

Custom Widget: Calendar

Contents

Introduction.....	1
Interfaces	1
Properties:	1
Events:	2
Remarks.....	2
Building a Calendar.....	2
Remarks:	4
Map of Revisions	5

Introduction

The Calendar custom widget allows the user to build and manipulate events on a calendar with the precision of minutes.

Interfaces

Properties:

Property Name	Action	Description
Format	Set	This property defines the format of the data handled by the custom widget. The following values are supported: 0 (default): CSV 1: JSON
ConfigurationFile	Set	When this property changes of value, it will load the configuration settings from the respective file name to build the Calendar.
ConfigurationText	Set	When this property changes of value, it will build the Calendar according to this property value.
InsertEvents	Set	Add one or more events into calendar following the format specified below
SelectedEvents	Get/Set	List of the selected events following the format specified below.

DeletedEvents	Get	List of the deleted events following the format specified below.
StartedEvents	Get	List of the started events following the format specified below. The value will be updated every time one or more events start.
FinishedEvents	Get	List of the finished events following the format specified below. The value will be updated every time one or more events finish
CurrentEvents	Get	List of the current events following the format specified below. It includes all of the events which were started but not finished yet.
ChangedEvent	Get	The recently changed event (when user change date/time of an event) following the format specified below.
DisableUser	Set	1 = disable editing and deleting actions 0 = enable editing and deleting actions
Font	Set	Set the event font and follows the format specified below.
DateFormat	Set	Changes the default date format.
TimeFormat	Set	Changes the default time format.
View	Get/Set	Specify the zoom level on the calendar. Possible values will be: "month", "week", "day", "list"

Events:

Event	Description
onViewChange	This event is triggered when user changes the view of the calendar
onEventChanged	This event is triggered when any event date/time is changed
onEventDeleted	This event is triggered when one or more event(s) is/are deleted
onEventStarted	This event is triggered when one or more event(s) is/are started
onEventFinished	This event is triggered when one or more event(s) is/are finished

Remarks

Building a Calendar

In order to build the Calendar, the user can configure the settings either through the ConfigurationFile property or through the ConfigurationText property. Even though both properties can be configured, it seldom be a valid configuration from a practical point of view, since the last one modified will overwrite the settings from the previous one.

The configuration will follow the following syntax, depending on the value of the Format property:

Format Value	Syntax	Example
0 (csv)	<label >,<start timestamp>,<end timestamp>,<data>,<color>	My first event, 11/23/2017 00:00, 11/23/2017 23:59, some description, red
1 (json)	{ "label": "<label>", "startTS": "<start timestamp>", "endTS": "<end timestamp>", "data": "<data>", "color": "<color>" }	{ "label": "My first event", "startTS": "11/23/2017 00:00", "endTS": "11/23/2017 23:59", "data": "some description", "color": "red" }

label The text associated with the event. This text will appear in the calendar indicating the event label.

startTS Event start date and time. The default format must be MM/DD/YYYY HH:MM

endTS Event end date and time. The default format must be MM/DD/YYYY HH:MM

data String data associated with the event.

color Event color. The color that will represent the event over the calendar.

The Font format:

0 (csv)	,,,	Arial, 12px, BIU, red
1 (json)	{ "type": "", "size": "<size in px>", "biu": "<bold italic underline>", "color": "<color>" }	{ "type": "Arial", "size": "12px", "biu": "biu", "color": "red" }

The graphical representation of the Calendar described in the previous example should look like the following picture:

	6/5/2016	6/6/2016	6/7/2016	6/8/2016
9 ^{AM}				
10 ^{AM}	Event 1		Event 3	
11 ^{AM}		Event 2		
12 ^{PM}				
1 ^{PM}				
2 ^{PM}			Event 4	
3 ^{PM}				
4 ^{PM}				
5 ^{PM}				

Remarks:

- The syntax is the same both for the content of the file pointed by the “ConfigurationFile” property as well as for the actual value of the “ConfigurationText” property.
- When using the “ConfigurationText” property in the CSV format, the separator between two nodes is a CR and LF (ASCII 13 and 10) characters.
- If the “ConfigurationFile” property extension is omitted, it should be considered “.csv” or “.json” by default, depending on the value of the “Format” property. Optionally, the user can customize the file extension in the value of the “ConfigurationFile” property.
- If the path is omitted for the “ConfigurationFile” property, the configuration file must be found in the custom widget root directory. Optionally, the “ConfigurationFile” property can support a relative path from the custom widget root directory.
- The only mandatory values in the syntax for one event <label>, <start ts> <end ts>.
- In json format, to insert an array of events, the json string should include brackets character “[” “]” to specify an array of json objects.
- The object must support UNICODE characters.

Map of Revisions

Revision	Date	Author	Description
A	November 16 rd , 2017	Vicente Teodoro	Initial Revision
B	December 14, 2017	Huy Le	Add ChangedEvent, Font format
C	March 09, 2017	Huy Le	Fix data format starTS, endTS, add data field, "list" view, change CW name to "Calendar"