graphomate

Scripting Documentation of the graphomate tables 2.2

for SAP Lumira Designer 2.x

Version 2.2 - as of June 2019

https://www.graphomate.com

1 Scripting Documentation

Scripting Documentation of the graphomate tables for SAP Lumira Designer

Rev 2.2 as of June 2019

1.1 Introduction

The graphomate tables supports the SAP Lumira Designer scripting language. This enables the user to dynamically interact with the add-on during runtime. Script commands include manipulation of the graphomate tables layout or setting of data. This document features a list of all currently available script commands. Please do not hesitate to contact us if any feature is missing. For this, you can use our online form.

1.2 Events

The following events are available. User-made scripts can be deposited on the Property Sheet under the group Events.

Event Name	Description
On Element Clicked	Is triggered when the user clicks on an element of the table at runtime. In order to allow a drill-down, 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: getSelectedMember() getSelectedMemberKey() getSelectedMemberText() getClickedElementIndex() getClickedSeriesIndex()
On Input Changed	This event is triggered after some input was performed on the table. To enable manual input the Row Format Configuration can be used by enabling the style Content Editable. To retrieve the changed values from the table the method getChangedInput can be used. Using getSelectedMember matching this to the members from the Datasource is possible.
On Hierarchy State Changed	As soon as some hierarchy node is expanded or minimized this event is triggered. The manipulated node can be retrieved by using getSelectedMember. The method getHierarchyStateChange enables you to check if a certain node is expanded or minimized.

1.3 Functions

The following functions are available. They can be addressed in the scripting editor of Designer. We differentiate between getter and setter functions as they get information from the extension or set its properties.

Getter Name	Example	Return type	Example Return Value	Description
getAxisWidth()	GRAPHOMATETABLE_1.getAxisWidth();	int	1	Returns the current axis width.
getChangedInput()	GRAPHOMATETABLE_1.getChangedInput();	String	Hello World!	Returns the last changed content of a table cell. For further information take a look at the On Input Changed event.
getClickedElementIndex()	GRAPHOMATETABLE_1.getClickedElementIndex();	int	3	Returns the index of the last clicked element. Returns -1 if a cell was clicked that is not part of a data series.
getClickedSeriesIndex()	GRAPHOMATETABLE_1.getClickedSeriesIndex();	int	1	Returns the index of the last clicked Data Series. Returns -1 if a cell was clicked that is not part of a data series.
getCollapsibleHierarchy()	GRAPHOMATETABLE_1.getCollapsibleHierarchy();	boolean	true	Returns whether the hierarchy control elements (+ and - before the line items) are displayed.
getColumnOrder()	GRAPHOMATETABLE_1.getColumnOrder();	String	[{"identifier":"series1","position":0},{"identifier":"deviation1", "position":1},{"identifier":"series2","position":2}, {"identifier":"deviation2","position":3}]	Returns the current order of the columns (data series and deviations).
getColumnTitleVisible()	GRAPHOMATETABLE_1.getColumnTitleVisible();	boolean	false	Returns whether the Column Title is visible.
getColumnWidth()	GRAPHOMATETABLE_1.getColumnWidth();	int	20	Returns the Column Width as an integer value if the Column Width is fixed.
getComparisonGroup()	GRAPHOMATETABLE_1.getComparisionGroup();	String	A	Returns the Comparison Group as a string value.
getDatatypes()	GRAPHOMATETABLE_1.getDatatypes();	String	AC,PP,BU	This function returns the Data Types of the selected Data Series seriesIndex as a comma-separated string.
getDeviationBadColor()	GRAPHOMATETABLE_1.getDeviationBaddColor();	String	#FF0000	Returns the color for negative deviations.
getDeviationGoodColor()	GRAPHOMATETABLE_1.getDeviationGoodColor();	String	#00FF00	Returns the color for positive deviations.
getDeviationIds(String id)	GRAPHOMATETABLE_1.getDevationIds();	String	deviation1,deviation2	Returns a list of IDs of all deviation series. The elements of the list are separated by a comma.
getDeviationLabels()	GRAPHOMATETABLE_1.getDeviationLabels();	String	Percent,Absolute	Returns the currently assigned column headers of the deviation series, if these were set via the standard property Deviation Labels, which overwrites the column headers for deviation columns regardless of the setting of Deviation Labels in the Additional Properties Sheet or Standard Property Deviations Config.

getDeviationLabelsandIds()	GRAPHOMATETABLE_1.getDeviationLabelsAndIds();	String	deviation1:abs,deviation2:perc	Returns the list of pairs of IDs and labels for all deviation columns. These are the labels assigned by using the Additional Properties Sheet or the Standard Property Deviations Config. The ID of the Deviation column is separated from the label by a colon, the individual pairs by a comma.
getDeviationLabelById(String id)	GRAPHOMATETABLE_1.getDeviationLabelById("deviation1");	String	perc	Returns the label of the deviation column whose ID was passed in the method call. This is the label assigned using the Additional Properties Sheet or the Standard Property Deviations Config.
getDeviationVisibility(String name)	GRAPHOMATETABLE_1.getDeviationVisibility("perc");	boolean	true	Returns the visibility of the deviation definition with the label name as a boolean value.
getExceptionPosition()	GRAPHOMATETABLE_1.getExceptionPosition();	String	leading	Return the manner in which the exceptions are positioned (at the beginning of the cell (leading), before or after the value).
getExceptionVisibility(string value name)	GRAPHOMATETABLE_1.getExceptionVisibility("good");	boolean	true	Returns the visibility of the Exception definition with the label name as a boolean value.
getExpandTo()	GRAPHOMATETABLE_1.getExpandTo();	int	3	ExpandTo determines up to which hierarchy level the hierarchy is to be expanded in the start drilldown (starting with level 0). This get-method returns this level.
getExtendedNumberFormat()	GRAPHOMATETABLE_1.getExtendedNumberFormat();	String	-||.|,|1|1|	Returns the number format for the Extended Mode. The format string can be found in the manual.
getExtendedNumberFormatPercentage()	GRAPHOMATETABLE_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.
getFilterString()	GRAPHOMATETABLE_1.getFilterString();	String	revenue$	Returns the Filter String.
getFixedColumnWidth()	GRAPHOMATETABLE_1.getFixedColumnWidth();	boolean	true	This function returns a boolean value whether the Column Width is fixed (otherwise the width is set automatically).
getFixedHeader()	GRAPHOMATETABLE_1.getFixedHeader();	boolean	true	Returns whether the table header should be fixed and therefore also visible while scrolling.
getFontFamily()	GRAPHOMATETABLE_1.getFontFamily();	String	Arial	Returns the current font family.
getHierarchyStateChange()	GRAPHOMATETABLE_1.getHierarchyStateChange();	String	EXPANDED	At the moment a user clicks a hierarchy element (+/-) it is saved, if the node was opened (EXPANDED) or closed (CLOSED). This information is available for the last click by calling this script method. Use this method in combination with the On Hierarchy State Change event.
getHierarchyStates()	GRAPHOMATETABLE_1.getHierarchyStates();	String	"EXPANDED", "COLLAPSED", "","","","", "COLLAPSED", "","","", ""	Returns the current hierarchy state, i. e. which nodes are expanded and which are collapsed. The return value is an array whose length corresponds to the number of rows. For each row, it is stored whether its "children" are displayed or not. Note: The property Use Hierarchy Marker needs to be deactivated for this script command to work properly.
getLabelFormatMode()	GRAPHOMATETABLE_1.getLabelFormatMode();	String	extended	Returns which mode is used for the label format.
getLocale()	GRAPHOMATETABLE_1.getLocale();	String	en	Returns the setting that is currently selected for locale.
getNegativeDeviationIsGood()	GRAPHOMATETABLE_1.getNegativeDeviationIsGood();	boolean	true	Returns whether negative deviations are visualized as positive.
getNoClip()	GRAPHOMATETABLE_1.getNoClip();	boolean	true	Returns the value for the noClip property.
getNumberFormat()	GRAPHOMATETABLE_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()	GRAPHOMATETABLE_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).
getNumberFormatPerColumn()	GRAPHOMATETABLE_1.getNumberFormatPerColumn();	String	["","-||.|,|1|1|%25",""]	Returns the array with the number format of the column.
getRowTitleVisible()	GRAPHOMATETABLE_1.getRowTitleVisible();	boolean	false	Returns whether the Row Title is visible.
getSelectedMember(Dimension dimensionKey)	GRAPHOMATETABLE_1.getSelectedMember("0D_NWI_RCOD");	Member	undefined	After the user clicked on an element/a category in the table, this function returns the member of the specified dimension dimensionKey. The member provides different properties such as text, internalKey and externalKey. If a non-valid dimension was specified or an element that does not represent a data value was clicked, undefined is returned.
getSelectedMemberKey(Dimension dimensionKey)	GRAPHOMATETABLE_1.getSelectedMemberKey("0D_NWI_RCOD");	String	10274	After the user clicked on an element/a category in the table, 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(Dimension dimensionKey)	GRAPHOMATETABLE_1.getSelectedMemberText("0D_NWI_RCOD");	String	North West	After the user clicked on an element/a category in the table, this function returns the Text of the member of the selected dimension dimensionKey. The return value is similar to the function getSelectedMemberKey().
getSemanticAxis()	GRAPHOMATETABLE_1.getSemanicAxis();	boolean	true	Returns whether an semantic axis is shown for deviations.
getSeries(int seriesIndex)	GRAPHOMATETABLE_1.getSeries(2);	String	{"Keyfigures":"0D_NWI_NSAL"}	Returns the data description of the selected Data Series seriesIndex as JSON string.
getSeriesLabels()	GRAPHOMATETABLE_1.getSeriesLabels();	String	Jan,Feb	Returns the current series labels (shown as column titles).
getShowDeviationLabels()	GRAPHOMATETABLE_1.getShowDeviationLabels();	boolean	false	Returns whether the values of the deviations are displayed (if false, only the bars are displayed).
getShowHierarchy()	GRAPHOMATETABLE_1.getShowHierarchy();	boolean	true	Returns whether the row headers are displayed with a hierarchy.

getShowScalingFactor()	GRAPHOMATETABLE_1.getShowScalingFactor();	boolean	true	Returns whether the scaling factor (determined in the Designer settings) is displayed below the column title.
getShowTitle()	GRAPHOMATETABLE_1.getShowTitle();	boolean	true	Returns whether the title is displayed.
getShowUnits()	GRAPHOMATETABLE_1.getShowUnits();	boolean	true	Returns whether the units (coming from the query) are displayed below the column title.
getSumBorders()	GRAPHOMATETABLE_1.getSumBorders();	boolean	true	Returns whether Sum Borders are displayed.
getSumBorderColor()	GRAPHOMATETABLE_1.getSumBorderColor();	String	#00FFAA	Returns which color is used for the Sum Border.
getSumBorderThickness()	GRAPHOMATETABLE_1.getSumBorderThickness();	float	0.1	Returns the thickness of the Sum Border.
getSumPadding()	GRAPHOMATETABLE_1.getSumPadding();	float	0.0	Returns the padding between the sum and the corresponding child elements.
getSuppressZeroLabels()	GRAPHOMATETABLE_1.getSuppressZeroLabels();	boolean	true	Returns whether zero values are displayed.
getTextualDeviation()	GRAPHOMATETABLE_1.getTextualDeviation();	boolean	true	Returns whether textualDeviation is active. If textualDeviation is true, no deviation charts are displayed, but the values are represented as numbers.
getTitle()	GRAPHOMATETABLE_1.getTitle();	String	Sales in Mio. EUR|2013 ACT and BUD|North West	Returns the title of the chart as a string value. A line break is represented with a pipe (|).
getUseFormattedData()	GRAPHOMATETABLE_1.getUsedFormattedData();	boolean	true	Returns whether the preformatted data from Designer is used for the entries of the table.
getUserHighlightColor()	GRAPHOMATETABLE_1.getUserHighlightColor();	String	red,green	Returns a list of deposited colors for the User Hightlights as comma separated string-value. The colors are switched at each click (during runtime) until the desired color is set.
getUserHighlightList()	GRAPHOMATETABLE_1.getUserHighlightList();	String	[[1,3],[2,3][1,5]]	Returns a list of the current user highlights (highlighted values at runtime). Returns a JSON-string, which passes the positions of each highlight with two numbers; the first number indicates the data series, while the second number represents the position of the value in the data series.
getWidthPerColumn()	GRAPHOMATETABLE_1.getWidthPerColumn();	String	[100,300,0,"max200"]	Returns the current setting for the Width Per Column-property.
getFontSize()	GRAPHOMATETABLE_1.getFontSize();	float	12	Returns the currently selected font size.
getNoClip()	GRAPHOMATETABLE_1.getNoClip();	boolean	false	Returns the current value of No Clip.
getZeroValueReplacement()	GRAPHOMATETABLE_1.getZeroValueReplacement();	String	-	Returns the Zero Value Replacement. Empty if there is no Zero Value Replacement.
getInfinityValueReplacement()	GRAPHOMATETABLE_1.getInfinityValueReplacement();	String	~	Returns the Infinity Value Replacement. Empty if there is no Infinity Value Replacement.
getEmptyValueReplacement()	GRAPHOMATETABLE_1.getEmptyValueReplacement();	String	¯_()_/¯	Returns the Empty Value Replacement. Empty if there is no Empty Value Replacement.
getShowHeatmap()	GRAPHOMATETABLE_1.getShowHeatmap();	boolean	true	Returns true if the table is displayed as a heat map, false otherwise.
getHeatmapColScale()	GRAPHOMATETABLE_1.getHeatmapColScale();	String	Reds,4	Returns the selected color scheme of the heatmap.
getIncellColumns()	GRAPHOMATETABLE_1.getIncellColumns();	String	["series1","series2"]	Returns the columns that have In Cell Charts.
getIncellBarHeightFactor()	GRAPHOMATETABLE_1.getIncellBarHeightFactor();	float	0.7	Returns the factor used to calculate the height of the bars in the In Cell Charts.
getIncellBarOpactiy()	GRAPHOMATETABLE_1.getIncellBarOpactiy();	float	0.2	Returns the opacity of the In Cell Charts.
getIncellPositiveColor()	GRAPHOMATETABLE_1.getIncellPositiveColor();	String	#ff0000	Returns the color of the positive bars of the In Cell Charts.
getIncellNegativeColor()	GRAPHOMATETABLE_1.getIncellNegativeColor();	String	#00ff00	Returns the color of the negative bars of the In Cell Charts.
getUseOutlierThreshold()	GRAPHOMATETABLE_1.getUseOutlierThreshold();	boolean	true	Returns whether the Outlier Threshold is used.
getOutlierThreshold()	GRAPHOMATETABLE_1.getOutlierThreshold();	float	1000000	Returns the Outlier Threshold for absolute values.
getOutlierThresholdPercent()	GRAPHOMATETABLE_1.getOutlierThresholdPercent();	float	0.4	Returns the Outlier Threshold for percentage values.
getIbcsPinHeads()	GRAPHOMATETABLE_1.getIbcsPinHeads();	boolean	false	Returns whether the deviations are drawn according to the IBCS.
getApplyPinColor()	GRAPHOMATETABLE_1.getApplyPinColor();	boolean	false	Returns whether the manual pin color should be applied. If false, the good or bad color will be applied.
getApplyPinHeadColor()	GRAPHOMATETABLE_1.getApplyPinHeadColor();	boolean	false	Returns whether the manual pin head color should be applied. If false, the good or bad color will be applied.
getApplyDataTypesToCells()	GRAPHOMATETABLE_1.getApplyDataTypesToCells();	boolean	false	Returns whether the data types should be applied to the columns' cells.
getContextMenuAvailable()	GRAPHOMATETABLE_1.getContextMenuAvailable();	boolean	false	Returns whether the context menu is available via right click.
getDeviationLabelSizeFactor()	GRAPHOMATETABLE_1.getDeviationLabelSizeFactor();	float	0.7	Returns the deviation label font size factor.
getDataCellAlignmentRight()	GRAPHOMATETABLE_1.getDataCellAlignmentRight();	boolean	false	Returns whether the text of data cells is aligned right.
getDeviationsFromNull()	GRAPHOMATETABLE_1.getDeviationsFromNull();	boolean	false	Returns whether NULL values are interpreted as zero values (0) for deviation calculations.
getExceptionsFromNull()	GRAPHOMATETABLE_1.getExceptionsFromNull();	boolean	false	Returns whether NULL values are interpreted as zero values (0) for exception calculations.
getExceptionFillNoneInTextColor()	GRAPHOMATETABLE_1.getExceptionFillNoneInTextColor();	boolean	false	Returns whether the color of exception symbols with a fill type of 'none' should be used for coloring the cells' text instead of its background.
getShowFilterHelper()	GRAPHOMATETABLE_1.getShowFilterHelper();	boolean	false	Returns the visibility of the filter helper input box.
getHorizontalGridLineThickness()	GRAPHOMATETABLE_1.getHorizontalGridLineThickness();	float	0.1	Returns the thickness of horizontal grid lines in em.
getHighlightThickness()	GRAPHOMATETABLE_1.getHighlightThickness();	float	3	Returns the thickness of user highlight borders in px.

getHierarchyIndent()	GRAPHOMATETABLE_1.getHierarchyIndent();	float	1.2	Returns the size of the hierarchy indent in em.
getManualScalePercentMax()	GRAPHOMATETABLE_1.getManualScalePercentMax();	float	50	Returns the manual maximal value of the percentage scale.
getManualScalePercentMin()	GRAPHOMATETABLE_1.getManualScalePercentMin();	float	-50	Returns the manual minimal value of the percentage scale.
getNodeStyleBold()	GRAPHOMATETABLE_1.getNodeStyleBold();	boolean	false	Returns whether the hierarchy nodes are styled using bold or normal font weight.
getNoWrap()	GRAPHOMATETABLE_1.getNoWrap();	boolean	false	Returns whether the row title texts should be prevented from wrapping when they are too long.
getNodeStyleItalic()	GRAPHOMATETABLE_1.getNodeStyleItalic();	boolean	false	Returns whether the hierarchy nodes are styled using italic or normal font style.
getRemoveRepeatedLabels()	GRAPHOMATETABLE_1.getRemoveRepeatedLabels();	boolean	false	Returns whether repeating dimension members across the columns should only show up at the column of their first occurrence.
getRowTitleAlignmentRight()	GRAPHOMATETABLE_1.getRowTitleAlignmentRight();	boolean	false	Returns whether the row title texts are aligned right.
getShowHorizontalGridLines()	GRAPHOMATETABLE_1.getShowHorizontalGridLines();	boolean	false	Returns the visibility of the horizontal grid lines.
getShowDataTypesInHeader()	GRAPHOMATETABLE_1.getShowDataTypesInHeader();	boolean	false	Returns whether the column headers should contain horizontal bars styled by their data type.
getOutlierMode()	GRAPHOMATETABLE_1.getOutlierMode();	String	short	Returns a string representing the outlier mode.
getUseHierarchyMarker()	GRAPHOMATETABLE_1.getUseHierarchyMarker();	boolean	false	Returns the value of the property ’Use Hierarchy Marker’. If this property is set to true only the Hierarchy Marker (+/-) can be used to open or close hierarchy levels. Otherwise (set to false) the user may also click on the row title to open or close hierarchy levels.
getUseHierarchyStates()	GRAPHOMATETABLE_1.getUseHierarchyStates();	boolean	false	Returns the value of the property ’Use Hierarchy States’. This property is used to switch between the usage of ’Hierarchy States’ and ’Expand To’, as otherwise the properties may contradict each other.
getShowHeaderDimensions()	GRAPHOMATETABLE_1.getShowHeaderDimensions();	String	1,3	Returns a string containing a comma separated list of integer values. Each value is an index of a column header dimension level. All header dimension texts addressed by an index of this list are meant to be visible while the other ones aren’t displayed. A value of ” leads to all dimensions being visible.
getVisibleSeries()	GRAPHOMATETABLE_1.getVisibleSeries();	String	2,3	Returns a comma separated list of the visible data series indices.
getSaveSelectedMemberOnRightClick()	GRAPHOMATETABLE_1.getSaveSelectedMemberOnRightClick();	boolean	false	Returns whether a right click stores member information of the clicked cell retrievable by using the script method getSelectedMember.
getManualScaleMaximum()	GRAPHOMATETABLE_1.getManualScaleMaximum();	float	150	Returns the maximum value of the absolute scale.
getManualScaleMinimum()	GRAPHOMATETABLE_1.getManualScaleMinimum();	float	-150	Returns the minimum value of the absolute scale
getContextMenuCustomEventVisibility(int index)	GRAPHOMATETABLE_1.getContextMenuCustomEventVisibility(1);	boolean	false	Returns the visibility of a specified custom event in the Context Menu.
getContextMenuCustomEventName(int index)	GRAPHOMATETABLE_1.getContextMenuCustomEventName(1);	String	Custom Event 1	Returns the name of a specified custom event in the Context Menu.
getContextMenuOptionVisibility(int index)	GRAPHOMATETABLE_1.getContextMenuOptionVisibility(1);	boolean	false	Returns the visibility of a specified option in the Context Menu. Use the following indices to access the options: 1 for Sorting, 2 for Filters, 3 for Context Sensitive Filters, 4 for Filter and Hide, 5 for Clear Filters, 6 for Drilldown, 7 for Drillup, 8 for Zero Display, 9 for Member Display, 10 for Position of Totals, 11 for Total Display.

Setter Name	Example	Description
setAxisWidth(int value)	GRAPHOMATETABLE_1.setAxisWidth(1);	Setzt eine neue Achsenbreite.
setCollapsibleHierarchy(boolean value)	GRAPHOMATETABLE_1.setCollapsibleHierarchy(false);	Sets the visibility of the hierarchy control elements (+ and - before the line items).
setColumnOrder(String value)	var neworder = ’[{"identifier":"series1","position":2},{"identifier":"tableDeviations1","position":1},{"identifier":"series2", "position":0}]’ GRAPHOMATETABLE_1.setColumnOrder(neworder);	Sets the column order.
setColumnTitleVisible(boolean value)	GRAPHOMATETABLE_1.setColumnTitleVisible(false);	Sets the visibility of the column titles: If the value is false, the column titles are hidden.
setColumnWidth(int value)	GRAPHOMATETABLE_1.setColumnWidth(100);	This function sets the width of the columns. But it is only interpreted when fixedColumnWidth is set.
setComparisonGroup(String id)	GRAPHOMATETABLE_1.setComparisonGroup("A");	Sets the Comparison Group of the deviation charts in the table as a string. All charts in tables with the same Comparison Group are equal scaled and aligned.
setDatatypes(String value)	GRAPHOMATETABLE_1.setDatatypes("AC,PP,BU");	This function sets the Data Types for the entire table. Expects value as comma-separated string.
setDeviationBadColor(String value)	GRAPHOMATETABLE_1.setDeviationBadColor("#FF0000");	Sets the color of the negative deviations.
setDeviationGoodColor(String value)	GRAPHOMATETABLE_1.setDeviationGoodColor("#00FF00");	Sets the color for the positive deviations.
setDeviationLabel(String id, String label)	GRAPHOMATETABLE_1.setDeviationLabel("deviation2", "Percent");	Allows you to change the label of a deviation column. The ID of the column to be changed and the new label are expected as parameters of the method. The label of the respective column in the Property Deviations Config get changed. This is equivalent to changing the label from the Additional Properties Sheet.
setDeviationLabels(String value)	GRAPHOMATETABLE_1.setDeviationLabels("Absolute,Percent");	Sets new column headers for the deviations and changes the value of the Property Deviation Labels, which overwrites the column headers separately from the setting in the Additional Properties Sheet or the Standard Property Deviations Config.
setDeviationVisibility(String name, boolean visible)	GRAPHOMATETABLE_1.setDeviationVisibility("abs", true);	Sets the visibility of the deviation definition with the label name.
setExceptionPosition(String value)	GRAPHOMATETABLE_1.setExceptionPosition("leading");	Sets the value for positioning the exceptions. Possible values are leading (at the beginning oft he cell), before and after the value.
setExceptionVisibility(String name, boolean visible)	GRAPHOMATETABLE_1.setExceptionVisibility("good", true);	Sets the visibility of the Exception definition with the label name.

setExpandTo(int value)	GRAPHOMATETABLE_1.setExpandTo(2);	Sets the hierarchy level up to which the hierarchy is to be expanded in the start drilldown (starting with level 0). In conjunction with setting the visibility of the control elements for the hierarchy (see collapsibleHierarchy ()) a restriction in the visibility is possible (see Examples).
setExtendedNumberFormat(String value)	GRAPHOMATETABLE_1.setExtendedNumberFormat("-||.|,|1|1|");	Sets the number format for the Extended Mode. The format string can be found in the manual.
setExtendedNumberFormatPercentage(String value)	GRAPHOMATETABLE_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.
setFilterString(String value)	GRAPHOMATETABLE_1.setFilterString("revenue$");	Sets an initial text or regular expression into the filter in order to show a certain drilldown of the table at the start of the application.
setFixedColumnWidth(boolean value)	GRAPHOMATETABLE_1.setFixedColumnWidth(true);	This functions sets whether the Column Width should be fixed. If it is not fixed, the width is determined automatically.
setFixedHeader(boolean value)	GRAPHOMATETABLE_1.setFixedHeader(true);	This function sets whether the table header (column titles) are fix and therefore also visible while scrolling.
setFontFamily(String value)	GRAPHOMATETABLE_1.setFontFamily("Comic Sans");	Sets the font family.
setHierarchyStates(String states)	GRAPHOMATETABLE_1.setHierarchyStates('["EXPANDED", "COLLAPSED", "","","","", "COLLAPSED", "","","",""] ');	This function can be used to set the hierarchy state, i. e. automatically expand or collapse hierarchy nodes. As parameter, an array is expected which contains one value per row in the table. The array needs to hold either the string "EXPANDED" (child nodes are visible) or the string "COLLAPSED" (child nodes are hidden). An empty string ("") changes nothing.
setLabelFormatMode(String value)	GRAPHOMATETABLE_1.setLabelFormatMode("extended");	Sets a string value for the used label format mode. Available values are basic and extended.
setLocale(String value)	GRAPHOMATETABLE_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 value)	GRAPHOMATETABLE_1.setNegativeDeviationIsGood(false);	This function sets whether negative deviations are shown as positive or negative.
setNoClip(boolean value)	GRAPHOMATETABLE_1.setNoClip(false);	Sets the value for the noClip property.
setNumberFormat(String value)	GRAPHOMATETABLE_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 value)	GRAPHOMATETABLE_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).
setNumberFormatPerColumn(String value)	GRAPHOMATETABLE_1.setNumberFormatPerColumn('["","-||.|,|1|1|%25",""]');	Enables the setting of an extended number format for each column. The array contains different format strings. Empty strings are ignored and the number format of the table is used.
setRowTitleVisible(boolean value)	GRAPHOMATETABLE_1.setRowTitleVisible(false);	Sets the visibility of the row titles: If the value is false, the row titles are hidden.
setSemanticAxis(boolean value)	GRAPHOMATETABLE_1.setSemanticAxis(true);	Sets whether semantic axis for the devitations are displayed.
setSeries(int seriesIndex, String value)	GRAPHOMATETABLE_1.setSeries(1, '{"Keyfigures":"0D_NWI_NSAL"}');	Set the data description of the selected Data Series seriesIndex. Expects value as JSON string. This string either describes a row or a column in the Initial View. The string can be read out after data for a Data Series has been selected on the Property Sheet with the graphomate tables extension.
setSeriesLabels(String value)	GRAPHOMATETABLE_1.setSeriesLabels("Oct,Nov");	Sets the column titles of the data series.
setShowDeviationLabels(boolean value)	GRAPHOMATETABLE_1.setShowDeviationLabels(true);	Sets whether the values of the deviations are displayed.
setShowHierarchy(boolean value)	GRAPHOMATETABLE_1.setShowHierarchy(true);	Sets whether the row titles are displayed in a hierarchical structure.
setShowScalingFactor(boolean value)	GRAPHOMATETABLE_1.setShowScalingFactor(true);	Sets whether the scaling factor (coming from the Designer settings) is displayed.
setShowTitle(boolean value)	GRAPHOMATETABLE_1.setShowTitle(true);	Sets whether the title is displayed.
setShowUnits(boolean value)	GRAPHOMATETABLE_1.setShowUnits(true);	Sets whether the units (coming from the query) are displayed below the column title.
setSumBorders(boolean value)	GRAPHOMATETABLE_1.setSumBorders(true);	Sets whether Sum Borders are displayed.
setSumBorderColor(String value)	GRAPHOMATETABLE_1.setSumBorderColor("#00FFCC");	Sets the color of the Sum Borders.
setSumBorderThickness(float value)	GRAPHOMATETABLE_1.setSumBorderThickness(0.2);	Sets the thickness of the Sum Borders.
setSumPadding(float value)	GRAPHOMATETABLE_1.setSumPadding(0.3);	Sets the padding between the sums and the corresponding child elements.
setSuppressZeroLabels(boolean value)	GRAPHOMATETABLE_1.setSuppressZeroLabels(true);	Sets whether zero values are displayed in the table.
setTextualDeviation(boolean value)	GRAPHOMATETABLE_1.setTextualDeviation(true);	Sets whether textualDeviation is active. If it is true, no deviation charts are displayed, but the values are represented as numbers.
setTitle(String value)	GRAPHOMATETABLE_1.setTitle("Sales in Mio. EUR|2013 ACT and BUD|North West");	Sets the title of the table. Expects value as a string. Line breaks (\n) are interpreted as such.
setUseFormattedData(boolean value)	GRAPHOMATETABLE_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.
setUserHighlightColor(String value)	GRAPHOMATETABLE_1.setUserHighlightColor("red,#00ff00,#0000ff");	Sets a list of disposited colors for the User Highlights. Expects a comma separated string-value. The colors of the list are switched at each click (during runtime) until the desired color is set.
setUserHighlightList(String value)	GRAPHOMATETABLE_1.setUserHighlightList("[[1,3],[2,3][1,5]]");	Sets the list of user highlights (highlighted values at runtime). Expects a JSON-string consisting of one or more arrays with two numbers; the first number indicates the data row, while the second number represents the position of the value in the data series.
setWidthPerColumn(String val)	GRAPHOMATETABLE_1.getWidthPerColumn("[100,300,0,\"max200\"]");	Sets the Width Per Column-property to the given value.
setFontSize(...)	GRAPHOMATETABLE_1.setFontSize(15);	Sets the font size to the passed value.
setNoClip(...)	GRAPHOMATETABLE_1.setNoClip(true);	The setting of the property prevents the pots of the percentage deviations on the axis from being cut off. This will result in a less readable version, but will prevent a bug that may otherwise occur with Safari.
setZeroValueReplacement(...)	GRAPHOMATETABLE_1.setZeroValueReplacement("-");	The passed string is displayed instead of 0 values.
setInfinityValueReplacement(...)	GRAPHOMATETABLE_1.setInfinityValueReplacement("~");	The passed string is displayed instead of infinity values.
setEmptyValueReplacement(...)	GRAPHOMATETABLE_1.setEmptyValueReplacement("ŕ_()_/ŕ");	The passed string is displayed instead of null values.

setShowHeatmap(...)	GRAPHOMATETABLE_1.setShowHeatmap(true);	Ensures that the heatmap representation of the table is switched on or off.
setHeatmapColScale(...)	GRAPHOMATETABLE_1.setHeatmapColScale("Blues,5");	Sets the color scheme of the heat map.
setIncellColumns(...)	GRAPHOMATETABLE_1.setIncellColumns("[\"series1\",\"series2\"]");	Specifies the columns that should have In Cell Charts.
setIncellBarHeightFactor(...)	GRAPHOMATETABLE_1.setIncellBarHeightFactor(0.5);	Sets the factor used to calculate the height of the bars in the In Cell Charts.
setIncellBarOpactiy(...)	GRAPHOMATETABLE_1.setIncellBarOpactiy(0.1);	Sets the opacity of the bars of the In Cell Charts.
setIncellPositiveColor(...)	GRAPHOMATETABLE_1.setIncellPositiveColor("#0000ff");	Sets the color of the positive bars of the In Cell Charts.
setIncellNegativeColor(...)	GRAPHOMATETABLE_1.setIncellNegativeColor("#00ff00");	Sets the color of the negative bars of the In Cell Charts.
setUseOutlierThreshold(...)	GRAPHOMATETABLE_1.setUseOutlierThreshold(false);	Determines whether to use the Outlier Threshold.
setOutlierThreshold(...)	GRAPHOMATETABLE_1.setOutlierThreshold(5000000);	Sets the Outlier Threshold for absolute values.
setOutlierThresholdPercent(...)	GRAPHOMATETABLE_1.setOutlierThresholdPercent(0.5);	Sets the Outlier Threshold for percentage values.
setIbcsPinHeads(...)	GRAPHOMATETABLE_1.setIbcsPinHeads(true);	Represents the pin heads of the percentage deviations according to the IBCS.
setApplyPinColor(boolean applyPinColor)	GRAPHOMATETABLE_1.setApplyPinColor(true);	Sets whether the manual pin color should be applied. If false, the good or bad color will be applied.
setApplyPinHeadColor(boolean applyPinHeadColor)	GRAPHOMATETABLE_1.setApplyPinHeadColor(true);	Sets whether the manual pin head color should be applied. If false, the good or bad color will be applied.
setApplyDataTypesToCells(boolean applyDataTypesToCells)	GRAPHOMATETABLE_1.setApplyDataTypesToCells(true);	Sets whether the data types should be applied to the columns’ cells.
setContextMenuAvailable(boolean contextMenuAvailable)	GRAPHOMATETABLE_1.setContextMenuAvailable(true);	Sets whether the context menu is available via right click.
setDeviationLabelSizeFactor(float deviationLabelSizeFactor)	GRAPHOMATETABLE_1.setDeviationLabelSizeFactor(0.7);	Sets the deviation label font size factor.
setDataCellAlignmentRight(boolean dataCellAlignmentRight)	GRAPHOMATETABLE_1.setDataCellAlignmentRight(true);	Sets whether the text of data cells is aligned right.
setDeviationsFromNull(boolean deviationsFromNull)	GRAPHOMATETABLE_1.setDeviationsFromNull(true);	Sets whether NULL values are interpreted as zero values (0) for deviation calculations.
setExceptionsFromNull(boolean exceptionsFromNull)	GRAPHOMATETABLE_1.setExceptionsFromNull(true);	Sets whether NULL values are interpreted as zero values (0) for exception calculations.
setExceptionFillNoneInTextColor(boolean exceptionFillNoneInTextColor)	GRAPHOMATETABLE_1.setExceptionFillNoneInTextColor(true);	Sets whether the color of exception symbols with a fill type of ’none’ should be used for coloring the cells’ text instead of its background.
setShowFilterHelper(boolean showFilterHelper)	GRAPHOMATETABLE_1.setShowFilterHelper(true);	Sets the visibility of the filter helper input box.
setHorizontalGridLineThickness(float thickness)	GRAPHOMATETABLE_1.setHorizontalGridLineThickness(0.1);	Sets the thickness of horizontal grid lines in em.
setHighlightThickness(float thickness)	GRAPHOMATETABLE_1.setHighlightThickness(2);	Sets the thickness of user highlight borders in px.
setHierarchyIndent(float hierarchyIndent)	GRAPHOMATETABLE_1.setHierarchyIndent(1.2);	Sets the size of the hierarchy indent in em.
setManualScalePercentMax(float max)	GRAPHOMATETABLE_1.setManualScalePercentMax(50);	Sets the manual maximal value of the percentage scale.
setManualScalePercentMax(float min)	GRAPHOMATETABLE_1.setManualScalePercentMax(-50);	Sets the manual minimal value of the percentage scale.
setNodeStyleBold(boolean bold)	GRAPHOMATETABLE_1.setNodeStyleBold(true);	Sets whether the hierarchy nodes are styled using bold or normal font weight.
setNoWrap(boolean noWrap)	GRAPHOMATETABLE_1.setNoWrap(true);	Sets whether the row title texts should be prevented from wrapping when they are too long.
setNodeStyleItalic(boolean italic)	GRAPHOMATETABLE_1.setNodeStyleItalic(true);	Sets whether the hierarchy nodes are styled using italic or normal font style.
setRemoveRepeatedLabels(boolean removeRepeatedLabels)	GRAPHOMATETABLE_1.setRemoveRepeatedLabels(true);	Sets whether repeating dimension members across the columns should only show up at the column of their first occurrence.
setRowTitleAlignmentRight(boolean rowTitleAlignmentRight)	GRAPHOMATETABLE_1.setRowTitleAlignmentRight(true);	Sets whether the row title texts are aligned right.
setShowHorizontalGridLines(boolean visible)	GRAPHOMATETABLE_1.setShowHorizontalGridLines(true);	Sets the visibility of the horizontal grid lines.
setShowDataTypesInHeader(boolean showDataTypesInHeader)	GRAPHOMATETABLE_1.setShowDataTypesInHeader(true);	Sets whether the column headers should contain horizontal bars styled by their data type.
setOutlierMode(String outlierMode)	GRAPHOMATETABLE_1.setOutlierMode("short");	Sets a string representing the outlier mode.
setUseHierarchyMarker(boolean useHierarchyMarker)	GRAPHOMATETABLE_1.setUseHierarchyMarker(true);	Sets the value of the property ’Use Hierarchy Marker’. If this property is set to true only the Hierarchy Marker (+/-) can be used to open or close hierarchy levels. Otherwise (set to false) the user may also click on the row title to open or close hierarchy levels.
setUseHierarchyStates(boolean useHierarchyStates)	GRAPHOMATETABLE_1.setUseHierarchyStates(true);	Sets the value of the property ’Use Hierarchy States’. This property is used to switch between the usage of ’Hierarchy States’ and ’Expand To’, as otherwise the properties may contradict each other.
setShowHeaderDimensions(String visibleHeaderDimensionIndices)	GRAPHOMATETABLE_1.setShowHeaderDimensions("1,3");	Sets a string containing a comma separated list of integer values. Each value is an index of a column header dimension level. All header dimension texts addressed by an index of this list are meant to be visible while the other ones aren’t displayed. A value of ” leads to all dimensions being visible.
setVisibleSeries(String value)	GRAPHOMATETABLE_1.setVisibleSeries("2,3");	Sets a comma separated list of the visible data series indices.
setSaveSelectedMemberOnRightClick(boolean saveSelectedMemberOnRightClick)	GRAPHOMATETABLE_1.setSaveSelectedMemberOnRightClick(true);	Sets whether a right click stores member information of the clicked cell retrievable by using the script method getSelectedMember.
setContextMenuCustomEventVisibility(int index, boolean visibility)	GRAPHOMATETABLE_1.setContextMenuCustomEventVisibility(1, true)	Sets the visibility of a specified custom event in the Context Menu.
setContextMenuCustomEventName(int index, String name)	GRAPHOMATETABLE_1.setContextMenuCustomEventName(1, "My Custom Event")	Sets the name of a specified custom event in the Context Menu.

setContextMenuOptionVisibility(int index, boolean visibility)	GRAPHOMATETABLE_1.setContextMenuOptionVisibility(1, true)	Sets the visibility of a specified option in the Context Menu. Use the following indices to access the options: 1 for Sorting, 2 for Filters, 3 for Context Sensitive Filters, 4 for Filter and Hide, 5 for Clear Filters, 6 for Drilldown, 7 for Drillup, 8 for Zero Display, 9 for Member Display, 10 for Position of Totals, 11 for Total Display.
hide()	GRAPHOMATETABLE_1.hide()	An alternative to setVisible(false), which hides the table in a more performant way. The component gets hidden using CSS and does not need to be re-rendered after getting shown again. However, this can result in the table not reflecting possible changes that occurred while it was hidden.
show()	GRAPHOMATETABLE_1.show()	A table that was hidden by using the method hide() can only be displayed again with the help of this method.

1.4 Overview: Scripting language

The graphomate tables extension fully supports the scripting language of Designer. Therefore it is possible to control graphomate tables 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.

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.setItems(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);
```

Example: Save User Highlights by bookmarks

At runtime, you can set so-called User Highlights to selected values (see Interactivity at Runtime). To save this state of the dashboard, we need two buttons:

To save the id of the bookmark, for simplicity we use a text field (in the example TEXT_2), in which the id is written. This step can also hide visually using a variable for the id.

The following skript is assigned to the save button in order to save the actual state of the dashboard to variable named id. The variable is then being backed-up into a text box:

```
var id = Bookmark.saveBookmark("Bookmark");
TEXT_2.setText(id);
```

The next script is assigned to the load button in order to read the id from the text box and restore this bookmark state:

```
var id = TEXT_2.getText();
Bookmark.loadBookmark(id);
```

You may find more explanations and examples for the scripting language in our Scripting Documentation.