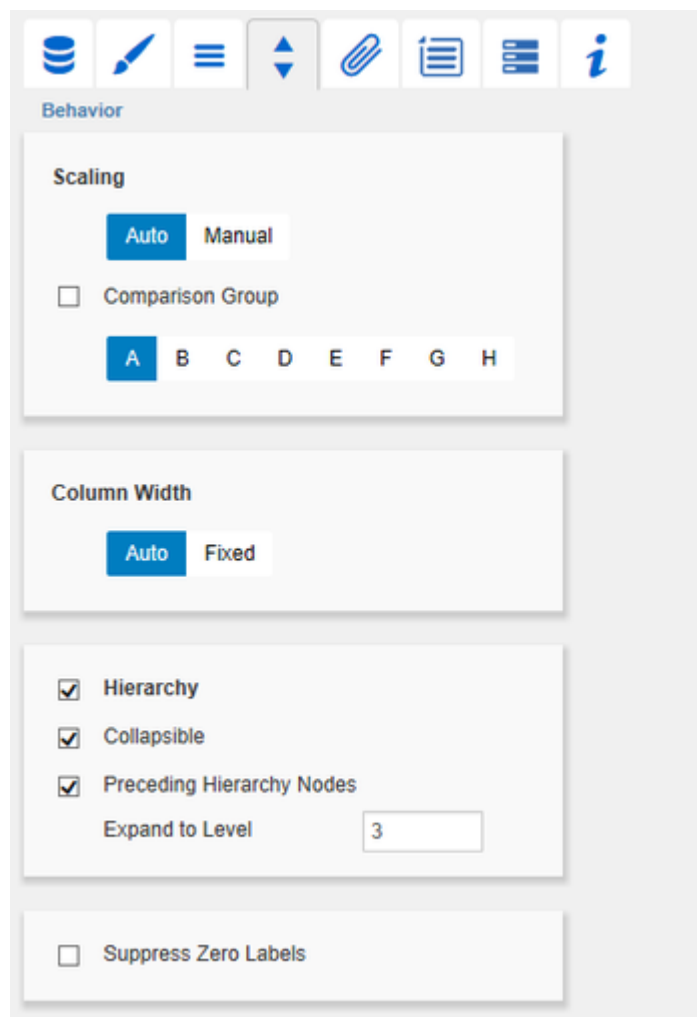
- *Auto*  
In Auto mode the available space (width of the component) is divided equally between the columns.
- *Fixed*  
Enter a fixed width for the columns in [px] here.

Use the property *Width Per Column* in the *Standard Properties Sheet* to set the width of each column independently.

### Hierarchy

Activate the Checkbox *Hierarchy* to show the hierarchical order of the dimension. If you want to drilldown and roll-up within the hierarchy during runtime, activate the Checkbox *Collapsible*.

*Preceding Hierarchy Nodes* controls the position of the totals in the table: above or below the table values.



*Expand to Level* determine up to which hierarchy level the hierarchy is to be expanded in the start drilldown (starting with level 0).

It's imperative that external BW-hierarchies used in a Data Source are fully expanded, and all (sub-)totals are displayed.

Failure to do so may result in misstatements!

### Suppress Zero Labels

If the Checkbox is activated, all zero values are hidden – both in the table itself and in the deviation charts.

### In Cell Charts

Use drag-and-drop to move the desired series from *Available Series* to *Series with In Cell Charts* so that bar graphs are displayed in the corresponding columns to visualize the value. Positive values are colored according to the *Positive Color*, negative values corresponding to the *Negative Color*. The bars are provided with a transparency. The strength of the transparency can be influenced by the *Bar Opacity* (values between 0 and 1 are possible). In addition, you can adjust the height of the bars in relation to the size of the table cell using the *Bar Height Factor* (values between 0 and 1 are possible).

### Heatmap

If this option is activated, the value columns of the table are displayed as heat maps, i. e. with a colored background. For this purpose, all values of the table are mapped to a color scale, which can be set under *Color Scheme*.

In addition, all color scales defined via ColorBrewer can be realized in the Standard Properties. The definition of a color scheme in the *Standard Properties Sheet* consists of two comma-separated parts: The abbreviated color scale (e.g. YlOrRd for "Yellow to Orange to Red") and the number of gradations.

**In Cell Charts**

Available Series:

Series 1 Series 2

Series with In Cell Charts:

Bar Height Factor: 0.8

Bar Opacity: 0.5

Positive Color: ■

Negative Color: ■

Heatmap

Color Scheme: Red to Yel ▼

**Highlight Group**

Highlight Group

Highlight Group Color: ■

Match Measures

Both the In Cell Charts and the Heatmap can cause the bars of different columns to overlap. If this is the case, this is probably due to the fact that the width of the columns has been configured too small. Please check whether an enlargement of the column width (in particular the line title) causes the overlap to disappear.

### **Highlight Group**

Activate the *Highlight Group* to match corresponding bubbles. For detailed information see Runtime Interactivity.





## Helper Tab

### Horizontal Grid Lines

Activate the Checkbox *Horizontal Grid Lines* to display these. Set the *Thickness* and *Color* of the grid lines with the following two parameters.

### Sum Borders

Activate the Checkbox in order to display the Sum Borders. You can specify the *Thickness* and *Color* of the Sum Borders. With *Padding* a distance between the sums and the child elements is added.

We recommend to deactivate the option *Preceding Hierarchy Nodes* (see Behavior Tab) when using the *Sum Borders*, so that the sums of elements are listed below the values.

### Deviation Axis

In this section adjustments to the Deviation Axis can be done. You can specify the *Thickness* of the Deviation Axis.

*Semantic Axis* enables to draw the deviation axis in accordance with the Data Type associated to Data Series2. The Checkbox in front of *Deviation Labels* controls whether the textual values should be displayed. If this option is deactivated, there are only graphical bars.

### Filter Box

In the top left cell of the first row of the graphomate tables we implemented a filter function.

Activate the Checkbox *Filter Box* to make the Filter Box visible for the users. Just enter some letters or regular expressions in the Filter Box and only those table rows are shown that contain these letters. Upper and lower case is not considered.

### Exception Position

*Exception Position* describes where the exceptions are displayed in the table. The conditions for displaying exceptions can be configured via the following Row Format Configuration. Selectable positions are: at the beginning of the cell (*leading*), *before* or *after* the value.

The screenshot shows the 'Helper' configuration panel with the following settings:

- Horizontal Grid Lines:**
  - Horizontal Grid Lines
  - Thickness (in em): 0.1
  - Color: [Grey swatch]
- Sum Borders:**
  - Sum Borders
  - Thickness: 0.1
  - Color: [Black swatch]
  - Padding (in em): 0
- Deviation Axis:**
  - Thickness (in px): 2
  - Semantic Axis
  - Deviation Labels
- Filter Box:**
  - Filter Box
- Exception Position:** leading (dropdown menu)

## Row Format Configuration

The *Row Format Configuration* enables detailed layouting of rows and columns.

This new layouting feature replaces the *Exception Configuration*, which could have been found at the same position in previous versions. The Exceptions are now just one of many styles (see below) provided by the *Row Format Configuration*.

The Checkboxes in front activate the visibility of each Configuration. Enter a *Name* for the special Row Format Configuration in the next column. As it will be used internally and for the scripting language you should choose a unique name. *Target* determines the target column in which the *Row Format Configuration* should be displayed. Target refers to the index of the column (Target 0 describes the first column containing the row titles).

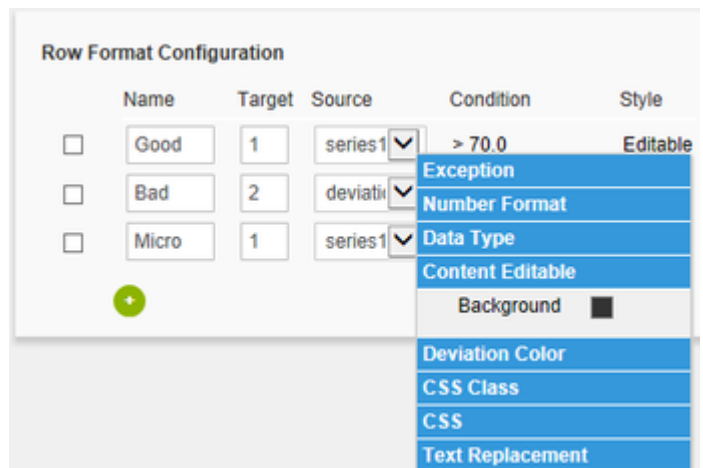
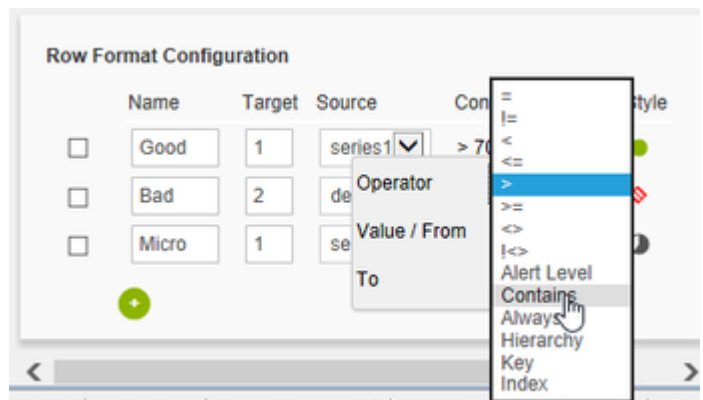
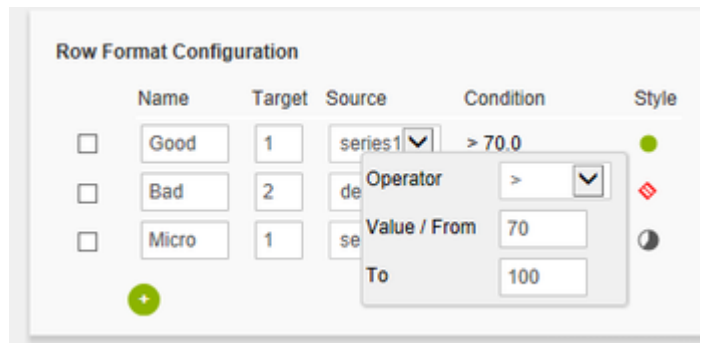
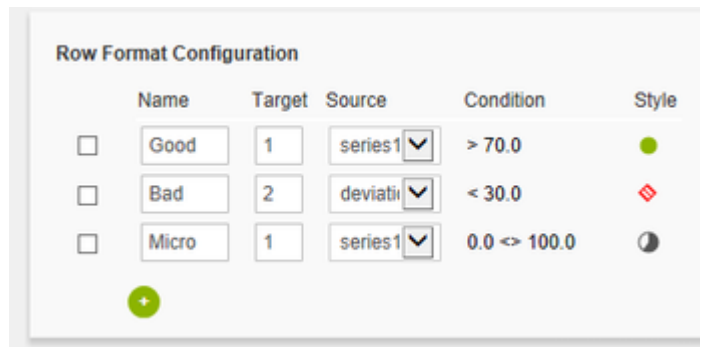
Apart from setting a single column you can even chose more columns as *Target* for the used Style. There are two other methods to set a range of columns: The keyword *all* is used to pick all columns. The second way is to enter a list of explicit column indices separated by comma (e.g. 2,3).

As there are some Styles which are not applied to all columns, please see below if the chosen Style does support layouting all columns.

The *Source* column is used to calculate or check the chosen *Condition*. You can pick a column using a single index. Unlike in older versions you can now also choose columns containing titles, attribute or deviations as *Source* column.

To specify the *conditions* that lead to the display of an Row Format click on the value under the heading *Condition*. Select the condition by choosing an Operator and a span.

When applying comparison operators to data cells, the underlying numerical value is always considered, never the formatted string. For percentage deviations, it is the converted decimal value (50% corresponds to 0.5). In addition to the usual mathematical operators for value comparisons, a few other options are available:



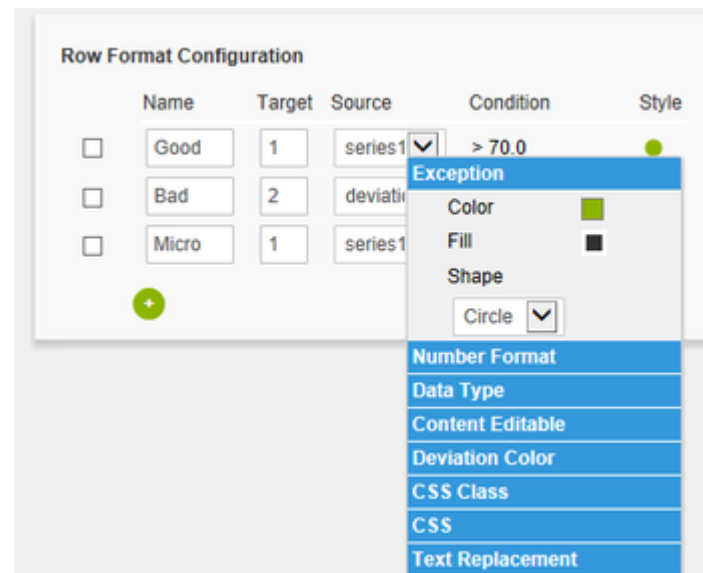
- **Alert Level**  
Can be used to take over an alert level from the exceptions of a BW query. Additionally you need to input an alert level index into the text field named 'Value/From' (e.g.: use '1' to apply alert level 'good 1' or '7' to apply alert level 'bad 1')
- **Contains**  
Can be used to check if a specific string (which can be specified in Value/ From) is contained with another string. It is also possible to define conditions using regular expressions.
- **Always**  
This condition will always be applied, no matter which value is specified.
- **Hierarchy**  
Can be used to select hierarchy levels. In doubt, the hierarchy level of a row can be read from the assigned CSS class.
- **Key**  
If the key entered in the value field is identical to the key of the source cell, the row format configuration is applied.
- **Index**  
Applies the configuration directly to the row with the specified index. It is possible to enter either a single index (3), a comma separated list (1,3,4,5) or a range (1..5).

This *Condition* together with the *Source* cell is used to restrict the layouting of the *Target* cell, so it directly controls if the *Style* is applied or not.

In order to define the *Style* of an *Row Format Configuration* click on the icon under the heading *Style* and choose between: *Exception*, *Number Format*, *Data Type*, *Content Editable*, *Deviation Color*, *CSS Class* or *CSS*.

- **Exception**: By clicking *Exception* you are able to set the *Color*, the *Fill* and the *Shape* of the displayed *Exception*.

The *Shape* "micro pies" visualizes a part-to-whole relation. The colored part is calculated as the percentage of each cell value of the total value. The total and cell value is based on the exception config's source.



- *Number Format:*  
Choose which *Value Format* (*basic* or *extended*) should be applied to the target column. If you want to use the *basic Value Format*, it should look like a Number Format according to numeral.js, i.e. "0.00a". (13 423 253 13.42m)  
The use of the *extended Value Format* requires an input statement with this format: "-||,| |10^6|2|mio" .
- *Data Type*  
Draw the notation of the Data Types into the headers. **Restrictions:** Only works for the row title, so the Target needs to be 0 or all.
- *Content Editable*  
Enables the user to change the content of the chosen cell and signals that with a colored background. Changing the cell content triggers the event *On Input Changed*.
- *Deviation Color*  
Offers the possibility to invert the *Deviation Color* for chosen cells. There is no need for further configurations with this Style. **Restrictions:** This options only affects columns containing deviations..
- *CSS Class*  
Sets the given CSS class or classes to the chosen cells. More than one class can be set by using a list separated by commas (i.e. *class1 ,class2,class3*).
- *CSS*  
Applies the given CSS properties to the chosen cells. **Restrictions:** There is no guarantee that all properties are applied. Some settings like the Text Color can be set in several locations, which may be considered with a higher priority (the setting with the highest priority is applied in the end). The CSS must be specified as follows, making sure that no spaces after the colons and no semicolons are used:
  - A single property: *color:red*
  - Several properties at once: *color:red, background:grey,font-family:MS Comic Sans*
- *Text Replacement*  
Replaces the selected cells with the specified text.

With the red button you delete Row Formats and with the green one you add new ones.

## Context Menu Tab

In this tab you will be able to choose which options are available in the context menu. An activated checkbox means this options will be shown. The first checkbox enables or disables the context menu altogether. Once activated the context menu will be accessible during runtime by right clicking on the component. The active operations will be usable as follows.

### Sorting

With Sorting data from the available dimensions can be sorted in ascending or descending order.

### Filters

Allows you to filter the data by member for all available dimensions.

### Context Sensitive Filters

Shows filters corresponding to the currently clicked dimension.

### Filter and Hide

In addition to the normal filtering this operation also removes the corresponding dimension from our initial view.

### Clear Filters

Resets previously set filters.

### Drilldown

Adds another dimension to the component.

### Drillup

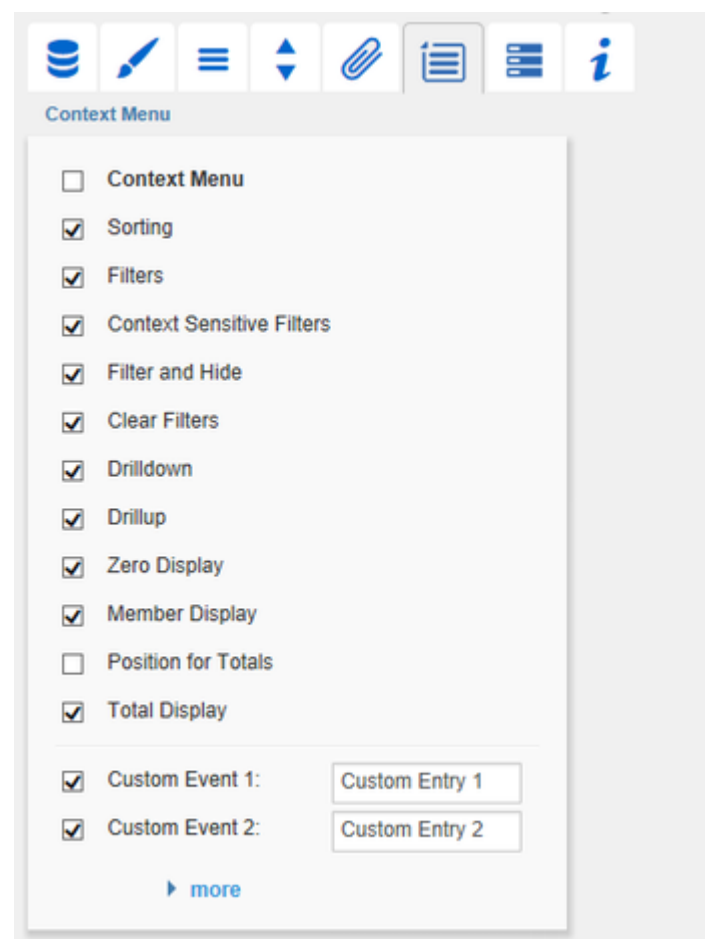
Removes a dimension from the chart (this can cause an error which makes a reload necessary).

### Zero Display

Sets the way zero values are displayed.

### Member Display

Lets you choose the way member names are displayed (text or key).



### Position For Totals

Is disabled by default, since this option has no effect on the *graphomate tables*. Also activating the option has no effect.

*Position For Totals* affects only the *graphomate charts*.

### Total Display

Enable or disable totals for each dimension.

### Custom Events

There are 10 possible events which can be linked to your own scripts. These can be defined in the *Standard Properties Sheet* and triggered using the corresponding button in the context menu. In the *Additional Properties Sheet* the names for these buttons is customizable.

All operations are executed directly on the data source, but without changing it.

A reload resets all changes made with the context menu.

## Server Tab

### Server Configuration

Here you can enter the *Server URL* to which the component should connect and from which it should retrieve the templates.

If you have specified the URL to the server, a check mark indicates that a connection could be established.

If another server is used later, you can also press the button next to the Save button and edit the *Server URL*.

The hyperlink below the buttons allows you to jump to the admin.

### Templating Configuration

If the current settings are to be saved as a template, a new template name can be entered in the input field with the name *Create Template* and confirmed by clicking the *Save current State* button.

If the naming convention entered is correct, the template is saved on the server and a toast with the corresponding message is displayed in the lower part of the Additional Property Sheet. However, if the name entered does not meet the expected naming conventions, a toast with the permitted characters is displayed accordingly.

Templates stored on the server can be retrieved and applied via the Template List. Just select the desired template and press the *Apply* button. If you want to delete a template, select the template in the list and press the *Delete* button. If changes have been made to the templates on the server side, the list must be updated via the *Refresh List* button. Changes should then be visible.

The screenshot shows a user interface for server and template management. At the top, there is a toolbar with icons for database, edit, menu, expand/collapse, link, list, server, and help. Below the toolbar, the interface is divided into three main sections:

- Server Configuration:** Contains a text input field for "Server URL" and a "Save" button with an edit icon.
- Create Template:** Contains a text input field for "Name" and a "Save current State" button.
- Template List:** Contains a dropdown menu for selecting a template and three buttons: "Apply", "Delete", and "Refresh List".




### Info Tab

Here you can find the version number of our *graphomate tables* as well as a link where you may report bugs or propose wishes for new features.

A list of all open-source libraries that we used can be found under *Credits*.

graphomate .ll tables

Info

graphomate .ll 

tables Version: 2.1.0.7 Build: 201808021401 (IE 11)

Please use [this link](#) to submit errors or ideas for improvement.

The [General Terms of Licence and Maintaining of the graphomate GmbH](#) apply.

© 2017 graphomate GmbH, D-Kiel,  
[www.graphomate.com](http://www.graphomate.com)

Credits

▶ [more](#)

## Standard Properties

Some (expert) features can only be controlled in the *Standard Properties Sheet*.

### Context Menu

#### Save Selected Member On Right Click

When clicking a data cell with the left mouse button, member information about the clicked cell gets stored and will be retrievable using the script method `getSelectedMember`. If this property is activated, a right click will store member information as well. This is helpful for processing the member information of a right clicked cell in custom context menu script events.

### Hierarchy

#### Apply Node Style to Nodes Without Sums

If totals are hidden for a dimension of implicit hierarchies (= hierarchies composed of several simple dimensions), the hierarchy nodes of this dimension are displayed in bold. This is the new default behavior since v2.2.0 of the tables. To restore the previous behavior in which the corresponding hierarchy nodes were not displayed in bold, this property can be set to `false`.

#### Save States

If this setting is activated, the state of the hierarchy is restored after a redraw (for example if triggered by a script function). This option makes complex workarounds with the script language and the *Hierarchy States* obsolete. To use this feature, the Additional Properties Sheet must be opened once to generate a unique identifier for the table. This happens automatically in the background of the Additional Properties Sheet.

#### Unique Identifier

This property is not relevant in most cases. It contains a unique identifier of the table that is used to store the hierarchy states in the browser memory when the *Save States* property is enabled. Normally, an ID is automatically generated when the Additional Properties Sheet of a table is opened for the first time. However, if two tables for which the *Save States* property is enabled have the same identifier (e.g. because one table is a copy of the other), it is necessary to adjust it manually so that the two tables differ in their identifiers. Otherwise they would influence each other when saving their hierarchy states.

#### Use Hierarchy Marker

If this property is set to `true` only the *Hierarchy Marker* (+/-) can be used to open or close hierarchies. Otherwise (set to `false`) the user may also click on the row title to open or close hierarchies.

#### Use Hierarchy States

The property *Use Hierarchy States* is used to switch between the usage of *Hierarchy States* and *Expand To* (see Behavior Tab), as otherwise, the latter properties may contradict each other. This effectively means that Hierarchy States are only applied if *Use Hierarchy States* is actively set to `true` (it is deactivated by default). Otherwise, *Expand To* is used.

### Deviations

#### Apply Pin Color, Pin Color

By default the deviation pins are colored with the negative or positive deviation color (depending on the sign). If the pins should have a different color, the property *Apply Pin Color* must be set to *true* and a *Pin Color* has to be defined.

### Apply Pin Head Color, Pin Head Color

Same as *Apply Pin Color* and *Pin Color*, but for pin heads.

### Deviaton Label Size Factor

According to IBCS the text size of the labels in the deviation columns corresponds to the text size of the value columns of the table. The default factor for this is 0.7, that means the tables' text size times 0.7 results in the text size for the deviations. This factor can be changed with this property. Please be aware that a factor lower than 0.3 or higher than 1.2 may result in wrong representations.

### Deviatons From Null

This property affects the way NULL values are interpreted when calculating the deviations. If the property is set to *false* a NULL value is considered to be *not available*, so nothing is drawn. Otherwise the value is used as a *Zero*, so the deviation is the difference between zero and the value.

### Textual Deviaton

This property controls whether the deviations should be displayed as plain values or as deviation charts.

### No Clip

This option disables *clipping* for SVGs, a feature used to prevent pin heads to show up behind both sides of a deviation axis. This behaviour is not recommended, but necessary if you plan to deploy your table on iOS: Newer Safari versions contain a bug and therefore do not support clipping, which leads to rendering errors.

### Ignore Multiple Currency Values

If this property is activated, no deviations are displayed for rows in which a value included in the calculation of a deviation would have '\*' as formatted data, regardless of whether '\*' is displayed in the table or not. In addition, switching on this property also causes '\*' values to be displayed as such, even if the *Use Formatted Data* property is off.

### Labels

#### Data Cell Alignment Right

By default the cell contents are aligned to the right (the property value is *true*). To get the data aligned to the left, this option has to be set to *false*.

#### Number Format Per Column

This property allows to define (different) *Extended Number Formats* per column. Therefore an array of string values (with the corresponding *Extended Number Format*) is used. An empty string value sets the *Number Format* to this one, which is used for the table. This property can also be controlled via scripting language.

Example for two columns: ["-|.|,|10^6|1|", "-|.|,|1|1|%25"]

### No Wrap

If the row headers are too long to fit the available space, they are wrapped by default. If this options is activated, row headers are not wrapped. To furthermore abbreviate them, use the following CSS snippet on the table:

```
.rowHeader {
  max-width: <target width>px;
}
```

### Suppress Repeating Units

If formatted data is used for the table and a currency is displayed for each cell of a column, this property can be turned on to hide the currencies in each cell. The currency can then be displayed once in the column header using the property *Show Units*.

### Remove Repeated Labels

If several dimension members are repeated in the column headers, this property can be used to suppress the repetitions.

	Net Sales		Feb 09	
	Jan 09		South East	West
	South East	West		
Internet	3.6m	1.6m	4.6m	1.2m
EDI	4.5m	1.2m	6.1m	1.2m
Phone	6.4m	687.9k	6.5m	1.5m

fig. 1: *Remove Repeated Labels* is active

	Net Sales	Net Sales	Net Sales	Net Sales
	Jan 09	Jan 09	Feb 09	Feb 09
	South East	West	South East	West
Internet	3.6m	1.6m	4.6m	1.2m
EDI	4.5m	1.2m	6.1m	1.2m
Phone	6.4m	687.9k	6.5m	1.5m

fig. 2: *Remove Repeated Labels* is not active

### Zero, Empty und Infinity Value Replacement

Use this property to replace certain values ('0', 'null' oder '') with other values.

If you e. g. would like to replace all '' in deviations with '~', set the *Infinity Value Replacement* to contain '~'. If you would like to show '—' for all empty cells (that contain 'null' values), set the *Empty Value Replacement* correspondingly.

### Show Row Titels

If the row titles should not be displayed, they can be hidden using this property.

### Show Units

This property allows to display the units (coming from the query) below the column titles.

### Show Scaling Factor

This property allows to display the scaling factor (coming from the Designer settings) below the column titles.

### Show Header Dimensions

If several dimensions are present within the columns, individual dimension members are separated with a line break in the column headers. This property holds an array of integers, which correspond to the indices of the dimensions to be shown.

Example: the columns comprise three dimensions. Setting this property to [1,3] results in column headers which show only the members of the first and the third column dimensions.

### Show Column Titels

If the column titles should not be displayed, they can be hidden using this property.

### Use Formatted Data

This property allows using the preformatted data of the Data Source for cells of the table. This is possible only for those values that are passed from the Data Source. Values, which are calculated in the table itself, e.g. deviations, are formatted still either in basic or extended number format.

### Exceptions

#### Exception Fill None In Text Color

The *Exception* filltype *None* causes that no Exception symbol is displayed but the corresponding cell background is filled. By activating the property *Exception Fill None In Text Color* the text color is changed instead of the background.

### Highlight

#### Highlight Thickness, Highlight Color, User Highlight List

With these properties the design of the *User Highlights* (highlighted values at runtime) can be changed and the list of *User Highlights* can be customized and readout. The user highlights can be set at runtime by pressing the *Alt-Key* and clicking with the mouse. That way the table cell is highlighted with a colored border. Change the thickness of the Highlight with the property *Highlight Thickness* and chose a color (*Highlight Color*). The *User Highlight List* contains the list of set highlights. Details about the format of the *User Highlight List* can be found in the script documentation.

### Outliers

#### Outlier Threshold, Outlier Threshold Percent, Use Outlier Threshold

The use of Outlier Thresholds is activated with *Use Outlier Threshold*. The usage is only intended for a scenario with *Comparison Group* switched off, otherwise Thresholds and Comparison Group would both affect the scaling, which can lead to an undefined behavior.

If usage is activated, all values that are greater than the threshold or smaller than the negative threshold are displayed as outliers. The thresholds therefore indicate the absolute value of the positive and negative limits. Values for the property *Outlier Threshold Percent* must be entered as decimal percentage representation, e.g. 0.6 for "60%".

## Display

### Table Header Position Fixed

If the table header (column titles) should be displayed permanently (even when scrolling), this can be done using this property.

If this property is activated, horizontal scrolling is suppressed because the width of the table is required to calculate the column header widths and the column header widths are required to determine the table width. Otherwise, there would be a 'the chicken or the egg' causality dilemma.

### Width Per Column

Via this property a custom width for each column of the table can be set. The value must be given as an array. The indices of the array correspond to the column indexes in the table. The column at position 0 are the row titles. The width is specified in pixels. For a table with three data series and one deviation series a possible array could look like this: [200,100,100,0,300]

You do not have to set fixed values for all columns: If for one or more columns the value 0 is set, then the width of the corresponding column/s is determined automatically, i.e. the remaining space is provided for the columns and divided evenly.

It is also possible to specify a number with a 'max' instead of a number or '0', for example: [200,"max200",100,0,300]. This will allow the column as much space as possible (similar to the setting with '0'), but maximally as much as specified.

The *Width Per Column* setting overrides the *Fixed Column Width* setting.

HTML tables are meant to distribute the total width cleverly between the columns, so the browser does not always consider the manually defined width of the individual columns. This can be the case, for example, with long words in RowHeader cells that should not be wrapped in the middle of the word. The positive difference between the manually specified width and the actual width of a column gets subtracted from the remaining columns by the browser (unless they are In Cell or Deviation columns). As a result, two columns to which you manually assign the same width may be displayed with different widths. This property should therefore only be considered as a rough approximation and should only be used if modern browsers do not provide a satisfactory result in the distribution of column widths.

## Interactivity at Runtime

The *graphomate tables* extension supports some features that can be used interactively at runtime. Many (but not all) of these functions can be configured on the *Additional Properties Sheet*.

### Collapsible Hierarchies

If the option *Collapsible* in *Hierarchy* in the Behavior tab is activated, hierarchical orders of dimensions can be used interactively with drilldown and roll-up during runtime.

### Context Menu

If the option *Context Menu* in the corresponding tab is activated, the Context Menu will be accessible during runtime by right clicking on the component. It allows special filtering (according to the settings made). A reload resets all changes made with the context menu.

### Filter Box

In the top left cell of the first row of the *graphomate tables* a filter function is implemented. Just enter some letters or regular expressions in the Filter Box and only those table rows are shown that contain these letters. Upper and lower case is not considered.

If the option *Show Filter Box* in the Helper tab is activated, the Filter Box is visible for the users.

### User Highlights

Furthermore there is the possibility of creating so-called *User Highlights* at runtime and to store those with bookmarks. Simply click on the desired value in the dashboard while pressing the Alt-key to set the *User Highlights*. The *User Highlights* can be configured via the properties *Highlight Thickness* and *Highlight Color* on the Property Sheet.

### Highlight Group

If the Highlight Group property is activated in the Behavior tab, all cells within the table that have the same member combination (exclusive Keyfigure Dimension) as a hovered element are highlighted by a colored background. If there are further charts or tables with an activated Highlight Group property in the application, all elements (bars of a chart or cells of a table) whose member combinations are a superset of the member combination of the hovered cell (exclusive Keyfigure Dimension) will be highlighted in these. The Highlight Group Color option can be used to customize the highlighting color for each component.

Example: In Table 1 a cell with the member combination  $A := \{\text{"Atlantic", "New Brunswick"}\}$  hovered with the cursor. Table 2 has cells which contain the member combinations  $B := \{\text{"Atlantic", "New Brunswick", "Home Office"}\}$ . Since B represents a superset of A (all elements of A are contained in B: this includes the equivalence of A and B), these cells are highlighted accordingly. Conversely, if an element from table 2 with member combination B is hovered, the elements from table 1 with member combination A are no longer highlighted, since A is not a superset of B.

If you also want to consider the Keyfigure Dimension (Measure) in the member combinations, you can activate the Match Measures options for each table. As an example we add a keyfigure dimension to the member combinations of the first example:  $A := \{\text{"Atlantic", "New Brunswick", "Order Quantity"}\}$  and  $B := \{\text{"Atlantic", "New Brunswick", "Home Office", "Profit"}\}$ . Here the behavior described above applies: when hovering a cell with the member combination A all cells are highlighted whose member combination is a superset of A. In this case B is no longer a superset of A, so the elements with member combination B are no longer highlighted.

Since the Match Measures option can be activated individually for each table, a final example of the different behavior between the above examples should be given. In this case, the Match Measures option is disabled in

Table 1 and enabled in Table 2. If you now hover in table 1 over a cell with the member combination  $A := \{\text{"Atlantic"}, \text{"New Brunswick"}\}$  (with the keyfigure "**Order Quantity**"), all cells whose member combination is a superset of  $A$  are highlighted within Table 1. In Table 2, however, the situation is different, since the Match Measures option is activated there. Here only those cells are highlighted whose member combinations are a superset of  $A_{\text{Measures}} := \{\text{"Atlantic"}, \text{"New Brunswick"}, \text{"Order Quantity"}\}$ .

### Accessibility

The tables are compatible with screen readers. This means that table cells, rows, columns and the table itself as well as the title are annotated with metadata that can be used by screen readers to meaningfully summarize the content. Therefore use the standardized WAI-ARIA annotations. Screenreaders differ in the way they use this information, on which the table has no influence, so different screenreaders can read different information.



## Known Issues

- If hatched or dotted patterns are used, the **Lumira Designer PDF export** may have problems to correctly generate a PDF.
- **Editable content in the table with Internet Explorer:** When using the Row Format Configuration to make cells content editable the background color will be set correctly in the Internet Explorer. But due to a problem in IE versions 9 through 11 it is not possible to edit the content in the cells. If you need this feature and are using the Internet Explorer please contact us.
- **Offset in the table when using Safari:** Due to a rendering error in the Safari version 9 and higher the tables will be drawn with a slight offset, about 10px at the top.
- In Design Mode, the first time the Additional Properties Sheet is opened after a restart of Designer, the server URL may not be displayed correctly, although it is displayed correctly in the standard properties sheet and the server is connected.
- If column headers have been renamed, the original name of the measure from the data source is still displayed in the context menu instead of the newly assigned name, because the context menu is implemented purely on the basis of data sources.
- Rows without a value in any cell are not displayed. This can be prevented by assigning an explicit height using CSS or by using the property Null Replacement with a white space character such as " ".
- Data selections may only cover entire columns in order to be processed correctly by the component. Restricting the selection to certain row members is not supported.
- Structures or explicit hierarchies are supported, but only if they are the only dimensions of the Rows axis. The combination of structures or hierarchies with other dimensions is not supported.

## Number Formatting

### Possible Inputs for the numeral.js Format String

Floating point		
Number	Format String	Output
10000	'0,0.0000'	10.000,0000
10000.23	'0,0'	10
-10000	'0,0.0'	-10.000,0
-0.23	'00'	-,23
-0.23	'(00)'	(,23)
0.23	'0.00000'	0,23000
0.23	'0.0[0000]'	0,23
1230974	'0.0a'	1,2m
1460	'0 a'	1 k
1	'0o'	1 <sup>st</sup>
Currency		
Number	Format String	Output
1.000.234	'\$0,0.00'	\$1.000,23
1000.2	'0,0[.]00 \$'	1.000,20 \$
1001	'\$ 0,0[.]00'	\$ 1.001
Percentage		
Number	Format String	Output
1	'0%'	100%
-0.43	'0 %'	-43%

Source

### Possible input for the Extended Number format string

Percentages and numbers are formatted separately. For this two format string are used, which follow the same rules and structure. In general the strings of characters in the format string are simply replaced in the input number. Exceptions to this rule would be scaling factors and the number of decimal places.

The separate elements of the format string are separated by the "|" character. Format strings with less than 7 elements (i.e. 6 "|" characters) are considered invalid and therefore ignored. In this case the standard format without scaling and with one position after the decimal point is used.

The format string consists of the following elements, the ordering is obligatory and cannot be changed:  
*Negative|Prefix|Thousands|Decimal|Scaling|PointsAfterDecimal|Suffix*

Negative	The prefix for negative numbers. The string "(" will result in the number will be placed inside parentheses, including eventual prefixes and suffixes. If <i>None</i> is used the absolute number will be shown.
Prefix	A prefixed string of characters, for example a currency symbol or an identifier for the scaling factor.
Thousands	A separator for thousands.
Decimal	A separator for the decimal places.
Scaling	The scaling factor for the value. This can be given as a whole number or a power, formatted as $n^m$ . A whole number will be calculated as a power to the basis 10 internally. In case of a given power the value will be divided by it directly. Even negative numbers are possible, for example "-3" or "10^-3"; in this case the value will get adequately higher. Non numeric inputs are ignored. <i>Please note:</i> given a "0" or a "1" will result in a scaling factor of 1. The only way to achieve a scaling by 10 is to input "10^1".
PointsAfter Decimal	The number of places after the decimal.
Suffix	A suffix for the value, analogous to the prefix.

# CSS Selectors

## Overview of css classes

The following illustrations provides some of the css selectors, which can be addressed by using the *Custom CSS* in SAP Lumira Designer.

