



Scripting Dokumentation der graphomate sparklines 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 sparklines für SAP Lumira Designer

Version 2.1 – Stand September 2018

1.1 Einführung

Die graphomate sparklines 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.

Hinweis zu Umlauten, Sonder- und Leerzeichen

Generell gilt: Umlaute, Sonder- und Leerzeichen werden aus technischen Gründen im Advanced Property Sheet prozentkodiert (siehe <http://de.wikipedia.org/wiki/URL-Encoding>).

Beispiel:

Müsli - Entwicklung der Verkäufe in % gegenüber dem Vorjahr

wird zu:

M%C3%BCsli%20Entwicklung%20der%20Verk%C3%A4ufe%20in%20%25%20gegen%C3%BCber%20dem%20Vorjahr

Dies macht es erforderlich, dass auch bei der Übertragung von String-Werten aus der Skriptsprache oder dem Standard Property Sheet eine Prozentkodierung vorgenommen wird. (Im Internet sind dazu zahlreiche Tools erhältlich, bspw. <http://www.url-encoder.de>.)

1.2 Ereignisse

Die nachfolgend aufgelisteten Ereignisse stehen zur Verfügung. Auf diesen können im Standard-Properties-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: getSelectedMember() getSelectedMemberKey() getSelectedMemberText()

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
getBarWidthFactor()	GRAPHOMATESPARKLINES_1.getBarWidthfactor();	float	0.5	Gibt den aktuellen Faktor zurück, mit dem die Balkenbreite berechnet wird.
getDisplayCategoryLabels()	GRAPHOMATESPARKLINES_1.getDisplayCategoryLabels();	boolean	true	Gibt zurück, ob die Category Label aktuell angezeigt werden. Im Falle von true werden sie dargestellt, andernfalls nicht.
getDisplayLeadingValues()	GRAPHOMATESPARKLINES_1.getDisplayLeadingValue();	boolean	true	Gibt zurück, ob die Leading Values (erster Datenpunkt der individuellen Sparklines) aktuell angezeigt werden oder nicht.
getDisplayOverallRange()	GRAPHOMATESPARKLINES_1.getDisplayOverallRange();	boolean	true	Gibt zurück, ob Overall Range und Relative Range aktuell angezeigt werden.
getDisplayTrailingValues()	GRAPHOMATESPARKLINES_1.getDisplayTrailingValues();	boolean	true	Gibt zurück, ob die Trailing Values (letzter Datenpunkt der individuellen Sparklines) aktuell angezeigt werden.
getDisplayType()	GRAPHOMATESPARKLINES_1.getDisplayType();	String	line	Gibt die aktuell gewählte Darstellungsart zurück; entweder "line" oder "bars".
getEnableRuler()	GRAPHOMATESPARKLINES_1.getEnableRuler();	boolean	true	Gibt zurück, ob die vertikale Hilfslinie aktuell aktiviert ist oder nicht.
getEnableZoom()	GRAPHOMATESPARKLINES_1.getEnableZoom();	boolean	true	Gibt zurück, ob die Zoomfunktion zur Runtime aktuell aktiviert ist.
getEnableZoomAnimation()	GRAPHOMATESPARKLINES_1.getEnableZoomAnimation();	boolean	true	Gibt zurück, ob die Animation des Zoomfensters aktiviert ist oder nicht.
getHighlightMinMax()	GRAPHOMATESPARKLINES_1.getHighlightMinMax();	boolean	true	Gibt zurück, ob die Minimum/Maximum-Hervorhebungen aktuell angezeigt werden.

getIsDataInRows()	GRAPHOMATESPARKLINES_1.getIsDataInRows();	boolean	true	Gibt die aktuell verwendete Ausrichtung der Dateninterpretation zurück. Wird true zurück gegeben, wird die Selektion zeilenweise interpretiert, bei false werden die Sparklines aus den Spalten der Selektion generiert.
getSelectedMember(String dimensionKey)	GRAPHOMATESPARKLINES_1.getSelectedMember();	String	0D_NWI_RCOD	Nachdem der User auf eine Sparkline geklickt hat (unabhängig davon, ob die Zoomfunktion aktiviert oder deaktiviert ist), gibt diese Funktion den Member der spezifizierten Dimension dimensionKey zurück. Der Member stellt verschiedene Properties wie text, internalKey und externalKey zur Verfügung.
getSelectedMemberKey(String dimensionKey)	GRAPHOMATESPARKLINES_1.getSelectedMemberKey("0D_NWI_RCOD");	String	10274	Nachdem der User auf eine Sparkline geklickt hat (unabhängig davon, ob die Zoomfunktion aktiviert oder deaktiviert ist), gibt diese Funktion den Schlüssel (Key) des Members der spezifizierten Dimension dimensionKey zurück.
getSelectedMemberText(String dimensionKey)	GRAPHOMATESPARKLINES_1.getSelectedMemberText("0D_NWI_RCOD");	String	North West	Nachdem der User auf eine Sparkline geklickt hat (unabhängig davon, ob die Zoomfunktion aktiviert oder deaktiviert ist), gibt diese Funktion den Text des Members der spezifizierten Dimension dimensionKey zurück. Die Rückgabe ist analog zu der Funktion getSelectedMemberKey().
getSelection()	GRAPHOMATESPARKLINES_1.getSelection();	String	{}	Gibt die aktuell verwendete Selektion zurück. Wird "{}" als Parameter zurück gegeben, bedeutet es, dass das gesamte Resultset zur Generierung der Sparklines verwendet wird.
getTitle()	GRAPHOMATESPARKLINES_1.getTitle();	String	Switzerland,%202015%0A	Gibt den aktuellen Titel zurück.
getZoomDuration()	GRAPHOMATESPARKLINES_1.getZoomDuration();	Float	2000	Gibt die aktuell eingestellte Dauer der Zoom-Animation in ms zurück.
getTitleFontSize()	GRAPHOMATESPARKLINES_1.getTitleFontSize();	Float	18	Gibt die aktuell eingestellte Schriftgröße des Titels zurück.
getSparklinePadding()	GRAPHOMATESPARKLINES_1.getSparklinePadding();	Float	4	Gibt das aktuell eingestellte Padding in px zurück.
getDisplayTrailingValues()	GRAPHOMATESPARKLINES_1.getDisplayTrailingValues();	boolean	true	Gibt an, ob der Endwert der Datenserie hinter der Sparkline angezeigt wird.
getSparklineLineThickness()	GRAPHOMATESPARKLINES_1.getSparklineLineThickness();	Float	2	Gibt die Dicke der Linie in px an.
getFontSize()	GRAPHOMATESPARKLINES_1.getFontSize();	Float	12	Gibt die aktuell eingestellte Schriftgröße zurück.
getFontFamily()	GRAPHOMATESPARKLINES_1.getFontFamily();	String	Consolas	Gibt die aktuell eingestellte Schriftart zurück.
getNumeralJsFormatString()	GRAPHOMATESPARKLINES_1.getNumeralJsFormatString();	String	0.0,0	Gibt den aktuell eingestellten numeral.js Format-String zurück.
getLocale()	GRAPHOMATESPARKLINES_1.getLocale();	String	en	Gibt die aktuell eingestellte Ländereinstellung zurück.
getLabelFormatMode()	GRAPHOMATESPARKLINES_1.getLabelFormatMode();	String	extended	Gibt das aktuell eingestellte Label Format wieder.
getExtendedNumberFormat()	GRAPHOMATESPARKLINES_1.getExtendedNumberFormat();	String	() € . . 10^1 2 ?	Gibt den aktuell eingestellten Extended-Number-Format-String zurück.
getExtendedNumberFormatPercent()	GRAPHOMATESPARKLINES_1.getExtendedNumberFormatPercent();	String	None 4 . . 10^2 4 !	Gibt den aktuell eingestellten Extended-Number-Format-Percent-String zurück.
getSuppressRepeatingCategoryLabels()	GRAPHOMATESPARKLINES_1.getSuppressRepeatingCategoryLabels();	String	[true,true,false,false,false]	Gibt den aktuell eingestellten Wert der zu unterdrückenen Labels zurück.
getZoomHeight()	GRAPHOMATESPARKLINES_1.getZoomHeight();	Float	200	Gibt den aktuellen Wert der Zoom-Größe zurück in px.
getZoomFontSize()	GRAPHOMATESPARKLINES_1.getZoomFontSize();	Float	8	Gibt den aktuellen Wert der Schriftgröße für das Pop-Up-Fenster wieder.
getZoomLineThickness()	GRAPHOMATESPARKLINES_1.getZoomLineThickness();	Float	2	Gibt die aktuelle Dicke der Linie im Pop-Up-Fenster an.
getEnableZoomTooltip()	GRAPHOMATESPARKLINES_1.getEnableZoomTooltip();	Boolean	true	Gibt an, ob der Zoom-Tooltip aktiviert ist.
getEnableZoomCategoryLabels()	GRAPHOMATESPARKLINES_1.getEnableZoomCategoryLabels();	Boolean	true	Gibt an, ob die Labels im Pop-Up aktiviert sind.
getTextColor()	GRAPHOMATESPARKLINES_1.getTextColor();	String	#ff99e6	Gibt die aktuelle Farbe der Textfarbe als Hexadezimalzahl zurück.
getGoodColor()	GRAPHOMATESPARKLINES_1.getGoodColor();	String	#ff99e6	Gibt die aktuelle Farbe der positiven Zahlen als Hexadezimalzahl zurück.
getBadColor()	GRAPHOMATESPARKLINES_1.getBadColor();	String	#ff99e6	Gibt die aktuelle Farbe der negativen Zahlen als Hexadezimalzahl zurück.
getSparklineColor()	GRAPHOMATESPARKLINES_1.getSparklineColor();	String	#ff99e6	Gibt die aktuelle Farbe der Linien als Hexadezimalzahl zurück.
getSparklineBackgroundColor()	GRAPHOMATESPARKLINES_1.getSparklineBackgroundColor();	String	#ff99e6	Gibt die aktuelle Farbe des Hintergrundes der Sparklines als Hexadezimalzahl zurück.
getSparklineRangeColor()	GRAPHOMATESPARKLINES_1.getSparklineRangeColor();	String	#ff99e6	Gibt die aktuelle Farbe Range als Hexadezimalzahl zurück.
getZoomLineColor()	GRAPHOMATESPARKLINES_1.getZoomLineColor();	String	#ff99e6	Gibt die aktuelle Farbe der Linien im Pop-Up als Hexadezimalzahl zurück.
getZoomHelplineColor()	GRAPHOMATESPARKLINES_1.getZoomHelplineColor();	String	#ff99e6	Gibt die aktuelle Farbe der Hilfslinie im Pop-Up als Hexadezimalzahl zurück.
getZoomHighlightColor()	GRAPHOMATESPARKLINES_1.getHighlightColor();	String	#ff99e6	Gibt die aktuelle Farbe beim Rüberfahren über die Linien im Pop-Up-Fenster als Hexadezimalzahl zurück.
getRulerLineColor()	GRAPHOMATESPARKLINES_1.getRulerLineColor();	String	#ff99e6	Gibt die aktuelle Farbe des Striches (Ruler) als Hexadezimalzahl zurück.
getRulerBoxColor()	GRAPHOMATESPARKLINES_1.getRulerBoxColor();	String	#ff99e6	Gibt die aktuelle Farbe des Hintergrundes der Zahlen beim Nutzen des Rulers als Hexadezimalzahl zurück.

Setter Name	Beispiel	Beschreibung
setBarWidthFactor(float value)	GRAPHOMATESPARKLINES_1.setBarWidthFactor(0.8);	Setzt die Berechnungsgrundlage für die Balkenbreite; der übergebene Wert muss zwischen 0.01 und 1 liegen.
setDisplayCategoryLabels(Boolean value)	GRAPHOMATESPARKLINES_1.setDisplayCategoryLabels(true);	Mit diesem Skriptbefehl lässt sich die Anzeige der Category Label an- oder ausstellen.
setDisplayLeadingValues(Boolean value)	GRAPHOMATESPARKLINES_1.setDisplayLeadingValues(true);	Schaltet die Anzeige der Leading Values (erster Datenpunkt der individuellen Sparklines) an oder aus.
setDisplayOverallRange(Boolean value)	GRAPHOMATESPARKLINES_1.setDisplayOverallRange(true);	Schaltet die Anzeige der Overall Range und Relative Range im Hintergrund der Sparklines an oder aus.
setDisplayTrailingValues(Boolean value)	GRAPHOMATESPARKLINES_1.setDisplayTrailingValues(true);	Schaltet die Anzeige der Trailing Values (letzter Datenpunkt der individuellen Sparklines) an oder aus.
setDisplayType(String type)	GRAPHOMATESPARKLINES_1.setDisplayType("line");	Setzt die Darstellungsart; der Wert des Parameters muss entweder "line" oder "bars" sein.

setEnableRuler(Boolean value)	GRAPHOMATESPARKLINES_1.setEnableRuler(true);	Schaltet die vertikale Hilfslinie zur Runtime an oder aus.
setEnableZoom(Boolean value)	GRAPHOMATESPARKLINES_1.setEnableZoom(true);	Schaltet die Zoomfunktion zur Runtime an oder aus.
setEnableZoomAnimation(Boolean value)	GRAPHOMATESPARKLINES_1.setEnableZoomAnimation(true);	Mit dieser Funktion kann die Animation des Zoomfensters zur Runtime de- bzw. aktiviert werden. Bei Übergabe von false erscheint das Zoom-Popup direkt und wird nicht animiert.
setHighlightMinMax(Boolean value)	GRAPHOMATESPARKLINES_1.setHighlightMinMax(true);	Schaltet die Anzeige der Minimum/Maximum-Hervorhebungen an oder aus.
setIsDataInRows(Boolean value)	GRAPHOMATESPARKLINES_1.setIsDataInRows(true);	Bestimmt die Ausrichtung der Dateninterpretation. Bei true wird die Selektion zeilenweise interpretiert, bei false werden die Sparklines aus den Spalten der Selektion generiert.
setSelection(String value)	GRAPHOMATESPARKLINES_1.setSelection("{}");	Setzt die Selektion auf den übergebenen String. Im Falle von "{}" als Parameter wird das gesamte Resultset zur Generierung der Sparklines verwendet.
setTitle(String value)	GRAPHOMATESPARKLINES_1.setTitle("Schweiz,%202015");	Setzt den Titel der Sparklines-Komponente
setZoomDuration(Float Value)	GRAPHOMATESPARKLINES_1.setZoomDuration(2000);	Mit dieser Funktion wird die Dauer der Zoom-Animation in Millisekunden festgelegt.
setTitleFontSize(Float Value)	GRAPHOMATESPARKLINES_1.setTitleFontSize(18);	Setzt die Schriftgröße des Titels.
setSparklinePadding(Float Value)	GRAPHOMATESPARKLINES_1.setSparklinePadding(4);	Setzt das Padding der Sparklines.
setDisplayTrailingValues(Boolean Value)	GRAPHOMATESPARKLINES_1.setDisplayTrailingValues(true);	Gibt an, ob der Endwert der Datenreihe hinter der Sparkline angezeigt werden soll.
setSparklineLineThickness(Float Value)	GRAPHOMATESPARKLINES_1.setSparklineLineThickness(3);	Setzt die Dicke der Linien.
setFontSize(Float Value)	GRAPHOMATESPARKLINES_1.setFontSize(9);	Setzt die Schriftgröße in den Sparklines.
setFontFamily(String Value)	GRAPHOMATESPARKLINES_1.setFontFamily("Consolas");	Setzt die Schriftart. Z.B.: Consolas, Arial, Verdana, Wingdings, ...
setNumeralJsFormatString(String Value)	GRAPHOMATESPARKLINES_1.setNumeralJsFormatString("0.0,0");	Setzt den numeral.js String für die Daten.
setLocale(String Value)	GRAPHOMATESPARKLINES_1.setLocale("en");	Setzt die Ländereinstellung. Mögliche Werte: "de", "en", "fr" oder "auto".
setLabelFormatMode(String Value)	GRAPHOMATESPARKLINES_1.setLabelFormatMode("extended");	Setzt das Label Format. Mögliche Werte: "basic" oder "extended".
setExtendedNumberFormat(String Value)	GRAPHOMATESPARKLINES_1.setExtendedNumberFormat("{} EUR , 10^1 2 ?");	Setzt das Extended-Number-Format. Z.B.: "{} € , 10^1 2 ?".
setExtendedNumberFormatPercent(String Value)	GRAPHOMATESPARKLINES_1.setExtendedNumberFormatPercent("None 4 , 10^2 4 !");	Setzt das Extended-Number-Format für prozentuale Werte. Z.B.: "None 4 , 10^2 4 !".
suppressRepeatingCategoryLabels(String Value)	GRAPHOMATESPARKLINES_1.setSuppressRepeatingCategoryLabels("[true,true,false,false,false]");	Setzt die zu ausblendenden doppelten Category-Labels. Z.B.: "[true,true,false,false,false]".
setZoomHeight(Float Value)	GRAPHOMATESPARKLINES_1.setZoomHeight(238);	Setzt die Höhe des Pop-Up-Fensters in px. Ist die Sparklines Komponente kleiner, wird auch das Pop-Up-Fenster entsprechend kleiner dargestellt.
setZoomFontSize(Float Value)	GRAPHOMATESPARKLINES_1.setZoomFontSize(7);	Setzt die Größe der Schriftart im Pop-Up-Fenster.
setZoomLineThickness(Float Value)	GRAPHOMATESPARKLINES_1.setZoomLineThickness(2);	Setzt die Dicke der Linie im Pop-Up-Fenster.
setEnableZoomTooltip(Boolean Value)	GRAPHOMATESPARKLINES_1.setEnableZoomTooltip(true);	Schaltet die Tooltips im Pop-Up-Fenster ein oder aus.
setEnableZoomCategoryLabels(Boolean Value)	GRAPHOMATESPARKLINES_1.setEnableZoomCategoryLabels(true);	Schaltet den Titel im Pop-Up-Fenster ein oder aus. Der Wert der Linie wird angezeigt, über den gefahren wird, ansonst der Wert der zum Öffnen des Fenster ausgewählt worden ist.
setTextColor(String Value)	GRAPHOMATESPARKLINES_1.setTextColor("#ff99e6");	Setzt die Schriftfarbe.
setGoodColor(String Value)	GRAPHOMATESPARKLINES_1.setGoodColor("#ff99e6");	Setzt die Farbe die auf positive Werte angewendet werden soll.
setBadColor(String Value)	GRAPHOMATESPARKLINES_1.setBadColor("#ff99e6");	Setzt die Farbe die auf negative Werte angewendet werden soll.
setSparklineColor(String Value)	GRAPHOMATESPARKLINES_1.setSparklineColor("#ff99e6");	Setzt die Farbe für die Linie.
setSparklineBackgroundColor(String Value)	GRAPHOMATESPARKLINES_1.setSparklineBackgroundColor("#ff99e6");	Setzt die Farbe für den Hintergrund der Sparkline.
setSparklineRangeColor(String Value)	GRAPHOMATESPARKLINES_1.setSparklineRangeColor("#ff99e6");	Setzt die Farbe für den Wert der Range.
setZoomLineColor(String Value)	GRAPHOMATESPARKLINES_1.setZoomLineColor("#ff99e6");	Setzt die Farbe Linie im Pop-Up-Fenster.
setZoomHelplineColor(String Value)	GRAPHOMATESPARKLINES_1.setZoomHelplineColor("#ff99e6");	Setzt die Farbe für die Hilfslinie im Pop-Up-Fenster.
setZoomHighlightColor(String Value)	GRAPHOMATESPARKLINES_1.setHighlightColor("#ff99e6");	Setzt die Farbe für die Linie über die mit den Zeiger gefahren wird im Pop-Up-Fenster.
setRulerLineColor(String Value)	GRAPHOMATESPARKLINES_1.setRulerLineColor("#ff99e6");	Setzt die Farbe für den Trennstrich der auftaucht, wenn man über die Punkte der Linie fährt.
setRulerBoxColor(String Value)	GRAPHOMATESPARKLINES_1.setRulerBoxColor("#ff99e6");	Setzt die Farbe für den Hintergrund der Werte die auftauchen, wenn man über die Punkte der Linie fährt.