



Scripting Documentation of the graphomate treemaps 2.2 for SAP Lumira Designer 2.x

Version 2.2 - as of June 2019

<https://www.graphomate.com>

1 Scripting Documentation

Scriptin Documentation of graphomate treemaps for SAP Lumira Designer

Rev 2.2 – as of June 2019

1.1 Introduction

The graphomate treemaps extension supports the scripting language of Designer. Therefore, it is possible to determine properties of the extension at run-time. New data can be set or titles can be adapted. Furthermore, it is possible to use drilldown functionalities of the extension and thus create interactive dashboards.

This document provides a list of available functions, events and examples. We are steadily working on expanding the range of functions. If you miss certain functions, please give us a note. Therefore you can use our online form under <http://www.graphomate.com/en/feedback>.

1.2 Events

The following events are available. User-made scripts can be deposited on the Standard Properties Sheet under the point Events.

Ereignis Name	Beschreibung
On Element Clicked	Is triggered if the user clicks on an element of the treemap at run-time. In order to allow a drilldown, it is necessary to react on the user input at this point. The return values of the following functions are updated at the moment of the event: getSelectedMemberKey(), getSelectedMemberText(), getClickedElementIndex(), getClickedSeriesIndex()
On New Data Arrival	Is triggered when the treemap receives updated data. This is happening, for example, when the dimension of a Data Source or if a filter has been changed.

1.3 Functions

The following functions are available in the graphomate treemaps extension. They can be addressed in the scripting editor of Designer.

Getter Name	Example	Returntype	Example Returntype	Description
getColorScheme()	GRAPHOMATETREEMAP_1.getColorScheme();	String	Spectral,9	Returns the current ColorBrewer scheme for the treemap. The return value contains the scheme name – an seperated with a comma – the number of classes.
getColorCalculationMethod()	GRAPHOMATETREEMAP_1.getColorCalculationMethod();	String	hierarchy	Returns the method used for coloring the rectangles of the treemap.
getDepth()	GRAPHOMATETREEMAP_1.getDepth();	int	3	Returns on how many levels the restriction of the hierarchy depth is fixed.
getDescriptionArea()	GRAPHOMATETREEMAP_1.getDescriptionArea();	boolean	false	Returns whether the rectangle labels contain the area dimension and the value of the members.
getDescriptionDeviation()	GRAPHOMATETREEMAP_1.getDescriptionDeviation();	boolean	false	Returns whether the rectangle labels contain the area dimension and the value of the members.
getDescriptionName()	GRAPHOMATETREEMAP_1.getDescriptionName();	boolean	false	Returns whether the rectangle labels contain the name of the memeber.
getDescriptionPath()	GRAPHOMATETREEMAP_1.getDescriptionPath();	boolean	true	Returns whether the rectangle labels contain the hierarchy levels.
getDescriptionVisible()	GRAPHOMATETREEMAP_1.getDescriptionVisible();	boolean	true	Returns whether the rectangle labels are displayed.
getEmbedded()	GRAPHOMATETREEMAP_1.getEmbedded();	boolean	false	Return whether the embedded mode (frames to display the hierarchy levels) of the treemaps is active.
getExtendedNumberFormat()	GRAPHOMATETREEMAP_1.getExtendedNumberFormat();	String	- . , 1 1	Returns the number format for the extended mode. The format string can be found in the manual.
getExtendedNumberFormatPercentage()	GRAPHOMATETREEMAP_1.getExtendedNumberFormatPercentage();	String	- . , 1 1 %25	Returns the number format for the percentage values in the extended mode. The format string can be found in the manual.
getHideOverflowingDescriptions()	GRAPHOMATETREEMAP_1.getHideOverflowingDescription();	boolean	true	Returns whether the option to remove overflowing labels has been selected.
getLabelFormatMode()	GRAPHOMATETREEMAP_1.getLabelFormatMode();	String	extended	Returns which mode is used for the label format.
getLimitDepth()	GRAPHOMATETREEMAP_1.getLimitDepth();	boolean	true	Returns whether it is possible to use the limitation of the depth of the hierarchy, which is represented in the treemap.
getLocale()	GRAPHOMATETREEMAP_1.getLocale();	String	fr	Returns the setting that is currently selected for locale.
getNegativeDeviationIsGood()	GRAPHOMATETREEMAP_1.getNegativeDeviationIsGood();	boolean	true	Returns whether the negative deviations are interpreted as positive or negative.
getNumberFormat()	GRAPHOMATETREEMAP_1.getNumberFormat();	String	0,0.0	Returns the number format for the basic mode. The format string corresponds to the format numeral.js (see http://www.numeraljs.com).

getNumberFormatPercentage()	GRAPHOMATETREEMAP_1.getNumberFormatPercentage();	String	0 %	Returns the number format for the percentage values in the basic mode. The format string corresponds to the format numeral.js (see http://www.numeraljs.com).
getRestPadding()	GRAPHOMATETREEMAP_1.getRestPadding();	int	3	Returns the width of the spacing between the frames and the treemap (for the left, right and bottom side), which is used in the embedded mode.
getSelectedMember(Dimension dimensionKey)	GRAPHOMATETREEMAP_1.getSelectedMember("0D_NWI_RCOD");	Member		After the user clicked on an element/a category in the treemap, this function returns the member of the specified dimension dimensionKey. The member provides different properties such as text, internalKey and externalKey.
getSelectedMemberKey(String dimensionKey)	GRAPHOMATETREEMAP_1.getSelectedMemberKey("0D_NWI_RCOD");	String	10274	After the user clicked on an element/a category in the treemap, this function returns the Key of the member of the selected dimension dimensionKey. The Keys of the dimensions are shown in the Initial View.
getSelectedMemberText(String dimensionKey)	GRAPHOMATETREEMAP_1.getSelectedMemberText("0D_NWI_RCOD");	String	North West	After the user clicked on an element/a category in the treemap, this function returns the Text of the member of the selected dimension dimensionKey. The return value is similar to the function getSelectedMemberKey().
getShowLegend()	GRAPHOMATETREEMAP_1.getShowLegend();	boolean	true	Returns whether the legend (on the upper left above the treemap) is displayed.
getShowTitle()	GRAPHOMATETREEMAP_1.getShowTitle();	boolean	true	Returns whether the title is displayed.
getTopPadding()	GRAPHOMATETREEMAP_1.getTopPadding();	int	15	Returns the width of the spacing between the top side of the frames and the treemap, which is used in the embedded mode.
getTitle()	GRAPHOMATETREEMAP_1.getTitle();	String	Sales in Mio. EUR 2013 ACT and BUD North West	Returns the title of the treemap as a string value. A line break is represented with a pipe ().
getTooltipArea()	GRAPHOMATETREEMAP_1.getTooltipArea();	boolean	false	Returns whether the tooltips contain the area dimension and the value of the members.
getTooltipDeviation()	GRAPHOMATETREEMAP_1.getTooltipDeviation();	boolean	false	Returns whether the tooltips contain the color dimension and the value of the members.
getTooltipName()	GRAPHOMATETREEMAP_1.getTooltipName();	boolean	false	Returns whether the tooltips contain the name of the member.
getTooltipPath()	GRAPHOMATETREEMAP_1.getTooltipPath();	boolean	true	Returns whether the tooltips contain the hierarchy levels.
getTooltipVisible()	GRAPHOMATETREEMAP_1.getTooltipVisible();	boolean	true	Returns whether the tooltips are displayed.
getUseFormattedData()	GRAPHOMATETREEMAP_1.getUsedFormattedData();	boolean	true	Returns whether the preformatted data from Designer is used for the entries of the table.

Setter Name	Example	Description
setColorCalculationMethod(String val)	GRAPHOMATETREEMAP_1.setColorCalculationMethod("hierarchy");	Sets the method by which the rectangles of the treemap are colored. Valid values for method are: hierarchy, arealdentity, colorIdentity, absoluteDeviation, percentDeviation.
setColorScheme(String val)	GRAPHOMATETREEMAP_1.setColorScheme("Set1, 4");	Sets the new ColorBrewer scheme for the treemap and the number of classes. Both values must be separated by a comma and passed in a string.
setDepth(int level)	GRAPHOMATETREEMAP_1.setDepth(2);	Limits the number of hierarchy levels displayed in the treemap to the value level.
setDescriptionpArea(boolean val)	GRAPHOMATETREEMAP_1.setDescriptionArea(true);	Sets whether the rectangle labels contain the area dimension and the value of the member.
setDescriptionDeviation(boolean val)	GRAPHOMATETREEMAP_1.setDescriptionDeviation(true);	Sets whether the rectangle labels contain the color dimension and the value of the member.
setDescriptionName(boolean val)	GRAPHOMATETREEMAP_1.setDescriptionName(true);	Sets whether the rectangle labels contain the name of the member.
setDescriptionPath(boolean val)	GRAPHOMATETREEMAP_1.setDescriptionPath(false);	Sets whether the rectangle labels contain the area hierarchy levels.
setDescriptionVisible(boolean val)	GRAPHOMATETREEMAP_1.setDescriptionVisible(true);	Sets whether the rectangle lables are displayed.
setEmbedded(boolean val)	GRAPHOMATETREEMAP_1.setEmbedded(true);	Allows to turn on and off the embedded mode (frame to display the hierarchy levels).
setExtendedNumberFormat(String val)	GRAPHOMATETREEMAP_1.setExtendedNumberFormat("- . , 1 1 ");	Sets the number format for the extended mode. The format string can be found in the manual.
setExtendedNumberFormatPercentage(String val)	GRAPHOMATETREEMAP_1.setExtendedNumberFormatPercentage("- . , 1 1 %25");	Sets the number format for the percentages in the extended mode. The format string can be found in the manual.
setHideOverflowingDescriptions(boolean val)	GRAPHOMATETREEMAP_1.setHideOverflowingDesicription(false);	Sets whether overflowing labels are to be removed.
setLabelFormatMode(String val)	GRAPHOMATETREEMAP_1.setLabelFormatMode("extended");	Sets a string value for the used label format mode. Available values are basic and extended.
setLimitDepth(boolean val)	GRAPHOMATETREEMAP_1.setLimitDepth(false);	Sets whether the limitation of the depth of the hierarchy is activated.
setLocale(String val)	GRAPHOMATETREEMAP_1.setLocale("de");	Sets the value of the locale. Allowed values are: de, en, fr and auto. If auto is used, the locale from the query is used for formatting the data.
setNegativeDeviationIsGood(boolean val)	GRAPHOMATETREEMAP_1.setNegativeDeviationIsGood(true);	Sets whether the negative deviations are interpreted as positive or negative.
setNumberFormat(String val)	GRAPHOMATETREEMAP_1.setNumberFormat("0.0.0");	Sets the number format for the basic mode. The format string corresponds to the format numeral.js (see http://www.numeraljs.com).
setNumberFormatPercentage(String val)	GRAPHOMATETREEMAP_1.setNumberFormatPercentage("0 %");	Sets the number format for the percentages in the basic mode. The format string corresponds to the format numeral.js (see http://www.numeraljs.com).
setRestPadding(int val)	GRAPHOMATETREEMAP_1.setRestPadding(1);	Sets the width of the spacing between the frames and the treemap (for the left, right and bottom side) in the embedded mode.
setShowLegend(boolean val)	GRAPHOMATETREEMAP_1.setShowLegend(true);	Sets whether the legend is displayed.
setShowTitle(boolean val)	GRAPHOMATETREEMAP_1.setShowTitle(true);	Sets whether the title is displayed.
setTopPadding(int val)	GRAPHOMATETREEMAP_1.setTopPadding(1);	Sets the width of the spacing between the top side of the frames and the treemap in the embedded mode.
setTitle(String val)	GRAPHOMATETREEMAP_1.setTitle("Sales in Mio. EUR 2013 ACT and BUD North West");	Sets the title of the treemap. Expects value as a string. Line breaks (\n) are interpreted as such.

setTooltipArea(boolean val)	GRAPHOMATETREEMAP_1.setTooltipArea(true);	Sets whether the tooltips contain the area dimension and the value of the member.
setTooltipDeviation(boolean val)	GRAPHOMATETREEMAP_1.setTooltipDeviation(true);	Sets whether the tooltips contain the color dimension and the value of the member.
setTooltipName(boolean val)	GRAPHOMATETREEMAP_1.setTooltipName(true);	Sets whether the tooltips contain the name of the member.
setTooltipPath(boolean val)	GRAPHOMATETREEMAP_1.setTooltipPath(false);	Sets whether the tooltips contain the hierarchy levels.
setTooltipVisible(boolean val)	GRAPHOMATETREEMAP_1.setTooltipVisible(true);	Sets whether the tooltips are displayed.
setUseFormattedData(boolean val)	GRAPHOMATETREEMAP_1.setUseFormattedData(true);	Sets whether the preformatted data from Designer is used for the entries of the table. This is possible only for the values that are passed from the query. Values, which are calculated in the table itself, eg for the deviations, are formatted still either in the basic or extended number format.

1.4 Overview: Scripting language

The graphomate treemaps extension fully supports the scripting language of Designer. Therefore it is possible to control graphomate treemaps interactively at runtime.

For example, it is possible to set new data or change the title. Furthermore, it is possible to use drilldown functions of the extension and create interactive dashboards.

1.4.1 Example: Dynamic filtering of a Data Source with a dropdown box

In order to use a dynamic filter on a dimension of a Data Source, we first need to fill a dropdown box with all valid values to enable the user to choose from the data later on. We use `getMemberList` on the canvas-event On Startup to retrieve a list of all members of the dimension `0D_NWI_ACOD` of the selected Data Source. Autocomplete shows all parameters of `getMemberList`.

We can then fill the dropdown box with the retrieved list using `setItem`.

```
var memberList = DS_1.getMemberList("0D_NWI_ACOD",
MemberPresentation.INTERNAL_KEY, MemberDisplay.TEXT, 10);
DROPDOWN_1.setItem(memberList);
```

The dropdown box will be filled with the members of Dimension `0D_NWI_ACOD` on application start.

If a user selects an element from the list, we can adapt the filter of the Data Source by using the following script that is put on the events On Element Clicked:

```
var selectedMemberKey = DROPDOWN_1.getSelectedValue();
DS_1.setFilter("0D_NWI_ACOD", selectedMemberKey);
```

You may find more explanations and examples for the scripting language in our [Script Documentation](#).