



# Scripting Dokumentation der graphomate tables 2.2 für SAP Lumira Designer 2.x

Version 2.2 - Stand Juni 2019

<https://www.graphomate.com>

# 1 Scripting-Dokumentation

## Scripting-Dokumentation der graphomate tables für SAP Lumira Designer

Version 2.2 – Stand Juni 2019

### 1.1 Einführung

Die graphomate tables Extension unterstützt die Skriptsprache von Lumira Designer. Somit ist es möglich, die Extension zur Laufzeit interaktiv zu verwenden. Beispielsweise können neue Daten gesetzt, Titel bearbeitet und Skalierungen verändert werden. Eine Auflistung der gegenwärtigen Funktionen und Beispiele zur Verwendung liefert dieses Dokument. Wir sind bemüht, den Funktionsumfang ständig zu erweitern. Sollten Ihnen bestimmte Funktionen fehlen, teilen Sie uns dies gerne über unser [Kontaktformular](#) mit.

### 1.2 Ereignisse

Die nachfolgend aufgelisteten Ereignisse stehen zur Verfügung. Auf diesen können im Property Sheet unter dem Punkt Events vom Benutzer geschriebene Skripte hinterlegt werden.

Ereignis Name	Beschreibung
On Element Clicked	Wird ausgelöst, wenn der Benutzer zur Laufzeit auf ein Element der Komponente in der Zeichenfläche klickt. Um einen Drilldown zu ermöglichen, muss an dieser Stelle auf die Benutzereingabe reagiert werden. Zum Zeitpunkt dieses Ereignisses werden die Rückgabewerte folgender Funktionen aktualisiert: <code>getSelectedMember()</code> <code>getSelectedMemberKey()</code> <code>getSelectedMemberText()</code> <code>getClickedElementIndex()</code> <code>getClickedSeriesIndex()</code>
On Input Changed	Dieses Event wird ausgelöst, wenn eine Eingabe in die Tabelle erfolgt ist. Um die Tabelle eingabebereit zu machen, kann der Style Content Editable der Row Format Configuration genutzt werden. Zum Auslesen des geänderten Tabellen-Werts wird die Methode <code>getChangedInput</code> benutzt. Über <code>getSelectedMember</code> kann eine Zuordnung zu dem in der Data Source hinterlegten Member vorgenommen werden.
On Hierarchy State Changed	Sobald ein Hierarchie-Knoten ein- oder ausgeklappt wird, wird dieses Event ausgelöst. Über <code>getSelectedMember</code> kann der Knoten, der verändert wurde, ermittelt werden. Mit der Methode <code>getHierarchyStateChange</code> ist es möglich herauszufinden, ob der Knoten ein- oder ausgeklappt wurde.

### 1.3 Funktionen

Folgende Funktionen stehen zur Verfügung. Diese können innerhalb des Skript-Editors von Designer angesprochen werden.

Getter Name	Beispiel	Rückgabotyp	Beispiel Rückgabewert	Beschreibung
<code>getAxisWidth()</code>	<code>GRAPHOMATETABLE_1.getAxisWidth();</code>	int	1	Gibt die aktuelle Achsenbreite zurück.
<code>getChangedInput()</code>	<code>GRAPHOMATETABLE_1.getChangedInput();</code>	String	Hello World!	Gibt den Wert der zuletzt geänderten Tabellen-Zelle zurück. Nähere Informationen dazu sind unter dem Event On Input Changed aufgeführt.
<code>getClickedElementIndex()</code>	<code>GRAPHOMATETABLE_1.getClickedElementIndex();</code>	int	3	Gibt den Index des als letztes geklickten Elements zurück. Gibt -1 zurück, wenn auf eine Zelle geklickt wurde, die nicht Teil einer Datenserie ist.
<code>getClickedSeriesIndex()</code>	<code>GRAPHOMATETABLE_1.getClickedSeriesIndex();</code>	int	1	Gibt den Index der als letztes geklickten Datenserie zurück. Gibt -1 zurück, wenn auf eine Zelle geklickt wurde, die nicht Teil einer Datenserie ist.
<code>getCollapsibleHierarchy()</code>	<code>GRAPHOMATETABLE_1.getCollapsibleHierarchy();</code>	boolean	true	Gibt zurück, ob die Steuerelemente der Hierarchie (+ und - vor dem Zeilentitel) angezeigt werden.
<code>getColumnOrder()</code>	<code>GRAPHOMATETABLE_1.getColumnOrder();</code>	String	<code>[{"identifier":"series1","position":0}, {"identifier":"deviation1", "position":1}, {"identifier":"series2", "position":2}, {"identifier":"deviation2", "position":3}]</code>	Gibt die aktuelle Sortierung der Spalten (Serien und Abweichungen) zurück.
<code>getColumnTitleVisible()</code>	<code>GRAPHOMATETABLE_1.getColumnTitleVisible();</code>	boolean	false	Gibt die Sichtbarkeit der Spaltentitel zurück.
<code>getColumnWidth()</code>	<code>GRAPHOMATETABLE_1.getColumnWidth();</code>	int	20	Gibt die Spaltenbreite als Integer-Wert zurück.
<code>getComparisonGroup()</code>	<code>GRAPHOMATETABLE_1.getComparisonGroup();</code>	String	A	Gibt die Comparison Group als String zurück.
<code>getDatatypes()</code>	<code>GRAPHOMATETABLE_1.getDatatypes();</code>	String	AC,PP,BU	Diese Funktion liefert die Datenarten (Data Types) als kommaseparierten String zurück.
<code>getDeviationBadColor()</code>	<code>GRAPHOMATETABLE_1.getDeviationBadColor();</code>	String	#FF0000	Gibt die aktuelle Farbe für negative Abweichungen zurück.
<code>getDeviationGoodColor()</code>	<code>GRAPHOMATETABLE_1.getDeviationGoodColor();</code>	String	#00FF00	Gibt die aktuelle Farbe für positive Abweichungen zurück.
<code>getDeviationIds()</code>	<code>GRAPHOMATETABLE_1.getDeviationIds();</code>	String	deviation1,deviation2	Gibt eine Liste der IDs aller Abweichungsspalten zurück. Die Elemente der Liste sind durch ein Komma getrennt.

getDeviationLabels()	GRAPHOMATETABLE_1.getDeviationLabels();	String	Prozent,Absolut	Gibt die aktuell vergebenen Spaltentitel der Abweichungsserien zurück, wenn diese über die Standardproperty Deviation Labels gesetzt wurden, die unabhängig von der Einstellung den Labels aus Additional Properties Sheet oder Standardproperty Deviations Config die Spaltenköpfe für Abweichungsspalten überschreibt.
getDeviationLabelsAndIds()	GRAPHOMATETABLE_1.getDeviationLabelsAndIds();	String	deviation1:Absolut,deviation2:Prozent	Gibt die Liste der Paare von IDs und Label aller Abweichungsspalten zurück. Dies sind die Labels, die auf dem Additional Properties Sheet oder durch die Standardproperty Deviations Config vergeben wurden. Die ID der Abweichungsspalte ist von dem Label durch einen Doppelpunkt getrennt, die einzelnen Paare durch ein Komma.
getDeviationLabelById(String id)	GRAPHOMATETABLE_1.getDeviationLabelById("deviation2");	String	Prozent	Gibt das Label der Abweichungsspalte zurück, deren ID bei dem Methodenaufruf übergeben wurde. Dies ist das Label, das auf dem Additional Properties Sheet oder durch die Standardproperty Deviations Config vergeben wurde.
getDeviationVisibility(String name)	GRAPHOMATETABLE_1.getDeviationVisibility("Prozent");	boolean	true	Gibt die Sichtbarkeit der Deviation-Definition mit dem Label name zurück.
getExceptionPosition()	GRAPHOMATETABLE_1.getExceptionPosition();	String	leading	Gibt zurück, in welcher Weise die Exceptions positioniert werden (zu Beginn der Tabellenzelle, vor oder hinter dem Wert).
getExceptionVisibility(String name)	GRAPHOMATETABLE_1.getExceptionVisibility("good");	boolean	true	Gibt die Sichtbarkeit der Exception-Definition mit dem Label name zurück.
getExpandTo()	GRAPHOMATETABLE_1.getExpandTo();	int	3	Mit expandTo kann festgelegt werden, welches Aufriss-Level der Hierarchie dem Nutzer zu Beginn angezeigt werden soll. Diese get-Methode gibt das gewählte Level zurück.
getExtendedNumberFormat()	GRAPHOMATETABLE_1.getExtendedNumberFormat();	String	- . , 1 1	Gibt das Nummernformat für den Extended-Modus zurück. Die Bestandteile und Bedeutung des Format-Strings können im <a href="#">Handbuch</a> nachgelesen werden.
getExtendedNumberFormatPercentage()	GRAPHOMATETABLE_1.getExtendedNumberFormatPercentage();	String	- . , 1 1 %25	Gibt das Nummernformat für die Prozentwerte im Extended-Modus zurück. Die Bestandteile und Bedeutung des Format-Strings können im <a href="#">Handbuch</a> nachgelesen werden.
getFilterString()	GRAPHOMATETABLE_1.getFilterString();	String	kosten\$	Gibt den Filter-Text zurück.
getFixedColumnWidth()	GRAPHOMATETABLE_1.getFixedColumnWidth();	boolean	true	Diese Funktion gibt zurück, ob die Spaltenbreite fix ist (andernfalls wird die Breite automatisch anhand des zur Verfügung stehenden Platzes gesetzt).
getFixedHeader()	GRAPHOMATETABLE_1.getFixedHeader();	boolean	true	Gibt zurück, ob der Tabellenkopf (sprich die Spaltentitel) auch beim Scrollen immer sichtbar sein soll.
getFontFamily()	GRAPHOMATETABLE_1.getFontFamily();	String	Comic Sans MS	Gibt die aktuell ausgewählte Schriftart zurück.
getHierarchyStateChange()	GRAPHOMATETABLE_1.getHierarchyStateChange();	String	EXPANDED	Bei einem Klick auf ein Hierarchy-Element (+/-) wird gesichert, ob der Knoten geöffnet (EXPANDED) oder geschlossen (CLOSED) wurde. Diese Information kann über diese Script-Methode für den letzten Klick ausgelesen werden. Die Methode sollte zusammen mit dem Event On Hierarchy State Change verwendet werden.
getHierarchyStates()	GRAPHOMATETABLE_1.getHierarchyStates();	String	"EXPANDED","COLLAPSED","","","","","COLLAPSED","","","",""	Gibt den aktuellen Status der Hierarchie zurück, also welche Knoten eingeklappt sind und welche nicht. Der Rückgabe-Wert ist ein Array dessen Länge der Anzahl der Zeilen entspricht. Für jede Zeile ist entsprechend vermerkt, ob deren "Kinder" dargestellt werden oder nicht. Hinweis: Damit diese Funktion reibungslos funktioniert, muss die Property Use Hierarchy Marker deaktiviert werden.
getLabelFormatMode()	GRAPHOMATETABLE_1.getLabelFormatMode();	String	extended	Gibt zurück, welcher Modus für die Label-Formatierung verwendet wird.
getLocale()	GRAPHOMATETABLE_1.getLocale();	String	en	Gibt die Einstellung zurück, die aktuell für locale gewählt ist.
getNegativeDeviationIsGood()	GRAPHOMATETABLE_1.getNegativeDeviationIsGood();	boolean	true	Gibt zurück, ob negative Abweichungen als positiv abgebildet werden.
getNoClip()	GRAPHOMATETABLE_1.getNoClip();	boolean	true	Gibt den Wert der noClip-Option zurück.
getNumberFormat()	GRAPHOMATETABLE_1.getNumberFormat();	String	0,0.0	Gibt das Nummernformat für den Basic-Modus zurück. Der Formatstring entspricht dem Format von numeral.js, siehe <a href="#">Handbuch</a> .
getNumberFormatPercentage()	GRAPHOMATETABLE_1.getNumberFormatPercentage();	String	0 %	Gibt das Nummernformat für die Prozentwerte im Basic-Modus zurück. Der Formatstring entspricht dem Format von numeral.js, siehe <a href="#">Handbuch</a> .
getRowTitleVisible()	GRAPHOMATETABLE_1.getRowTitleVisible();	boolean	false	Gibt die Sichtbarkeit der Zeilentitel zurück.
getSelectedMember(String dimensionKey)	GRAPHOMATETABLE_1.getSelectedMember("0D_NWI_RCOD");	Member	undefined	Nachdem der User ein Element/eine Kategorie der Tabelle geklickt hat, gibt diese Funktion den Member der spezifizierten Dimension dimensionKey zurück. Der Member stellt verschiedene Properties wie text, internalKey und externalKey zur Verfügung. Wenn eine nicht valide Dimension angegeben wurde oder auf ein Element geklickt wurde, das keinen Datenwert repräsentiert, wird undefined zurückgegeben.
getSelectedMemberKey(String dimensionKey)	GRAPHOMATETABLE_1.getSelectedMemberKey("0D_NWI_RCOD");	String	10274	Nachdem der User ein Element/eine Kategorie der Tabelle geklickt hat, gibt diese Funktion den Schlüssel (Key) des Members der spezifizierten Dimension dimensionKey zurück. Die Schlüssel der Dimensionen sind in der Initial View ersichtlich.
getSelectedMemberText(String dimensionKey)	GRAPHOMATETABLE_1.getSelectedMemberText("0D_NWI_RCOD");	String	North West	Nachdem der User ein Element/eine Kategorie der Tabelle geklickt hat, gibt diese Funktion den Text des Members der spezifizierten Dimension dimensionKey zurück. Die Rückgabe ist analog zu der Funktion getSelectedMemberKey().

getSemanticAxis()	GRAPHOMATETABLE_1.getSemanticAxis();	boolean	true	Gibt zurück, ob für die Abweichungen eine semantische Achse gezeichnet wird.
getSeries(int seriesIndex)	GRAPHOMATETABLE_1.getSeries(2);	String	{"Keyfigures":"0D_NWI_NSAL"}	Liefert die Datenbeschreibung der angegebenen Datenserie seriesIndex als JSON-String zurück.
getSeriesLabels()	GRAPHOMATETABLE_1.getSeriesLabels();	String	Jan,Feb	Gibt die aktuell vergebenen Labels der Serien zurück (die als Spaltentitel angezeigt werden).
getShowDeviationLabels()	GRAPHOMATETABLE_1.getShowDeviationLabels();	boolean	false	Gibt zurück, ob die Werte der Abweichungen angezeigt werden (falls nicht, erscheinen nur die Balken).
getShowHierarchy()	GRAPHOMATETABLE_1.getShowHierarchy();	boolean	true	Gibt zurück, ob die Zeilentitel der Tabelle hierarchisch dargestellt werden sollen.
getShowScalingFactor()	GRAPHOMATETABLE_1.getShowScalingFactor();	boolean	true	Gibt zurück, ob der aus den Designer-Einstellungen kommende Skalierungsfaktor unter dem Spaltentitel angezeigt werden soll.
getShowTitle()	GRAPHOMATETABLE_1.getShowTitle();	boolean	true	Gibt zurück, ob der Titel angezeigt wird.
getShowUnits()	GRAPHOMATETABLE_1.getShowUnits();	boolean	false	Gibt zurück, ob unter den Spaltentiteln die aus der Query kommenden Einheiten angezeigt werden sollen.
getSumBorders()	GRAPHOMATETABLE_1.getSumBorders();	boolean	false	Gibt zurück, ob Summen-Separatoren angezeigt werden.
getSumBorderColor()	GRAPHOMATETABLE_1.getSumBorderColor();	String	#00FFAA	Gibt zurück, welche Farbe für die Summen-Separatoren verwendet wird.
getSumBorderThickness()	GRAPHOMATETABLE_1.getSumBorderThickness();	float	0.1	Gibt die aktuelle Dicke des Summen-Separators zurück.
getSumPadding()	GRAPHOMATETABLE_1.getSumPadding();	float	1.2	Gibt den Abstand zwischen Summe und Kind-Elementen der Summe zurück.
getSuppressZeroLabels()	GRAPHOMATETABLE_1.getSuppressZeroLabels();	boolean	true	Gibt zurück, ob 0-Werte in der Tabelle angezeigt werden.
getTextualDeviation()	GRAPHOMATETABLE_1.getTextualDeviation();	boolean	true	Gibt zurück, ob textualDeviation gesetzt ist. Wenn textualDeviation gesetzt ist, dann werden keine Abweichungsspalten gezeichnet, sondern die Spalte wird wie eine reguläre Tabellenspalte (mit Texten angezeigt).
getTitle()	GRAPHOMATETABLE_1.getTitle();	String	Sales in Mio. EUR 2013 ACT and BUD North West	Gibt den Titel der Tabelle als String wieder. Zeilenumbrüche werden durch einen senkrechten Strich ( ) repräsentiert.
getUseFormattedData()	GRAPHOMATETABLE_1.getUseFormattedData();	boolean	false	Gibt zurück, ob die aus der Query kommenden, vorgeformatierten Daten zur Darstellung der Werte verwendet werden.
getUserHighlightColor()	GRAPHOMATETABLE_1.getUserHighlightColor();	String	red,green	Gibt eine Liste der hinterlegten User Highlight-Farben als kommagetrennten String zurück. Die Farben werden in der angegebenen Reihenfolge bei den User Highlights (zur Laufzeit hervorgehobene Werte) durchgeschaltet, bis die Wunschfarbe eingestellt ist.
getUserHighlightList()	GRAPHOMATETABLE_1.getUserHighlightList();	String	[[1,3],[2,3][1,5]]	Gibt eine Liste der aktuellen User Highlights (zur Laufzeit hervorgehobene Werte) als JSON-String zurück, der die Positionen der Highlights jeweils mit zwei Zahlen übergibt. Dabei gibt die erste Zahl die Datenreihe an, während die zweite Zahl die Position des Wertes in der Datenreihe darstellt.
getWidthPerColumn()	GRAPHOMATETABLE_1.getWidthPerColumn();	String	[100,300,0,"max200"]	Gibt die aktuell gesetzte Einstellung für die Width Per Column-Property zurück.
getFontSize()	GRAPHOMATETABLE_1.getFontSize();	float	12	Gibt die aktuell gewählte Schriftgröße zurück.
getNoClip()	GRAPHOMATETABLE_1.getNoClip();	boolean	false	Gibt den aktuellen Wert von No Clip zurück.
getZeroValueReplacement()	GRAPHOMATETABLE_1.getZeroValueReplacement();	String	-	Gibt das Zero Value Replacement zurück. Leer, wenn es kein Zero Value Replacement gibt.
getInfinityValueReplacement()	GRAPHOMATETABLE_1.getInfinityValueReplacement();	String	~	Gibt das Infinity Value Replacement zurück. Leer, wenn es kein Infinity Value Replacement gibt.
getEmptyValueReplacement()	GRAPHOMATETABLE_1.getEmptyValueReplacement();	String	~\_()\_/'	Gibt das Empty Value Replacement zurück. Leer, wenn es kein Empty Value Replacement gibt.
getShowHeatmap()	GRAPHOMATETABLE_1.getShowHeatmap();	boolean	true	Gibt true zurück, wenn die Tabelle als Heatmap dargestellt wird, sonst false.
getHeatmapColScale()	GRAPHOMATETABLE_1.getHeatmapColScale();	String	Reds,4	Gibt das gewählte Farbschema der Heatmap zurück.
getIncellColumns()	GRAPHOMATETABLE_1.getIncellColumns();	String	["series1","series2"]	Gibt die Spalten zurück, die über In Cell Charts verfügen.
getIncellBarHeightFactor()	GRAPHOMATETABLE_1.getIncellBarHeightFactor();	float	0.7	Gibt den Faktor zurück, der zur Berechnung der Höhe der Balken der In Cell Charts verwendet wird.
getIncellBarOpacity()	GRAPHOMATETABLE_1.getIncellBarOpacity();	float	0.2	Gibt die Opazität der In Cell Charts zurück.
getIncellPositiveColor()	GRAPHOMATETABLE_1.getIncellPositiveColor();	String	#ff0000	Gibt die Farbe der positiven Balken der In Cell Charts zurück.
getIncellNegativeColor()	GRAPHOMATETABLE_1.getIncellNegativeColor();	String	#00ff00	Gibt die Farbe der negativen Balken der In Cell Charts zurück.
getUseOutlierThreshold()	GRAPHOMATETABLE_1.getUseOutlierThreshold();	boolean	true	Gibt zurück, ob der Outlier Threshold verwendet wird.
getOutlierThreshold()	GRAPHOMATETABLE_1.getOutlierThreshold();	float	1000000	Gibt den Outlier Threshold für absolute Werte zurück.
getOutlierThresholdPercent()	GRAPHOMATETABLE_1.getOutlierThresholdPercent();	float	0.4	Gibt den Outlier Threshold für prozentuale Werte zurück.
getIbcsPinHeads()	GRAPHOMATETABLE_1.getIbcsPinHeads();	boolean	false	Gibt zurück, ob die Abweichungen gemäß der IBCS gezeichnet werden.
getApplyPinColor()	GRAPHOMATETABLE_1.getApplyPinColor();	boolean	false	Gibt zurück, ob die manuelle Pin Color auf Pins der Abweichungscharts angewendet wird. Ansonsten wird die jeweilige Abweichungsfarbe angewendet.



getApplyPinHeadColor()	GRAPHOMATETABLE_1.getApplyPinHeadColor();	boolean	false	Gibt zurück, ob die manuelle Pin Head Color auf die Pinköpfe der Abweichungscharts angewendet wird. Ansonsten wird die jeweilige Abweichungsfarbe angewendet.
getApplyDataTypesToCells()	GRAPHOMATETABLE_1.getApplyDataTypesToCells();	boolean	false	Gibt zurück, ob die die Data Types der Spalten auf alle ihre Zellen angewendet werden sollen.
getContextMenuAvailable()	GRAPHOMATETABLE_1.getContextMenuAvailable();	boolean	false	Gibt zurück, ob das Kontextmenü durch Rechtsklick erreichbar ist.
getDeviationLabelSizeFactor()	GRAPHOMATETABLE_1.getDeviationLabelSizeFactor();	float	0.7	Gibt den Deviation Label Font Size Factor zurück, der die Schriftgröße der Deviation Labels in Abhängigkeit von der Schriftgröße des normalen Textes beschreibt.
getDataCellAlignmentRight()	GRAPHOMATETABLE_1.getDataCellAlignmentRight();	boolean	false	Gibt zurück, ob der Inhalt von Datenzellen rechtsbündig ist.
getDeviationsFromNull()	GRAPHOMATETABLE_1.getDeviationsFromNull();	boolean	false	Gibt zurück, ob NULL-Werte bei der Abweichungsberechnung als 0-Werte interpretiert werden sollen.
getExceptionsFromNull()	GRAPHOMATETABLE_1.getExceptionsFromNull();	boolean	false	Gibt zurück, ob NULL-Werte bei der Berechnung von Exceptions als 0-Werte interpretiert werden sollen.
getExceptionFillNoneInTextColor()	GRAPHOMATETABLE_1.getExceptionFillNoneInTextColor();	boolean	false	Gibt zurück, ob die Farbe von Exceptions mit dem fill type 'none' auf den Text einer Zelle angewendet werden wird, anstatt auf den Hintergrund.
getShowFilterHelper()	GRAPHOMATETABLE_1.getShowFilterHelper();	boolean	false	Gibt die Sichtbarkeit der Filter Helper Inputbox zurück.
getHorizontalGridLineThickness()	GRAPHOMATETABLE_1.getHorizontalGridLineThickness();	float	0.1	Gibt die Dicke der horizontalen Gitterlinien in em zurück.
getHighlightThickness()	GRAPHOMATETABLE_1.getHighlightThickness();	float	3	Gibt die Dicke der User Highlight Umrandungen in px zurück.
getHierarchyIndent()	GRAPHOMATETABLE_1.getHierarchyIndent();	float	1.2	Gibt die Größe der Einrückung von Hierarchieleveln in em zurück.
getManualScalePercentMax()	GRAPHOMATETABLE_1.getManualScalePercentMax();	float	50	Gibt den manuell festgelegten Maximalwert für die Skalierung von Prozentwerten zurück.
getManualScalePercentMin()	GRAPHOMATETABLE_1.getManualScalePercentMin();	float	-50	Gibt den manuell festgelegten Minimalwert für die Skalierung von Prozentwerten zurück.
getNodeStyleBold()	GRAPHOMATETABLE_1.getNodeStyleBold();	boolean	false	Gibt zurück, ob die Hierarchieknoten fettgedruckt oder normal dargestellt werden.
getNoWrap()	GRAPHOMATETABLE_1.getNoWrap();	boolean	false	Gibt zurück, ob die Zeilentitel daran gehindert werden umzubrechen, wenn sie zu lang sind.
getNodeStyleItalic()	GRAPHOMATETABLE_1.getNodeStyleItalic();	boolean	false	Gibt zurück, ob die Hierarchieknoten kursiv oder normal dargestellt werden.
getRemoveRepeatedLabels()	GRAPHOMATETABLE_1.getRemoveRepeatedLabels();	boolean	false	Gibt zurück, ob mehrmals auftretende Dimensionsbeschreibungen nur einmal dargestellt werden. Für den Wert false werden dagegen bei mehreren Vorkommnissen gleicher Dimensionstexte in unterschiedlichen Spalten alle Texte redundant angezeigt.
getRowTitleAlignmentRight()	GRAPHOMATETABLE_1.getRowTitleAlignmentRight();	boolean	false	Gibt zurück, ob die Zeilentitel rechtsbündig sind.
getShowHorizontalGridLines()	GRAPHOMATETABLE_1.getShowHorizontalGridLines();	boolean	false	Gibt die Sichtbarkeit der horizontalen Gitterlinien zurück.
getShowDataTypesInHeader()	GRAPHOMATETABLE_1.getShowDataTypesInHeader();	boolean	false	Gibt zurück, ob die Spaltenüberschriften horizontale Balken enthalten, die den Datentyp der jeweiligen Spalte darstellen.
getOutlierMode()	GRAPHOMATETABLE_1.getOutlierMode();	String	short	Gibt die Darstellungsart von Ausreißern zurück.
getUseHierarchyMarker()	GRAPHOMATETABLE_1.getUseHierarchyMarker();	boolean	false	Gibt den Wert der Property 'Use Hierarchy Marker' zurück. Wenn diese Property auf true gesetzt ist, können nur die Hierarchy Marker (+/-) verwendet werden, um ein Hierarchielevel auf- und zuzuklappen. Ansonsten kann auch auf den gesamten Text des Hierarchielevels geklickt werden.
getUseHierarchyStates()	GRAPHOMATETABLE_1.getUseHierarchyStates();	boolean	false	Gibt den Wert der Property 'Use Hierarchy States' zurück. Diese Property dient dazu zwischen der Verwendung der 'Hierarchy States' und 'Expand To' zu wählen, da beide Properties den Zustand der Hierarchieaufklappung verändern und sonst interferieren würden.
getShowHeaderDimensions()	GRAPHOMATETABLE_1.getShowHeaderDimensions();	String	1,3	Gibt einen String zurück, der eine kommaseparierte Liste aus Ganzzahlen darstellt. Jeder dieser Werte stellt einen Index dar, der eine Zeile der Spaltenüberschriften adressiert. Alle so adressierten Texte der Spaltenüberschriften werden dargestellt, alle anderen werden ausgeblendet. Eine leere Liste ("" ) führt dazu, dass alle Dimensionsbeschriftungen in den Spaltenüberschriften angezeigt werden.
getVisibleSeries()	GRAPHOMATETABLE_1.getVisibleSeries();	String	2,3	Gibt einen String zurück, der eine kommaseparierte Liste aus Datenspalten-Indices darstellt. Alle damit adressierten Spalten sind sichtbar, die anderen werden versteckt. Eine leere Liste ("" ) führt dazu, dass alle Spalten sichtbar sind.
getSaveSelectedMemberOnRightClick()	GRAPHOMATETABLE_1.getSaveSelectedMemberOnRightClick();	boolean	false	Gibt zurück, ob bei einem Rechtsklick auf eine Zelle ihre Memberinformationen gespeichert werden sollen oder nicht. Die Informationen sind dann über getSelectedMember() abrufbar.
getManualScaleMaximum()	GRAPHOMATETABLE_1.getManualScaleMaximum();	float	150	Gibt den Maximalwert der absoluten Skala wieder, die für die Skalierung der Abweichungsspalten verwendet wird.
getManualScaleMinimum()	GRAPHOMATETABLE_1.getManualScaleMinimum();	float	-150	Gibt den Minimalwert der absoluten Skala wieder, die für die Skalierung der Abweichungsspalten verwendet wird.
getContextMenuCustomEventVisibility(int index)	GRAPHOMATETABLE_1.getContextMenuCustomEventVisibility(1);	boolean	false	Gibt die Sichtbarkeit des angefragten Custom-Events zurück. Der übergebene Index ist 1-indiziert.

getContextMenuCustomEventName(int index)	GRAPHOMATETABLE_1.getContextMenuCustomEventName(1);	String	Custom Event 1	Gibt den Namen des angefragten Custom-Events zurück. Der übergebene Index ist 1-indiziert.
getContextMenuOptionVisibility(int index)	GRAPHOMATETABLE_1.getContextMenuOptionVisibility(1);	boolean	false	Gibt die Sichtbarkeit des angefragten Kontextmenüeintrags zurück. Der übergebene Index ist 1-indiziert und richtet sich nach der Reihenfolge der Einträge im Additional Properties Sheet auf dem Tab "Context Menu".

Setter Name	Beispiel	Beschreibung
setAxisWidth(int val)	GRAPHOMATETABLE_1.setAxisWidth(1);	Setzt eine neue Achsenbreite.
setCollapsibleHierarchy(boolean val)	GRAPHOMATETABLE_1.setCollapsibleHierarchy(true);	Setzt, ob die Steuerelemente der Hierarchie (+ und - vor dem Zeilentitel angezeigt werden sollen).
setColumnOrder(String val)	var neworder = '[{"identifier":"series1","position":2},{"identifier":"tableDeviations1","position":1},{"identifier":"series2","position":0}]' GRAPHOMATETABLE_1.setColumnOrder(neworder);	Setzt eine neue Sortierung der Spalten.
setColumnTitleVisible(boolean val)	GRAPHOMATETABLE_1.setColumnTitleVisible(false);	Setzt die Sichtbarkeit der Spaltentitel: Wenn der Wert true ist, werden die Zeilentitel ausgeblendet.
setColumnWidth(int val)	GRAPHOMATETABLE_1.setColumnWidth(100);	Setzt die Breite der Spalten auf den Integer-Wert val (Angabe in Pixeln). Die Breite kommt nur dann zur Anwendung, wenn fixedColumnWidth aktiviert ist.
setComparisonGroup(String id)	GRAPHOMATETABLE_1.setComparisonGroup("A");	Setzt die Comparison Group der Abweichungsdiagramme der Tabelle. Tabellen mit der gleichen Comparison Group werden gleich skaliert und ausgerichtet.
setDatatypes(String val)	GRAPHOMATETABLE_1.setDatatypes("AC,PP,BU");	Setzt die Datentypen für die gesamte Tabelle. Erwartet val als kommaseparierten String.
setDeviationBadColor(String color)	GRAPHOMATETABLE_1.setDeviationBadColor("#FF0000");	Setzt eine neue Farbe für die negativen Abweichungen.
setDeviationGoodColor(String color)	GRAPHOMATETABLE_1.setDeviationGoodColor("#00FF00");	Setzt eine neue Farbe für die positiven Abweichungen.
setDeviationLabel(String id, String label)	GRAPHOMATETABLE_1.setDeviationLabel("deviation2", "Prozent");	Ermöglicht das Neusetzen des Labels einer Abweichungsspalte. Als Parameter der Methode werden die ID der zu ändernden Spalte und das neue Label erwartet. Es wird das Label der jeweiligen Spalte im Wert der Property Deviations Config verändert. Dies kommt einer Änderung des Labels aus dem Additional Properties Sheet gleich.
setDeviationLabels(String val)	GRAPHOMATETABLE_1.setDeviationLabels("Absolut,Prozent");	Setzt neue Spaltentitel für die Abweichungen und verändert den Wert der Property Deviation Labels, die separat von der Einstellung im Additional Properties Sheet oder der Standardproperty Deviations Config die Spaltenköpfe überschreibt.
setDeviationVisibility(String name, boolean visible)	GRAPHOMATETABLE_1.setDeviationVisibility("Absolut", true);	Setzt die Sichtbarkeit der Deviation-Definition mit dem Label name auf den Wert visible.
setExceptionPosition(String val)	GRAPHOMATETABLE_1.setExceptionPosition("leading");	Setzt den Wert für die Positionierung der Exceptions. Mögliche Werte sind leading (zu Beginn der Tabellenzelle), before (direkt vor dem Wert) und after (direkt hinter dem Wert).
setExceptionVisibility(String name, boolean visible)	GRAPHOMATETABLE_1.setExceptionVisibility("good", true);	Setzt die Sichtbarkeit der Exception-Definition mit dem Label name auf den Wert visible.
setExpandTo(int val)	GRAPHOMATETABLE_1.setExpandTo(2);	Setzt das Aufriss-Level der Tabelle auf den Wert value; der Wert ist 0-indexiert (d. h. die erste Hierarchie-Ebene hat den Wert 0, die zweite den Wert 1 usw.). Im Zusammenspiel mit dem Setzen der Sichtbarkeit der Steuerelemente für die Hierarchie (siehe collapsibleHierarchy()) kann dadurch eine Beschränkung in der Sichtbarkeit eingeführt werden.
setExtendedNumberFormat(String format)	GRAPHOMATETABLE_1.setExtendedNumberFormat("- . , 1 1 ");	Setzt das Nummernformat für den Extended-Modus. Der Formatstring kann im <a href="#">Handbuch</a> nachgelesen werden.
setExtendedNumberFormatPercentage(String format)	GRAPHOMATETABLE_1.setExtendedNumberFormatPercentage("- . , 1 1 %25");	Setzt das Nummernformat für die Prozentwerte im Extended-Modus. Der Formatstring kann im <a href="#">Handbuch</a> nachgelesen werden.
setFilterString(String filter)	GRAPHOMATETABLE_1.setFilterString("kosten\$");	Setzt einen initialen Text oder regulären Ausdruck in den Filter, um dem Nutzer beim Start der Applikation einen bestimmten Aufriss der Tabelle zu zeigen.
setFixedColumnWidth(boolean val)	GRAPHOMATETABLE_1.setFixedColumnWidth(true);	Diese Funktion setzt, ob die Spaltenbreite fix sein soll. Ist dieser Wert nicht gesetzt, wird die Breite automatisch ermittelt.
setFixedHeader(boolean val)	GRAPHOMATETABLE_1.setFixedHeader(true);	Diese Funktion setzt, ob der Tabellenkopf (die Spaltentitel) auch beim Scrollen immer sichtbar sein soll.
setFontFamily(String font)	GRAPHOMATETABLE_1.setFontFamily("Comic Sans");	Setzt eine neue Schriftart.
setHierarchyStates(String states)	GRAPHOMATETABLE_1.setHierarchyStates('["EXPANDED", "COLLAPSED", "", "", "", "", "COLLAPSED", "", "", "", ""]');	Erlaubt das Setzen des Status der Hierarchie. Dadurch können Knoten innerhalb der Hierarchie ein- und ausgeklappt werden. Die Methode erwartet als Parameter ein Array dessen Länge der Anzahl der Zeilen entspricht. Das Array kann die String-Werte EXPANDED (Kinder des Knotens sind sichtbar) oder COLLAPSED (Kinder des Knotens sind nicht sichtbar) enthalten. Ein Leerstring ("") bewirkt keine Änderung am Hierarchiezustand.
setLabelFormatMode(String val)	GRAPHOMATETABLE_1.setLabelFormatMode("extended");	Setzt als String, welcher Modus für die Label-Formatierung verwendet wird. Hierbei stehen basic und extended zur Verfügung.
setLocale(String val)	GRAPHOMATETABLE_1.setLocale("de");	Setzt einen neuen locale-Wert. Zulässige Werte für das locale sind: de, en, fr und auto. Wird auto ausgewählt, dann wird das locale aus der Query für die Formatierung der Texte hergenommen.
setNegativeDeviationIsGood(boolean val)	GRAPHOMATETABLE_1.setNegativeDeviationIsGood(false);	Diese Funktion setzt, ob negative Abweichungen als positiv oder negativ abgebildet werden.
setNoClip(boolean val)	GRAPHOMATETABLE_1.setNoClip(false);	Setzt den Wert der noClip-Option.
setNumberFormat(String val)	GRAPHOMATETABLE_1.setNumberFormat("0,0.0");	Setzt das Nummernformat für den Basic-Modus. Der Formatstring entspricht dem Format von numeral.js, siehe <a href="#">Handbuch</a> .
setNumberFormatPercentage(String val)	GRAPHOMATETABLE_1.setNumberFormatPercentage("0 %");	Setzt das Nummernformat für die Prozentwerte im Basic-Modus. Der Formatstring entspricht dem Format von numeral.js, siehe <a href="#">Handbuch</a> .
setNumberFormatPerColumn(String vals)	GRAPHOMATETABLE_1.setNumberFormatPerColumn('["", "- . , 1 1 %25", ""]');	Ermöglicht das Setzen eines extended number formats pro Spalte. Die Übergabe der Formate erfolgt in einem Array aus Format-Strings, Leer-Strings werden ignoriert und zur Formatierung wird dann das Format der Tabelle herangezogen. Beispiel: Die zweite Spalte der Tabelle mit einem Nummernformat auszeichnen (alle anderen Spalten werden in dem für die Tabelle vorgegebenen Nummernformat ausgezeichnet).
setRowTitleVisible(boolean val)	GRAPHOMATETABLE_1.setRowTitleVisible(false);	Setzt die Sichtbarkeit der Zeilentitel: Wenn der Wert false ist, werden die Zeilentitel ausgeblendet.
setSemanticAxis(boolean val)	GRAPHOMATETABLE_1.setSemanticAxis(true);	Setzt, ob semantische Achsen für die Abweichungen de- oder aktiviert werden.

setSeries(Integer seriesIndex, String value)	GRAPHOMATETABLE_1.setSeries(1, '{"Keyfigures":"0D_NWI_NSAL"}');	Setzt die Datenbeschreibung der spezifizierten Datenserie seriesIndex. Erwartet value als JSON-String. Dieser String beschreibt entweder eine Zeile oder eine Spalte in der Initial View. Der String kann ausgelesen werden, nachdem man mit der graphomate tables Extension im Property Sheet Daten für eine Datenserie selektiert hat. Beispiel: Die Datenbeschreibung der ersten Datenserie auf den übergebenen Wert setzen.
setSeriesLabels(String val)	GRAPHOMATETABLE_1.setSeriesLabels("Oct,Nov");	Setzt neue Spaltentitel.
setShowDeviationLabels(boolean val)	GRAPHOMATETABLE_1.setShowDeviationLabels(true);	Setzt, ob die Werte der Abweichungen angezeigt werden.
setShowHierarchy(boolean val)	GRAPHOMATETABLE_1.setShowHierarchy(true);	Setzt, ob die Zeilentitel hierarchisch dargestellt werden sollen.
setShowScalingFactor(boolean val)	GRAPHOMATETABLE_1.setShowScalingFactor(true);	Setzt, ob der aus den Designer-Einstellungen kommende Skalierungsfaktor angezeigt werden soll.
setShowTitle(boolean val)	GRAPHOMATETABLE_1.setShowTitle(true);	Setzt, ob der Titel angezeigt werden soll.
setShowUnits(boolean val)	GRAPHOMATETABLE_1.setShowUnits(true);	Setzt, ob die aus der Query kommenden Einheiten einer Spalte unter dem Spaltentitel angezeigt werden sollen.
setSumBorders(boolean val)	GRAPHOMATETABLE_1.setSumBorders(true);	Setzt, ob die Anzeige von Summen-Separatoren aktiviert oder deaktiviert ist.
setSumBorderColor(String val)	GRAPHOMATETABLE_1.setSumBorderColor("#00FFCC");	Setzt eine neue Farbe für den Summen-Separator.
setSumBorderThickness(float val)	GRAPHOMATETABLE_1.setSumBorderThickness(0.2);	Setzt die Dicke des Summen-Separators.
setSumPadding(float val)	GRAPHOMATETABLE_1.setSumPadding(0.3);	Setzt den Abstand zwischen Summe und den Kind-Elementen der Summe.
setSuppressZeroLabels(boolean val)	GRAPHOMATETABLE_1.setSuppressZeroLabels(true);	Setzt, ob 0-Werte in der Tabelle angezeigt werden oder deren Anzeige unterdrückt wird.
setTextualDeviation(boolean val)	GRAPHOMATETABLE_1.setTextualDeviation(true);	Setzt, ob die textualDeviation aktiv ist. Ist dies der Fall, werden keine Abweichungsdiagramme gezeichnet, sondern die Spalte wird wie eine reguläre Tabellenspalte (mit Texten) angezeigt.
setTitle(String val)	GRAPHOMATETABLE_1.setTitle("Sales in Mio. EUR 2013 ACT and BUD North West");	Setzt den Titel der Tabelle. Erwartet val als String. Zeilenumbrüche (\n) werden auch als solche interpretiert.
setUseFormattedData(boolean val)	GRAPHOMATETABLE_1.setUseFormattedData(true);	Setzt, ob die aus Designer vorformatierten Daten für die Tabelle verwendet werden. Dies ist nur möglich für die Werte, die aus der Query übergeben werden. Werte, die in der Tabelle selbst berechnet werden z.B. für die Abweichungen verwenden zur Formatierung weiterhin entweder das Basic oder das Extended <a href="#">Nummernformat</a> .
setUserHighlightColor(String val)	GRAPHOMATETABLE_1.setUserHighlightColor("red,#00ff00,#0000ff");	Setzt eine Liste der hinterlegten User Highlight-Farben. Erwartet einen kommaseparierten String. Die gesetzten Farben werden dabei in der angegebenen Reihenfolge bei den User Highlights (zur Laufzeit hervorgehobene Werte) durchgeschaltet, bis die Wunschfarbe eingestellt ist.
setUserHighlightList(String val)	GRAPHOMATETABLE_1.setUserHighlightList("[[1,3],[2,3][1,5]]");	Setzt ein oder mehrere User Highlights (zur Laufzeit hervorgehobene Werte). Erwartet einen JSON-String bestehend aus einem oder mehreren Arrays mit jeweils zwei Zahlen. Dabei gibt die erste Zahl die Datenreihe an, während die zweite Zahl die Position des Wertes in der Datenreihe darstellt.
setWidthPerColumn(String val)	GRAPHOMATETABLE_1.setWidthPerColumn("[100,300,0,\"max200\"]");	Setzt die Width Per Column-Property neu.
setFontSize(...)	GRAPHOMATETABLE_1.setFontSize(15);	Setzt die Schriftgröße auf den übergebenen Wert
setNoClip(...)	GRAPHOMATETABLE_1.setNoClip(true);	Das Setzen der Property verhindert, dass die Pinköpfe der prozentualen Abweichungscharts an der Achse abgeschnitten werden. Das führt zwar zu einer schwerer lesbaren Darstellung, verhindert aber einen Fehler, der sonst in Zusammenhang mit Safari auftreten kann.
setZeroValueReplacement(...)	GRAPHOMATETABLE_1.setZeroValueReplacement("-");	Der übergebene String wird anstatt von 0-Werten angezeigt.
setInfinityValueReplacement(...)	GRAPHOMATETABLE_1.setInfinityValueReplacement("~");	Der übergebene String wird anstatt von Infinity-Werten angezeigt.
setEmptyValueReplacement(...)	GRAPHOMATETABLE_1.setEmptyValueReplacement("f_/r");	Der übergebene String wird anstatt von null-Werten angezeigt.
setShowHeatmap(...)	GRAPHOMATETABLE_1.setShowHeatmap(true);	Sorgt dafür, dass die Heatmap-Darstellung der Tabelle an- oder ausgeschaltet wird.
setHeatmapColScale(...)	GRAPHOMATETABLE_1.setHeatmapColScale("Blues,5");	Setzt das Farbschema der Heatmap.
setIncellColumns(...)	GRAPHOMATETABLE_1.setIncellColumns("[\"series1\",\"series2\"]");	Legt die Spalten fest, die über In Cell Charts verfügen sollen.
setIncellBarHeightFactor(...)	GRAPHOMATETABLE_1.setIncellBarHeightFactor(0.5);	Legt den Faktor fest, der für die Berechnung der Höhe der Balken der In Cell Charts verwendet wird.
setIncellBarOpactiy(...)	GRAPHOMATETABLE_1.setIncellBarOpactiy(0.1);	Legt die Opazität der Balken der In Cell Charts fest.
setIncellPositiveColor(...)	GRAPHOMATETABLE_1.setIncellPositiveColor("#0000ff");	Legt die Farbe der positiven Balken der In Cell Charts fest.
setIncellNegativeColor(...)	GRAPHOMATETABLE_1.setIncellNegativeColor("#00ff00");	Legt die Farbe der negativen Balken der In Cell Charts fest.
setUseOutlierThreshold(...)	GRAPHOMATETABLE_1.setUseOutlierThreshold(false);	Legt fest, ob die Outlier Threshold Anwendung finden sollen.
setOutlierThreshold(...)	GRAPHOMATETABLE_1.setOutlierThreshold(5000000);	Setzt den Outlier Threshold für absolute Werte.
setOutlierThresholdPercent(...)	GRAPHOMATETABLE_1.setOutlierThresholdPercent(0.5);	Setzt den Outlier Threshold für prozentuale Werte.
setIbcsPinHeads(...)	GRAPHOMATETABLE_1.setIbcsPinHeads(true);	Stellt die Pinköpfe der prozentualen Abweichungen gemäß der IBCS dar.
setApplyPinColor(boolean applyPinColor)	GRAPHOMATETABLE_1.setApplyPinColor(true);	Bestimmt, ob die manuelle Pin Color auf Pins der Abweichungscharts angewendet wird. Ansonsten wird die jeweilige Abweichungsfarbe angewendet.
setApplyPinHeadColor(boolean applyPinHeadColor)	GRAPHOMATETABLE_1.setApplyPinHeadColor(true);	Bestimmt, ob die manuelle Pin Head Color auf die Pinköpfe der Abweichungscharts angewendet wird. Ansonsten wird die jeweilige Abweichungsfarbe angewendet.
setApplyDataTypesToCells(boolean applyDataTypesToCells)	GRAPHOMATETABLE_1.setApplyDataTypesToCells(true);	Bestimmt, ob die die Data Types der Spalten auf alle ihre Zellen angewendet werden sollen.
setContextMenuAvailable(boolean contextMenuAvailable)	GRAPHOMATETABLE_1.setContextMenuAvailable(true);	Bestimmt, ob das Kontextmenü durch Rechtsklick erreichbar ist.
setDeviationLabelSizeFactor(float deviationLabelSizeFactor)	GRAPHOMATETABLE_1.setDeviationLabelSizeFactor(0.7);	Setzt den Deviation Label Font Size Factor, der die Schriftgröße der Deviation Labels in Abhängigkeit von der Schriftgröße des normalen Textes beschreibt.
setDataCellAlignmentRight(boolean dataCellAlignmentRight)	GRAPHOMATETABLE_1.setDataCellAlignmentRight(true);	Bestimmt, ob der Inhalt von Datenzellen rechtsbündig ist.
setDeviationsFromNull(boolean deviationsFromNull)	GRAPHOMATETABLE_1.setDeviationsFromNull(true);	Bestimmt, ob NULL-Werte bei der Abweichungsberechnung als 0-Werte interpretiert werden sollen.



setExceptionsFromNull(boolean exceptionsFromNull)	GRAPHOMATETABLE_1.setExceptionsFromNull(true);	Bestimmt, ob NULL-Werte bei der Berechnung von Exceptions als 0-Werte interpretiert werden sollen.
setExceptionFillNoneInTextColor(boolean exceptionFillNoneInTextColor)	GRAPHOMATETABLE_1.setExceptionFillNoneInTextColor(true);	Bestimmt, ob die Farbe von Exceptions mit dem fill type 'none' auf den Text einer Zelle angewendet werden wird, anstatt auf den Hintergrund.
setShowFilterHelper(boolean showFilterHelper)	GRAPHOMATETABLE_1.setShowFilterHelper(true);	Setzt die Sichtbarkeit der Filter Helper Inputbox.
setHorizontalGridLineThickness(float thickness)	GRAPHOMATETABLE_1.setHorizontalGridLineThickness(0.1);	Setzt die Dicke der horizontalen Gitterlinien in em.
setHighlightThickness(float thickness)	GRAPHOMATETABLE_1.setHighlightThickness(2);	Setzt die Dicke der User Highlight Umrandungen in px.
setHierarchyIndent(float hierarchyIndent)	GRAPHOMATETABLE_1.setHierarchyIndent(1.2);	Setzt die Größe der Einrückung von Hierarchieleveln in em.
setManualScalePercentMax(float max)	GRAPHOMATETABLE_1.setManualScalePercentMax(50);	Setzt den manuell festgelegten Maximalwert für die Skalierung von Prozentwerten.
setManualScalePercentMax(float min)	GRAPHOMATETABLE_1.setManualScalePercentMax(-50);	Setzt den manuell festgelegten Minimalwert für die Skalierung von Prozentwerten.
setNodeStyleBold(boolean bold)	GRAPHOMATETABLE_1.setNodeStyleBold(true);	Bestimmt, ob die Hierarchieknoten fettgedruckt oder normal dargestellt werden.
setNoWrap(boolean noWrap)	GRAPHOMATETABLE_1.setNoWrap(true);	Bestimmt, ob die Zeilentitel daran gehindert werden umzuberechnen, wenn sie zu lang sind.
setNodeStyleItalic(boolean italic)	GRAPHOMATETABLE_1.setNodeStyleItalic(true);	Bestimmt, ob die Hierarchieknoten kursiv oder normal dargestellt werden.
setRemoveRepeatedLabels(boolean removeRepeatedLabels)	GRAPHOMATETABLE_1.setRemoveRepeatedLabels(true);	Bestimmt, ob mehrmals auftretende Dimensionsbeschreibungen nur einmal dargestellt werden. Für den Wert false werden dagegen bei mehreren Vorkommnissen gleicher Dimensionstexte in unterschiedlichen Spalten alle Texte redundant angezeigt.
setRowTitleAlignmentRight(boolean rowTitleAlignmentRight)	GRAPHOMATETABLE_1.setRowTitleAlignmentRight(true);	Bestimmt, ob die Zeilentitel rechtsbündig sind.
setShowHorizontalGridLines(boolean visible)	GRAPHOMATETABLE_1.setShowHorizontalGridLines(true);	Setzt die Sichtbarkeit der horizontalen Gitterlinien.
setShowDataTypesInHeader(boolean showDataTypesInHeader)	GRAPHOMATETABLE_1.setShowDataTypesInHeader(true);	Bestimmt, ob die Spaltenüberschriften horizontale Balken enthalten, die den Datentyp der jeweiligen Spalte darstellen.
setOutlierMode(String outlierMode)	GRAPHOMATETABLE_1.setOutlierMode("short");	Setzt die Darstellungsart von Ausreißern.
setUseHierarchyMarker(boolean useHierarchyMarker)	GRAPHOMATETABLE_1.setUseHierarchyMarker(true);	Setzt den Wert der Property 'Use Hierarchy Marker'. Wenn diese Property auf true gesetzt ist, können nur die Hierarchy Marker (+/-) verwendet werden, um ein Hierarchielevel auf- und zuzuklappen. Ansonsten kann auch auf den gesamten Text des Hierarchielevels geklickt werden.
setUseHierarchyStates(boolean useHierarchyStates)	GRAPHOMATETABLE_1.setUseHierarchyStates(true);	Setzt den Wert der Property 'Use Hierarchy States'. Diese Property dient dazu zwischen der Verwendung der 'Hierarchy States' und 'Expand To' zu wählen, da beide Properties den Zustand der Hierarchieaufklappung verändern und sonst interferieren würden.
setShowHeaderDimensions(String visibleHeaderDimensionIndices)	GRAPHOMATETABLE_1.setShowHeaderDimensions("1,3");	Setzt einen String, der eine kommaseparierte Liste aus Ganzzahlen darstellt. Jeder dieser Werte stellt einen Index dar, der eine Zeile der Spaltenüberschriften adressiert. Alle so adressierten Texte der Spaltenüberschriften werden dargestellt, alle anderen werden ausgeblendet. Eine leere Liste ("") führt dazu, dass alle Dimensionsbeschriftungen in den Spaltenüberschriften angezeigt werden.
setVisibleSeries(String value)	GRAPHOMATETABLE_1.setVisibleSeries("2,3");	Setzt einen String, der eine kommaseparierte Liste aus Datenspalten-Indices darstellt. Alle damit adressierten Spalten sind sichtbar, die anderen werden versteckt. Eine leere Liste ("") führt dazu, dass alle Spalten sichtbar sind.
setSaveSelectedMemberOnRightClick(boolean saveSelectedMemberOnRightClick)	GRAPHOMATETABLE_1.setSaveSelectedMemberOnRightClick(true);	Bestimmt, ob bei einem Rechtsklick auf eine Zelle ihre Memberinformationen gespeichert werden sollen oder nicht. Die Informationen sind dann über getSelectedMember() abrufbar.
setContextMenuCustomEventVisibility(int index, boolean visibility)	GRAPHOMATETABLE_1.setContextMenuCustomEventVisibility(1, true)	Setzt die Sichtbarkeit des angegebenen Custom-Events. Der übergebene Index ist 1-indiziert.
setContextMenuCustomEventName(int index, String name)	GRAPHOMATETABLE_1.setContextMenuCustomEventName(1, "My Custom Event")	Setzt den Namen des angegebenen Custom-Events. Der übergebene Index ist 1-indiziert.
setContextMenuOptionVisibility(int index, boolean visibility)	GRAPHOMATETABLE_1.setContextMenuOptionVisibility(1, true)	Setzt die Sichtbarkeit des angegebenen Kontextmenüeintrags. Der übergebene Index ist 1-indiziert und richtet sich nach der Reihenfolge der Einträge im Additional Properties Sheet auf dem Tab "Context Menu".
hide()	GRAPHOMATETABLE_1.hide()	Alternative zu setVisible(false), die ein performanteres Ausblenden der Tabelle bewirkt. Dazu wird die Komponente mittels CSS ausgeblendet und muss nach dem Einblenden nicht neu gerendert werden. Dies kann allerdings dazu führen, dass die Tabelle mögliche Änderungen, die auftraten während sie ausgeblendet war, nach dem Einblenden nicht reflektiert.
show()	GRAPHOMATETABLE_1.show()	Eine Tabelle, die durch die Methode hide() ausgeblendet wurde, kann nur mit Hilfe dieser Methode wieder eingeblendet werden.

### 1.3.1 Beispiele

Die Sichtbarkeit von Exceptions "togglen":

```
var exc1 = GRAPHOMATETABLE_1.getExceptionVisibility("good");
var exc2 = GRAPHOMATETABLE_1.getExceptionVisibility("bad");
GRAPHOMATETABLE_1.setExceptionVisibility("good", !exc1);
GRAPHOMATETABLE_1.setExceptionVisibility("bad", !exc2);
```

Nur die ersten beiden Hierarchie-Ebenen anzeigen und die Navigation ausschalten (fixieren):



```
GRAPHOMATETABLE_1.setExpandTo(1);
GRAPHOMATETABLE_1.setCollapsibleHierarchy(false);
```

### Weitere Beispiele

Für weitere Beispiele schauen Sie bitte in unsere [Scripting-Dokumentation](#) für die graphomate charts Extension.

## 1.4 Übersicht Skriptsprache

Die graphomate tables Extension unterstützt die Skriptsprache von Designer. Somit ist es möglich graphomate tables zur Laufzeit interaktiv zu steuern.

Beispielsweise können neue Daten gesetzt und Titel bearbeitet verändert werden. Weiterhin ist es möglich, Drilldown-Funktionalitäten der Extension zu verwenden und interaktive Dashboards zu erstellen.

### Beispiel: Mit Hilfe einer Dropdown-Box eine Data Source dynamisch Filtern

Um auf einer Dimension einer Data Source dynamisch zu filtern, muss zuerst eine Dropdown-Box mit allen zulässigen Werten befüllt werden, damit der User aus dieser später dann seine Wahl treffen kann.

Dafür wird auf dem Canvas-Event On Startup von der gewünschten Data Source mit `getMemberList` eine Liste aller Member der Dimension `OD_NWI_ACOD` geholt. Die Parameter von `getMemberList` werden dabei vom Autocomplete vorgeschlagen.

Die so erlangte Liste kann dann per `setItem` auf die Dropdown-Box gesetzt werden.

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

Nun wird die Dropdown-Box beim Applikationsstart mit den Mitgliedern der Dimension `OD_NWI_ACOD` befüllt.

Selektiert nun ein User ein Element aus der Liste, kann mit folgendem Skript, welches auf das Ereignis On Element Clicked der Dropdown-Box gelegt wird, der Filter der Data Source angepasst werden:

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

### Beispiel: User Highlights mittels Bookmarks speichern

Zur Laufzeit lassen sich sogenannte User Highlights um ausgewählte Werte setzen (siehe [Interaktivität zur Laufzeit](#)). Um diesen Zustand zu speichern, bedarf es zweier Schaltflächen:

Um die ID des Bookmarks zwischen zu speichern, bedienen wir uns der Einfachheit halber eines Textfeldes (im Beispiel `TEXT_2`), in das die ID geschrieben wird. Diesen Schritt kann man mittels Variablen auch visuell ausblenden.

Der für das Speichern zuständige Button erhält folgendes Skript, um in die Variable `id` den Bookmark-Zustand zu speichern. Die Variable wird zur Sicherung dann in ein Textfeld geschrieben:

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

Der für das Laden zuständige Button erhält folgendes Skript und liest damit die ID aus dem Textfeld aus und stellt diesen Bookmark-Zustand wieder her:

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

Weitere Erläuterungen und Beispiele zur Skriptsprache finden Sie in unserer [Scripting-Dokumentation](#).