



Scripting Dokumentation der graphomate trees 2.1 für SAP Lumira Designer 2.x

Version 2.1 - Stand Oktober 2018

<https://www.graphomate.com>

1 Scripting-Dokumentation

Scripting-Dokumentation der graphomate trees für SAP Lumira Designer

Version 2.1 – Stand September 2018

1.1 Einführung

Die graphomate trees unterstützt die Skriptsprache von 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>getSelectedMemberKey()</code> , <code>getSelectedMemberText()</code> , <code>getClickedElementIndex()</code> , <code>getClickedSeriesIndex()</code>
On New Data Arrival	Wird ausgelöst, wenn die tree aktualisierte Daten erhält. Dies geschieht zum Beispiel, nachdem sich die Dimensionen einer Data Source geändert haben oder wenn Filter verändert wurden. Eine Anpassung der Datenarten (Data Types) wäre an dieser Stelle zum Beispiel möglich.

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>getColorScheme()</code>	<code>GRAPHOMATETREE_1.getColorScheme();</code>	String	Spectral,9	Gibt das aktuell für die tree ausgewählte ColorBrewer-Farbschema zurück. Die Rückgabe enthält das Schema und – durch ein Komma separiert – die Anzahl der Klassen.
<code>getColorCalculationMethod()</code>	<code>GRAPHOMATETREE_1.getColorCalculationMethod();</code>	String	hierarchy	Gibt zurück, welche Methode zur Einfärbung der Rechtecke der tree verwendet wird.
<code>getDepth()</code>	<code>GRAPHOMATETREE_1.getDepth();</code>	int	3	Gibt zurück, auf wie viele Level die Beschränkung der Hierarchie-Tiefe festgelegt ist.
<code>getExtendedNumberFormat()</code>	<code>GRAPHOMATETREE_1.getExtendedNumberFormat();</code>	String	- . , 1 1	Gibt das Nummernformat für den Extended-Modus zurück. Der Formatstring kann im Handbuch nachgelesen werden.
<code>getExtendedNumberFormatPercentage()</code>	<code>GRAPHOMATETREE_1.getExtendedNumberFormatPercentage();</code>	String	- . , 1 1 %25	Gibt das Nummernformat für die Prozentwerte im Extended-Modus zurück. Der Formatstring kann im Handbuch nachgelesen werden.
<code>getLabelFormatMode()</code>	<code>GRAPHOMATETREE_1.getLabelFormatMode();</code>	String	extended	Gibt zurück, welcher Modus für die Label-Formatierung verwendet wird.
<code>getLimitDepth()</code>	<code>GRAPHOMATETREE_1.getLimitDepth();</code>	boolean	true	Gibt zurück, ob die Begrenzung der Hierarchie-Tiefe, die in der tree dargestellt wird, möglich ist.
<code>getLocale()</code>	<code>GRAPHOMATETABLE_1.getLocale();</code>	String	fr	Gibt die Einstellung zurück, die aktuell für locale gewählt ist.
<code>getNegativeDeviationIsGood()</code>	<code>GRAPHOMATETREE_1.getNegativeDeviationIsGood();</code>	boolean	true	Gibt zurück, ob die negativen Abweichungen als positiv, oder als negativ interpretiert werden.
<code>getNumberFormat()</code>	<code>GRAPHOMATETREE_1.getNumberFormat();</code>	String	0,0.0	Gibt das Nummernformat für den Basic-Modus zurück. Der Formatstring entspricht dem Format von numeral.js, siehe Handbuch .
<code>getNumberFormatPercentage()</code>	<code>GRAPHOMATETREE_1.getNumberFormatPercentage();</code>	String	0 %	Gibt das Nummernformat für die Prozentwerte im Basic-Modus zurück. Der Formatstring entspricht dem Format von numeral.js, siehe Handbuch .
<code>getRestPadding()</code>	<code>GRAPHOMATETREE_1.getRestPadding();</code>	int	3	Gibt die Breite der linken, rechten und unteren Seite des Rahmens zurück, der im embedded-Modus gezeichnet wird.
<code>getSelectedMember(Dimension dimensionKey)</code>	<code>GRAPHOMATETREE_1.getSelectedMember("0D_NWI_RC0D");</code>	Member		Nachdem der User ein Element/eine Kategorie der tree geklickt hat, gibt diese Funktion den Member der spezifizierten Dimension <code>dimensionKey</code> zurück. Der Member stellt verschiedene Properties wie <code>text</code> , <code>internalKey</code> und <code>externalKey</code> zur Verfügung.

getSelectedMemberKey(String dimensionKey)	GRAPHOMATETREE_1.getSelectedMemberKey("0D_NWI_RCOD");	String	10274	Nachdem der User ein Element/eine Kategorie der tree 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)	GRAPHOMATETREE_1.getSelectedMemberText("0D_NWI_RCOD");	String	North West	Nachdem der User ein Element/eine Kategorie der tree geklickt hat, gibt diese Funktion den Text des Members der spezifizierten Dimension dimensionKey zurück. Die Rückgabe ist analog zu der Funktion getSelectedMemberKey().
getTooltipVisible()	GRAPHOMATETREE_1.getTooltipVisible();	boolean	true	Gibt zurück, ob zur Laufzeit Tooltips zur tree angezeigt werden.
getUseFormattedData()	GRAPHOMATETREE_1.getUsedFormattedData();	boolean	true	Gibt zurück, ob in der Query vorformatierte Daten zur Anzeige in den trees verwendet werden.
getTreeType()	GRAPHOMATETREE_1.getTreeType();	String	vertical	Gibt den gewählten Typ der Visualisierung zurück. Der Typ ist einer der folgenden Werte: circle, horizontal, vertical
getFontSize()	GRAPHOMATETREE_1.getFontSize();	float	12	Gibt die Schriftgröße zurück.
getTextColor()	GRAPHOMATETREE_1.getTextColor();	String	#000000	Gibt die Schriftfarbe zurück.
getNodeFillColor()	GRAPHOMATETREE_1.getNodeFillColor();	String	#c0ffee	Gibt die Füllfarbe der Knoten des Baumes zurück.
getNodeStrokeColor()	GRAPHOMATETREE_1.getNodeStrokeColor();	String	#c0ffee	Gibt die Farbe zurück, in der die Umrandung der Knoten des Baumes gezeichnet ist.
getNodeStrokeThickness()	GRAPHOMATETREE_1.getNodeStrokeThickness();	float	0.7	Gibt die Dicke der Umrandung zurück.
getLinkStrokeColor()	GRAPHOMATETREE_1.getLinkStrokeColor();	String	#c0ffee	Gibt die Farbe der Verknüpfungen zwischen den Knoten zurück.
getLinkStrokeThickness()	GRAPHOMATETREE_1.getLinkStrokeThickness();	float	0.9	Gibt die Dicke der Verknüpfungen zurück.
getShowValue()	GRAPHOMATETREE_1.getShowValue();	boolean	true	Gibt zurück, ob neben der Beschriftung eines Kontens auch dessen Wert angezeigt wird.
getFontFamily()	GRAPHOMATETREE_1.getFontFamily();	String	Comic Sans	Gibt die gewählte Schriftart zurück.
getLinkLayout()	GRAPHOMATETREE_1.getLinkLayout();	String	cross	Gibt das Layout der Verknüpfungen zurück. Einer der folgenden Werte ist möglich: swung, straight, elbow, cross
getRootNodeDescription()	GRAPHOMATETREE_1.getRootNodeDescription();	String	Summe	Gibt die Beschriftung des Wurzelknoten zurück, falls manuell eine Beschriftung angelegt wurde.
getInfinityColor()	GRAPHOMATETREE_1.getInfinityColor();	String	#c0ffee	Gibt die Farbe zurück, die für Knoten vergeben wird, deren Wert NaN oder Infinity (zum Beispiel bei Teilung durch 0) ist.
getNullColor()	GRAPHOMATETREE_1.getNullColor();	String	#0ff1ce	Gibt die Farbe zurück, die für Knoten vergeben wird, deren Wert Null (bspw. weil kein Wert in der Data Source vorhanden ist) ist.
getAscendingSortOrder()	GRAPHOMATETREE_1.getAscendingSortOrder();	boolean	false	Gibt die gewählte Sortierung der Knoten zurück.
getAreaScale()	GRAPHOMATETREE_1.getAreaScale();	String	log	Gibt zurück, welche Methode für die Skalierung der Flächen gewählt worden ist. Mögliche Werte umfassen: linear, log
getMinNodeRadius()	GRAPHOMATETREE_1.getMinNodeRadius();	float	23	Gibt den minimalen Radius eines Knotens zurück.
getMaxNodeRadius()	GRAPHOMATETREE_1.getMaxNodeRadius();	float	42	Gibt den maximalen Radius eines Knotens zurück.
getDescriptionDelimiter()	GRAPHOMATETREE_1.getDescriptionDelimiter();	String		Gibt zurück, welches Trennzeichen für die Darstellung von Label und Wert an einem Knoten verwendet wird.

Setter Name	Beispiel	Beschreibung
setColCalculationMethod(String val)	GRAPHOMATETREE_1.setColorCalculationMethod("hierarchy");	Setzt die Methode nach der die Rechtecke der TREE eingefärbt werden. Zulässige Werte für method sind: hierarchy, arealIdentity, colorIdentity, absoluteDeviation, percentDeviation.
setColorSchema(String val)	GRAPHOMATETREE_1.setColorScheme("Set1, 4");	Ermöglicht das Setzen eines neuen ColorBrewer-Farbschemas und der Anzahl der Klassen. Beide Angaben müssen durch ein Komma separiert und in einem String übergeben werden.
setDepth(int level)	GRAPHOMATETREE_1.setDepth(2);	Beschränkt die in der TREE dargestellte Hierarchie-Tiefe auf den Wert level.
setExtendedNumberFormat(String val)	GRAPHOMATETREE_1.setExtendedNumberFormat("- . , 1 1 ");	Setzt das Nummernformat für den Extended Modus. Der Formatstring kann im Handbuch nachgelesen werden.
setExtendedNumberFormatPercentage(String val)	GRAPHOMATETREE_1.setExtendedNumberFormatPercentage("- . , 1 1 %25");	Setzt das Nummernformat für die Prozentwerte im Extended -Modus. Der Formatstring kann im Handbuch nachgelesen werden.
setLabelFormatMode(String val)	GRAPHOMATETREE_1.setLabelFormatMode("extended");	Setzt als String, welcher Modus für die Label-Formatierung verwendet wird. Hierbei stehen basic und extended zur Verfügung.
setLimitDepth(boolean val)	GRAPHOMATETREE_1.setLimitDepth(false);	Setzt, ob die Begrenzung der Hierarchie-Tiefe angeschaltet ist.
setLocale(String val)	GRAPHOMATETREE_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)	GRAPHOMATETREE_1.setNegativeDeviationIsGood(true);	Setzt, ob die negativen Abweichungen als positiv, oder als negativ interpretiert werden sollen.
setNumberFormat(String val)	GRAPHOMATETREE_1.setNumberFormat("0,0.0");	Setzt das Nummernformat für den Basic-Modus. Der Formatstring entspricht dem Format von numeral.js, siehe Handbuch .
setNumberFormatPercentage(String val)	GRAPHOMATETREE_1.setNumberFormatPercentage("0 %");	Setzt das Nummernformat für die Prozentwerte im Basic-Modus. Der Formatstring entspricht dem Format von numeral.js, siehe Handbuch .
setTooltipVisible(boolean val)	GRAPHOMATETREE_1.setTooltipVisible(true);	Setzt, ob in den TREES Tooltips angezeigt werden.

setUseFormattedData(boolean val)	GRAPHOMATETREE_1.setUseFormattedData(true);	Setzt, ob die aus Design Studio 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 Nummernformat.
setTreeType(String val)	GRAPHOMATETREE_1.setTreeType("circle");	Setzt den gewählten Typ der Visualisierung. Der Typ ist einer der folgenden Werte: circle, horizontal, vertical
setFontSize(float val)	GRAPHOMATETREE_1.setFontSize(14);	Setzt die Schriftgröße.
setTextColor(String val)	GRAPHOMATETREE_1.setTextColor("#c0ffee");	Setzt die Schriftfarbe.
setNodeFillColor(String val)	GRAPHOMATETREE_1.setNodeFillColor("#c0ffee");	Setzt die Füllfarbe der Knoten des Baumes.
setNodeStrokeColor(String val)	GRAPHOMATETREE_1.setNodeStrokeColor("#c0ffee");	Setzt die Farbe, in der die Umrandung der Knoten des Baumes gezeichnet ist.
setNodeStrokeThickness(float val)	GRAPHOMATETREE_1.setNodeStrokeThickness(3);	Setzt die Dicke der Umrandung.
setLinkStrokeColor(String val)	GRAPHOMATETREE_1.setLinkStrokeColor("#0ff1ce");	Setzt die Farbe der Verknüpfungen zwischen den Knoten.
setLinkStrokeThickness(float val)	GRAPHOMATETREE_1.setLinkStrokeThickness(1.2);	Setzt die Dicke der Verknüpfungen.
setShowValue(boolean val)	GRAPHOMATETREE_1.setShowValue(false);	Setzt, ob neben der Beschriftung eines Knotens auch dessen Wert angezeigt wird.
setFontFamily(String val)	GRAPHOMATETREE_1.setFontFamily("Wingdings");	Setzt die Schriftart.
setLinkLayout(String val)	GRAPHOMATETREE_1.setLinkLayout("elbow");	Setzt das Layout der Verknüpfungen. Einer der folgenden Werte ist möglich: swung, straight, elbow, cross
setRootNodeDescription(String val)	GRAPHOMATETREE_1.setRootNodeDescription("Welt");	Setzt die Beschriftung des Wurzelknotens, falls manuell eine Beschriftung angelegt wurde.
setInfinityColor(String val)	GRAPHOMATETREE_1.setInfinityColor("#0ff1ce");	Setzt die Farbe, die für Knoten vergeben wird, deren Wert NaN oder Infinity (zum Beispiel bei Teilung durch 0) ist.
setNullColor(String val)	GRAPHOMATETREE_1.setNullColor("#0ff1ce");	Setzt die Farbe, die für Knoten vergeben wird, deren Wert Null (bspw. weil kein Wert in der Data Source vorhanden ist) ist.
setAscendingSortOrder(boolean val)	GRAPHOMATETREE_1.setAscendingSortOrder(true);	Setzt die Sortierung der Knoten.
setAreaScale(String val)	GRAPHOMATETREE_1.setAreaScale("linear");	Setzt, welche Methode für die Skalierung der Flächen verwendet wird. Mögliche Werte umfassen: linear, log
setMinNodeRadius(float val)	GRAPHOMATETREE_1.setMinNodeRadius("3");	Setzt den minimalen Radius eines Knoten.
setMaxNodeRadius(float val)	GRAPHOMATETREE_1.setMaxNodeRadius("30");	Setzt den maximalen Radius eines Knoten.
setDescriptionDelimiter(String val)	GRAPHOMATETREE_1.setDescriptionDelimiter("~");	Setzt, welches Trennzeichen für die Darstellung von Label und Wert an einem Knoten verwendet wird.