

# NETRONIC HTML5 Visual Scheduling Widget - Standard Edition (VSW SE)

---

Valid for the VSW SE as of version 7.0.0  
2023.07.10-09:13

## Contents

1	Changelog.....	4
2	System Requirements .....	25
2.1	Supported Browsers and Versions .....	25
2.2	Needed 3 <sup>rd</sup> Party Libraries and Versions .....	25
2.3	Imported 3 <sup>rd</sup> Party Libraries and Versions.....	27
3	Object Model.....	27
3.1	Activities .....	29
3.1.1	Activity.....	29
3.1.2	ActivityEntry .....	48
3.2	Allocations .....	49
3.2.1	Allocation.....	50
3.2.2	AllocationEntry .....	61
3.3	Calendars.....	62
3.3.1	Calendar .....	62
3.3.2	CalendarEntry .....	62
3.4	Curves.....	63
3.4.1	Curve.....	64
3.4.2	CurvePointEntry .....	65
3.5	DateLine.....	66
3.6	Entity .....	67
3.7	HierarchySupplementaryDefinitions.....	70
3.7.1	HierarchySupplementaryDefinition.....	71
3.7.2	HierarchyLevelSupplementaryDefinition .....	72
3.7.3	GroupingLevelDefinition .....	73
3.8	Link .....	74
3.9	PeriodHighlighters.....	76

3.9.1	PeriodHighlighter .....	76
3.9.2	PeriodHighlighterEntry .....	77
3.10	Resource .....	77
3.11	Skill .....	81
3.12	Symbol .....	84
3.13	TableRowDefinitions .....	85
3.13.1	TableRowDefinition .....	85
3.13.2	TableCellDefinition .....	86
3.14	TooltipTemplate .....	89
4	Widget .....	92
4.1	Options .....	92
4.2	Callbacks .....	130
4.3	Methods .....	149
5	Enumerations .....	160
5.1	ActivityBarDragModes .....	160
5.2	ActivityBarShape .....	160
5.3	AllocationBarDragModes .....	160
5.4	AllocationBarShape .....	161
5.5	BarDesigns .....	161
5.6	BarSortMode .....	161
5.7	CollapseState .....	161
5.8	CurveInterpolationType .....	162
5.9	CurveType .....	162
5.10	DateLineCaptionOrientation .....	162
5.11	DateLineCaptionPosition .....	162
5.12	DateLineGridModes .....	164
5.13	HorizontalAlignment .....	164
5.14	HorizontallyScrollableViewArea .....	164
5.15	HorizontalScrollPosition .....	164
5.16	LinkMarker .....	164
5.17	LinkRoutingType .....	164
5.18	ObjectType .....	164
5.19	PageOrientation .....	165
5.20	PanningMode .....	165
5.21	PatternType .....	165
5.22	PrintingMode .....	165
5.23	ProgressBarWidthCalculationMode .....	165
5.24	RelationType .....	166
5.25	RowDesigns .....	166

5.26	RowDragModes .....	166
5.27	RowInsertionMode .....	167
5.28	RowSortMode .....	167
5.29	SelectionChangedReason .....	167
5.30	SnapTargets .....	167
5.31	TableType .....	167
5.32	TargetPositions .....	168
5.33	TextWrapMode .....	168
5.34	TimescaleInteractionModes .....	168
5.35	TimescaleNavigationMode .....	168
5.36	TimeType .....	169
5.37	TimeUnit .....	169
5.38	TreeVisualizationMode .....	169
5.39	UpdateModes .....	169
5.40	VerticallyScrollableViewArea .....	170
5.41	VerticalScrollPosition .....	170
5.42	ViewArea .....	170
5.43	ViewType .....	170
5.44	VisualSubtype .....	170
5.45	VisualType .....	170
5.46	WarningCode .....	171
5.47	WorldViewPosition .....	171
APPENDIX .....		172
Information Material on Specific Topics .....		172
Past Release Notes .....		172

# 1 Changelog

Version	Description of changes
<b>7.0.0</b> (See also <a href="#">release notes!</a> )	<p>MAJOR: New view type SkilledResourcesView:</p> <ul style="list-style-type: none"> <li>New object type Skill, new methods add/update/removeSkills and setResourcePropertiesForSkills, new enum value Skill for ObjectType, new property SkillID for Allocation objects, and new property SkillIDs for Resource objects.</li> <li>New options allocationRowsVisibleInSkilledResourcesView, definedAllocationLinksVisibleInSkilledResourcesView, entitiesTableVisibleInSkilledResourcesView, linksVisibleInSkilledResourcesView, tableRowDefinitionIDForTitleInSkilledResourcesView, tableViewWidthInSkilledResourcesView in analogy to the resources view.</li> <li>New options defaultValuesForSkillProperties, defaultSkillMinimumRowHeight, defaultSkillRowCollapsible, defaultSkillRowSelectable, defaultSkillRowTooltipTemplateID, defaultSkillTableRowDefinitionID.</li> <li>New callback argument SkillID on callbacks onClicked, onDoubleClicked, canDrag, onDragStart, onDrag, onDragEnd, onDrop, onCollapseStateChanged, onCurveCollapseStateChanged, and onShowContextMenu when referencing a resource row or an allocation row or bar.</li> <li>New possibility to select, highlight, or scroll to a skilled object (see methods selectObjects, highlightObjects, scrollToObject and callback onSelectionChanged).</li> <li>New properties SkilledRowTooltipTemplateID on resources and allocations, and SkilledBarTooltipTemplateID on allocations. New options defaultSkilledResourceRow/SkilledAllocationRow/SkilledAllocationBarTooltipTemplateID.</li> <li>New option allocationBarDesignOfOtherSkill for showing allocation bars differently when they belong to another skill than the resource shown below a skill row.</li> <li>New accessor &gt;Skill for formats used on resources in the new view type (see property BarTextFormat of Allocation objects, property TextFormat of TableCellDefinition objects, or property HTMLFormat of TooltipTemplates) and new accessor #Skill for TooltipTemplate objects on resource rows for referencing the current skill the cursor is hovering above.</li> </ul> <p>MAJOR (BREAKING CHANGE):</p> <ul style="list-style-type: none"> <li>Now by default there is no fallback of allocation property values to activity property values anymore. We decided to change the behavior since this improves the performance when updating activities. Also, many customers did not use these fallbacks at all. If the old behavior is needed for your application, you can set the option decouplingOfAllocationPropertiesFromActivities to false.</li> </ul> <p>MINOR: Streamlining and simplification of API:</p> <ul style="list-style-type: none"> <li>The prefix "PM_" has been removed from object property names and analogously "pm_" from option names. However, there is no need to change existing code immediately as the former notation will continue to be supported.</li> <li>New properties BarPatternType and BarPatternColor for Allocation objects analog to Activity objects.</li> <li>Property TextColor renamed to BarTextColor on Allocation and Activity objects.</li> <li>New property callbackArgs.code in callback onLogWarning and new enumeration WarningCode to help developers to understand why the warning has occurred. Also, new warnings are established.</li> </ul>

Version	Description of changes
	<ul style="list-style-type: none"> <li>The callback canDrag now can handle a promise. This replaces the now deprecated options forcedActivity/Allocation/Resource/EntityAllowedBar/RowDragModes.</li> <li>The word “nonworking” now consequently is documented and usable with this spelling. In the code it is allowed to use “nonWorking” everywhere.</li> <li>The options onCollapseStateChangedTriggeredByUpdateCalls and clickCallbackTriggeringOnRowInTimeArea were renamed to triggeringOfOnCollapseStateChangedByUpdateCallsEnabled and triggeringOfOnClickedInTimeAreaOfRow, resp., for better differentiation to callback options beginning with the prefix “on” and for unified naming.</li> </ul> <p>MINOR: Possibility to switch off complexity of bars for gaining performance:</p> <ul style="list-style-type: none"> <li>New options defaultActivityBarDesign, defaultAllocationBarDesign, tonedDownOverlayColor, reducedBarTopOffsetAndHeightScaleFactor.</li> <li>New property BarDesign on Activity and Allocation objects.</li> <li>New enumeration BarDesigns.</li> </ul> <p>MINOR: More flexibility for showing text:</p> <ul style="list-style-type: none"> <li>New option application Variables Map for flexible replacements in formatted text.</li> <li>New options intlDateTimeFormatOptionsMap and intlNumberFormatOptionsMap for flexible formatting of dates and numbers.</li> <li>New property TextFormat for TableCellDefinition objects.</li> <li>New property BarTextFormat for Activity and Allocation objects. New options defaultActivity/AllocationBarTextFormat.</li> <li>(Property InnerHTML of TooltipTemplate objects renamed to HTMLFormat.)</li> <li>New formatting options for property HTMLFormat of TooltipTemplate objects.</li> <li>New accessor [...] for property accessor strings to get array or map content with dynamic value inside the brackets (see property BarTextFormat of Activities/Allocations, property TextFormat of TableCellDefinitions, or property HTMLFormat of TooltipTemplates).</li> </ul> <p>MINOR: Extended graphical representation:</p> <ul style="list-style-type: none"> <li>New property TableColorVisibleInTimeArea for the GroupingLevelDefinition object.</li> <li>New option separationLinesInColoredIndentation.</li> <li>New properties SymbolHeight and SymbolWidth for TableCellDefinition objects.</li> <li>New option applicationStyleDefinition e.g. for defining CSS variables.</li> <li>The color of the three dots in the table symbol column was always black. To improve the readability, the dots now are automatically colored white if the background color of the symbol column cell is a darker one.</li> <li>New option defaultLoadCurvePaneHeight and Resource property LoadCurvePaneHeight.</li> </ul> <p>MINOR: Open-source libraries File-Saver and css-element-queries are not included in the code anymore because they were replaced by own code.</p> <p>TODO: This document: Now also the important method "option" is described with some hints, despite it is implemented by the base widget within jQueryUI.</p>
6.4.4	<p>PATCH: Several fixes for issues concerning collapsing or expanding child rows, allocation rows, or curve panes in an invisible view.</p> <p>PATCH: The widget did not resize anymore when the parent of its DIV element was changed.</p> <p>PATCH: Bars with zero-width were not visible anymore (since version 6.4.2).</p>

Version	Description of changes
<b>6.4.3</b>	<p>PATCH: Sometimes rows were vanishing when the appropriate data objects were removed and added again together with option asynchronousRendering set to true (since version 6.4.0).</p> <p>PATCH: Links eventually were not updated correctly in the chart when row objects were removed and added again within the vertical range of the appropriate link.</p> <p>PATCH: Fixed a hanging cursor on Firefox when clicking on a bar once or twice and then a context menu or a dialog got visible (since version 6.4.0).</p> <p>PATCH: The property PM_CollapseStateInLoadsView of Resource objects did not work anymore.</p>
<b>6.4.2</b>	<p>PATCH: The callback onDoubleClicked was not triggered on Firefox when the chart was bigger and vertically scrolled to the end.</p> <p>PATCH: The option preventDefaultOnContextMenuEvents did not work for the callback onShowContextMenu referencing the timescale.</p> <p>PATCH: Fixed display glitches when using the bar shape Regular on activities and allocations with entries that are shown within the coordinate range of the bevel.</p>
<b>6.4.1</b>	<p>PATCH: Setting options ending with ...AllowedRow/BarDragModes with value null falsely set the value to 0. This partly led to an inability to drag rows or bars, resp. (since version 6.4.0).</p> <p>PATCH: When bars overlapped in time only when considering the constraint or the predicted end date, then they were not placed in different sub rows.</p> <p>PATCH: Bars were not visible, when only a constraint or the predicted end date of a bar was inside the time range between options start and end.</p>
<b>6.4.0</b> (See also <a href="#">release notes!</a> )	<p>MINOR: New possibility to sort the table interactively by table columns, see new options interactiveSwitchingOfSortOrderEnabled, sortingIndicatorVisible, rowSortModeNoneEnabledOnInteractiveSwitchingOfSortOrder, and new callback onRowSortingChangeRequested.</p> <p>MINOR: New option nonWorkingTimesCalendarIDs.</p> <p>MINOR: New triggering of callback onShowContextMenu when the user clicks the secondary button of the mouse or presses the finger on the timescale.</p> <p>MINOR: New options asynchronousRendering and asynchronousInteractiveTimeAreaStretching for more performance when many objects are visible on the screen.</p> <p>MINOR: New method determineObjectByPageCoordinates.</p> <p>MINOR: Additional keywords &gt;SourceAllocation and &gt;TargetAllocation in tooltip templates for links.</p> <p>PATCH/MINOR: Because of an unwanted change of behavior with version 6.1.0, a new option triggeringOfOnShowTooltipForEntriesInBarsEnabled is implemented.</p> <p>PATCH: In Firefox the resize cursor for the vertical splitters hung under certain circumstances (since version 6.3.7).</p> <p>PATCH: Eventually a bar could not be resized interactively at the end date when it started before the date in widget option "start".</p> <p>PATCH: The keyword #Entry did not work in TooltipTemplate objects for activity bars or allocation bars.</p> <p>PATCH: The method cancelSaveAsPDF did not work correctly when used while the first page was</p>

Version	Description of changes
	<p>saved to the PDF document.</p> <p>PATCH: When dropping a row near the horizontal separation line, it could happen that the callback onDrop mentioned the neighbored row instead of the targeted one.</p> <p>PATCH: Fixed exception that occurred when hovering over a period highlighter entry in activities view with allocation rows visible.</p> <p>PATCH: The output parameter allowedDragModes of callback canDrag did not work fully.</p>
<b>6.3.7</b>	<p>PATCH: The callbacks onDragStart, onDrag, onDrop, onDragEnd did not contain the property named 'event' like in other interaction callbacks.</p> <p>PATCH: The later addition or update of row objects did not update the horizontal scroll bar of the appropriate table.</p> <p>PATCH: Fixed problem when user presses Escape key while dragging a vertical splitter, a column separator in the table title, or a dateline.</p> <p>PATCH: Fixed exception in Safari when a warning was generated internally (see callback onLogWarning).</p> <p>PATCH: On macOS the cursor for row drag&amp;drop was not the correct one (only an up arrow instead of an up-and-down arrow).</p>
<b>6.3.6</b>	<p>PATCH: Row dragging did not work when using the callback canDrag to enable DragVertically for individual row objects exclusively.</p> <p>PATCH: Interactive resizing of columns did not work correctly when using a visual zoom factor unequal to 1.</p> <p>PATCH: The callback onShowTooltip was not called for every table cell again when moving the mouse cursor.</p> <p>PATCH: In all callbacks where the table cell index was included in the arguments (onShowTooltip, onShowContextMenu, onClicked, onDoubleClicked), the horizontal table scroll offset and the visual zoom factor were not considered.</p> <p>PATCH: Dragging date lines within Firefox was not possible anymore and fixed missing cursor image when hovering a collapse/expand button within Firefox (since version 6.2.3).</p> <p>PATCH: Scrolling using a trackpad was not accurate.</p>
<b>6.3.5</b>	<p>PATCH: Method getSelectedObjects did not return selected links.</p> <p>PATCH: Row dragging did not work for entities when using option pm_defaultEntityAllowedRowDragModes set to DragVertically only.</p> <p>PATCH: When using one of the options for filtering visibility of row objects it could happen, that bars for invisible rows were drawn. Since version 6.3.3.</p>
<b>6.3.4</b>	<p>PATCH: The callback canDrag was triggered too often.</p> <p>PATCH: The callback onShowTooltip was not called always, when the mouse cursor left a bar.</p> <p>PATCH: When using a bar shape for an allocation different from the default one, this was not always visible on the first allocation entry after updating an allocation object.</p> <p>PATCH: Fixed rare exception when positioning links between invisible bars.</p>

Version	Description of changes
<b>6.3.3</b>	<p>PATCH: Performance improvements for several remove... method calls and for artificial links from activity links in resources view.</p> <p>PATCH: Fixed not working press gesture directly following a double-click.</p> <p>PATCH: Fixed missing cursor icon when hovering an application-defined release or due date symbol of an activity bar.</p> <p>PATCH: The library html2canvas was required at start-up of VSW but is only necessary when saving a PDF file and using the PDF options topHTML or bottomHTML.</p> <p>Additionally: This document now lists polyfill-library as included resource (see chapter 2.3).</p>
<b>6.3.2</b>	Internal release.
<b>6.3.1</b>	<p>PATCH: Performance improvements for several update... method calls.</p> <p>PATCH: The curve values displayed in the tooltips did not exactly match the definition in the curve entries due to JavaScript side effects when adding and subtracting values.</p> <p>PATCH: When dragging a row with allowed drag mode DragOnSameLevelOnly target rows are not expanded automatically when the level is below the one of the dragged row.</p> <p>PATCH: Period highlighters were not visible on allocation rows in activities view.</p> <p>PATCH: Rows were too high, when more than one zero-width bar with same start date existed there.</p> <p>PATCH: Fixed exception when calling one of the methods selectObjects and highlightObjects with an allocation object in activities view.</p> <p>Additionally: Added missing return values (Promise objects) to the description of methods fitTimeAreaIntoView, scrollToDate, scrollToObject in this document.</p>
<b>6.3.0</b> (See also <a href="#">release notes!</a> )	<p>MINOR: Activity, entity, and resource rows optionally can be dragged and dropped vertically inside the appropriate table now:</p> <ul style="list-style-type: none"> <li>• New options pm_defaultActivity/ResourceAllowedRowDragModes.</li> <li>• New options pm_forcedActivity/Entity/ResourceAllowedRowDragModes (suitable for cases where the callback handler of the application for canDrag cannot provide property changes on callback arguments on return).</li> <li>• New property PM_AllowedRowDragModes in Activity and Resource objects.</li> <li>• New enum values DragVertically and DragOnSameLevelOnly for RowDragModes.</li> <li>• New properties in callbackArgs when callbacks onDrag and onDrop are triggered. Furthermore, for this now exists the new enum RowInsertionMode.</li> <li>• Precalculation of sort code value when sort mode is set to be ascending (see new options activity/entity/resourceRowSortMode).</li> </ul> <p>MINOR: Sorting now also is possible by using a definable property in objects instead of implementing an appropriate compare callback:</p> <ul style="list-style-type: none"> <li>• New options activity/allocation/entity/resourceRowSortCodePropertyName and activity/allocation/entity/resourceRowSortMode.</li> <li>• New enum RowSortMode.</li> <li>• New property PM_SortCode for Activity/Allocation/Entity/Resource objects serving as default property for new options above.</li> </ul>



Version	Description of changes
	<p>MINOR: New property <code>PM_BorderDashArray</code> of Activity and Allocation objects.</p> <p>MINOR: New callback options <code>visibilityFilterForActivities/Allocations/Resources/Entities</code>. These ones replace the now deprecated option <code>visibilityFilter</code> for performance reasons.</p> <p>MINOR: New callback options <code>compareActivities/Allocations/Resources/Entities</code>. These ones replace the now deprecated option <code>compareObjects</code> for performance reasons.</p> <p>MINOR: New options <code>defaultValuesForObjectProperties</code> with <i>Object</i> standing for Activity, ActivityEntry, Allocation, AllocationEntry, Entity, Link, Resource.</p> <p>MINOR: New property <code>entitiesTableViewWidth</code> for callback <code>onTimeAreaViewParameters-Changed</code>.</p> <p>MINOR: New option <code>loggingVerboseLevel</code>.</p> <p>MINOR: New property <code>commandCounter</code> for callbacks <code>onLogWarning</code> and <code>onLogError</code>.</p> <p>MINOR: Symbols on date lines are now also placed optimized. See option <code>pm_dateLineCaptionOptimizedPositioningEnabled</code>.</p> <p>MINOR: Automatic recognition of duplicate IDs and cycles in hierarchy when using ParentIDs on adding or updating objects, see callback options <code>onLogWarning</code> and <code>onLogError</code>.</p> <p>PATCH: Fixed issues with cursor icon when hovering draggable splitters, column separators, and date lines.</p> <p>PATCH: Fixed internal exception after calling method <code>saveAsPDF</code>.</p>
6.2.8	<p>PATCH: Fixed missing tooltips on curves when using Firefox (since 7.5.3).</p> <p>PATCH: Sometimes it was not possible to drag the bar onto the original position when interactively dragging bars with options <code>timeStepUnit</code> and <code>timeStepUnitFactor</code> set to coarser values (since 7.5.2).</p> <p>PATCH: It was impossible to gain keyboard focus when using Firefox and clicking or tapping into widget element (since 7.5.3).</p> <p>PATCH: When starting a web application from file system, the PDF export failed in whole when a URL was used inside top or bottom HTML strings. Now only a warning is triggered.</p> <p>PATCH: Dragging of bars was not always possible when using a hatch pattern.</p> <p>PATCH: The callback <code>onShowContextMenu</code> was not triggered on period highlighter entries.</p> <p>PATCH: It was not possible to access neither row object properties using <code>#RowObject</code> nor period highlighter properties in tooltip templates assigned to period highlighter entries.</p> <p>PATCH: When showing allocation rows in activities view, selecting bars by drawing a rectangle optically selected activity and allocation bars.</p> <p>PATCH: When option <code>editable</code> was set to false, you could nevertheless drag entities into the time area. Additionally, changing the option <code>editable</code> at run-time had no effect.</p> <p>PATCH: When dragging entities into the time area with options <code>timeStepUnit</code> and <code>timeStepUnitFactor</code> set to coarser values, then the calendar was not continuously considered.</p> <p>PATCH: Dragging bars with option <code>timeStepUnit</code> set to "year" did not work at all.</p> <p>PATCH: Option <code>pm_barSortModeForOptimizedRowDesign</code> did not work.</p> <p>PATCH: When changing option <code>pm_bottomRowMarginInTimeArea</code>, row heights were not</p>

Version	Description of changes
	<p>updated concerning a currently visible curve pane.</p> <p>PATCH: When clicking/tapping onto a curve, sometimes an exception occurred, and sometimes selected elements were not deselected.</p> <p>PATCH: Neither the callback onSelectionChanged was triggered nor selected entities were deselected, when the user clicked/tap on the time area background or on curves.</p> <p>Attention: When using option pm_dateLineCaptionOptimizedPositioningEnabled, the optimization of caption positions does not work when symbols are used on the date lines! This will be fixed with upcoming version 7.6.0.</p>
<b>6.2.7</b>	<p>PATCH: More performance when using add methods for Activity, Entity, Resource objects.</p> <p>PATCH: Fixed additional horizontal scrollbar for top view area in Firefox (since version 6.2.3).</p>
<b>6.2.6</b>	<p>PATCH: Fixed functionality of property PM_BorderColor of Activity objects when entries are existing.</p> <p>PATCH: Fixed missing values when evaluating a TooltipTemplate for a tooltip (since version 6.2.3).</p> <p>In this document now the number of possible hierarchy levels are explicitly described as being limited to a maximum of approx. 100 (see property ParentID of Activity, Entity, and Resource objects) because of possible performance issues.</p>
<b>6.2.5</b>	<p>PATCH: Fixed false empty triggering of callback onShowTooltip when moving the mouse pointer from one bar to another one (since version 6.2.1).</p> <p>PATCH: Fixed internal exception when adding resources, entities, or activities with IDs that were removed before under some circumstances (since version 6.2.1).</p> <p>PATCH: Fixed not visible links in resources view when using activity links and adding allocation objects again after removing all existing allocation objects before (since version 6.2.1).</p> <p>PATCH: Fixed a false triggering of a UI event "contextmenu" when using Firefox additional to triggering the callback onShowContextMenu (since version 6.2.3).</p> <p>PATCH: Fixed issue of rows remaining invisible after updating the property ParentID of Activity, Entity, or Resource objects (since version 6.2.1).</p> <p>PATCH: Fixed false interpretation of property permissionToPrint in options object of method saveAsPDF.</p>
<b>6.2.4</b>	<p>PATCH: When dragging bars vertically, the drag date lines showed the dates temporarily (since version 6.2.3).</p> <p>PATCH: Fixed exception when dragging entity object into the time area at the lower end of the chart.</p> <p>PATCH: Fixed an update issue when using the callback visibilityFilter to hide activity bars on collapsed activity rows and showing the bars in ancestor rows.</p> <p>PATCH: Fixed a performance issue when using the callback visibilityFilter with a big number of allocations.</p> <p>PATCH: The about dialog now shows the open-source library "core-js" as an included component (see also chapter 2.3).</p>

Version	Description of changes
<b>6.2.3</b>	<p>PATCH: The widget now supports showing scroll bars when using Firefox beginning with version 100 on Windows 11.</p> <p>PATCH: When using Firefox the inner DIV elements of the widget were focusable by using the Tab key. Now this is prevented.</p> <p>PATCH: Fixed an exception when the user clicked into the background of a row within the time area.</p> <p>PATCH: The callback onShowTooltip was triggered too often after leaving the curve area of a row.</p>
<b>6.2.2</b>	<p>PATCH: The property cellIndex was not working as expected in callback onShowTooltip. Additionally, it was not documented by accident.</p> <p>PATCH: When showing the context menu on a currently selected object all selected objects were deselected (since version 6.2.1).</p> <p>PATCH: When using a newer version of D3, the dragged bar was not surrounded by a flashing rectangle anymore.</p> <p>PATCH: The shown symbol for the ReleaseDate property of Activity objects was overdrawn by the activity bar when the dates in properties Start and End were in same range.</p> <p>PATCH: Interactively resizing bars with options timeStepUnit and timeStepUnitFactor set to coarser values was not showing a smooth phantom.</p> <p>Attention: The widget does not show scroll bars when using Firefox beginning with version 100 on Windows 11. This will be fixed in a later patch release. As a workaround you can change the setting “Always show scrollbars” in System Preferences &gt; Accessibility &gt; Visual Effects.</p>
<b>6.2.1</b>	<p>PATCH: Much more performance when changing the property ParentID of Activity, Entity, or Resource objects.</p> <p>PATCH: Activity links in resources view under circumstances remained invisible when added in startup phase of widget.</p> <p>PATCH: When using activity links in resources view, these were not selectable interactively.</p> <p>PATCH: When using multiple bar dragging with one of the options or properties concerning ...AllowedBarDragModes set to DragHor+DragVer, horizontal dragging of bars in more than one resource row at the same time did not work as expected.</p> <p>PATCH: In some cases, the mouse cursor was not cleared when leaving a bar on screen.</p> <p>PATCH: The context menu of the browser is suppressed now on the timescale and in the fixed symbol column of the table when using the secondary mouse button.</p> <p>Additionally: The predefined text for topHTML in the PDF options dialog was corrected.</p> <p>Additionally: This reference guide now shows small class diagrams for each data object type.</p>
<b>6.2.0</b> (See also <a href="#">release notes!</a> )	<p>MINOR: New options topText, bottomText, topHTML, bottomHTML, ownerPassword, userPassword, and permissionTo..., author, title, subject, keywords for method saveAsPDF. (For using topHTML and bottomHTML the additional open-source library “html2canvas” is needed.)</p> <p>MINOR: New option pm_dateLineCaptionOptimizedPositioningEnabled.</p> <p>MINOR: New property PredefinedGroups for HierachyLevelSupplementaryDefinition objects.</p>

Version	Description of changes
	<p>MINOR: Renaming of the following properties in GroupingLevelDefinition objects for alignment with a fallback to the older property names:</p> <ul style="list-style-type: none"> <li>• CodeToTextMap to GroupingCodeToTextMap,</li> <li>• CodeSource to GroupingCodeSource,</li> <li>• TableBackgroundColor to TableColor.</li> </ul> <p>MINOR: New properties ScaleMinimumValue and ScaleMaximumValue for Curve objects.</p> <p>MINOR: New option pm_linesShownInLoadCurvePanels.</p> <p>MINOR: New property SuitableResourceIDs for Entity objects and new property SuitableActivityIDs for Allocation and Entity objects.</p> <p>MINOR: New callbacks onLogError and onLogWarning.</p> <p>PATCH: Fixed positioning when dragging a date line or the vertical splitter and the option visualZoomFactor was set to value unequal to 1.</p> <p>PATCH: When saving a PDF document, symbols shown at the top of date lines were not exported.</p> <p>PATCH: More performance when updating activity objects.</p> <p>Additionally: The Sample App now contains an example for a PDF options dialog.</p> <p>Additionally: The Sample App now references current versions of 3rd party libraries.</p>
<b>6.1.11</b>	<p>PATCH: The world view sometimes was scaled to high, so that the view rectangles were not visible fully.</p> <p>PATCH: When an active HierarchySupplementaryDefinition object was updated the changes did not get visible in the entities.</p> <p>PATCH: The mouse cursor changed too often on allocation bars with allocation entries since version 6.1.10.</p>
<b>6.1.10</b>	<p>PATCH: When dragging allocation bars horizontally erroneously the dragMode property in callback onDrop had flag DragVertically switched on.</p> <p>PATCH: A defined symbol on a date line was invisible when either width or height left undefined.</p> <p>PATCH: All resources were grayed on dragging multiple allocation bars when the property SuitableResourceIDs was empty on at least one of the Allocation objects. Now an empty value is interpreted consistently on single and multiple bar dragging.</p> <p>PATCH: The mouse cursor did not change to “resize” in certain cases on allocation bars.</p> <p>PATCH: When dragging an allocation bar vertically the non-working time was not updated, when it should be visible inside the bar.</p> <p>PATCH: Setting the option pm_symbolColumnNameTitleSymbolIDs with the same value as before took too much time.</p>
<b>6.1.9</b>	<p>PATCH: After calling scrollToObject with an allocation the bar representing this allocation was not visible fully when it was out of sight before and is positioned below the first sub row.</p> <p>PATCH: When the option loggingEnabled was set to true in the initiation options of the widget, the red recording button was not shown on screen.</p> <p>PATCH: When using saveAsPDF without setting zoomFactorInPercent, horPageCountLimit, and</p>

Version	Description of changes
	verPageCountLimit, the resulting PDF document eventually contained more than page.
<b>6.1.8</b>	<p>PATCH: After calling method selectObjects the time area did not work normally, e.g. rescaling by using the timescale did not update the time area anymore.</p> <p>PATCH: When a row object shows a selection frame the sensible area around the collapse/expand button was smaller than without the selection frame.</p> <p>PATCH: In a special case the call to fitTimeAreaIntoView started an animation for horizontal scrolling and hindered a following call to scrollToObject.</p>
<b>6.1.7</b>	<p>PATCH: Fixed false scroll position when method scrollToObject was called with option pm_scrollToObjectAnimationEnabled set to true.</p> <p>PATCH: Now horizontal scrolling to begin of the time area is avoided when scrollToObject is called for an activity or an allocation object that has no defined start date. In this case now only vertical scrolling is done.</p> <p>PATCH: Setting a non-existent ID into the property ParentID of an activity, resource, or entity object using an update method was not working anymore.</p> <p>PATCH: Internal exceptions now are visible in browser again.</p> <p>PATCH: After calling the method scrollToObject with an entity object, highlighting was not working anymore.</p> <p>PATCH: In TooltipTemplate objects used for curve tooltips the reserved words #Load, #Capacity, #Date, #SingleLoads did not show the appropriate values.</p>
<b>6.1.6</b>	PATCH: Fixed missing triggering of callbacks onShowTooltip, onClicked, and onDoubleClicked when visualType is PeriodHighlighter and mouse cursor is on allocation rows.
<b>6.1.5</b>	PATCH: Fixed missing attributes periodHighlighter and entryIndex on callbacks onShowTooltip, onClicked, and onDoubleClicked when visualType is PeriodHighlighter.
<b>6.1.4</b>	<p>PATCH: When using Allocation.SuitableResourceIDs on a bigger data model, it took too long to start dragging on bars.</p> <p>PATCH: When adding allocation rows right on expanding the containing row these remained invisible.</p>
<b>6.1.3</b>	PATCH: Fixed exception after updating activity objects while allocation rows are shown in activities view.
<b>6.1.2</b>	<p>PATCH: Fixed hanging mouse cursor when leaving links.</p> <p>PATCH: Fixed issue with actual time resolution when options maximumTimeResolutionUnit/-Factor were modified again after widget instantiation.</p> <p>PATCH: Fixed exception when dragging an activity bar where the property PM_HasAllocationRows of the activity is set to true without being assigned to existing allocations.</p>
<b>6.1.1</b>	<p>PATCH: The property cellIndex was missing in callback onShowTooltip at least for table rows representing entities.</p> <p>PATCH: Fixed missing update of allocation rows in activities view after updating</p>

Version	Description of changes
	<p>PeriodHighlighter objects.</p> <p>PATCH: Time axis could not be shrunk enough anymore when using a big time resolution unit step.</p> <p>PATCH: Fixed internal exception handling.</p> <p>PATCH: Fixed crash, when start and end were set to null.</p>
<b>6.1.0</b> (See also <a href="#">release notes!</a> )	<p>MINOR: New property Entries for Activity objects and new object type ActivityEntry.</p> <p>MINOR: New options defaultUpdateMode and resetValueForDifferentialUpdate. New value for enumeration UpdateModes.</p> <p>MINOR: New option pm_timescaleInteractionMode.</p> <p>MINOR: New property DefaultCode for GroupingLevelDefinition objects.</p> <p>MINOR: New option pm_resourceHierarchySupplementaryDefinitionIDInLoadsView.</p> <p>MINOR: New options maximumTimeResolutionUnit and maximumTimeResolutionUnitFactor. New values for options maximumTimeResolutionUnit, timeStepUnit, and for unit parameter of setTimeResolutionInView method. New enumeration TimeUnit as an alternative for string values.</p> <p>MINOR: In TooltipTemplates single curve values are now accessible by using #SingleLoads.curveID.</p> <p>MINOR: New option onCollapseStateChangedTriggeredByUpdateCalls.</p> <p>PATCH: Switching the grouping on or modifying the grouping now is much faster.</p> <p>PATCH: Fixed an issue with missing animation on grouping modifications.</p> <p>PATCH: Fixed an issue when modifying property TableRowDefinitionID in HierarchyLevelSupplementaryDefinition objects.</p> <p>PATCH: Fixed the visibility of allocation object values in tooltips generated by tooltip templates on allocation bars in resources view.</p> <p>PATCH: When an allocation referenced a non-existing resource, it could not be filtered in activities view.</p> <p>PATCH: Fixed world view issues concerning scaling and scroll bar visibility.</p> <p>PATCH: About box now only contains the libraries incorporated inside of VSW library files prefixed with "nwaf-". See revised chapter "System Requirements" for more details.</p>
<b>6.0.5</b>	<p>PATCH: When using the callback visibilityFilter for filtering allocation rows, then the containing row showed a collapse or expand button even when all allocation rows were invisible. Now a special symbol appears instead.</p> <p>PATCH: Fixed an issue in PDF export when the time range is huge.</p> <p>PATCH: Fixed an issue when modifying the property PM_ViewArea on row objects that have visible allocation rows.</p> <p>PATCH: Fixed an issue when the visibilityFilter was modified and allocation rows therefore became visible again.</p>
<b>6.0.4</b>	<p>PATCH: When changing the "start" or "end" option, the view start date is now preserved, if possible.</p>



Version	Description of changes
	PATCH: Fixed exception and malfunction concerning animation when changing objects in fast sequence.
<b>6.0.3</b>	<p>PATCH: Period highlighter grids assigned to resources are now shown also in allocation rows of these resources, in analogy to calendar grids.</p> <p>PATCH: Fixed an issue with PDF export newly appeared with version 6.0.2.</p>
<b>6.0.2</b>	<p>PATCH: Parameters “entry” and “entryIndex” in callbacks onShowTooltip and onShowContextMenu working again.</p> <p>PATCH: The colorization for the grouping and hierarchy levels in the table was not working fully caused by vertical virtualization that is internally used to get more performance.</p> <p>PATCH: Sorting of rows now is triggered correctly when setting the callback option compareObjects using the same Function object.</p> <p>PATCH: Callback onVerticalScrollOffsetChanged now also works correctly when grouping is used.</p>
<b>6.0.1</b>	<p>PATCH: Missing parameter “date” in onClicked callback when time area background was clicked.</p> <p>PATCH: After changing the group criteria in the data of a row object with active grouping, the grouping was not updated.</p> <p>PATCH: When changing the data of an activity object while allocation rows are visible in the activities view, then the allocation rows for this activity object disappeared.</p> <p>PATCH: The new tree view feature did not work correctly when the top view area is visible and the property PM_ViewArea was changed.</p> <p>Additionally: The Sample App now demonstrates the new feature “grouping by criteria” (see also context menu of resource table rows).</p>
<b>6.0.0</b> (See also <a href="#">release notes</a> !)	<p>MINOR: New additional grouping by criteria within the current parent-child hierarchy:</p> <ul style="list-style-type: none"> <li>• New objects HierarchySupplementaryDefinition, HierarchyLevelSupplementaryDefinition, GroupingLevelDefinition to specify grouping.</li> <li>• New options pm_activity/resource/entityHierarchySupplementaryDefinitionID to specify the active hierarchy supplementary definition object for additional grouping of activity/resource/entity objects appearing as rows in the appropriate table.</li> <li>• New parameters for callbacks onClicked, onDoubleClicked, onShowContextMenu, onShowTooltip, compareObjects when grouping rows are affected.</li> <li>• New callback determineGroupingCode to specify grouping information.</li> </ul> <p>MINOR: New method highlightObjects and therefore renamed options pm_scrollToObjectHighlightingColor to pm_objectHighlightingColor and pm_scrollToObjectHighlightFlashingEnabled to pm_objectHighlightFlashingEnabled.</p> <p>MINOR: New bar shape named Symbol for allocation and activity bars:</p> <ul style="list-style-type: none"> <li>• New enum value named Symbol in enumerations AllocationBarShape and ActivityBarShape used in property PM_BarShape of allocation and activity objects.</li> <li>• New properties PM_BarShapeSymbolID and PM_BarShapeSymbolWidth for allocation and activity objects.</li> </ul> <p>MINOR: The bar shape named Diamond now is usable additionally for allocation bars.</p> <p>MINOR: To improve a more compact layout optionally there are new options</p>

Version	Description of changes
	<p>pm_allocation/activityBarTopOffsetAndHeightScaleFactor, pm_entitiesTableCellContentTopOffset, pm_tableCellContentTopOffset, pm_tableTitleAndTimescaleHeight, pm_entitiesTableTitleHeight, pm_progressBarHeight.</p> <p>MINOR: New options treeVisualizationMode, pm_treeViewLineColor/DashArray, entitiesTableTreeVisualizationMode, pm_entitiesTableTreeViewLineColor/DashArray and new enumeration TreeVisualizationMode.</p> <p>MINOR: New property PM_StrokeDashArray for Curve objects.</p> <p>MINOR: New option pm_barSortModeForOptimizedRowDesign, new enumeration BarSortMode.</p> <p>MINOR: New option pm_clickCallbackTriggeringOnRowInTimeArea.</p> <p>MINOR: New property promise for callbackArgs object in callback onShowTooltip.</p> <p>MINOR: New options pm_symbolColumnTitleBackgroundColor and pm_entitiesTableSymbolColumnTitleBackgroundColor.</p> <p>MINOR: New property PM_RowSymbolColumnBackgroundColor for Activity, Allocation, Entity, and Resource objects and new property SymbolColumnBackgroundColor for TableRowDefinition objects.</p> <p>MINOR: Options pm_top/mainViewAreaVisible for resources view extended to pm_top/mainViewAreaVisibleInActivities/Loads/ResourcesView to cover activities view and loads view additionally.</p>
<b>5.3.7</b>	Internal release.
<b>5.3.6</b>	PATCH: More performance for callback compareObjects by reducing the number of calls to the minimum. Additionally the callback arguments now contains the property viewType.
<b>5.3.5</b>	<p>PATCH: PDF export fixed (issue since 5.3.4).</p> <p>PATCH: The callback compareObjects was not called for allocation rows.</p> <p>PATCH: The callback onClicked was not called on curves anymore.</p> <p>PATCH: In case where allocation rows are visible in resources view, collapsing a resource row did not lead allocation links disappear where needed.</p>
<b>5.3.4</b>	<p>PATCH: Fixed graphical issue when canceling dragging of a date line.</p> <p>PATCH: Bars without start and end dates are not shown anymore.</p> <p>PATCH: Snapping while dragging a bar now also works for date line grids in mode Automatic.</p> <p>PATCH: World view now cannot get higher/wider than widget extent anymore.</p> <p>PATCH: Now update calls should be possible when in callback handler function for onSelectionChanged.</p> <p>PATCH: Fixed issue when dragging more than one allocation bar and the property SuitableResourceIDs is used at least on some of the allocations.</p> <p>PATCH: Setting one of the properties PM_CollapseState and PM_CollapseStateInLoadsView for Resource objects is now also working when resources view or loads view, resp., is not visible.</p>
<b>5.3.3</b>	<p>PATCH: Fixes snapping to start and end dates of other allocations when dragging an allocation.</p> <p>PATCH: Fixes an issue for the options pm_ignoreCalendarOnActivity/AllocationBarInteractions</p>



Version	Description of changes
	<p>when updating objects while dragging.</p> <p>PATCH: Fixes an eventual exception when option pm_activityBaselinesVisible was set.</p>
<b>5.3.2</b>	<p>PATCH: Fixes issue for not showing about dialog anymore when pressing Shift+Ctrl+Alt+F12 since 5.3.1.</p>
<b>5.3.1</b>	<p>PATCH: Property visualSubtype added to the argument of onShowTooltip.</p> <p>PATCH: The option pm_activityBaselineBarsVisible now works correctly when it is modified with resources view open and then switched to activities view.</p> <p>PATCH: Fixed issue when dragging a release date symbol or due date symbol in Firefox, when the symbol is user-defined.</p> <p>PATCH: Fixed issue when using method scrollToObject with an allocation object.</p>
<b>5.3.0</b> (See also <a href="#">release notes!</a> )	<p>MINOR: New property PM_CollapseStateInLoadsView for Resource objects.</p> <p>MINOR: New options pm_symbolColumnBackgroundColor and pm_entitiesTableSymbolColumnBackgroundColor.</p> <p>MINOR: New option pm_ignoreCalendarOnActivityBarInteractions.</p> <p>MINOR: New callback onSaveAsPDFProgress and new method cancelSaveAsPDF.</p> <p>MINOR: New enumeration PatternType, new properties PM_BarPatternType and PM_BarPatternColor on Activity objects, and new properties PM_PatternType and PM_PatternColor on AllocationEntry objects.</p> <p>MINOR: New options pm_dateLineGridColor, pm_dateLineGridDashArray, and pm_dateLineGridWidth.</p> <p>MINOR: New option pm_activityBaselineBarsVisible.</p> <p>MINOR: New link property PM_TargetMarker and new enumeration LinkMarker.</p> <p>PATCH: Fixed exception when modifying the property ParentID of an Activity, Entity, or Resource object with an ID of a non-existing object.</p> <p>PATCH: Fixed issues with colored background rectangles in the table when saving a PDF document.</p> <p>PATCH: Fixed issue when canceling dragging of a date line by pressing Escape key.</p> <p>PATCH: Fixed positioning issues when using bar diamond shapes.</p>
<b>5.2.12</b>	<p>PATCH: Fixed performance issue for removeAll(ObjectType.Resource) again and additionally removeAll(ObjectType.Allocation).</p>
<b>5.2.11</b>	<p>PATCH: Fixed performance issue for removeAll(ObjectType.Resource).</p>
<b>5.2.10</b>	<p>PATCH: Issue fixed for invisible curve pane when resource has set property PM_CurveCollapseState to 0 and was added before its parent resource.</p> <p>PATCH: The scrollToDate method was missing the offset parameter.</p> <p>PATCH: Fixed issues when using curves of type List.</p> <p>PATCH: Fixed issues concerning symbols in table title.</p>

Version	Description of changes
	<p>PATCH: Watermarks were not scaled on screen anymore since version 5.2.9.</p> <p>PATCH: When using the secondary mouse button while dragging the action now is canceled.</p> <p>PATCH: Fixed issue when sizing a table column interactively that has a background color.</p> <p>PATCH: Sometimes the saved PDF file showed collapsed allocation rows and vice versa.</p> <p>PATCH: Fixed exception when user clicked into timescale with visible world view.</p> <p>PATCH: Fixed issue of resolving object references in an applied tooltip template.</p>
<b>5.2.9</b>	<p>PATCH: Exception fixed concerning adding allocations after first refresh in resources view.</p> <p>PATCH: Issue fixed in method saveAsPDF concerning referenced SVG images not visible in PDF.</p> <p>PATCH: Infinite loop fixed in method saveAsPDF when many images are not loadable.</p>
<b>5.2.8</b>	<p>PATCH: Performance issue fixed concerning allocation rows in resources view.</p>
<b>5.2.7</b>	<p>PATCH: Issue fixed for callback onShowContextMenu.</p> <p>PATCH: Issue fixed for callback visibilityFilter.</p>
<b>5.2.6</b>	<p>PATCH: Issue fixed concerning the callback visibilityFilter used with allocation. Additionally setting the filter did not re-render the widget content.</p> <p>PATCH: The callback arguments for the callbacks onClicked, onDoubleClicked, onShowContextMenu did not contain the property cellIndex when called for a table row.</p>
<b>5.2.5</b>	<p>PATCH: Fixed issue of not recognized setting option pm_allocationRowsVisibleInActivities/ResourcesView before first call to render method. Workaround was to change the view type twice.</p> <p>PATCH: Fixed issue with option visibilityFilter not being called immediately when set (issue was existent since 4.0.0).</p>
<b>5.2.4</b>	<p>PATCH: Supplemented missing property tableViewWidth in onTimeAreaViewParametersChanged callback.</p> <p>PATCH: In some situations, the allocation bar was not vanishing on the source row in resources view after dragging vertically to another row.</p> <p>PATCH: Delivered JavaScript files now are prefixed with a UTF8-BOM since in one case concerning Firefox they were misinterpreted as being encoded in ANSI.</p> <p>PATCH: Fixed hanging issue when updating resource objects or period highlighters within drag &amp; drop interaction of allocation bars.</p>
<b>5.2.3</b>	<p>PATCH: Property PM_HasAllocationRows for Resource objects was missing in code and documentation.</p> <p>PATCH: Option pm_defaultResourceAllocationRowsCollapsible was missing in code and documentation.</p> <p>PATCH: Bars within a row object disappeared when the grandparent row object was collapsed, and the bars should remain visible (see PM_CollapsedRowDesign) and the parent row object was not collapsed.</p> <p>PATCH: Property PM_AllocationRowsCollapseState of Resource objects was not working.</p>

Version	Description of changes
	This document lacked documentation for the properties PM_AllocationRowsCollapsible, PM_AllocationRowsCollapseState for Resource objects introduced with version 5.1.0.
<b>5.2.2</b>	PATCH: Crash fixed when calling saveAsPDF without any links in the chart.
<b>5.2.1</b>	<p>PATCH: New default for property TextSource in TableCellDefinition objects is "", when property SymbolIDSource is set, else the default is "TableText" as before. This is for convenience.</p> <p>PATCH: Animation on expanding/collapsing rows in loads view is enabled again.</p> <p>PATCH: The vertical splitter can now be dragged to the left until the table has a width of 0 even if fixed columns exist.</p> <p>PATCH: When dragging a bar to the border of time than the user cannot drag it out of sight anymore.</p> <p>PATCH: Fixed false property value false of property newRowObjectIsSuitableResource in onDrag callbacks.</p> <p>PATCH: The method selectObjects did not work for allocation bars anymore.</p> <p>PATCH: Symbols and status fields on bars now are stabilized in z-order also concerning the texts in the bars.</p> <p>PATCH: Enumeration ObjectType now is correctly documented.</p>
<b>5.2.0</b> (See also <a href="#">release notes</a> !)	<p>MINOR: Now additional dates on allocation and activities can be defined either as link source or link target:</p> <ul style="list-style-type: none"> <li>• New properties LinkSourceDate/LinkTargetDate on activity and allocation objects.</li> <li>• New values for property RelationType on link objects: SourceDateStart, SourceDateEnd, EndTargetDate, StartTargetDate, SourceDateTargetDate.</li> </ul> <p>MINOR: Now it is possible to click and double click on symbols in the left fixed symbol column in a table. Therefore a new property symbolIndex was added to the callback arguments of the callbacks onClicked and onDoubleClicked.</p> <p>MINOR: New property PM_TooltipTemplateID on PeriodHighlighterEntry objects.</p> <p>MINOR: New option firstDayOfWeek.</p> <p>MINOR: New options to specify default tooltip templates: pm_defaultActivityBar/RowTooltipTemplateID, pm_defaultAllocationBar/RowTooltipTemplateID, pm_defaultEntityRowTooltipTemplateID, pm_defaultLinkTooltipTemplateID, pm_defaultPeriodHighlighterEntryTooltipTemplateID, pm_defaultResourceRow/CurveTooltipTemplateID.</p> <p>MINOR: New enumeration RelationType for links.</p> <p>MINOR (is a MAJOR change when updating from 5.1.0): New properties start/endPropertyName in callbackArgs of callbacks canDrag, onDragStart, onDrag, onDragEnd, onDrop.</p> <p>PATCH: Fixed performance issue that was existent since 5.1.0 because of implementation of allocation rows when using links.</p> <p>PATCH: Fixed issues with tooltip template markup using keywords beginning with #, and concerning date formatting.</p> <p>PATCH: Fixed issue concerning eventually false week numbering in timescale.</p> <p>PATCH: Fixed issue in world view that occurred under certain circumstances when changing time</p>

Version	Description of changes
	<p>resolution in main view.</p> <p>PATCH: Fixed issues concerning graphical links and missing animations after updating the data model.</p>
<b>5.1.0</b> (WAS NOT OFFICIALLY PUBLISHED)	<p>MINOR: Now it is possible to show allocations in own rows also in the resources view using the new option <code>pm_allocationRowsVisibleInResourcesView</code>.</p> <p>MINOR: Symbols shown for properties <code>ReleaseDate</code> and <code>DueDate</code> on Activity objects are now specifiable and draggable:</p> <ul style="list-style-type: none"> <li>• New properties <code>PM_ReleaseDateAllowedDragModes</code>, <code>PM_ReleaseDateSymbolHeight</code>, <code>PM_ReleaseDateSymbolID</code>, and <code>PM_ReleaseDateSymbolWidth</code> for the release date.</li> <li>• New properties <code>PM_DueDateAllowedDragModes</code>, <code>PM_DueDateSymbolHeight</code>, <code>PM_DueDateSymbolID</code>, and <code>PM_DueDateSymbolWidth</code> for the due date.</li> <li>• New property <code>propNames</code> in <code>callbackArgs</code> of callback <code>onDrop</code>. See MAJOR change in 5.2.0 to properties <code>startPropertyName</code> and <code>endPropertyName</code>.</li> </ul> <p>MINOR: New properties <code>Background/TextColor</code> on <code>TableRowDefinition</code> objects.</p> <p>MINOR: New argument <code>cellIndex</code> at callbacks <code>onClicked</code> and <code>onDoubleClicked</code>.</p> <p>MINOR: New methods <code>scrollViewAreaHorizontally</code> and <code>scrollViewAreaVertically</code>.</p> <p>MINOR: New locales added for Japanese, Russian, Thai, and Chinese.</p> <p>MINOR: Clarification of options and properties concerning title, a.o. renaming <code>Header</code> to <code>Title</code> (compatibility is given):</p> <ul style="list-style-type: none"> <li>• New option names are <code>pm_tableTitleBackgroundColor</code>, <code>pm_tableTitleTextColor</code>, <code>pm_tableTitleColumnSeparatorColor</code>, <code>pm_tableTitleHighlightingColor</code>, <code>pm_entitiesTableTitleBackgroundColor</code>, <code>pm_entitiesTableTitleTextColor</code>, <code>pm_entitiesTableTitleColumnSeparatorColor</code>, <code>pm_entitiesTableTitleHighlightingColor</code>. Old option names remain in the interface, but are marked as deprecated.</li> </ul> <p>MINOR: Property <code>Title</code> of <code>TableCellDefinition</code> objects renamed to <code>TitleText</code> (compatibility is given).</p> <p>PATCH: Method <code>selectObjects</code> did not work for allocation objects in activities view.</p>
<b>5.0.2</b>	<p>PATCH: Wrong coloring of allocation bars.</p> <p>PATCH: After updating links, they have not been rendered correctly.</p>
<b>5.0.1</b>	<p>PATCH: Fix for issue when dragging an entity without a non-zero duration.</p>
<b>5.0.0</b> (see also <a href="#">release notes!</a> )	<p>MINOR: New method <code>saveAsPDF</code>.</p> <p>MINOR: Now allocation rows can be made visible in activities view by using the new option <code>pm_allocationRowsVisibleInActivitiesView</code>. Additionally other additions were made in this environment:</p> <ul style="list-style-type: none"> <li>• New options <code>pm_defaultAllocationTableRowDefinitionID</code>, <code>pm_defaultAllocationMinimumRowHeight</code>, <code>pm_defaultAllocationRowSelectable</code>, <code>pm_defaultActivityAllocationRowsCollapsible</code>, <code>pm_defaultAllocationRowDesign</code>, <code>pm_defaultAllocationAllowedBarDragModesInActivitiesView</code>, <code>pm_forcedAllocationAllowedBarDragModesInActivitiesView</code>.</li> <li>• New properties <code>PM_AllocationRowsCollapsible</code>, <code>PM_AllocationRowsCollapseState</code>,</li> </ul>

Version	Description of changes
	<p>PM_HasAllocationRows on Activity objects.</p> <ul style="list-style-type: none"> <li>New properties PM_MinimumRowHeight, PM_RowSelectable, PM_RowDesign, PM_AllowedBarDragModesInActivitiesView, TableText on Allocation objects.</li> <li>New callbackArgs property isForAllocationRows in callback onCollapseStateChanged.</li> </ul> <p>MINOR: Now links can be defined between allocations:</p> <ul style="list-style-type: none"> <li>New properties Source/TargetAllocationID for links.</li> <li>New options pm_definedAllocationLinksVisibleInActivitiesView/ResourcesView.</li> </ul> <p>MINOR: New optional parameter for method removeAll.</p> <p>MINOR: New options pm_watermarkSymbolID and pm_watermarkOpacity.</p> <p>MINOR: New properties PM_StatusFrameColor/Visible for activities and allocations. New options pm_defaultActivity/AllocationStatusFrameColor.</p> <p>MINOR: New property PM_BarOpacity for Activity and Allocation objects.</p> <p>MINOR: New properties Background/TextColor, Background/TextColorSource on TableCellDefinition objects.</p> <p>MINOR: New options tableViewWidthInActivitiesView/ResourcesView/LoadsView and tableViewWidthsSynchronized.</p> <p>MINOR: New options pm_symbolColumnTitleVisible/SymbolIDs as well as pm_entitiesTableSymbolColumnTitleVisible/SymbolIDs.</p> <p>MINOR: New option pm_scrollOffsetsChangedCallbackTimeDelay.</p> <p>MINOR: Additional values for callback onVerticalScrollOffsetChanged.</p> <p>MINOR: Constraint dates are now also considered in the summary and in the diamond bar shape of allocations and activities. For the diamond shape, the PredictedEnd property is also taken into account.</p> <p>MINOR: Options pm_activity/resourceTableRowDefinitionIDForTitle renamed to pm_tableRowDefinitionIDForTitleInActivitiesView/ResourcesView. Option pm_entityTableRowDefinitionIDForTitle renamed to pm_tableRowDefinitionIDForTitleInEntitiesTable. Old option names are deprecated but accepted for compatibility reasons.</p> <p>MINOR: New option pm_tableRowDefinitionIDForTitleInLoadsView.</p> <p>MINOR: New option tooltipDelay.</p> <p>MAJOR: After dragging and dropping a draggable date line, the application now needs to update the date line object within the onDrop callback handler to apply the changed date. In older versions the date line incorrectly remained on the new date.</p> <p>MAJOR: Activity rows and resource rows do not show the calendar of an ancestor anymore when the represented objects do not have an own calendar.</p> <p>PATCH: Fix for issue with hidden rows when using method scrollToObject.</p> <p>PATCH: Several fixes concerning tooltips and captions of PeriodHighlighterEntry objects.</p>
4.0.5	<p>PATCH: Fixed and improved appearance of bars while dragging and of dragged entities in time area also especially when using the options timeStepUnit and timeStepUnitFactor for a more intuitive user experience.</p> <p>PATCH: Fixed cursor issue on entities table, not being updated correctly when moving the</p>

Version	Description of changes
	<p>mouse.</p> <p>PATCH: Fixed issue concerning snapping when dragging bars in collapsed rows and bars of collapsed sub rows are shown.</p> <p>PATCH: Fixed issue concerning snapping when dragging bars and start dates or end dates of other bars contain millisecond values. Now these start dates and end dates are rounded down or up, resp., to full seconds.</p> <p>PATCH: Property date of callbackArgs in callback onShowTooltip was not existent since 4.0.3.</p>
<b>4.0.4</b>	<p>PATCH: Fixed crash when using world view together with bar symbols.</p> <p>PATCH: Canceling of deselection of all objects in a callback handler for callback onSelectionChanged now possible. Additionally, new attributes for callbackArgs named reason, reasonObject, reasonObjectType.</p> <p>PATCH: Zooming out of the currently visible time range by using the “up” button in the timescale resulted in an incorrect horizontal scroll offset.</p> <p>PATCH: Fixed issue when using the method updateDateLines (not all properties have been updated).</p> <p>PATCH: Fixed issue where the curves were not shown when adding a resource with property PM_CurveCollapseState set to 0.</p> <p>PATCH: Fixed issue with setting options timeStepUnit and timeStepUnitFactor.</p>
<b>4.0.3</b>	<p>PATCH: Fixed naming issues for external dependencies “hammerjs” and “tinycolor2”.</p>
<b>4.0.2</b>	<p>PATCH: Improved behavior for bar dragging.</p> <p>PATCH: Fixed issue concerning AMD for external dependency “jquery-ui/ui/widget” and “jquery”.</p>
<b>4.0.1</b>	<p>PATCH: Fixed a crash that occurred when the pm_defaultActivityBarHeight option was set when initializing the widget.</p> <p>PATCH: Default for option viewType now is ViewType.Activities again (since 4.0.0 it was set to ViewType.Resources).</p>
<b>4.0.0</b> (See also <a href="#">release notes</a> !)	<p>MAJOR: To be treated as a bug fix, the property dragMode in the callback onDrop now contains the dragMode of the interaction that took place and not all allowed drag modes on the object!</p> <p>MINOR: New options multipleBarDraggingEnabled, pm_forcedActivityAllowedBarDragModes, pm_forcedAllocationAllowedBarDragModes. New properties coupledObjects and startsAndEndsOfCoupledObjects in callback onDrop. New property selectedObjects in callback canDrag.</p> <p>MINOR: When dragging a bar vertically the visible time span of it now is adapted according to the calendar of the current target row.</p> <p>MINOR: World view implemented. See options worldViewVisible, worldViewPosition, worldViewExtent.</p> <p>MINOR: Improved loading performance.</p> <p>MINOR: New options loggingEnabled and interactiveActivationOfLoggingEnabled.</p> <p>MINOR: New property SymbolIDSource in TableCellDefinition object.</p>

Version	Description of changes
	<p>MINOR: New property newRowObjectIsSuitableResource for callbackArgs of callback onDrag.</p> <p>MINOR: When dropping a date line interactively, the resulting date is rounded to the best possible date that is represented by the X coordinate the line phantom is shown on.</p> <p>MINOR: New callback "visibilityFilter" triggered for filtering row objects of types Activity, Entity, Resource.</p> <p>MINOR: Additional parameters for method scrollToObject and new options pm_scrollToObjectAnimationEnabled, pm_scrollToObjectHighlightFlashingEnabled, and pm_scrollToObjectHighlightingColor.</p> <p>MINOR: New property HorizontalTitleAlignment in TableCellDefinition object.</p> <p>MINOR: New properties PM_BarTextPrefixSymbolID/Height/Width, PM_Left/RightBarSymbolID, PM_Left/RightBarSymbolWidth, PM_Left/RightBarSymbolHeight for Allocation and Activity objects.</p> <p>MINOR: Support for Polish (pl) and Portuguese (pt = pt-pt; pt-br) locales added.</p> <p>MINOR: New option pm_ignoreCalendarOnAllocationBarInteractions.</p> <p>MINOR: Option pm_commonViewAreaVisible renamed to pm_mainViewAreaVisible.</p> <p>PATCH: Many bug fixes.</p>
<b>3.2.1</b>	PATCH: A click on a curve now triggers the callback onClicked again.
<b>3.2.0</b> (See also <a href="#">release notes</a> !)	<p>MINOR: New options cursorDateLineVisible, pm_timeAreaPanningMode, pm_timescaleInteractionsEnabled, and pm_curvePanelsCollapsibleInResourcesView.</p> <p>MINOR: New options currentDate, pm_pastBackgroundFillColor/LineColor/LineWidth/LineDashArray.</p> <p>MINOR: New option timeZone.</p> <p>MINOR: New date line properties CaptionOrientation, CaptionPosition, InFrontOfBars, and Draggable.</p> <p>MINOR: New activity and allocation properties PM_Status4Color and PM_Status4Visible.</p> <p>MINOR: New allocation property SuitableResourceIDs and new options pm_suitableResourcesOverlayColor/pm_unsuitableResourcesOverlayColor.</p> <p>MINOR: New object types PeriodHighlighter/PeriodHighlighterEntry and new methods add/update/removePeriodHighlighters. New property PM_PeriodHighlighterID on activity and resource objects. New VisualType property PeriodHighlighter.</p>
<b>3.1.3</b>	<p>PATCH: Texts in first scrollable table column (in left table and in entities table) was clipped too much on the right side.</p> <p>PATCH: In some cases, the SVG content was drawn over the horizontal scrollbars.</p> <p>PATCH: It is now allowed to drag bars even when they are drawn inside a visible collapsed row and belong to a hidden row.</p>
<b>3.1.2</b>	PATCH: Updates to calendar and curve objects now updates also the activities view.
<b>3.1.1</b>	PATCH: Performance issue and memory leaks removed.



Version	Description of changes
<b>3.1.0</b> (See also <a href="#">release notes!</a> )	<p>MINOR: New options pm_topRowMarginInTimeArea, pm_bottomRowMarginInTimeArea, pm_subRowDistanceInTimeArea, pm_topBarSymbolsVisible.</p> <p>MINOR: New option pm_linksVisibleInActivitiesView</p> <p>MINOR: New option timescaleNavigationMode</p> <p>MINOR: New link property PM_RoutingType and new option pm_defaultLinkRoutingType</p> <p>MINOR: New option pm_selectionColor</p> <p>MINOR: New option pm_splitterHighlightingColor</p>
<b>3.0.0</b>	<p>MINOR: New objects TooltipTemplate, TableRowDefinition/TableCellDefinition, DateLine including add/update/remove methods and properties named PM_(Bar/Curve)TooltipTemplateID and PM_TableRowDefinitionID on several objects.</p> <p>MINOR: New properties like PM_RowSelectable/PM_BarSelectable, PM_RowCollapsible on several objects.</p> <p>MINOR: New property PM_ViewArea on Resource objects.</p> <p>MINOR: New properties BaselineStart/BaseLineEnd, DueDate, ReleaseDate plus color properties on Activity objects.</p> <p>MINOR: New properties PM_BarHeight, PM_BarTextWrapMode, PM_EndIsSnapTarget/ PM_StartIsSnapTarget, PM_SnapTargetsForStart/ PM_SnapTargetsForEnd on Activity and Allocation objects.</p> <p>MINOR: New properties PM_CollapsedRowDesign/ PM_ExpandedRowDesign, PM_CollapseState/PM_CurveCollapseState, PM_MinimumRowHeight on Activity and Resource objects.</p> <p>MINOR: New properties EarliestEnd/EarliestStart, LatestEnd/LatestStart, MustEndOn/MustStartOn plus color properties, and PM_EarliestDragStart/ PM_LatestDragEnd on Activity and Allocation objects.</p> <p>MINOR: New method setTimeResolutionForView.</p> <p>MINOR: Many new color options e.g. for coloring the timescale.</p> <p>MINOR: New callbacks onClicked, onCollapseStateChanged/ onCurveCollapseStateChanged, onTableCellDefinitionWidthChanged, onTimeAreaViewParametersChanged, onVerticalScrollOffsetChanged.</p> <p>MINOR: And some more object properties and options.</p>
<b>2.1.0</b>	<p>MINOR: New method about.</p> <p>MINOR: New message boxes for invalid, expiring, expired, not existing license.</p>
<b>2.0.0</b>	<p>MAJOR: Now the setting of a license key is mandatory.</p> <p>MINOR: New method removeAll.</p> <p>MINOR: New option locale.</p> <p>MINOR: New allocation properties PM_ProgressColor and PM_ProgressNonworkingColor.</p> <p>PATCH: Activity property Editable now marked as deprecated.</p> <p>MINOR: New option pm_linksVisibleInResourcesView.</p>



Version	Description of changes
<b>1.0.0</b>	Initial release.

## 2 System Requirements

### 2.1 Supported Browsers and Versions

Google Chrome (current version at delivery date of library)

Mozilla Firefox (current version at delivery date of library)

Apple Safari (current version at delivery date of library)

Microsoft Edge (current version at delivery date of library)

### 2.2 Needed 3<sup>rd</sup> Party Libraries and Versions

The libraries with the prefix nwaf contain the VSW SE and are copyrighted by NETRONIC. They depend on the following 3rd party, open-source libraries.

In your application you can include these libraries directly or you can download these from a Content Delivery Network (CDN). For an example see the VSW SE Sample App.

**Your application must obey the license conditions of each 3rd party library!**

Library Name NPM Module Name	Supported Versions	Comment
<b>jQuery</b> "jquery"	2.x.x/3.x.x (last one tested: 3.7.0)	Required. Needed for HTML handling. Versions 2.x.x support older Internet Explorer versions (but these are not supported by VSW SE!).  License: MIT.  URL: <a href="https://jquery.com/">https://jquery.com/</a>
<b>jQuery UI</b> "jquery-ui/ui/..."	1.11.x/1.12.x/ 1.13.x (last one tested 1.13.1)	Required. Needed parts are the Widget Factory (widget) and Dialogs (widgets/dialog).  License: MIT.  URL: <a href="https://jqueryui.com/">https://jqueryui.com/</a>
<b>D3.js</b> "d3"	4.0.0-7.x.x (last one tested: 7.8.5)	Required. Needed for SVG handling. Beginning with version 5.0.0 Internet Explorer is not supported officially anymore.  License: ISC.  URL: <a href="https://d3js.org/">https://d3js.org/</a>  Used modules: d3-axis, d3-ease, d3-format, d3-scale, d3-selection, d3-shape, d3-time-format, d3-timer, d3-transition.
<b>Hammer.js</b>	2.0.8	Required. Needed for touch and mouse gesture handling.

<b>"hammerjs"</b>		License: MIT. URL: <a href="https://hammerjs.github.io/">https://hammerjs.github.io/</a>
<b>TinyColor</b> <b>"tinycolor2"</b>	1.4.1	Required. Needed for calculating derived colors e.g. for coloring non-working times. License: MIT. URL: <a href="https://bgrins.github.io/TinyColor/">https://bgrins.github.io/TinyColor/</a>
<b>Moment.js</b> <b>"moment"</b>	2.x.x (last one tested: 2.29.4)	Optional. Needed only, when using option "timeZone". The developer can decide, which data to serve with Moment Timezone. License: MIT. URL: <a href="https://momentjs.com/">https://momentjs.com/</a>
<b>Moment.Timezone</b> <b>"moment-timezone"</b>	0.5.x (last one tested: 0.5.43)	Optional. Needed only, when using option "timeZone". License: MIT. URL: <a href="https://momentjs.com/">https://momentjs.com/</a>
<b>SVG-to-PDFKit</b> <b>"svg-to-pdfkit"</b>	0.1.8 *	Optional. Needed only, when using method saveAsPDF. License: MIT. URL: <a href="https://github.com/alafr/SVG-to-PDFKit">https://github.com/alafr/SVG-to-PDFKit</a>
<b>PDFKit</b> <b>"pdfkit"</b>	0.11.0-0.13.0	Optional. Needed only, when using method saveAsPDF (needed by SVG-to-PDFKit). Provided standalone version contains libraries "stream" and "emitter-component". Licenses: MIT. URL: <a href="https://pdfkit.org/">https://pdfkit.org/</a>
<b>blob-stream</b> <b>"blob-stream"</b>	0.1.3	Optional. Needed only, when using method saveAsPDF (needed for receiving the PDF content generated by PDFKit). Provided standalone version contains libraries "stream", "emitter-component", and "blob". Licenses: MIT. URL: <a href="https://github.com/devongovett/blob-stream">https://github.com/devongovett/blob-stream</a>
<b>html2canvas</b> <b>"html2canvas"</b>	1.4.x (last one tested: 1.4.1)	Optional. Needed only, when using method saveAsPDF together with the options topHTML and/or bottomHTML. License: MIT. URL: <a href="https://html2canvas.hertzen.com/">https://html2canvas.hertzen.com/</a>

Note: The jQuery plugin jquery.mousewheel, which was required up to VSW SE 3.1, is no longer required as of VSW SE 3.2.

\* There is a bug in all versions of SVG-to-PDFKit that lowers the performance from about 1 second per exported page to about 1 minute per page. We were able to find and fix this bug (see pull request here: <https://github.com/alafr/SVG-to-PDFKit/pull/143>). The patch is included in master version of the

source code on GitHub, but there is no release yet. Therefore, we deliver the downloaded master version for your convenience.

## 2.3 Imported 3<sup>rd</sup> Party Libraries and Versions

These libraries are imported into our library file nwaf-apptools.min.js.

Library Name NPM Module Name	Version	Comment
<b>CSS Element Queries</b> "css-element-queries"	1.2.3	The included resize sensor triggers a callback when the size of a DIV element changes in the DOM. License: MIT. URL: <a href="https://github.com/marcj/css-element-queries">https://github.com/marcj/css-element-queries</a>
<b>FileSaver</b> "file-saver"	2.0.5	Imported to support logging (see option loggingEnabled) and PDF export (see method saveAsPDF). License: MIT. URL: <a href="https://github.com/eligrey/FileSaver.js">https://github.com/eligrey/FileSaver.js</a>
<b>core-js</b> "core-js"	3.23.1	Parts of this polyfill library are compiled into NWAF libraries by Babel.js to allow developing with current JavaScript standards while remaining compatible to the supported browsers and versions. License: MIT. URL: <a href="https://github.com/zloirock/core-js">https://github.com/zloirock/core-js</a>
<b>Polyfill-library</b> "polyfill-library"	4.5.0	Parts of this polyfill library are used to allow using web standard methods while remaining compatible to the supported browsers and versions. License: MIT. URL: <a href="https://github.com/Financial-Times/polyfill-library">https://github.com/Financial-Times/polyfill-library</a>

## 3 Object Model

The object model of the Visual Scheduling Widget Base is designed for resource planning in general, but is extended to cover presentations all views, activities view, resources view, and loads view.

The model is extensible on any JavaScript object. There are no special constructor functions for creating objects of a specific VSW SE type. Hence, the objects can be easily created with or without using the *new* keyword. The objects must provide the properties required by the corresponding VSW type and optionally those that are to deviate from the default values.

In this document you will find some UML diagrams that illustrate at a glance the relationships between the object type currently under consideration and the associated ones. **Only those object properties are listed that are essential for understanding the concept of this data model.** The **dark blue shaped types** in these diagrams are the ones that can be processed with the methods add..., update... and

remove... of the widget. To do this, they explicitly provide identifiers in the form of the ID property. In contrast, the **light blue types** include dependent objects without their own identifiers.

A compact description of the model can be found in the document "A Model for Resource Planning HTML5 Gantt Charts" delivered with this product (please see the file ResourcePlanningModel.pdf).

**A note on the dates in attributes:**

Browsers did not handle date strings consistently in the past. So it is recommended to use the simplified ISO 8601 standard see <http://www.ecma-international.org/ecma-262/5.1/#sec-15.9.1.15> for defining unambiguously: Examples: 2019-05-03T08:13:28Z (UTC) or 2019-05-03T10:13:28+02:00 (MEST) for the same time point. Using date objects in the object is recommended, since then the creation can be done on several ways and internally the dates can be used immediately without conversion.

**A note on using object references in add, update, and remove methods:** Internally object references are preserved by their ID property values. Therefore, you can or even should use new object references when updating an object. Internally the object reference for an existing ID then is updated. When a callback is triggered, object references within the callback arguments are the same as the last given reference on an add or update method. When using one of the remove methods with object references, these will be reduced to their ID property values, so it is not important to handle over the same object reference as in the last add or update method.

**A note on the old "PM\_" prefix of the object properties:**

The old "PM\_" prefix has been removed from the object properties for simplicity. However, there is no need to change existing code as the old notation of the properties will continue to be supported.

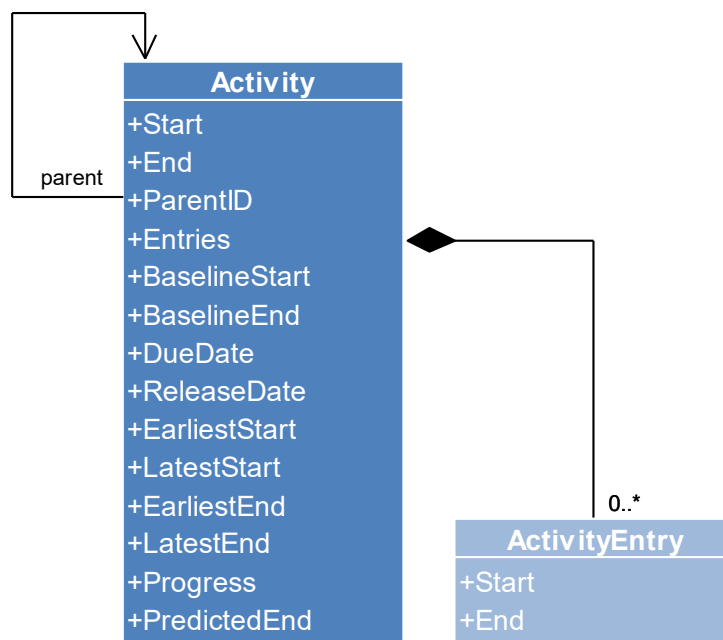
**A note on using CSS custom properties for coloring:**

Each object property that controls the color of an element in VSW can be set a value of type string that represents a "CSS color value". This means that you have, for instance, the following options to specify a color value:

- a predefined color name that CSS supports, such as "red", "green", or "blue"
- a hexadecimal notation, such as "#FF0000", "#00FF00", or "#0000FF"
- an RGB or RGBA notation, such as "rgb(255, 0, 0)", "rgb(0, 255, 0)", or "rgba(0, 0, 255, 0.5)"
- an HSL and HSLA notation, such as "hsl(0, 100%, 50%)", " hsl(120, 100%, 50%)", or " hsla(240, 100%, 50%, 0.5)"

In addition, you can also use [CSS Custom Properties](#). How to do that is described [in a blog post](#).

### 3.1 Activities



#### 3.1.1 Activity

An Activity object defines the properties of a single activity.

Activity Property Name	Type	Description
<b>AllocationRowsCollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: -1</b> – Specifies whether the allocation rows of the activity should be expanded or collapsed when displayed. See also callback <code>onCollapseStateChanged</code> and options <code>allocationRowsVisibleInActivitiesView</code> , <code>allocationRowsVisibleInResourcesView</code> .  -1: no change 0: display activity row in an expanded way for allocation rows. 1: display activity row in a collapsed way for allocation rows.
<b>AllocationRowsCollapsible</b>	boolean	<b>Optional, default: true</b> – If set to true, then the row representing this activity row will be interactively collapsible when allocation rows exist.
<b>AllowedBarDragModes</b>	number (see enum <a href="#">ActivityBarDragModes</a> )	<b>Optional, default: value of option <code>defaultActivityAllowedBarDragM</code></b>

Activity Property Name	Type	Description
		<b>odes</b> – This option determines the allowed bar drag modes for this activity in the activities view (these can be overwritten using the callback <code>canDrag</code> ).
<b>AllowedRowDragModes</b>	number (see enum <a href="#">RowDragModes</a> )	<b>Optional, default: value of option <code>defaultActivityAllowedRowDragModes</code></b> – This option determines the allowed row drag modes for this activity (these can be overwritten using the callback <code>canDrag</code> ).
<b>BarDesign</b>	number (see enum <a href="#">BarDesigns</a> )	<b>Optional, default: value of option <code>defaultActivityBarDesign</code></b> – This property determines the default design for activity bars including or excluding entries, complex shape, symbols, status, constraints, baseline, progress, and text.
<b>BarHeight</b>	number ( $\geq 0, \leq 1000$ )	<b>Optional, default: value in option <code>defaultActivityBarHeight</code></b> – Height of the bars in pixels. This attribute is useful, when more than one line of text is shown inside (see attribute <code>BarText</code> ). Proposal: For one line take 22, for two lines 38, for three lines 54, and so on. When no progress bar is needed, then you can subtract 4 from the value.
<b>BarOpacity</b>	number ( $\geq 0.0, \leq 1.0$ )	<b>Optional, default: undefined</b> – Specifies the opacity of the entire activity bar (including the visualization of the progress bar, symbols, constraint dates, and baseline bar).
<b>BarPatternColor</b>	string	<b>Optional, default: "white"</b> – Color for the pattern when this is visible by using property <code>BarPatternType</code> .
<b>BarPatternType</b>	number (see enum <a href="#">PatternType</a> )	<b>Optional, default: None</b> – If set, then a pattern is shown on top of the fill color and behind the text.
<b>BarSelectable</b>	boolean	<b>Optional, default: value of option <code>defaultActivityBarSelectable</code></b> – If set to true, then the bar

Activity Property Name	Type	Description
		representing this activity will be selectable.
<b>BarShape</b>	number (see enum <a href="#">ActivityBarShape</a> )	<b>Optional, default: value in option defaultActivityBarShape</b> – This option defines which shape should be used by default for the visualization activity bars.
<b>BarShapeSymbolID</b>	string	<b>Optional, default: undefined</b> – If the bar shape named Symbol is used (see property BarShape), then the symbol defined here will be shown. The symbol will be resized to the height defined in property BarHeight and to the width defined in property BarShapeSymbolWidth.
<b>BarShapeSymbolWidth</b>	number > 0	<b>Optional, default: value of property BarHeight</b> – Width of the symbol defined in property BarShapeSymbolID when the bar shape named Symbol is used.
<b>BarText</b>	string	<b>Optional, default: undefined</b> – Text to display in the bar.
<b>BarTextColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Color for the texts of the bar.
<b>BarTextFormat</b>	string	<p><b>Optional, default: value of option defaultActivityBarTextFormat</b> – This property describes the format of the bar text. If not set, then the value of property BarText is displayed.</p> <p>This string contains the placeholders for object values surrounded by double curly braces {{ }}. For example, based on the following string a tooltip with a table containing three rows of key-value pairs is created, where the values are taken from the properties "name" and "firstName" of the referenced object:</p> <pre>{{name}}, {{firstName}}</pre> <p>As an escape, the use of three open curly braces {{{ are displayed as {{.</p>

Activity Property Name	Type	Description
		<p>Additionally, the property name can be extended to contain the desired property type as in <code>{{Start:date}}</code>. At the moment only the types 'date' and 'number' are possible besides 'string' (other property types are converted automatically with <code>toString()</code>). The type 'date' converts date values by default using the same format as other dates in the timescale and at the dragging date line captions. You can add another colon followed by a format name, that is defined by the options <code>intlDateTimeFormatOptionsMap</code> or <code>intlNumberFormatOptionsMap</code>, resp.</p> <p>The referenced object is the object on which the tooltip will be shown. For period highlighter entries and allocation entries the referenced object is the main object and not the entry object.</p> <p>It is possible to access related objects by using the following keywords within the property accessor string: <code>&gt;Parent</code>, <code>&gt;Calendar</code>.</p> <p>It is also possible to access variables that are defined by the option <code>applicationVariablesMap</code> by using <code>?variableName</code>.</p> <p>If the value reached is an object, you can then access a property value by using a prefixed dot: <code>.propertyName</code> and you can use <code>[...]</code> to access a property value, a map entry or an array entry. Within <code>[...]</code> you can use a literal like 5 or A (with or without quotes) or even curly braces <code>{{...}}</code> with the same rules as above.</p>
<b>BarTextPrefixSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the bar symbol before the text



Activity Property Name	Type	Description
		(see property <code>BarTextPrefixSymbolID</code> ) in pixels at a zoom factor of 100%.
<b>BarTextPrefixSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown before the text inside of the activity bar. The symbol will be shown vertically centered inside the bar.
<b>BarTextPrefixSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the bar symbol before the text (see property <code>BarTextPrefixSymbolID</code> ) in pixels at a zoom factor of 100%.
<b>BarTextWrapMode</b>	number (see enum <a href="#">TextWrapMode</a> )	<b>Optional, default: <code>TextWrapMode.None</code></b> – Specifies whether the text inside the bar is wrapped.
<b>BarTooltipTemplateID</b>	string	<b>Optional, default: value of option <code>defaultActivityBarTooltipTemplateID</code></b> – ID of a tooltip template.  The template is used for tooltips that appear on the activity bars.
<b>BaselineBorderColor</b>	string (CSS color value)	<b>Optional, default: "#808080"</b> – Color for the border of the baseline bar.
<b>BaselineColor</b>	string (CSS color value)	<b>Optional, default: "#C8C8C8"</b> – Color for the working time periods of the baseline bar. The nonworking time periods of the bar will be colored with the same color as long as the property <code>BaselineNonworkingTimeColor</code> is undefined.
<b>BaselineEnd</b>	Date   string	<b>Optional, default: undefined</b> – Baseline end date of the activity.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.

Activity Property Name	Type	Description
		See also setting ActivityBaselineBarsVisible.
<b>BaselineNonworkingTimeColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: value of property BaselineColor</b> – Color for the nonworking time periods of the baseline bar.</p> <p>If set to "calculated", a color will be calculated using the color defined by the BaselineColor property.</p>
<b>BaselineStart</b>	Date   string	<p><b>Optional, default: undefined</b> – Baseline start date of the activity.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p> <p>See also setting ActivityBaselineBarsVisible.</p>
<b>BorderColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: "gray"</b> – Color for the border of the bar.</p> <p>If set to "calculated", a color will be calculated using the color defined by the Color property. This can be useful in situations where two bars are positioned next to each other, and a graphical indicator is needed to visually distinguish the two bars.</p>
<b>BorderDashArray</b>	string	<p><b>Optional, default: "none"</b> – Pattern of dashes and gaps for drawing the border of the bar. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a>.</p>

Activity Property Name	Type	Description
		The value "none" indicates that no dashing is used. In this case, the border is drawn solid.
<b>CalendarGridColor</b>	string (CSS color value)	<b>Optional, default: value of option calendarGridColor</b> – Specifies a color used to color the vertical stripes representing the nonworking times for the activity object inside the diagram. If allocation rows are visible the color is used for these rows, too.
<b>CalendarID</b>	string	<b>Optional, default: undefined</b> – Corresponding calendar. If undefined, then the default calendar specified by the option defaultCalendarID will be used. See also option activityCalendarsEnabled.
<b>CollapsedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default: value in option defaultActivityCollapsedRowDesign</b> – Specifies how the time area is filled when the row is collapsed and visible.  See explanation for possible values in enumeration chapter.
<b>CollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: -1</b> – Specifies whether the row of the activity should be expanded or collapsed when displayed. See also callback onCollapseStateChanged.  -1: no change 0: display activity row in an expanded way 1: display activity row in a collapsed way
<b>Color</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Color for the working time periods of the bar. The nonworking time periods of the bar will be colored with the same color as long as the property NonworkingTimeColor is undefined.
<b>CurveCollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: -1</b> – Specifies whether the curves in a activity row should be expanded or collapsed when displayed (only applicable, when option

Activity Property Name	Type	Description
		<p>curvePanelsVisibleInActivitiesView is set). See also callback <code>onCurveCollapseStateChanged</code>.</p> <p>-1: no change 0: display curves 1: hide curves</p>
<b>DueDate</b>	Date   string	<p><b>Optional, default: undefined</b> – Due date of the activity.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p> <p>See also option <code>releaseDueDateConnectionsVisible</code>, if you want the widget to draw a connection line between a due date and a release date.</p>
<b>DueDateAllowedDragModes</b>	number (see enum <a href="#">ActivityBarDragModes</a> )	<p><b>Optional, default: None</b> – This option determines the allowed drag mode for the due date of this activity in the activities view (these can be overwritten using the callback <code>canDrag</code>). In this context only <code>None</code> and <code>DragHorizontally</code> are used.</p>
<b>DueDateColor</b>	string (CSS color value)	<p><b>Optional, default: "black"</b> – Color for the due date symbol.</p>
<b>DueDateSymbolHeight</b>	number	<p><b>Optional, default: 12</b> – Height of the due date symbol (see property <code>DueDateSymbolID</code>) in pixels at a zoom factor of 100%. The default symbol is not sizable.</p>
<b>DueDateSymbolID</b>	string	<p><b>Optional, default: internal diamond symbol</b> – Identifier of the symbol to be shown at the due date of the activity. See also <code>DueDateSymbolHeight</code>, and <code>DueDateSymbolWidth</code>.</p>
<b>DueDateSymbolWidth</b>	number	<p><b>Optional, default: 12</b> – Width of the due date symbol (see property <code>DueDateSymbolID</code>) in</p>

Activity Property Name	Type	Description
		pixels at a zoom factor of 100%. The default symbol is not sizable.
<b>EarliestDragStart</b>	Date   string	<b>Optional, default: undefined</b> – If set, then the time before the given date is grayed, when beginning to drag the activity bar. If the option <code>dragDatesLimitingInteraction</code> is set to true, then the bar itself cannot be dragged before the date.
<b>EarliestEnd</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>EarliestEndColor</b>	string (CSS color value)	<b>Optional, default: value in option <code>defaultActivityConstraintSymbolColor</code></b> – Color for the EarliestEnd constraint symbol.
<b>EarliestStart</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>EarliestStartColor</b>	string (CSS color value)	<b>Optional, default: value in option <code>defaultActivityConstraintSymbolColor</code></b> – Color for the EarliestStart constraint symbol.
Editable		<b>Deprecated! Use <code>AllowedBarDragModes</code> instead.</b>
<b>End</b>	Date   string	<b>Optional, default: undefined</b> – End date of the activity.

Activity Property Name	Type	Description
		If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>Entries</b>	<a href="#">ActivityEntry[]</a>	<p><b>Optional, default: undefined</b> – array of activity entries.</p> <p>If used, then the entries will be shown as colored rectangles within the bar representation of the activity. Additionally the property BarShape must be set to Regular or Rectangle.</p>
<b>ExpandedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<p><b>Optional, default: value in option defaultActivityExpandedRowDesign</b> – Specifies how the time area is filled when the row is expanded and visible.</p> <p>See explanation for possible values in enumerations chapter.</p>
<b>HasAllocationRows</b>	boolean	<b>Optional, default: false</b> – If set to true, then the row representing this activity will be collapsible/expandable for allocation rows even when no allocations exist referencing this activity. This serves for lazy loading.
<b>HasChildren</b>	boolean	<b>Optional, default: false</b> – If set to true, then the row representing this activity will be collapsible/expandable even when there are no children defined. This serves for lazy loading.
<b>ID</b>	string	<b>Required</b> – Identifier of the activity. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.

Activity Property Name	Type	Description
<b>LatestDragEnd</b>	Date   string	<b>Optional, default: undefined</b> – If set, then the time after the given date is grayed, when beginning to drag the activity bar. If the option <code>dragDatesLimitingInteraction</code> is set to true, then the bar itself cannot be dragged after the date.
<b>LatestEnd</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>LatestEndColor</b>	string (CSS color value)	<b>Optional, default: value in option <code>defaultActivityConstraintSymbolColor</code></b> – Color for the LatestEnd constraint symbol.
<b>LatestStart</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>LatestStartColor</b>	string (CSS color value)	<b>Optional, default: value in option <code>defaultActivityConstraintSymbolColor</code></b> – Color for the LatestStart constraint symbol.
<b>LeftBarSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the left bar symbol (see property <code>LeftBarSymbolID</code> ) in pixels at a zoom factor of 100%.
<b>LeftBarSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the left side of the activity bar. The symbol will be

Activity Property Name	Type	Description
		shown vertically centered inside the bar. See also RightBarSymbolID, LeftBarSymbolHeight, and LeftBarSymbolWidth.
<b>LeftBarSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the left bar symbol (see property LeftBarSymbolID) in pixels at a zoom factor of 100%.
<b>LinkSourceDate</b>	Date   string	<b>Optional, default: undefined</b> – Additional date serving as an additional “start point” to connect a link. See also property “RelationType” of link object.
<b>LinkTargetDate</b>	Date   string	<b>Optional, default: undefined</b> – Additional date serving as an additional “end point” to connect a link. See also property “RelationType” of link object.
<b>MinimumRowHeight</b>	number	<b>Optional, default: value in option defaultMinimumActivityRowHeight</b> – Minimum height of the activity row in pixels. This attribute is useful, when more than one line of text is shown inside the table cells. Proposal: For one line take 36*, for two lines 52, for three lines 68, and so on. In order to have the same height also, when no bar is placed in the row, take the maximum bar height adding 20 (e.g. 42) as minimum.  For using word wrapping in table cells, it is necessary to use a table row definition by setting the property TableRowDefinitionID and setting the property WrapMode in a contained table cell definition.
<b>MustEndOn</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every



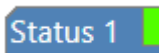
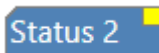
Activity Property Name	Type	Description
		formatted date string in a standardized way, one has to be careful about it.
<b>MustEndOnColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultActivityConstraintSymbol Color</b> – Color for the MustEndOn constraint symbol.
<b>MustStartOn</b>	Date   string	<p><b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>MustStartOnColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultActivityConstraintSymbol Color</b> – Color for the MustStartOn constraint symbol.
<b>NonworkingTimeColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: value of property Color</b> – Color for the nonworking time periods of the bar.</p> <p>If set to "calculated", a color will be calculated using the color defined by the Color property.</p>
<b>ParentID</b>	string	<p><b>Optional, default: undefined</b> – Identifier of the parent of the activity. This serves for setting up a hierarchy of activities.</p> <p>If this property is undefined the current activity will be considered as a root node of the activity hierarchy.</p> <p>We recommend using only a low number of hierarchy levels and we do not guarantee correct function beyond approx. 100 levels including hierarchy levels created by using Hierarchy-SupplementaryDefinitions.</p>
<b>PeriodHighlighterID</b>	string	<b>Optional, default: undefined</b> – Reference to a period highlighter

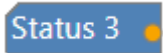
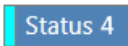
Activity Property Name	Type	Description
		object that contains colored time periods. This can be used to show shifts or exceptions to the calendar (see property CalendarID) that defines work and non-work times.
<b>PredictedEnd</b>	Date string	<p><b>Optional, default: undefined</b> – A date that indicates the predicted end of the activity. This date is used to display a bar between this date and the end of the activity.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>PredictedEndColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Color for the predicted end bar.
<b>Progress</b>	number (floating point; in percent; $\geq 0, \leq 100$ )	<b>Optional, default: 0.0</b> – Used to display a completion layer.
<b>ProgressBackgroundColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultActivityProgressBackgroundndColor</b> – Color for the background of the progress bar region.
<b>ProgressColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Color for the working time periods of the progress bar. The nonworking time periods of the bar will be colored with the same color as long as the property ProgressNonworkingTimeColor is undefined.
<b>ProgressNonworkingTimeColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: value of property ProgressColor</b> – Color for the nonworking time periods of the progress bar.</p> <p>If set to "calculated", a color will be calculated using the color defined by the ProgressColor property.</p>

Activity Property Name	Type	Description
<b>ReleaseDate</b>	Date   string	<p><b>Optional, default: undefined</b> – Release date of the activity.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p> <p>See also option <code>releaseDueDateConnectionsVisible</code>, if you want the widget to draw a connection line between a due date and a release date.</p>
<b>ReleaseDateAllowedDragModes</b>	number (see enum <a href="#">ActivityBarDragModes</a> )	<b>Optional, default: None</b> – This option determines the allowed drag mode for the release date of this activity in the activities view (these can be overwritten using the callback <code>canDrag</code> ). In this context only <code>None</code> and <code>DragHorizontally</code> are usable.
<b>ReleaseDateColor</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Color for the release date symbol.
<b>ReleaseDateSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the release date symbol (see property <code>ReleaseDateSymbolID</code> ) in pixels at a zoom factor of 100%. The default symbol is not sizable.
<b>ReleaseDateSymbolID</b>	string	<b>Optional, default: internal diamond symbol</b> – Identifier of the symbol to be shown at the due date of the activity. See also <code>ReleaseDateSymbolHeight</code> and <code>ReleaseDateSymbolWidth</code> .
<b>ReleaseDateSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the release date symbol (see property <code>ReleaseDateSymbolID</code> ) in pixels at a zoom factor of 100%. The default symbol is not sizable.
<b>RightBarSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the right bar symbol (see

Activity Property Name	Type	Description
		property RightBarSymbolID) in pixels at a zoom factor of 100%.
<b>RightBarSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the right side of the activity bar. The symbol will be shown vertically centered inside the bar. See also LeftBarSymbolID, RightBarSymbolHeight, and RightBarSymbolWidth.
<b>RightBarSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the right bar symbol (see property RightBarSymbolID) in pixels at a zoom factor of 100%.
<b>RowCollapsible</b>	boolean	<b>Optional, default: value of option defaultActivityRowCollapsible</b> – If set to true, then the row representing this activity will be interactively collapsible when children exist.
<b>RowSelectable</b>	boolean	<b>Optional, default: value of option defaultActivityRowSelectable</b> – If set to true, then the row representing this activity will be selectable.
<b>RowSymbolColumn-BackgroundColor</b>	string (CSS color value)	<b>Optional, default: value of property SymbolColumnBackgroundColor of assigned table row definition or option symbolColumnBackgroundColor</b> – Determines the color of the symbol column within this table row.
<b>RowSymbolIDs</b>	string[]	<b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the table symbol cell of the beginning of the table row. The symbols will be arranged one below the other. However, if the cell is not high enough to hold all symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.

Activity Property Name	Type	Description
		<p>An empty string ("" ) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve space for a symbol that may be shown at a later time.</p> <p>Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.</p>
<b>RowTooltipTemplateID</b>	string	<p><b>Optional, default: value of option defaultActivityRow-TooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the activity table rows.</p>
<b>SnapTargetsForEnd</b>	number (see enum <a href="#">SnapTargets</a> )	<p><b>Optional, default: value of widget option defaultActivitySnap-TargetsForEnd</b> – When dragging horizontally, then the visible end date of this allocation will optionally be snapping to date lines and calendar grids. The user can override an active snapping by pressing the ALT key while dragging. See also option maximumSnapDistance.</p>
<b>SnapTargetsForStart</b>	number (see enum <a href="#">SnapTargets</a> )	<p><b>Optional, default: value of widget option defaultActivitySnap-TargetsForStart</b> – When dragging horizontally, then the visible start date of this activity will optionally be snapping to date lines and calendar grids. The user can override an active snapping by pressing the ALT key while dragging. See also option maximumSnapDistance.</p>
<b>SortCode</b>	number string Date	<p><b>Optional, default: undefined</b> – If set, then the value will be used when sorting activity rows. The value type can be anyone that can be compared using JavaScript.</p>

Activity Property Name	Type	Description
		See also options <code>activityRowSortMode</code> and <code>activityRowSortCodePropertyName</code> .
<b>Start</b>	Date   string	<p><b>Optional, default: undefined</b> – Start date of the activity.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>Status1Color</b>	string (CSS color value)	<p><b>Optional, default: undefined</b> – Color for the status symbol to the right of the bar. If undefined, no symbol appears. Only visible, when property <code>Status1Visible</code> is true.</p>
<b>Status1Visible</b>	boolean	<p><b>Optional, default: false</b> – If set to true and the corresponding status color is set in property <code>Status1Color</code>, then a predefined symbol is displayed to the right of the bar.</p> <p> Status 1</p>
<b>Status2Color</b>	string (CSS color value)	<p><b>Optional, default: undefined</b> – Color for the status symbol to the right of the bar. If undefined, no symbol appears. Only visible, when property <code>Status2Visible</code> is true.</p>
<b>Status2Visible</b>	boolean	<p><b>Optional, default: false</b> – If set to true and the corresponding status color is set in property <code>Status2Color</code>, then a predefined symbol is displayed to the right of the bar.</p> <p> Status 2</p>
<b>Status3Color</b>	string (CSS color value)	<p><b>Optional, default: undefined</b> – Color for the status symbol to the right of the bar. If undefined, no symbol appears. Only visible, when property <code>Status3Visible</code> is true.</p>

Activity Property Name	Type	Description
Status3Visible	boolean	<b>Optional, default: false</b> – If set to true and the corresponding status color is set in property Status3Color, then a predefined symbol is displayed to the right of the bar. 
Status4Color	string (CSS color value)	<b>Optional, default: undefined</b> – Color for the status symbol to the left of the bar. If undefined, no symbol appears. Only visible, when property Status4Visible is true.
Status4Visible	boolean	<b>Optional, default: false</b> – If set to true and the corresponding status color is set in property Status4Color, then a predefined symbol is displayed to the left of the bar.  <b>Note: This property may be used with rectangle bar shapes only!</b> 
StatusFrameColor	string	<b>Optional, default: value of option defaultActivityStatusFrameColor</b> – Color for the status frame that will be shown when property StatusFrameVisible is set.
StatusFrameVisible	boolean	<b>Optional, default: false</b> – If set to true, then a frame is shown around the bar. See also property StatusFrameColor.
TableColor	string (CSS color value)	<b>Optional, default: level-dependent gray</b> – Color for the table row.
TableRowDefinitionID	string	<b>Optional, default: value of option defaultActivityTableRowDefinitionID</b> – Identifier of a TableRowDefinition object, that defines the composition of the table row.
TableText	string	<b>Optional, default: undefined</b> – Text to display in the table row (see also property TableRowDefinitionID).
TableTextColor	string (CSS color value)	<b>Optional, default: "black"</b> – Color for the table row texts.



Activity Property Name	Type	Description
TextColor		<b>Deprecated! Renamed to BarTextColor.</b>
TopLeftBarSymbolID	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the top left side of the activity bar. Please note: A symbol will be resized to an image with a width and height of 12 pixels each at a zoom level of 100%.
TopRightBarSymbolID	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the top right side of the activity bar. Please note: A symbol will be resized to an image with a width and height of 12 pixels each at a zoom level of 100%.
ViewArea	number (see enum <a href="#">ViewArea</a> )	<b>Optional, default: Main</b> – If set to Top, then the resource and its children are shown in a separate top view area in the resources view. Only settable on resource with no ParentID set. See also options mainViewAreaVisibleInActivitiesView and topViewAreaVisibleInActivitiesView.

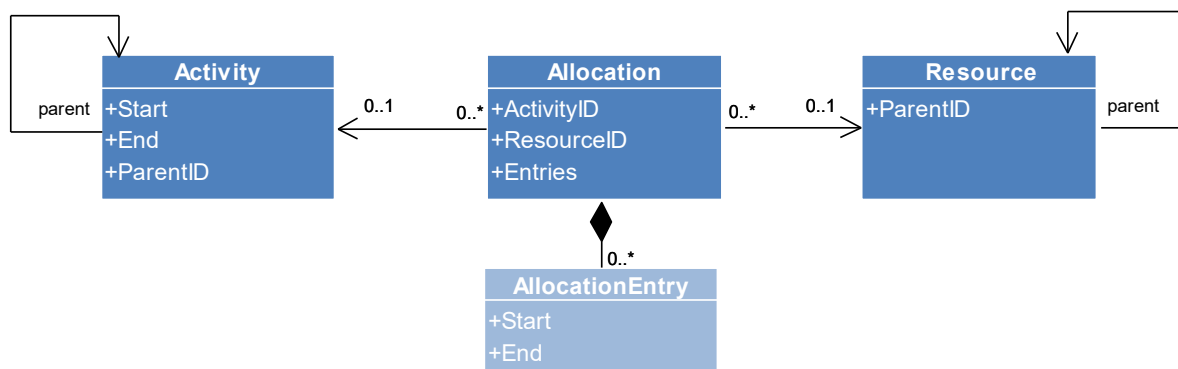
### 3.1.2 ActivityEntry

Objects of this type are only used within the array of the Entries property of Activity objects.

ActivityEntry Property Name	Type	Description
Color	string (CSS color value)	<b>Optional, default: value of property Color of activity</b> – Color for the working time periods of the bar.  The nonworking time periods of the bar will be colored with the same color as long as the property NonworkingTimeColor of the appropriate activity is undefined.
End	Date   string	<b>Optional, default: undefined</b> – End date of the activity entry. This date itself is not(!) part of the interval described by this entry.

ActivityEntry Property Name	Type	Description
		If data type is <i>string</i> , then the value has to be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC).
<b>Height</b>	number ( $\geq 0, \leq 1000$ )	<b>Optional, default: value in option defaultActivityBarHeight</b> – Height of the entry in pixels.
<b>NonworkingTimeColor</b>	string (CSS color value or "calculated")	<b>Optional, default: value of property NonworkingTimeColor of activity</b> – Color for the nonworking time periods of the bar.  If set to "calculated", a color will be calculated using the color defined by the Color property.
<b>PatternColor</b>	string	<b>Optional, default: "white"</b> – Color for the pattern when this is visible by using property PatternColor.
<b>PatternType</b>	number (see enum <a href="#">PatternType</a> )	<b>Optional, default: None</b> – If set, then a pattern is shown on top of the fill color and behind the text.
<b>RelativeTopOffset</b>	number	<b>Optional, default: 0</b> – Offset of the entry in pixels relative to the upper side of the corresponding activity bar. A negative number will shift the entry upwards, a positive number will shift the entry downwards.
<b>Start</b>	Date   string	<b>Optional, default: undefined</b> – Start date of the activity entry.  If data type is <i>string</i> , then the value has to be formatted this way: " YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC).

### 3.2 Allocations



### 3.2.1 Allocation

An Allocation object defines an allocation of one activity to one resource.

Allocation Property Name	Type	Description
<b>ActivityID</b>	string	<b>Optional, default: undefined</b> – Identifier of an <a href="#">Activity</a>
<b>AllowedBarDragModes</b>	number (see enum <a href="#">AllocationBarDragModes</a> )	<b>Optional, default: value of option defaultAllocation-AllowedBarDragModes</b> – This option determines the allowed bar drag modes for this allocation in the resources view (these can be overwritten using the callback canDrag).
<b>AllowedBarDragModesInActivitiesView</b>	number (see enum <a href="#">AllocationBarDragModes</a> )	<b>Optional, default: value of option defaultAllocation-AllowedBarDragModesInActivitiesView</b> – This option determines the allowed bar drag modes for this allocation in the activities view (these can be overwritten using the callback canDrag).
<b>BarDesign</b>	number (see enum <a href="#">BarDesigns</a> )	<b>Optional, default: value of option defaultAllocationBarDesign</b> – This property determines the default design for allocation bars including or excluding entries, complex shape, symbols, status, constraints, progress, and text.
<b>BarHeight</b>	number ( $\geq 0, \leq 1000$ )	<b>Optional, default: value in option defaultAllocationBarHeight</b> – Height of the bar in pixels. This is useful, when more than one line of text is shown inside (see attribute BarText). Proposal: For one line take 22, for two lines 38, for three lines 54, and so on. When no progress bar is needed, then you can subtract 4 from the value.
<b>BarOpacity</b>	number ( $\geq 0.0, \leq 1.0$ )	<b>Optional, default: undefined</b> – Specifies the opacity of the entire allocation bar (including the visualization of the progress bar, symbols, and constraint dates).
<b>BarPatternColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Color for the pattern when this is visible by using property BarPatternType.
<b>BarPatternType</b>	number (see enum <a href="#">PatternType</a> )	<b>Optional, default: None</b> – If set, then a pattern is shown on top of the fill color and behind the text.
<b>BarSelectable</b>	boolean	<b>Optional, default: value of option defaultAllocationBarSelectable</b> – If set to true, then the bar representing this allocation will be selectable.
<b>BarShape</b>	number (see enum <a href="#">AllocationBarShape</a> )	<b>Optional, default: value in option defaultAllocationBarShape</b> – This option defines which shape should be used by default for the visualization allocation bars.
<b>BarShapeSymbolID</b>	string	<b>Optional, default: undefined</b> – If the bar shape named Symbol is used (see property BarShape), then the symbol defined here will be shown. The symbol will be resized to the height defined in property BarHeight and to the width defined in property BarShapeSymbolWidth.

Allocation Property Name	Type	Description
<b>BarShapeSymbolWidth</b>	number > 0	<b>Optional, default: value of property BarHeight</b> – Width of the symbol defined in property BarShapeSymbolID when the bar shape named Symbol is used.
<b>BarText</b>	string	<b>Optional, default: undefined</b> – Text to display in the bar.
<b>BarTextColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Color for the texts of the bar.
<b>BarTextFormat</b>	string	<p><b>Optional, default: value of option defaultAllocationBarTextFormat</b> – This property describes the format of the bar text. If not set, then the value of property BarText is displayed.</p> <p>This string contains the placeholders for object values surrounded by double curly braces {{ }}. For example, based on the following string a tooltip with a table containing three rows of key-value pairs is created, where the values are taken from the properties "name" and "firstName" of the referenced object:</p> <pre>{{name}}, {{firstName}}</pre> <p>As an escape, the use of three open curly braces {{{ are displayed as {{.</p> <p>Additionally, the property name can be extended to contain the desired property type as in {{Start:date}}. At the moment only the types 'date' and 'number' are possible besides 'string' (other property types are converted automatically with toString()). The type 'date' converts date values by default using the same format as other dates in the timescale and at the dragging date line captions. You can add another colon followed by a format name, that is defined by the options intlDateTimeFormatOptionsMap or intlNumberFormatOptionsMap, resp.</p> <p>The referenced object is the object on which the tooltip will be shown. For period highlighter entries and allocation entries the referenced object is the main object and not the entry object.</p> <p>It is possible to access related objects by using the following keywords within the property accessor string:</p> <ul style="list-style-type: none"> <li>• On allocations: &gt;Activity, &gt;Resource</li> <li>• On activities: &gt;Parent, &gt;Calendar</li> <li>• On resources: &gt;Parent, &gt;Calendar, &gt;LoadCurve, &gt;CapacityCurve</li> </ul>

Allocation Property Name	Type	Description
		<ul style="list-style-type: none"> <li>Additionally on resources in SkilledResources view: &gt;Skill</li> </ul> <p>It is also possible to access variables that are defined by the option applicationVariablesMap by using <i>?variableName</i>.</p> <p>If the value reached is an object, you can then access a property value by using a prefixed dot: <i>.propertyName</i> and you can use [...] to access a property value, a map entry or an array entry. Within [...] you can use a literal like 5 or A (with or without quotes) or even curly braces {...} with the same rules as above.</p>
<b>BarTextPrefixSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the bar symbol before the text (see property BarTextSymbolSymbolID) in pixels at a zoom factor of 100%.
<b>BarTextPrefixSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown before the text inside of the allocation bar. The symbol will be shown vertically centered inside the bar.
<b>BarTextPrefixSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the bar symbol before the text (see property BarTextPrefixSymbolID) in pixels at a zoom factor of 100%.
<b>BarTextWrapMode</b>	number (see enum <a href="#">TextWrapMode</a> )	<b>Optional, default: TextWrapMode.None</b> – Specifies whether the text inside the bar is wrapped.
<b>BarTooltipTemplateID</b>	string	<p><b>Optional, default: value of option defaultAllocationBarTooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the allocation bars.</p>
<b>BarTopOffset</b>	number	<b>Optional, default: 0</b> – Offset of the bar in pixels relative to its upper side. A negative number will shift the bar upwards, a positive number will shift the bar downwards.
<b>BorderColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: "gray" *</b> – Color for the border of the bar.</p> <p>If set to "calculated", a color will be calculated using the color value of the property Color. This can be useful in situations where two bars are positioned next to each other, and a graphical indicator is needed to visually distinguish the two bars.</p> <p>* If the option decouplingOfAllocationPropertiesFromActivities is set</p>

Allocation Property Name	Type	Description
		to false, then the default value is the current value of property <code>BorderColor</code> of corresponding activity.
<b>BorderDashArray</b>	string	<p><b>Optional, default: "none"</b> – Pattern of dashes and gaps for drawing the border of the bar. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a>. The value "none" indicates that no dashing is used. In this case, the border is drawn solid.</p>
<b>Color</b>	string (CSS color value)	<p><b>Optional, default: "#646464" *</b> – Fallback color for the entries of the bar, see property <code>Color</code> of <code>AllocationEntry</code> objects.</p> <p>See also property <code>NonworkingTimeColor</code> for the coloring of the nonworking times.</p> <p>* If the option <code>decouplingOfAllocationPropertiesFromActivities</code> is set to false, then the default value is the current value of property <code>Color</code> of the referenced activity.</p>
<b>EarliestDragStart</b>	Date   string	<p><b>Optional, default: undefined</b> – If set, then the time before the given date is grayed, when beginning to drag the allocation bar. If the option <code>dragDatesLimitingInteraction</code> is set to true, then the bar itself cannot be dragged before the date.</p>
<b>EarliestEnd</b>	Date   string	<p><b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>EarliestEndColor</b>	string (CSS color value)	<p><b>Optional, default: value in option <code>defaultAllocationConstraintSymbolColor</code></b> – Color for the <code>EarliestEnd</code> constraint symbol.</p>
<b>EarliestStart</b>	Date   string	<p><b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>

Allocation Property Name	Type	Description
<b>EarliestStartColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultAllocationConstraintSymbolColor</b> – Color for the EarliestStart constraint symbol.
<b>EndIsSnapTarget</b>	boolean	<b>Optional, default: true</b> – If set to true, then the visible end date of this allocation in the resources view is used as a snap target for a dragged bar (see attributes SnapTargetsForStart and SnapTargetsForEnd and option maximumSnapDistance)
<b>Entries</b>	<a href="#">AllocationEntry</a> []	<b>Optional, default: undefined</b> – array of allocation entries.
<b>ID</b>	string	<b>Required</b> – Identifier of the allocation. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>LatestDragEnd</b>	Date   string	<b>Optional, default: undefined</b> – If set, then the time after the given date is grayed, when beginning to drag the allocation bar. If the option dragDatesLimitingInteraction is set to true, then the bar itself cannot be dragged after the date.
<b>LatestEnd</b>	Date   string	<p><b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>LatestEndColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultAllocationConstraintSymbolColor</b> – Color for the LatestEnd constraint symbol.
<b>LatestStart</b>	Date   string	<p><b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>LatestStartColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultAllocationConstraintSymbolColor</b> – Color for the LatestStart constraint symbol.
<b>LeftBarSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the left bar symbol (see property LeftBarSymbolID) in pixels at a zoom factor of 100%.
<b>LeftBarSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the left side of the allocation bar. The

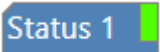
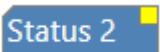
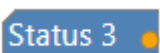
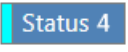
Allocation Property Name	Type	Description
		symbol will be shown vertically centered inside the bar. See also RightBarSymbolID, LeftBarSymbolHeight, and LeftBarSymbolWidth.
<b>LeftBarSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the left bar symbol (see property LeftBarSymbolID) in pixels at a zoom factor of 100%.
<b>LinkSourceDate</b>	Date   string	<b>Optional, default: undefined</b> – Additional date serving as an additional “start point” to connect a link. See also property “RelationType” of link object.
<b>LinkTargetDate</b>	Date   string	<b>Optional, default: undefined</b> – Additional date serving as an additional “end point” to connect a link. See also property “RelationType” of link object.
<b>MinimumRowHeight</b>	number	<b>Optional, default: value in option defaultAllocationMinimumRowHeight</b> – Minimum height of the allocation row in pixels. This attribute is useful when more than one line of text is shown inside the table cells. Proposal: For one line take 36*, for two lines 52, for three lines 68, and so on. To have the same height also, when no bar is placed in the row, take the maximum bar height adding 20 (e.g. 42) as minimum.  For using word wrapping in table cells, it is necessary to use a table row definition by setting the property TableRowDefinitionID and setting the property WrapMode in a contained table cell definition.
<b>MustEndOn</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>MustEndOnColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultAllocationConstraintSymbolColor</b> – Color for the MustEndOn constraint symbol.
<b>MustStartOn</b>	Date   string	<b>Optional, default: undefined</b> – If defined, an additional symbol will be displayed to indicate this date.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.



Allocation Property Name	Type	Description
<b>MustStartOnColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultAllocationConstraintSymbolColor</b> – Color for the MustStartOn constraint symbol.
<b>NonworkingTimeColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: undefined</b> – Color for the nonworking time periods of the bar.</p> <p>If set to "calculated", a color will be calculated using the color defined by the Color property.</p> <p>If undefined, nonworking times are colored like working times (taking the value of property Color).</p>
<b>PredictedEnd</b>	Date   string	<p><b>Optional, default: undefined</b> – A date that indicates the predicted end of the allocation. This date is used to display a bar between this date and the end of the allocation.</p> <p>If data type is <i>String</i>, then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.</p>
<b>PredictedEndColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Color for the predicted end bar.
<b>Progress</b>	number (floating point; in percent; $\geq 0$ , $\leq 100$ )	<p><b>Optional, default: 0.0 *</b> – Used to display a completion layer.</p> <p>* If the option decouplingOfAllocationPropertiesFromActivities is set to false, then the default value is the current value the property of same name in the reference activity.</p>
<b>ProgressBackgroundColor</b>	string (CSS color value)	<b>Optional, default: value in option defaultAllocationProgressBackgroundColor</b> – Color for the background of the progress bar region.
<b>ProgressColor</b>	string (CSS color value)	<p><b>Optional, "#646464" *</b> – Color for the working time periods of the progress bar.</p> <p>* If the option decouplingOfAllocationPropertiesFromActivities is set to false, then the default value is the current value the property of same name in the reference activity.</p>
<b>ProgressNonworking-TimeColor</b>	string (CSS color value or "calculated")	<p><b>Optional, default: undefined</b> – Color for the nonworking time periods of the progress bar.</p> <p>If set to "calculated", a color will be calculated using the color defined by the ProgressColor property.</p>

Allocation Property Name	Type	Description
		If undefined, then the nonworking times are colored like working times (taking the value of property <code>ProgressColor</code> ).
<b>ResourceID</b>	string	<b>Optional, default: undefined</b> – Identifier of a <a href="#">Resource</a> .
<b>RightBarSymbolHeight</b>	number	<b>Optional, default: 12</b> – Height of the right bar symbol (see property <code>RightBarSymbolID</code> ) in pixels at a zoom factor of 100%.
<b>RightBarSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the right side of the allocation bar. The symbol will be shown vertically centered inside the bar. See also <code>LeftBarSymbolID</code> , <code>RightBarSymbolHeight</code> , and <code>RightBarSymbolWidth</code> .
<b>RightBarSymbolWidth</b>	number	<b>Optional, default: 12</b> – Width of the right bar symbol (see property <code>RightBarSymbolID</code> ) in pixels at a zoom factor of 100%.
<b>RowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default: value in option <code>defaultAllocationRowDesign</code></b> – Specifies how the time area is filled when the row is visible.  See explanation for possible values in enumeration chapter.
<b>RowSelectable</b>	boolean	<b>Optional, default: value of option <code>defaultAllocationRowSelectable</code></b> – If set to true, then the row representing this allocation will be selectable.
<b>RowSymbolColumnBackgroundColor</b>	string (CSS color value)	<b>Optional, default: value of property <code>SymbolColumnBackgroundColor</code> of assigned table row definition or option <code>symbolColumnBackgroundColor</code></b> – Determines the color of the symbol column within this table row.
<b>RowSymbolIDs</b>	string[]	<b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the table symbol cell of the beginning of the table row. The symbols will be arranged one below the other. However, if the cell is not high enough to hold all symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.  An empty string ("" ) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve space for a symbol that may be shown at a later time.  Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.

Allocation Property Name	Type	Description
<b>RowTooltipTemplateID</b>	string	<p><b>Optional, default: value of option defaultAllocationRowTooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the entity table rows.</p>
<b>SkilledBarTooltipTemplateID</b>	string	<p><b>Optional, default: value of option defaultSkilledAllocationBarTooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the allocation bars in skilled resources view. It fallbacks to evaluation of the property BarTooltipTemplateID if not set.</p>
<b>SkilledRowTooltipTemplateID</b>	string	<p><b>Optional, default: value of option defaultSkilledAllocationRowTooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the entity table rows. It fallbacks to evaluation of the property RowTooltipTemplateID if not set.</p>
<b>SkillID</b>	string	<p><b>Optional, default: undefined</b> – When set to a valid ID of a Skill object, this influences the appearance of the bar within the resource rows in the skilled resources view.</p>
<b>SnapTargetsForEnd</b>	number (see enum <a href="#">SnapTargets</a> )	<p><b>Optional, default: value of widget option defaultAllocationSnap-TargetsForEnd</b> – When dragging horizontally, then the visible end date of this allocation will optionally be snapping to date lines, calendar grids, and start or end dates of other allocations in same row, when dragging lets these dates get near the end date (see attribute EndIsSnapTarget). The user can override an active snapping by pressing the ALT key while dragging.</p>
<b>SnapTargetsForStart</b>	number (see enum <a href="#">SnapTargets</a> )	<p><b>Optional, default: value of widget option defaultAllocationSnap-TargetsForStart</b> – When dragging horizontally, then the visible start date of this allocation will optionally be snapping to date lines, calendar grids, and start or end dates of other allocations in same row, when dragging lets these dates get near the start date (see attribute StartIsSnapTarget). The user can override an active snapping by pressing the ALT key while dragging.</p>
<b>SortCode</b>	number string Date	<p><b>Optional, default: undefined</b> – If set, then the value will be used when sorting allocation rows. The value type can be anyone that can be compared using JavaScript.</p>

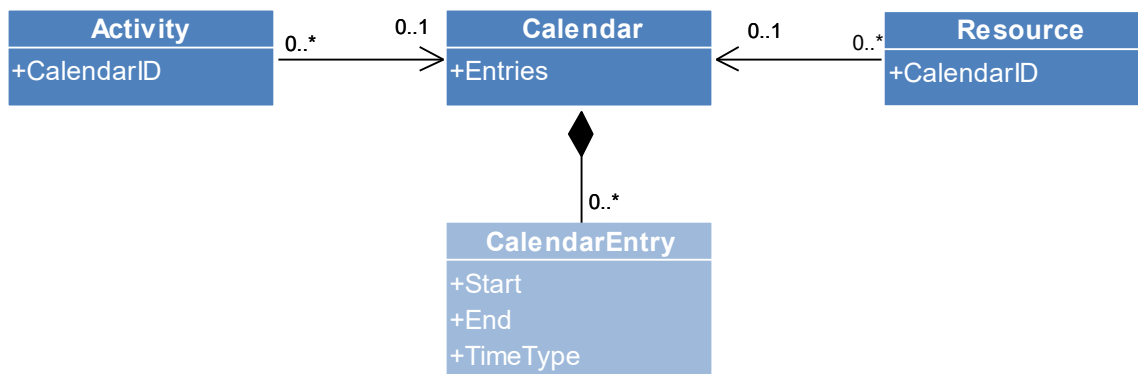
Allocation Property Name	Type	Description
		See also options allocationRowSortMode and allocationRowSortCodePropertyName.
<b>StartIsSnapTarget</b>	boolean	<b>Optional, default: true</b> – If set to true, then the visible start date of this allocation in the resources view is used as a snap target for a dragged bar (see attributes SnapTargetsForStart and SnapTargetsForEnd and option maximumSnapDistance)
<b>Status1Color</b>	string (CSS color value)	<b>Optional, default: undefined</b> – Color for the status symbol to the right of the bar. If undefined, no symbol appears. Only visible, when property Status1Visible is true.
<b>Status1Visible</b>	boolean	<b>Optional, default: false</b> – If set to true and the corresponding status color is set in property Status1Color, then a predefined symbol is displayed to the right of the bar. 
<b>Status2Color</b>	string (CSS color value)	<b>Optional, default: undefined</b> – Color for the status symbol to the right of the bar. If undefined, no symbol appears. Only visible, when property Status2Visible is true.
<b>Status2Visible</b>	boolean	<b>Optional, default: false</b> – If set to true and the corresponding status color is set in property Status2Color, then a predefined symbol is displayed to the right of the bar. 
<b>Status3Color</b>	string (CSS color value)	<b>Optional, default: undefined</b> – Color for the status symbol to the right of the bar. If undefined, no symbol appears. Only visible, when property Status3Visible is true.
<b>Status3Visible</b>	boolean	<b>Optional, default: false</b> – If set to true and the corresponding status color is set in property Status3Color, then a predefined symbol is displayed to the right of the bar. 
<b>Status4Color</b>	string (CSS color value)	<b>Optional, default: undefined</b> – Color for the status symbol to the left of the bar. If undefined, no symbol appears. Only visible, when property Status4Visible is true.
<b>Status4Visible</b>	boolean	<b>Optional, default: false</b> – If set to true and the corresponding status color is set in property Status4Color, then a predefined symbol is displayed to the left of the bar.  <b>Note: This property may be used with rectangle bar shapes only!</b> 

Allocation Property Name	Type	Description
StatusFrameColor	string	<b>Optional, default: value of option defaultAllocationStatusFrameColor</b> – Color for the status frame that will be shown when property StatusFrameVisible is set.
StatusFrameVisible	boolean	<b>Optional, default: false</b> – If set to true, then a frame is shown around the bar. See also property StatusFrameColor.
SuitableActivityIDs	string[]	<b>Optional, default: undefined</b> – An array of IDs of those activities to which the allocation could be assigned. If the array is defined, then all rows of activities that are not listed in that array will be covered by a half-transparent curtain. If the array is empty, all activity rows will be covered. If the array is not defined, then all rows are displayed in the normal way without any covering.  Also see options suitableActivityOverlayColor and unsuitableActivityOverlayColor.
SuitableResourceIDs	string[]	<b>Optional, default: undefined</b> – An array of IDs of those resources to which the allocation could be assigned. If the array is defined, then all rows of resources that are not listed in that array will be covered by a half-transparent curtain. If the array is empty, all resource rows will be covered. If the array is not defined, then all rows are displayed in the normal way without any covering.  Also see options suitableResourcesOverlayColor and unsuitableResourcesOverlayColor.
TableText	string	<b>Optional, default: undefined</b> – Text to display in the table row (see also property TableRowDefinitionID).
TextColor		<b>Deprecated! Renamed to BarTextColor.</b>
TopLeftBarSymbolID	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the top left side of the allocation bar. Please note: A symbol will be resized to an image with a width and height of 12 pixels each at a zoom level of 100%.
TopRightBarSymbolID	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown at the top right side of the allocation bar. Please note: A symbol will be resized to an image with a width and height of 12 pixels each at a zoom level of 100%.

### 3.2.2 AllocationEntry

AllocationEntry Property Name	Type	Description
<b>Color</b>	string (CSS color value)	<b>Optional, default: value of property Color of allocation</b> – Color for the working time periods of the bar. If undefined, the value of the corresponding allocation, if available, will be used.
<b>End</b>	Date string	<b>Optional, default: undefined</b> – End date of the allocation entry. This date itself is not(!) part of the interval described by this entry.  If data type is <i>String</i> , then the value has to be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC).
<b>Height</b>	number ( $\geq 0, \leq 1000$ )	<b>Optional, default: value in option defaultAllocationBarHeight</b> – Height of the entry in pixels.
<b>NonworkingTimeColor</b>	string (CSS color value or "calculated")	<b>Optional, default: value of property NonworkingTimeColor of allocation or if undefined then value of property Color</b> – Color for the nonworking time periods of the bar. If set to "calculated", a color will be calculated using the color defined by the Color property.
<b>PatternColor</b>	string	<b>Optional, default: "white"</b> – Color for the pattern when this is visible by using property PatternColor.
<b>PatternType</b>	number (see enum <a href="#">PatternType</a> )	<b>Optional, default: None</b> – If set, then a pattern is shown on top of the fill color and behind the text.
<b>RelativeTopOffset</b>	number	<b>Optional, default: 0</b> – Offset of the entry in pixels relative to the upper side of the corresponding allocation. A negative number will shift the entry upwards, a positive number will shift the entry downwards.
<b>Start</b>	Date string	<b>Optional, default: undefined</b> – Start date of the allocation entry.  If data type is <i>String</i> , then the value has to be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC).

### 3.3 Calendars



#### 3.3.1 Calendar

A Calendar object defines working and non-working times to be used with resources.

Calendar Property Name	Type	Description
<b>Entries</b>	<a href="#">CalendarEntry[]</a>	<b>Optional, default: undefined</b> – Array of calendar entry objects. The order of the entries inside the array is important! If undefined, the calendar consists of non-working times only.
<b>ID</b>	string	<b>Required</b> – Identifier of the calendar. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.

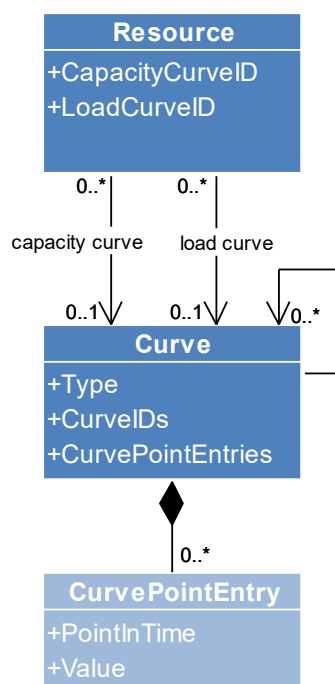
#### 3.3.2 CalendarEntry

A CalendarEntry object defines a single time period. It has to be referenced in the Entries array of a Calendar object. If several calendar entries describe the same time period, then the last entry wins.

CalendarEntry Property Name	Type	Description
<b>End</b>	Date   string	<b>Optional, default: undefined</b> – End of the working time period.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>Start</b>	Date   string	<b>Optional, default: undefined</b> – Start of the working time period.  If data type is <i>String</i> , then the value should be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is

CalendarEntry Property Name	Type	Description
		specified in UTC). Since the browsers do not interpret every formatted date string in a standardized way, one has to be careful about it.
<b>TimeType</b>	number (see enum <a href="#">TimeType</a> )	<b>Optional, default: 1</b> 1: WorkingTime, 2: NonworkingTime

### 3.4 Curves



Curve objects serve to define values over time that can be shown as capacity or load inside resource and activity rows (see properties **LoadCurveID** and **CapacityCurveID** in Resource object). Additionally, it is possible to stack curves when using curve object of stack type. Currently, there are no curve types that calculate their values automatically.

Curves are displayed in curves panes. Each pane can hold several curves (e.g. a capacity and a load curve). On the right side of a pane a numerical scale is displayed. By default, the minimum and maximum values for the tick markers are calculated automatically. However, this can be affected by setting the **ScaleMinimumValue** and the **ScaleMaximumValue** properties of the curves.



### 3.4.1 Curve

Curve Property Name	Type	Description
<b>CurveIDs</b>	string[]	<b>Optional, default: undefined</b> – Array of curve IDs (in case of CurveStack and CurveList only).
<b>CurvePointEntries</b>	<a href="#">CurvePointEntry</a> []	<b>Optional, default: undefined</b> – Array of point entries (in case of PointCurve only).
<b>FillColor</b>	string (CSS color value)	<b>Optional, default: "transparent"</b> – Color of the area below the curve.  Note: If a curve is used as an inventory curve, then the default is "transparent"
<b>ID</b>	string	<b>Required</b> – Identifier of the curve. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>InterpolationType</b>	number (see enum <a href="#">CurveInterpolationType</a> )	<b>Optional, default: StepAfter</b> – Type of interpolation.  At the moment there are restrictions concerning putting curves of linear interpolation type into curve stacks. It is recommended to use this interpolation type only inside curve lists.
<b>OverloadColor</b>	string (CSS color value)	<b>Optional, default: "#E01818"</b> – Used, when the curve is used as the load curve that referenced directly by the property LoadCurveID at the object. Then the area above the capacity curve will be colored by this color when the load is higher than the capacity.
<b>ScaleMaximumValue</b>	number	<b>Optional, default: –Infinity</b> – If this value here is greater than all values of this curve, then it defines the explicit maximum. Otherwise, the maximum of all curve values defines the overall maximum of this curve.  Hereby together with the property ScaleMinimumValue you can define the shown value range for the curve pane.
<b>ScaleMinimumValue</b>	number	<b>Optional, default: Infinity</b> – If this value here is lower than all values of this curve, then it defines the explicit minimum. Otherwise, the minimum of all curve values defines the overall minimum of this curve.  Hereby together with the property ScaleMaximumValue you can define the shown value range for the curve pane.

Curve Property Name	Type	Description
<b>StrokeColor</b>	string (CSS color value)	<b>Optional, default: "transparent"</b> – Color of the curve line itself.
<b>StrokeDashArray</b>	string (SVG stroke dash array value)	<b>Optional, default: "none"</b> – Pattern of dashes and gaps for drawing the curve line. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the line is drawn solid.
<b>Type</b>	number (see enum <a href="#">CurveType</a> )	<p><b>Optional, default: 0</b> – Type of the curve.</p> <p>A point curve contains a number of entries in the property CurvePointEntries.</p> <p>A curve stack and a curve list contains a number of IDs of other curves in the property CurveIDs. A curve stacks stacks the contained curves in the order inside the array optically.</p> <p>A curve list shows the contained curves one by one at the same space, so it is recommended to use translucent colors for filling the curves.</p> <p>Currently it is recommended not to put lists or stacks into other lists/stacks!</p>

### 3.4.2 CurvePointEntry

CurvePointEntry Property Name	Type	Description
<b>PointInTime</b>	Date   string	<p><b>Required</b> – This property serves as an identifier of the point entry.</p> <p>If data type is <i>String</i>, then the value has to be formatted this way: "YYYY-MM-DDThh:mm:ssZ" (this implies that the date is specified in UTC).</p>
<b>Value</b>	number (floating point)	<b>Optional, default: 0.0</b> – Value of the curve at the given point in time.

### 3.5 DateLine

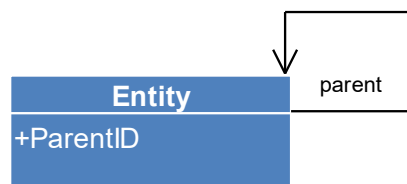
Date Line
+PointInTime
+Caption
+SymbolID

A DateLine object is a pure presentation object and defines the properties of a single date line.

DateLine Property Name	Type	Description
<b>Caption</b>	string	<b>Optional, default:</b> "" – Text for the caption of the date line.
<b>CaptionColor</b>	string (CSS color value)	<b>Optional, default:</b> "black" – Color of the caption.
<b>CaptionOrientation</b>	number (see enum <a href="#">DateLineCaptionOrientation</a> )	<b>Optional, default:</b> 2 – Specifies whether the caption should be oriented vertically or horizontally.  <b>Note:</b> If the caption position is TopCenter, TopLeft or TopRight (see the CaptionPosition property) and if the orientation is vertical, the caption orientation will still be positioned at the date line and not within the timescale.
<b>CaptionPosition</b>	number (see enum <a href="#">DateLineCaptionPosition</a> )	<b>Optional, default:</b> 1 – Specifies where the caption should be positioned relative to the date line.  See also property CaptionOrientation and option dateLineCaptionOptimizedPositioningEnabled.
<b>Color</b>	string (CSS color value)	<b>Optional, default:</b> "black" – Color of the line.
<b>DashArray</b>	string	<b>Optional, default:</b> "none" – Pattern of dashes and gaps for drawing the date line. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the line is drawn solid.
<b>Draggable</b>	boolean	<b>Optional, default:</b> false – If set to true, then the date line is draggable and the callback onDrop is triggered, when dropping it at a new date.
<b>ID</b>	string	<b>Required</b> – Identifier of this date line. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.

DateLine Property Name	Type	Description
<b>InFrontOfBars</b>	boolean	<b>Optional, default: true</b> – Determines how the date line is displayed. If set to false, the date line will be overlapped by the bars. Otherwise, the line will be displayed in front of the bars.
<b>PointInTime</b>	Date string	<b>Optional, default: undefined</b> – Date, where the date line should become visible. The date line only gets visible, when the date is set and the date lies between the values of the widget options start and end.
<b>SymbolHeight</b>	number > 0	<b>Optional, default: 12</b> – Height of the symbol referenced by property SymbolID.
<b>SymbolID</b>	string	<b>Optional, default: undefined</b> – When set, then the referenced symbol will be visible at the top of the date line. See also properties SymbolHeight/Width.
<b>SymbolWidth</b>	number > 0	<b>Optional, default: 12</b> – Width of the symbol referenced by property SymbolID.
<b>Width</b>	number ≥ 0	<b>Optional, default: 1</b> – Line width of the date line.

### 3.6 Entity



An Entity object defines the properties of a single entity. Entities are shown in a separate table on the right side.

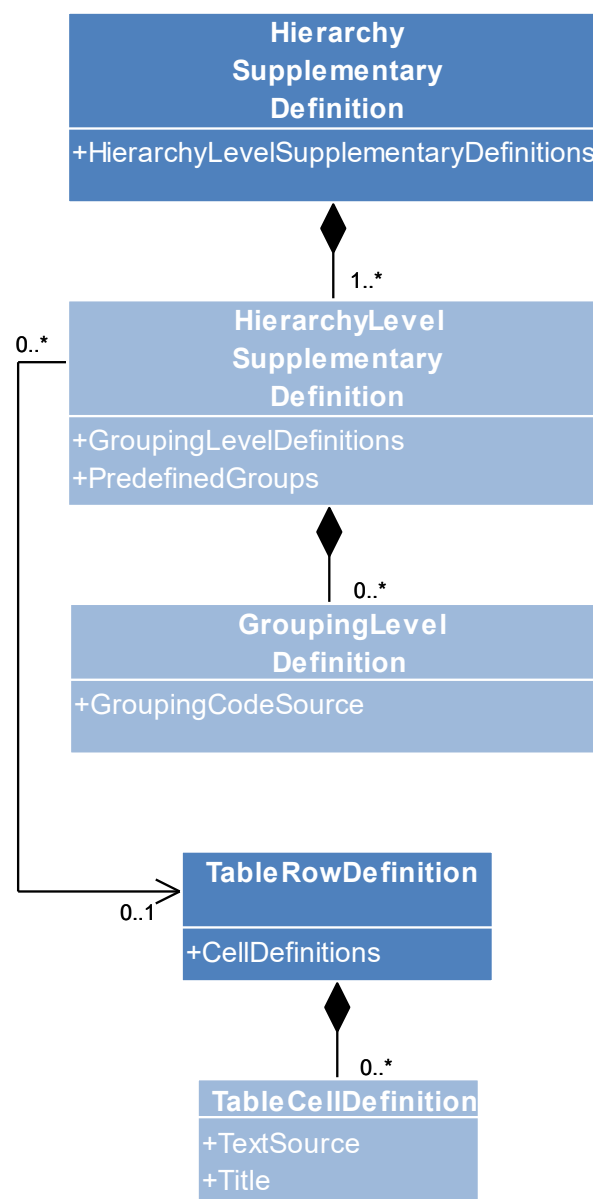
Entity Property Name	Type	Description
<b>AllowedRowDrag-Modes</b>	number (see enum <a href="#">RowDragModes</a> )	<b>Optional, default: value of option defaultEntityAllowedRowDragModes</b> – This option determines the allowed row drag modes for this entity when the entities table is visible (these can be overwritten using the callback canDrag).
<b>CollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: -1</b> – Specifies whether the row of the entity should be expanded or collapsed when displayed the very first time.
<b>Duration</b>	number (in milliseconds)	<b>Optional, default: undefined</b> – Duration of the pure working time of the entity. This property is used, for example, when moving the entity from the entities table to the Gantt diagram to display a bar of correct length during interaction.
<b>HasChildren</b>	boolean	<b>Optional, default: false</b> – If set to true, then the row representing this entity will be

Entity Property Name	Type	Description
		collapsible/expandable even when there are no children defined. This serves for lazy loading.
<b>ID</b>	string	<b>Required</b> – Identifier of this entity. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>MinimumRowHeight</b>	number	<p><b>Optional, default: value in option defaultMinimumEntityRowHeight</b> – Minimum height of the entity row in pixels. This attribute is useful, when more than one line of text is shown inside the table cells. Proposal: For one line take 36*, for two lines 52, for three lines 68, and so on. In order to have the same height also, when no bar is placed in the row, take the maximum bar height adding 20 (e.g. 42) as minimum.</p> <p>For using word wrapping in table cells, it is necessary to use a table row definition by setting the property <code>TableRowDefinitionID</code> and setting the property <code>WrapMode</code> in a contained table cell definition.</p>
<b>ParentID</b>	string	<p><b>Optional, default: undefined</b> – Identifier of the parent of the entity. This serves for setting up a hierarchy of entities.</p> <p>If this property is undefined the current entity will be considered as a root node of the entity hierarchy.</p> <p>We recommend using only a low number of hierarchy levels and we do not guarantee correct function beyond approx. 100 levels including hierarchy levels created by using <code>Hierarchy-SupplementaryDefinitions</code>.</p>
<b>RowCollapsible</b>	boolean	<b>Optional, default: value of option defaultEntityRowCollapsible</b> – If set to true, then the row representing this entity will be interactively collapsible when children exist.
<b>RowSelectable</b>	boolean	<b>Optional, default: value of option defaultEntityRowSelectable</b> – If set to true, then the row representing this entity will be selectable.
<b>RowSymbolColumn-BackgroundColor</b>	string (CSS color value)	<p><b>Optional, default: value of property SymbolColumnBackgroundColor of assigned table row definition or option entitiesTableSymbolColumnBackgroundColor</b> – Determines the color of the symbol column within this table row.</p>
<b>RowSymbolIDs</b>	string[]	<p><b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the table symbol cell of the beginning of the table row.</p> <p>The symbols will be arranged one below the other. However, if the cell is not high enough to hold all</p>

Entity Property Name	Type	Description
		<p>symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.</p> <p>An empty string (“”) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve space for a symbol that may be shown at a later time.</p> <p>Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.</p>
<b>RowTooltipTemplateID</b>	string	<p><b>Optional, default: value of option defaultEntityRowTooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the entity table rows.</p>
<b>SuitableActivityIDs</b>	string[]	<p><b>Optional, default: undefined</b> – An array of IDs of those activities to which the entity could be dropped.</p> <p>If the array is defined, then all rows of activities that are not listed in that array will be covered by a half-transparent curtain. If the array is empty, all activity rows will be covered. If the array is not defined, then all rows are displayed in the normal way without any covering.</p> <p>Also see options suitableActivityOverlayColor and unsuitableActivityOverlayColor and property SuitableActivityIDs of Allocation objects.</p>
<b>SuitableResourceIDs</b>	string[]	<p><b>Optional, default: undefined</b> – An array of IDs of those resources to which the entity could be dropped.</p> <p>If the array is defined, then all rows of resources that are not listed in that array will be covered by a half-transparent curtain. If the array is empty, all resource rows will be covered. If the array is not defined, then all rows are displayed in the normal way without any covering.</p> <p>Also see options suitableResourcesOverlayColor and unsuitableResourcesOverlayColor and property SuitableResourceIDs of Allocation objects.</p>
<b>TableColor</b>	string (CSS color value)	<p><b>Optional, default: level-dependent gray</b> – Color for the table row. If undefined, a default value of the widget will be used.</p>

Entity Property Name	Type	Description
<b>TableRowDefinitionID</b>	string	<b>Optional, default: value of option defaultEntityTableRowDefinitionID</b> – Identifier of a TableRowDefinition object that defines the composition of the table row.
<b>TableText</b>	string	<b>Optional, default: undefined</b> – Text to display in the table (see also property TableRowDefinitionID).
<b>TableTextColor</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Color for the table row texts. If undefined, a default value of the widget will be used.

### 3.7 HierarchySupplementaryDefinitions



### 3.7.1 HierarchySupplementaryDefinition

A HierarchySupplementaryDefinition object defines the additional grouping of a complete hierarchy of row objects by using criteria that are taken from property values within the row objects. Each HierarchyLevelSupplementaryDefinition object defines all groupings for a hierarchy level of row objects. The hierarchy levels are already built by using the property ParentID of the row objects of type Activity, Entity, Resource. The ID of a HierarchySupplementaryDefinition object is set into one of the options activity/entity/resourceHierarchySupplementaryDefinitionID. Currently the skilled resources view cannot be grouped.

A first simple definition for the first hierarchy level with only one grouping level consists of only a few set properties:

```
{
  "ID": "HSD1",
  "HierarchyLevelSupplementaryDefinitions": [
    // hierarchy level 0
    {
      "GroupingLevelDefinitions": [
        // first grouping level
        {
          "GroupingCodeSource": "_Grp1", // e.g., referencing an application-
            // defined property
          "TableBackgroundColor": "gold"
        }
      ]
    }
  ]
}
```

HierarchySupplementaryDefinition Property Name	Type	Description
<b>HierarchyLevel-SupplementaryDefinitions</b>	HierarchyLevel-SupplementaryDefinition[]	<b>Optional, default: undefined</b> – Array of hierarchy level supplementary definitions. The first object contains definitions for hierarchy level 0, the second defines level 1 and so on. If one hierarchy level should not be grouped additionally, then you can leave the array entry empty either by using an empty object or alternatively undefined/null.
<b>ID</b>	string	<b>Required</b> – Identifier of this hierarchy definition. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.



### 3.7.2 HierarchyLevelSupplementaryDefinition

Each HierarchyLevelSupplementaryDefinition object defines additional grouping for a level of row objects. Used in HierarchySupplementaryDefinition objects.

HierarchyLevel-SupplementaryDefinition Property Name	Type	Description
<b>GroupingLevelDefinitions</b>	GroupingLevel-Definition[]	<b>Optional, default: undefined</b> – Array of grouping level definitions. By using more than one GroupingLevelDefinition object in the array you can define multiple grouping criteria for one hierarchy level at once.
<b>InitiallyCollapsed</b>	boolean	<b>Optionally, default: false</b> – If this property is set to true, then the table rows in this hierarchy level will initially show collapsed. This property is only effective when the property CollapseState is not set to 0 or 1 on the affected table row object (Activity/Entity/Resource).
<b>PredefinedGroups</b>	Object[]	<p><b>Optionally, default: undefined</b> – If this property is set to an array, then each object in the array defines one or more predefined groups.</p> <p>In each object you can define which predefined group(s) to create by including some or all grouping codes up to the current hierarchy level. Additionally, you can set the following graphical properties for the predefined group: MinimumRowHeight, TableColor, TableTextColor.</p> <p>Example for second hierarchy level:</p> <pre>{   "GroupingCodeSourceOfLevel0": "a0",   "GroupingCodeSourceOfLevel1": "b1",   "TableColor": "lime" }</pre> <p>The property names in italics have to be replaced by the property names in your hierarchy level definitions on levels 0 and 1! This example defines a predefined group with code "b1" within another group with the code "a0". The group "a0" may be predefined or not. If it does not exist, it will be created additionally before creating "b1" using default coloring.</p> <p>In general, predefined groups are created before any other groups that are determined</p>

HierarchyLevel-SupplementaryDefinition Property Name	Type	Description
		by grouping codes within the grouped row objects.
<b>TableRowDefinitionID</b>	string	<b>Optional, default: undefined</b> – If set then the value overwrites the setting on an affected table row object (Activity/Entity/Resource).

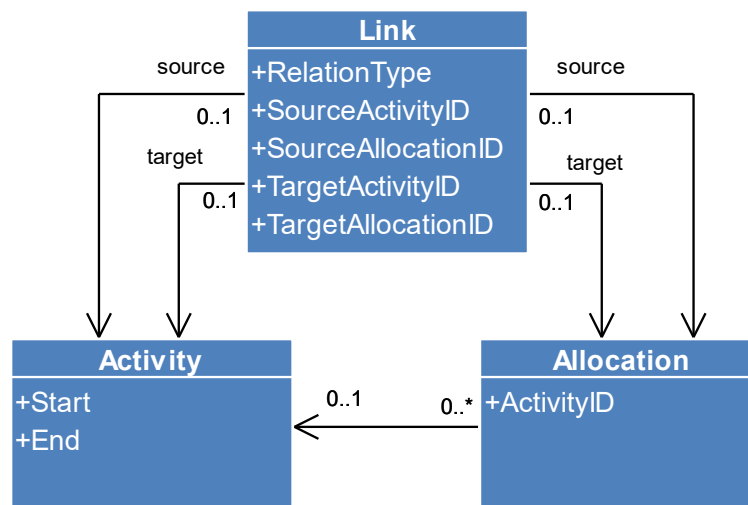
### 3.7.3 GroupingLevelDefinition

The GroupingLevelDefinition object defines the grouping criteria for all grouping levels of one hierarchy level as seen in a table on screen. The grouping level definition also defines the display of the resulting group lines. Used in a HierarchyLevelSupplementaryDefinition object.

GroupingLevelDefinition Property Name	Type	Description
CodeToTextMap		<b>Deprecated! Please use GroupingCodeToTextMap instead.</b>
DefaultCode		<b>Deprecated! Please use DefaultGroupingCode instead.</b>
<b>DefaultGroupingCode</b>	string	<b>Optional, default: ""</b> – If this property is set, the value serves as a default grouping criterion, if not otherwise defined.
<b>GroupingCodeSource</b>	string	<b>Optional, default: ""</b> – Names a property of the objects under consideration, the content of which is used as a grouping criterion.
<b>GroupingCodeToTextMap</b>	Object	<b>Optional, default: undefined</b> – If set then the object is used to map the group codes (key) to a long text (value) that will be shown on the table row representing the generated group.
<b>InitiallyCollapsed</b>	boolean	<b>Optional, default: false</b> – If this property is set to true, then the generated group rows initially are shown collapsed.
<b>MinimumRowHeight</b>	number > 0	<b>Optional, default: value of appropriate default option for the table</b> – Minimum height of the group row in pixels. This attribute is useful, when more than one line of text is shown inside the table cells. Proposal: For one line take 36, for two lines 52, for three lines 68, and so on. To have the same height also, when no bar is placed in the row, take the maximum bar height adding 20 (e.g. 42) as minimum.
TableBackgroundColor		<b>Deprecated! Please use TableColor instead.</b>

GroupingLevelDefinition Property Name	Type	Description
<b>TableColor</b>	string (CSS color value)	<b>Optional, default: value of the higher group row</b> – Color for the table row.
<b>TableColorVisibleInTimeArea</b>	boolean	<b>Optional, default: false</b> – If set to true, the time area row will be colored using the color defined by the TableColor property.
<b>TableTextColor</b>	string (CSS color value)	<b>Optional, default: value of the higher group row</b> – Color for the table row texts.
<b>TableTextFormat</b>	string	<b>Optionally, default: undefined</b> – If this property is set, then text in table on generated group rows is formatted using the value. The string can contain the keyword <code>{{&gt;MapText}}</code> to include the value of the map that is defined by the property GroupingCodeToTextMap else will show the group code itself.

### 3.8 Link



A Link object defines the properties of a single link between activities or or allocations.

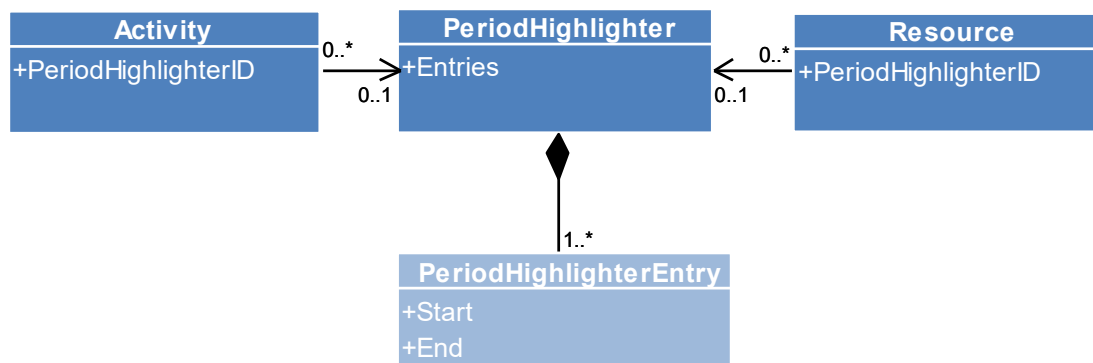
Links between activities are shown in activities view if the option `linksVisibleInActivitiesView` is true. Additionally, links between allocations are shown in activities view if the two options `definedAllocationLinksVisibleInActivitiesView` and `allocationRowsVisibleInActivitiesView` are also set to true.

Links between allocations in resources view are shown if the option `linksVisibleInResourcesView` is true. By default, the links between activities are shown as allocation links, but when the option `definedAllocationLinksVisibleInResourcesView` is true, then the defined allocation links are shown in resources view instead.

Link Property Name	Type	Description
<b>Color</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Color for the line.
<b>DashArray</b>	string	<b>Optional, default: "none"</b> – Pattern of dashes and gaps for drawing the link. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the link is drawn solid.
<b>ID</b>	string	<b>Required</b> – Identifier of this link. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>RelationType</b>	number (see enum <a href="#">RelationType</a> )	<b>Optional, default: 0</b> – The relation type is used for drawing: 0: Finish-Start, 1: Finish-Finish, 2: Start-Start, 3: Start-Finish, 4: SourceDate-Start, 5: SourceDate-Finish, 8: Finish-TargetDate, 10: Start-TargetDate, 12: SourceDate-TargetDate
<b>RoutingType</b>	number (see enum <a href="#">LinkRoutingType</a> )	<b>Optional, default: value of option defaultLinkRoutingType</b> – type of the link routing.  1: Curved, 2: Orthogonal
<b>Selectable</b>	boolean	<b>Optional, default: value of option defaultLinkSelectable</b> – If set to true, then the link will be selectable.
<b>SourceActivityID</b>	string	<b>Optional, default: undefined</b> – Identifier of the source activity. This property or SourceAllocationID has to be set.
<b>SourceAllocationID</b>	string	<b>Optional, default: undefined</b> – Identifier of the source allocation. This property or SourceActivityID has to be set. Please also note the explanations at the beginning of this <a href="#">"Link" chapter</a> .
<b>TargetActivityID</b>	string	<b>Optional, default: undefined</b> – Identifier of the source activity. This property or TargetAllocationID has to be set.
<b>TargetAllocationID</b>	string	<b>Optional, default: undefined</b> – Identifier of the target allocation. This property or TargetActivityID has to be set. Please also note the explanations at the beginning of this <a href="#">"Link" chapter</a> .
<b>TargetMarker</b>	number (see enum <a href="#">LinkMarker</a> )	<b>Optional, default: 1 (FilledArrow)</b> – allows to change the marker at the end (target) of a link.

Link Property Name	Type	Description
<b>TooltipTemplateID</b>	string	<b>Optional, default: value of option defaultLinkTooltipTemplateID</b> – ID of a tooltip template.  The template is used for tooltips that appear on the links.
<b>Width</b>	number $\geq 0$	<b>Optional, default: 1</b> – Line width of the link. The link arrow is also affected by this property.

### 3.9 PeriodHighlighters



A PeriodHighlighter object is a pure presentation object and defines the properties of a series of time periods that can be shown on each resource row and activity row (see property PeriodHighlighterID there). Each time period can be colored independently and can have a caption. Period highlighters also support the callbacks onShowTooltip, onDoubleClicked, and onShowContextMenu. In contrast to the grids created by Calendar objects, the time periods do not define work or non-work times, but only highlight time periods visually.

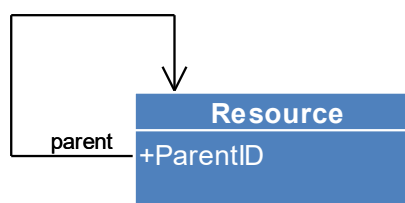
#### 3.9.1 PeriodHighlighter

PeriodHighlighter Property Name	Type	Description
<b>Entries</b>	<a href="#">PeriodHighlighterEntry[]</a>	<b>Required</b> – Array of entries that contain single time periods.
<b>ID</b>	string	<b>Required</b> – Identifier of this period highlighter. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.

### 3.9.2 PeriodHighlighterEntry

PeriodHighlighterEntry Property Name	Type	Description
<b>Caption</b>	string	<b>Optional, default:</b> "" – Text to show on the time period.
<b>CaptionColor</b>	string (CSS color value)	<b>Optional, default:</b> "white" – Color of the caption.
<b>Color</b>	string (CSS color value)	<b>Optional, default:</b> "rgba(0,0,0,0.1)" – Color of this time period.
<b>End</b>	Date   string	<b>Required</b> – End of the time period.
<b>Start</b>	Date   string	<b>Required</b> – Start of the time period.
<b>TooltipTemplateID</b>	string	<b>Optional, default:</b> value of option <b>defaultPeriodHighlighterEntryTooltipTemplateID</b> – ID of a tooltip template.  The template is used for tooltips that appear on the period highlighter entry.

### 3.10 Resource



A Resource object defines the properties of a single resource.

Resource Property Name	Type	Description
<b>AllocationRowsCollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default:</b> -1 – Specifies whether the allocation rows of the resource should be expanded or collapsed when displayed. See also callback <code>onCollapseStateChanged</code> and option <code>allocationRowsVisibleInResourcesView</code> .  -1: no change 0: display resource row in an expanded way for allocation rows. 1: display resource row in a collapsed way for allocation rows.
<b>AllocationRowsCollapsible</b>	boolean	<b>Optional, default:</b> value of option <b>defaultResourceAllocationRowsCollapsible</b> – If set to true, then the row representing this resource row will be interactively collapsible when allocation rows exist.

Resource Property Name	Type	Description
<b>AllowedRowDragModes</b>	number (see enum <a href="#">RowDragModes</a> )	<b>Optional, default: value of option defaultResourceAllowedRowDragModes</b> – This option determines the allowed row drag modes for this resource in resources view and loads view (these can be overwritten using the callback canDrag).
<b>CalendarID</b>	string	<b>Optional, default: undefined</b> – Corresponding calendar. If undefined, then the calendar specified by the option defaultCalendarID will be used.
<b>CapacityCurveID</b>	string	<b>Optional, default: undefined</b> – Identifier of any curve representing the capacity of this resource. If the identifier references a curve stack, then the summed curve is shown with the color settings of the curve stack.
<b>CollapsedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default: value in option defaultResourceCollapsedRowDesign</b> – Specifies how the time area is filled when the row is collapsed and visible.  See explanation for possible values in enumerations chapter.
<b>CollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: -1</b> – Specifies whether the row of the resource should be expanded or collapsed when displayed in resources view and eventually in loads view (see property CollapseStateInLoadsView).  See also callback onCollapseStateChanged.  -1: no change 0: display resource row in an expanded way 1: display resource row in a collapsed way
<b>CollapseStateInLoadsView</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: undefined</b> – Specifies whether the row of the resource should be expanded or collapsed when displayed in the loads view. If undefined, then the property CollapseState is used for compatibility reasons.  See also callback onCollapseStateChanged.  -1: no change 0: display resource row in an expanded way 1: display resource row in a collapsed way
<b>CurveCollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<b>Optional, default: -1</b> – Specifies whether the curves in a resource row should be expanded or collapsed when displayed. See also callback onCurveCollapseStateChanged.  -1: no change 0: display curves 1: hide curves

Resource Property Name	Type	Description
<b>CurveTooltipTemplateID</b>	string	<p><b>Optional, default: value of option <code>defaultResourceCurveTooltipTemplateID</code></b> – ID of a tooltip template.</p> <p>The template is used for tooltips that appear on the curve area of resources.</p>
<b>ExpandedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<p><b>Optional, default: value in option <code>defaultResourceExpandedRowDesign</code></b> – Specifies how the time area is filled when the row is expanded and visible.</p> <p>See explanation for possible values in enumerations chapter.</p>
<b>HasAllocationRows</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then the row representing this resource will be collapsible/expandable for allocation rows even when no allocations exist referencing this resource. This serves for lazy loading.</p>
<b>HasChildren</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then the row representing this resource will be collapsible/expandable even when there are no children defined. This serves for lazy loading.</p>
<b>HasCurves</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then the row representing this resource will be collapsible/expandable for curves even where there are no curves defined. This serves for lazy loading.</p>
<b>ID</b>	string	<p><b>Required</b> – Identifier of the resource. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.</p>
<b>LoadCurveID</b>	string	<p><b>Optional, default: undefined</b> – Identifier of any curve representing the load of this resource. If the identifier references a curve stack, then all curves within the curve stack are shown with their individual color settings as a stack.</p>
<b>LoadCurvePaneHeight</b>	number (> 0)	<p><b>Optional, default: value of option <code>defaultLoadCurvePaneHeight</code></b> – Height in pixels of the load curve pane.</p>
<b>MinimumRowHeight</b>	number	<p><b>Optional, default: value in option <code>defaultMinimumResourceRowHeight</code></b> – Minimum height of the resource row in pixels. This option is useful, when more than one line of text is shown inside the table cells. Proposal: For one line take 36*, for two lines 52, for three lines 68, and so on. In order to have the same height also, when no bar is placed in the row, take the maximum bar height adding 20 (e.g. 42) as minimum.</p>



Resource Property Name	Type	Description
		For using word wrapping in table cells, it is necessary to use a table row definition by setting the property <code>TableRowDefinitionID</code> and setting the property <code>WrapMode</code> in a contained table cell definition.
Name		<b>Deprecated!</b>
ParentID	string	<p><b>Optional, default: undefined</b> – Identifier of a parent resource this resource is assigned to. If this property is defined, the parent resource will become a resource group (if not yet a resource group) and it will keep its role as a resource with a capacity of its own.</p> <p>If this property is undefined the current resource will be considered as a root node of the resource hierarchy.</p> <p>We recommend to use only a low number of hierarchy levels and we do not guarantee correct function beyond approx. 100 levels including hierarchy levels created by using <code>Hierarchy-SupplementaryDefinitions</code>.</p>
PeriodHighlighterID	string	<b>Optional, default: undefined</b> – Reference to a period highlighter object that contains colored time periods. This can be used to show shifts or exceptions to the calendar (see property <code>CalendarID</code> ) that defines work and non-work times.
RowCollapsible	boolean	<b>Optional, default: value of option <code>defaultResourceRowCollapsible</code></b> – If set to true, then the row representing this resource will be interactively collapsible when children exist.
RowSelectable	boolean	<b>Optional, default: value of option <code>defaultResourceRowSelectable</code></b> – If set to true, then the row representing this resource will be selectable.
RowSymbolColumn-BackgroundColor	string (CSS color value)	<b>Optional, default: value of property <code>SymbolColumnBackgroundColor</code> of assigned table row definition or option <code>symbolColumnBackgroundColor</code></b> – Determines the color of the symbol column within this table row.
RowSymbolIDs	string[]	<p><b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the table symbol cell of the beginning of the table row.</p> <p>The symbols will be arranged one below the other. However, if the cell is not high enough to hold all symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.</p> <p>An empty string ("" ) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve</p>

Resource Property Name	Type	Description
		space for a symbol that may be shown at a later time.  Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.
<b>RowTooltipTemplateID</b>	string	<b>Optional, default: value of option defaultResourceRowTooltipTemplateID</b> – ID of a tooltip template.  The template is used for the tooltip that will appear on the table row when hovering the cursor above it.
<b>SkilledRowTooltipTemplateID</b>	string	<b>Optional, default: value of option defaultSkilledResourceRowTooltipTemplateID</b> – ID of a tooltip template.  The template is used for tooltips that appear on the entity table rows. It fallbacks to evaluation of the property RowTooltipTemplateID if not set.
<b>SkillIDs</b>	string[]	<b>Optional, default: undefined</b> – If set, then the resource object is shown as a row below the rows of the referenced Skill objects in the skilled resources view. When setting this property, it is not allowed to set the property ParentID!
<b>TableColor</b>	string (CSS color value)	<b>Optional, default: level-dependent gray</b> – Color for the table row. If undefined, a default value of the widget will be used.
<b>TableRowDefinitionID</b>	string	<b>Optional, default: value of option defaultResourceTableRowDefinitionID</b> – Identifier of a TableRowDefinition object, that defines the composition of the table row.
<b>TableText</b>	string	<b>Optional, default: undefined</b> – Text to display in the table row (see also property TableRowDefinitionID).
<b>TableTextColor</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Color for the table row texts. If undefined, a default value of the widget will be used.
<b>ViewArea</b>	number (see enum <a href="#">ViewArea</a> )	<b>Optional, default: Main</b> – If set to Top, then the resource and its children are shown in a separate top view area in the resources view. Only settable on resource with no ParentID set. See also options mainViewAreaVisibleInResourcesView/InLoadsView and topViewAreaVisibleInResourcesView/InLoadsView.

### 3.11 Skill



A Skill object is used for defining a skill for allocations and resources. It is visible when switching to the skilled resources view.

In this view, skills are shown as rows in the highest hierarchy level with the associated resources in rows of the next hierarchy level. Since the property SkillIDs of resources can hold references to more than one skill, resources show up more than once in this view. Only resources without parent are visible within the skilled resources view.

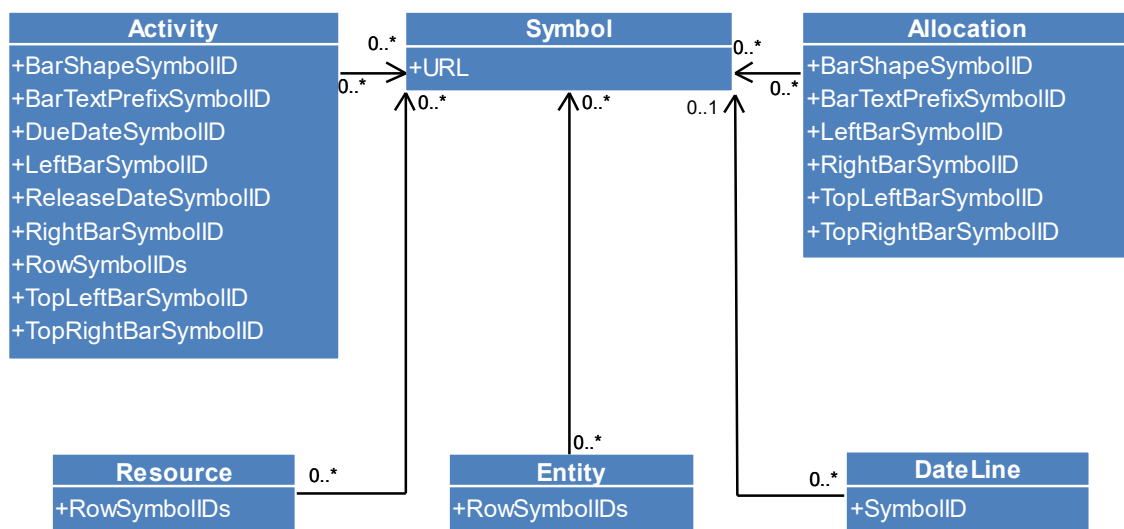
Each resource row shows the allocations assigned to it, but the allocations that reference the skill of the hierarchy parent row of the resource row are shown in full bar design while the others are shown with less details (see option allocationBarDesignOfOtherSkill). So, allocations also show up more than once in this view.

Skill Property Name	Type	Description
<b>CollapseState</b>	number (see enum <a href="#">CollapseState</a> )	<p><b>Optional, default: -1</b> – Specifies whether the row of the skill should be expanded or collapsed.</p> <p>See also callback <code>onCollapseStateChanged</code>.</p> <p>-1: no change 0: display resource row in an expanded way 1: display resource row in a collapsed way</p>
<b>ID</b>	string	<p><b>Required</b> – Identifier of this skill. Any content is allowed besides an empty string, a contained dot, or contained characters with code points 0 to 31 and 127. The first character should be a letter. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.</p>
<b>MinimumRowHeight</b>	number	<p><b>Optional, default: value in option <code>defaultSkillMinimumRowHeight</code></b> – Minimum height of the skill row in pixels. This option is useful, when more than one line of text is shown inside the table cells. Proposal: For one line take 36, for two lines 52, for three lines 68, and so on. To have the same height also, when no bar is placed in the row, take the maximum bar height adding 20 (e.g. 42) as minimum.</p> <p>For using word wrapping in table cells, it is necessary to use a table row definition by setting the property</p>

Skill Property Name	Type	Description
		TableRowDefinitionID and setting the property WrapMode in a contained table cell definition.
RowCollapsible	boolean	<b>Optional, default: value of option defaultSkillRowCollapsible</b> – If set to true, then the row representing this skill will be interactively collapsible when children exist.
RowSelectable	boolean	<b>Optional, default: value of option defaultSkillRowSelectable</b> – If set to true, then the row representing this skill will be selectable.
RowSymbolColumnBackgroundColor	string (CSS color value)	<b>Optional, default: value of property SymbolColumnBackgroundColor of assigned table row definition or option symbolColumnBackgroundColor</b> – Determines the color of the symbol column within the table row.
RowSymbolIDs	string[]	<p><b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the table symbol cell of the beginning of the table row.</p> <p>The symbols will be arranged one below the other. However, if the cell is not high enough to hold all symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.</p> <p>An empty string (“”) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve space for a symbol that may be shown later.</p> <p>Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.</p>
RowTooltipTemplateID	string	<p><b>Optional, default: value of option defaultSkillRowTooltipTemplateID</b> – ID of a tooltip template.</p> <p>The template is used for the tooltip that will appear on the table row when hovering the cursor above it.</p>
TableColor	string (CSS color value)	<b>Optional, default: level-dependent gray</b> – Color for the table row.
TableRowDefinitionID	string	<b>Optional, default: value of option defaultSkillTableRowDefinitionID</b> –

Skill Property Name	Type	Description
		Identifier of a TableRowDefinition object, that defines the composition of the table row.
<b>TableText</b>	string	<b>Optional, default: undefined</b> – Text to display in the table row (see also property TableRowDefinitionID).
<b>TableTextColor</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Color for the table row texts.

### 3.12 Symbol



A Symbol object is a pure presentation object and defines the properties of a single symbol. Symbols are used by resources, activities, and allocations. They can be displayed at different locations inside the table and the diagram area.

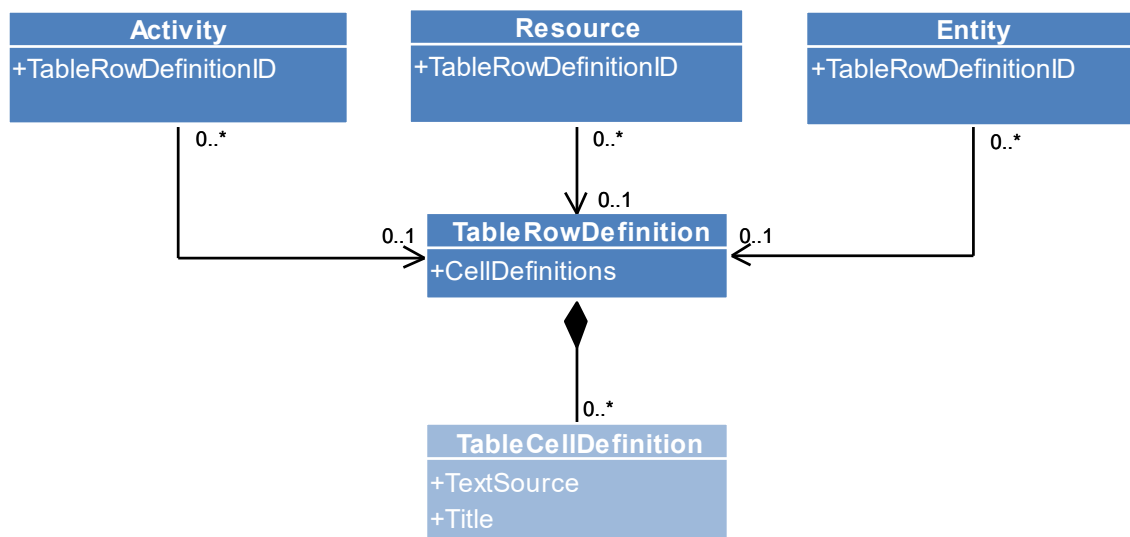
Please note: The symbols will be resized to an image with an appropriate width and height depending on their application. Therefore, when designing the symbols, you should ensure that they are clearly recognizable and visually distinguishable. For more details regarding the size, please see the descriptions of the properties related to symbols.

For some users maybe it is not possible to use paths in the property URL at all, but instead you have the possibility to use 'Data URIs', that can be created using an online service (e.g. <https://websemantics.uk/tools/image-to-data-uri-converter/>) to convert your SVG file to a string containing the SVG.

**A note regarding PDF export:** If you want to use our method saveAsPDF, then you will have to ensure that your SVG image files do not contain <style> tags, since the contained selectors may change the appearance of the exported SVG content. In case of existing <style> try to replace them by using style attributes on other tags. We can help if there are problems arising.

Symbol Property Name	Type	Description
<b>ID</b>	string	<b>Required</b> – Identifier of this symbol. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>URL</b>	string	<b>Required</b> – URL of a SVG image containing the symbol.  Two types of URLs are allowed: <ul style="list-style-type: none"> <li>absolute URL (e.g. “https://www.aaazzz.com/symbol.svg”)</li> <li>relative URL (e.g. “images/symbol.svg”) – In this case, the anchor path for the symbol directory is the application directory.</li> <li>Data URI (e.g. 'data:image/svg+xml;base64,...'). See <a href="https://en.wikipedia.org/wiki/Data_URI_scheme">https://en.wikipedia.org/wiki/Data_URI_scheme</a></li> </ul>

### 3.13 TableRowDefinitions



A **TableRowDefinition** object defines the composition of a table row containing one or more cells. You can reference these objects with the property **TableRowDefinitionID** of **Activity**, **Allocation**, **Entity**, and **Resource** objects. There are options for each property defining a default value for the corresponding property (see **defaultActivity/Allocation/Entity/ResourceTableRowDefinitionID**).

Additionally, it is possible to declare one table row definition to provide the table title for the views and the entities table by using the options **tableRowDefinitionIDForTitleInActivities/Resources/LoadsView** or **tableRowDefinitionIDForTitleInEntitiesTable**.

#### 3.13.1 TableRowDefinition

TableRowDefinition Property Name	Type	Description
<b>BackgroundColor</b>	string (CSS color value)	<b>Optional, default: undefined</b> – BackgroundColor of the table row. The value is only used when not undefined and is overlayed by the background color of the table row defined in the property TableColor of the row object.
<b>CellDefinitions</b>	<a href="#">TableCellDefinition</a> []	<b>Optional, default: [{ Title: "", TextSource: "TableText", Width: 200, HorizontalAlignment: HorizontalAlignment.Left }]</b> – Array of TableCellDefinition objects.
<b>ID</b>	string	<b>Required</b> – Identifier of this table row definition. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>SymbolColumnBackgroundColor</b>	string (CSS color value)	<b>Optional, default: value of option symbolColumnBackgroundColor or entitiesTableSymbolColumnBackgroundColor, resp.</b> – Determines the color of the symbol column within this table row when the property RowSymbolColumnBackgroundColor is not set on the object where this table row definition is applied to.
<b>TextColor</b>	string (CSS color value)	<b>Optional, default: undefined</b> – Text color of the table row definition. The value is only used when not undefined and is overlayed by the text color of the table row defined in the property TableTextColor of the row object.

### 3.13.2 TableCellDefinition

TableCellDefinition Property Name	Type	Description
<b>BackgroundColor</b>	string (CSS color value)	<b>Optional, default: undefined</b> – If set and property BackgroundColorSource is empty or the referenced property on a row object is empty, then this color overlays the background color of the table row defined in the property TableColor of the row object and the property BackgroundColor of the TableRowDefinition object.
<b>BackgroundColorSource</b>	string	<b>Optional, default: undefined</b> – If set to an object's property name and the value of the referenced property on a row object is not empty, then the

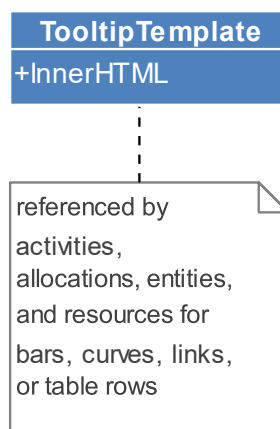
TableCellDefinition Property Name	Type	Description
		value there overlays the background color defined by property BackgroundColor and property TableColor of the row object.
<b>HorizontalAlignment</b>	number (see enum <a href="#">HorizontalAlignment</a> )	<b>Optional, default: Left</b> – Horizontal alignment of the shown text. The first column is always shown with left alignment because of the tree symbols on the left side.
<b>HorizontalTitleAlignment</b>	number (see enum <a href="#">HorizontalAlignment</a> )	<b>Optional, default: Center</b> – Horizontal alignment of the shown title text. In the entities table the last column is always shown with center alignment.
<b>MaximumWidth</b>	number	<b>Optional, default: Infinity</b> – Maximum width of the table cell, when cell width is interactively modified.
<b>MinimumWidth</b>	number	<b>Optional, default: 3</b> – Minimum width of the table cell, when cell width is interactively modified.
<b>SymbolHeight</b>	number	<b>Optional, default: undefined</b> – If set, then the symbol height is constant. If not set, then the symbol height is determined on each table row to fill the full height of that.
<b>SymbolIDSource</b>	string	<b>Optional, default: ""</b> – Property to take the symbol ID out of the referencing activity, resource, or entity object. The symbol will be displayed in the cell inside a square that has the size of the minimum row height. The symbol will obey the HorizontalAlignment property. It is also possible to use the TextSource property along with this property, but there are the following restrictions: If using left alignment, the text will be indented so that it is to the right of the symbol. If using center or right alignment, the symbol will be overlapped by the text.
<b>SymbolWidth</b>	number	<b>Optional, default: value of property SymbolHeight</b> – If set, then the symbol width is constant. If not set, then the symbol will be stretched to be square.
<b>TextColor</b>	string (CSS color value)	<b>Optional, default: undefined</b> – If set and property TextColorSource is empty or the referenced property on a row object is empty, then this color overlays the text color of the table row defined in the property TableTextColor of the row object and the property TextColor of the TableRowDefinition object.
<b>TextColorSource</b>	string	<b>Optional, default: undefined</b> – If set to an object's property name and the value of the referenced property on a row object is not empty, then the value there overlays the text color defined by property TextColor and property TableTextColor of the row object.



TableCellDefinition Property Name	Type	Description
<b>TextFormat</b>	string	<p><b>Optional, default: undefined</b> – String that describes the format of the content of the cell. This property overlays the property TextSource.</p> <p>This string contains the placeholders for object values surrounded by double curly braces {{ }}. For example, based on the following string a tooltip with a table containing three rows of key-value pairs is created, where the values are taken from the properties "name" and "firstName" of the referenced object:</p> <pre>{{name}}, {{firstName}}</pre> <p>As an escape, the use of three open curly braces {{{ are displayed as {{.</p> <p>Additionally, the property name can be extended to contain the desired property type as in {{Start:date}}. At the moment only the types 'date' and 'number' are possible besides 'string' (other property types are converted automatically with toString()). The type 'date' converts date values by default using the same format as other dates in the timescale and at the dragging date line captions. You can add another colon followed by a format name, that is defined by the options intlDateTimeFormatOptionsMap or intlNumberFormatOptionsMap, resp.</p> <p>The referenced object is the object on which the tooltip will be shown. For period highlighter entries and allocation entries the referenced object is the main object and not the entry object.</p> <p>It is possible to access related objects by using the following keywords within the property accessor string:</p> <ul style="list-style-type: none"> <li>• On activities: &gt;Parent, &gt;Calendar</li> <li>• On resources: &gt;Parent, &gt;Calendar, &gt;LoadCurve, &gt;CapacityCurve</li> <li>• Additionally on resources in SkilledResources view: &gt;Skill</li> <li>• On entities: &gt;Parent</li> <li>• On allocations: &gt;Activity, &gt;Resource</li> <li>• On links: &gt;SourceActivity, &gt;TargetActivity, &gt;SourceAllocation, &gt;TargetAllocation</li> </ul>

TableCellDefinition Property Name	Type	Description
		<p>It is also possible to access variables that are defined by the option <code>applicationVariablesMap</code> by using <code>?variableName</code>.</p> <p>If the value reached is an object, you can then access a property value by using a prefixed dot: <code>.propertyName</code> and you can use <code>[...]</code> to access a property value, a map entry or an array entry. Within <code>[...]</code> you can use a literal like <code>5</code> or <code>A</code> (with or without quotes) or even curly braces <code>{{...}}</code> with the same rules as above.</p>
<b>TextSource</b>	string	<p><b>Optional, default: "TableText"</b> – Property to take the text out of the referencing activity, resource, or entity object.</p> <p>See also property <code>TextFormat</code>.</p>
Title		<b>Deprecated! See property TitleText.</b>
<b>TitleText</b>	string	<p><b>Optional, default: ""</b> – When the table row definition, that contains this table cell definition, is referenced by one of the options <code>activity/resource/entityTableRowDefinitionIDForTitle</code>, then the title defined here will be shown on the table title.</p>
<b>Width</b>	number	<b>Optional, default: 200</b> – Width of the table cell.
<b>WrapMode</b>	number (see enum <a href="#">TextWrapMode</a> )	<b>Optional, default: None</b> – If set, then it possible to show more than one line of text using newline characters ( <code>'\n'</code> ).

### 3.14 TooltipTemplate



A `TooltipTemplate` object describes the appearance of a tooltip in the form of an HTML string. This string describes a DOM subtree and contains placeholders with references to the object properties to

be displayed. At runtime, the placeholders are replaced by the values of the referenced object properties.

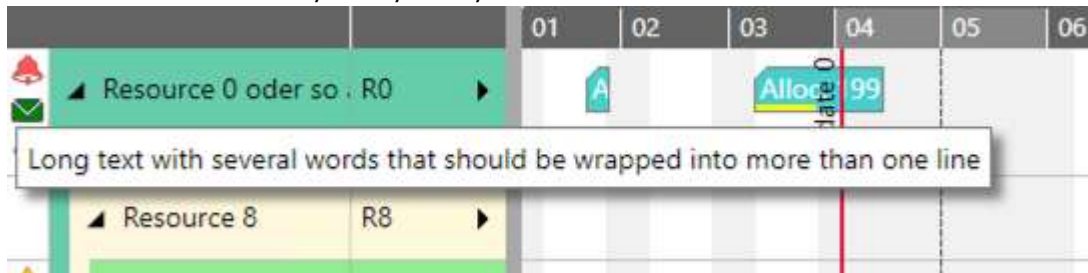
There are two ways to apply a template:

1. Either you can specify the template ID inside the out-parameter "tooltipTemplateID" of the onShowTooltip callback.
2. Or you can use the properties TooltipTemplateID, BarTooltipTemplateID, RowTooltipTemplateID, and CurveTooltipTemplateID of the activities, resources, allocations, links, and entities. Additionally, there exists the property TooltipTemplateID on period highlighter entries. All these properties have fallback options named:
  - defaultActivityBar/AllocationBarTooltipTemplateID
  - defaultActivityRow/AllocationRow/ResourceRow/EntityRowTooltipTemplateID
  - defaultResourceCurveTooltipTemplateID
  - defaultPeriodHighlighterEntryTooltipTemplateID

Here is an additional hint for designing the HTML markup:

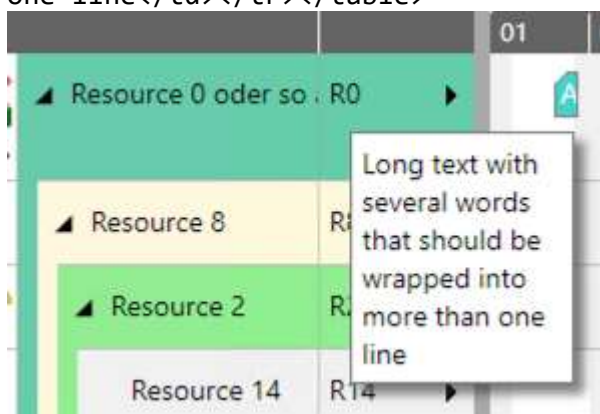
If you fill the markup with a normal table, you will get tooltips that are eventually very wide:

```
<table><tr><td>Long text with several words that should be wrapped into more than one line</td></tr></table>
```



In order to limit the width of the tooltip, you can set some attributes on the tags:

```
<table style="word-wrap: break-word;"><tr><td style="max-width: 100px;">Long text with several words that should be wrapped into more than one line</td></tr></table>
```



Of course, you are free to use other HTML tags within the markup, also including images by using data URIs.

TooltipTemplate Property Name	Type	Description
<b>ID</b>	string	<b>Required</b> – Identifier of this tooltip template. Any content is allowed besides an empty string and characters with code points 0 to 31 or 127. Each object type has its own identifier name space, so it is allowed to use the same ID for each object type separately.
<b>HTMLFormat</b>	string	<p><b>Optional, default:</b> "" – HTML string that describes the structure of a tooltip.</p> <p>This string contains the placeholders for object values surrounded by double curly braces {{ }}. For example, based on the following string a tooltip with a table containing three rows of key-value pairs is created, where the values are taken from the object properties "name", "firstName", and "age":</p> <pre>&lt;table&gt;   &lt;tr&gt;&lt;td&gt;Name: &lt;/td&gt;&lt;td&gt;{{name}}&lt;/td&gt;&lt;/tr&gt;   &lt;tr&gt;&lt;td&gt;First name: &lt;/td&gt;&lt;td&gt;{{firstName}}&lt;/td&gt;&lt;/tr&gt;   &lt;tr&gt;&lt;td&gt;Age: &lt;/td&gt;&lt;td&gt;{{age}}&lt;/td&gt;&lt;/tr&gt; &lt;/table&gt;</pre> <p>As an escape, the use of three open curly braces {{{ are displayed as {{.</p> <p>Additionally, the property name can be extended to contain the desired property type as in {{Start:date}}. At the moment only the type 'date' is possible besides 'string' (other property types are converted automatically with toString()). The type 'date' converts date values using the same format as other dates in the timescale and at the dragging date line captions.</p> <p>The referenced object is the object on which the tooltip will be shown. For period highlighter entries and allocation entries the referenced object is the main object and not the entry object.</p> <p>It is possible to access related objects by using the following keywords within the property accessor string:</p> <ul style="list-style-type: none"> <li>On activities: &gt;Parent, &gt;Calendar</li> <li>On resources: &gt;Parent, &gt;Calendar, &gt;LoadCurve, &gt;CapacityCurve</li> <li>On entities: &gt;Parent</li> <li>On allocations: &gt;Activity, &gt;Resource</li> <li>On links: &gt;SourceActivity, &gt;TargetActivity, &gt;SourceAllocation, &gt;TargetAllocation</li> </ul> <p>It is also possible to access other objects that are otherwise reachable by the callback arguments of the callback onShowTooltip by using the following keywords at the beginning of the property accessor string:</p> <ul style="list-style-type: none"> <li>On allocation bars: #Entry</li> <li>On period highlighters: #Entry, #RowObject.</li> <li>On curves: #Date, #Capacity, #Load, #SingleLoads. For #SingleLoads you have to add .curveID to get the curve value for a single curve.</li> <li>On resources and allocation rows in the skilled resources view: #Skill.</li> </ul>

TooltipTemplate Property Name	Type	Description
		<p>It is also possible to access variables that are defined by the option <code>applicationVariablesMap</code> by using <code>?variableName</code>.</p> <p>If the value reached is an object, you can then access a property value by using a prefixed dot: <code>.propertyName</code> and you can use <code>[...]</code> to access a property value, a map entry or an array entry. Within <code>[...]</code> you can use a literal like <code>5</code> or <code>A</code> (with or without quotes) or even curly braces <code>{{...}}</code> with the same rules as above.</p>
InnerHTML		<b>Deprecated. Renamed to HTMLFormat.</b>

## 4 Widget

This is the central object that an application talks to. Here are methods to add, update and remove the data objects meant above and there also are many options and callbacks to refine the appearance of the widget. Technically the widget is based on the widget factory of jQuery UI. Please see <https://learn.jquery.com/jquery-ui/> in order to learn how to work with jQuery and jQuery UI widgets in general.

At first the widget has to be instantiated using a call like `$("#gantttDiv").nXYZWidget(options)`, where 'options' is an optional object containing first settings if needed (otherwise it can be left undefined). After that you can set additional options and use the provided methods.

### 4.1 Options

The following options are settable and gettable by using the jQuery UI Widget command "option" at any time within a session.

#### A note regarding the old "pm\_" prefix of the widget options:

The old "pm\_" prefix has been removed from the widget options for simplicity. However, there is no need to change existing code as the old notation of the options will continue to be supported.

#### A note on using CSS custom properties for coloring:

Each option that controls the color of an element in VSW can be set a value of type string that represents a "CSS color value". This means that you have, for instance, the following options to specify a color value:

- a predefined color name that CSS supports, such as "red", "green", or "blue"
- a hexadecimal notation, such as "#FF0000", "#00FF00", or "#0000FF"
- an RGB or RGBA notation, such as "rgb(255, 0, 0)", "rgb(0, 255, 0)", or "rgba(0, 0, 255, 0.5)"
- an HSL and HSLA notation, such as "hsl(0, 100%, 50%)", "hsl(120, 100%, 50%)", or "hsla(240, 100%, 50%, 0.5)"

In addition, you can also use [CSS Custom Properties](#). How to do that is described [in a blog post](#).

Widget Option Name	Type	Description
<b>activityBarTopOffsetAndHeightScaleFactor</b>	number > 0 and <= 10	<b>Optional, default: 1</b> – This option modifies the top offset and height of activity bars. It can help to implement a compact layout.
<b>activityBaselineBarsVisible</b>	boolean	<b>Optional, default: true</b> – If set to false, no baseline bars are displayed for the activities.
<b>activityCalendarsEnabled</b>	boolean	<b>Optional, default: true</b> – If set to true, calendars assigned to activities by setting the activity property <code>CalendarID</code> are displayed in the Activities View.
<b>activityHierarchySupplementaryDefinitionID</b>	string	<b>Optional, default: undefined</b> – ID of a <code>HierarchySupplementaryDefinition</code> object that will be used to specify grouping parameters for hierarchy of activity objects.
<b>activityRowSortCodePropertyName</b>	string	<p><b>Optional, default: "SortCode"</b> – Name of a data property to be used as sort criteria while sorting activity rows. The values of the addressed property in the activities can contain strings, numbers, or date values.</p> <p>If using interactive vertical row dragging, the specified data property must contain values of number type.</p> <p>See also option <code>activityRowSortMode</code>.</p>
<b>activityRowSortMode</b>	number (see enum <a href="#">RowSortMode</a> )	<p><b>Optional, default: None</b> – If set to a mode unequal to <code>None</code> activity rows are sorted ascending or descending.</p> <p>It is a prerequisite to use the ascending mode for dragging activity rows vertically.</p> <p>See also option <code>activityRowSortCodePropertyName</code></p>
<b>activityTableRowDefinitionIDForTitle</b>		<b>Deprecated! See renamed option <code>tableRowDefinitionIDForTitleInActivitiesView</code>.</b>
<b>additionalDateInterpretedAs-Empty</b>	Date   string   null	<b>Optional, default: null</b> – If set, then on properties of date type the value can be set to the value given here and will be interpreted as being <code>null/undefined/""</code> . If given as a string,

Widget Option Name	Type	Description
		this string is converted to a Date object internally and each date will be checked by comparing the date values.
<b>additionalDateStringInterpreted-AsEmpty</b>	string	<b>Optional, default:</b> "" – If set, then on properties of date type the value can be set to the value given here and will be interpreted as being null/undefined/"". Each date string will be checked by comparing the strings.
<b>allocationBarDesignOfOtherSkill</b>	number (see enum <a href="#">BarDesigns</a> )	<b>Optional, default: DefaultReduced   Text</b> – The value is used for allocation bars that are shown in the view type SkilledResourcesView in a skill-specific resource row with a different skill. See also option defaultAllocationBarDesign.
<b>allocationBarTopOffsetAndHeightScaleFactor</b>	number > 0 and <= 10	<b>Optional, default: 1</b> – This option modifies the top offset and height of allocation bars in activities view and resources view. It can help to implement a compact layout.
<b>allocationRowSortCodePropertyName</b>	string	<b>Optional, default: "SortCode"</b> – Name of a data property to be used as sort criteria while sorting allocation rows. The values of the addressed property in the allocations can contain strings, numbers, or date values.  See also option allocationRowSortMode.
<b>allocationRowSortMode</b>	number (see enum <a href="#">RowSortMode</a> )	<b>Optional, default: None</b> – If set to a mode unequal to None allocation rows are sorted ascending or descending.  See also option allocationRowSortCodePropertyName.
<b>allocationRowsVisibleInActivitiesView</b>	boolean	<b>Optional, default: false</b> – If set to true, then allocations are shown as own rows below the row of the referenced activity in activities view.
<b>allocationRowsVisibleInResourcesView</b>	boolean	<b>Optional, default: false</b> – If set to true, then allocations are shown as own rows below the row of the referenced resource in resources view.
<b>allocationRowsVisibleInSkilledResourcesView</b>	boolean	<b>Optional, default: false</b> – If set to true, then allocations are shown as

Widget Option Name	Type	Description
		own rows below the row of the referenced resource in skilled resources view.
<b>applicationStyleDefinition</b>	string	<b>Option, default: null</b> – If set, then the contained text builds the content of a HTML style placed within the HEAD object inside the DOM. You can use {{?...}} to address application variables, see option applicationVariablesMap. This option allows it to define CSS variables, see <a href="https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties">https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties</a> . CSS variables can be used instead of a color name within color properties and options.
<b>applicationVariablesMap</b>	Map Object	<b>Optional, default: null</b> – If set, then the keys serve as variable names in text formatting (see property HTMLFormat of TooltipTemplate objects, property BarTextFormat of Activity and Allocation objects, and property TextFormat of TableCellDefinition objects). The values can be of type string, number, Date, boolean, Object. Every key name must start with a letter and must not contain a dot.
<b>asynchronousInteractiveTime-AreaStretching</b>	boolean	<b>Optional, default: false</b> – If set to true, rendering the time area will be delayed when the user stretches it by using the zoom out button of the timescale, by using the mouse wheel, or by using the appropriate touch gesture. This can be used to fasten the update behavior in case of diagrams with complex or many bars. That way, the diagram will become more reactive.
<b>asynchronousRendering</b>	boolean	<b>Optional, default: false</b> – If set to true, then the rows are filled with bars asynchronously when scrolling vertically or resizing a view.
<b>barSortModeForOptimizedRowDesign</b>	number (see enum <a href="#">BarSortMode</a> )	<b>Optional, default: StartAndEnd</b> – Determines how the bars are sorted in a row where the bars are shown vertically optimized, means that they do not overlap each other: <b>StartAndEnd (0)</b> : Bars are sorted such, that the earlier the start, the earlier the bar is placed. If two bars have the



Widget Option Name	Type	Description
		<p>same start, the longer bar is placed first.</p> <p><b>ByCompareObjects (1):</b> Bars are sorted by using the callback compareObjects.</p> <p><b>ByCompareObjectsOnSameStart (2):</b> Bars are sorted as in StartAndEnd, but for two bars with the same start, the result of callback compareFunc is used.</p>
<b>bottomRowMarginInTimeArea</b>	number > 0	<b>Optional, default: 5</b> – Height of the margin between the bottom row border and bars above in pixels. The value is also used for the vertical margins of curve panes. See also topRowMarginInTimeArea and subRowDistanceInTimeArea.
<b>calendarGridColor</b>	string (CSS color value)	<b>Optional, default: "#f0f0f0"</b> – Specifies a color used to color the vertical stripes representing the nonworking times inside the diagram.
clickCallbackTriggeringOnRowInTimeArea		<b>Deprecated! Renamed to triggeringOfOnClickedInTimeAreaOfRow.</b>
<b>currentDate</b>	Date   string   null	<b>Optional, default: null</b> – When set to a valid date, then a darkened area is positioned from the timescale start up to this date. The darkened area can be attributed by using the options pastBackgroundFillColor/ LineColor/ LineWidth/ LineDashArray.
<b>cursorDateLineVisible</b>	boolean	<b>Optional, default: false</b> – If this option is set to true, an additional labeled date line will follow the mouse cursor.
<b>curvePanelsCollapsibleInResourcesView</b>	boolean	<b>Optional, default: true</b> – Specifies whether the curve panes can be interactively collapsed or expanded.
<b>curvePanelsVisibleInActivitiesView</b>	boolean	<p><b>Optional, default: false</b> – If this option is set to true, a curve pane is displayed in the ActivitiesView for each activity row. In each pane the curves of the resource first found in an allocation related to the corresponding activity are displayed.</p> <p>Please note: This option has to be set when instantiating the widget. If it is set later, it has no effect.</p>
<b>dateLineCaptionOptimizedPositioningEnabled</b>	boolean	<b>Optional, default: false</b> – Specifies whether the captions of date lines

Widget Option Name	Type	Description
		should be arranged optimized to avoid overlapping.
<b>dateLineGridColor</b>	string (CSS color value)	<b>Optional, default: "#505050" (on weekly or daily grid) and "646464" (on automatic grid)</b> – Color of the date line grid (see also option dateLineGridMode).
<b>dateLineGridDashArray</b>	string	<b>Optional, default: "4,1" (on weekly or automatic grid) and "2,1" (on daily grid)</b> – Pattern of dashes and gaps for drawing the date line grid. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the grid lines are drawn solid (see also option dateLineGridMode).
<b>dateLineGridMode</b>	number (see enum <a href="#">DateLineGridModes</a> )	<b>Optional, default: Weekly</b> – This option determines the distance of the date lines shown.  See also options dateLineGridColor, dateLineGridDashArray, dateLineGridWidth.
<b>dateLineGridWidth</b>	number	<b>Optional, default: 1</b> – Width of the date line grid in pixels.
<b>decouplingOfAllocationProperties FromActivities</b>	boolean	<b>Optional, default: true</b> – If set to true, then there is no internal fallback of allocation property values of properties Color, NonworkingColor, BorderColor, Progress, ProgressColor, ProgressNonworkingColor from the values of the same properties of the assigned activity. This results in a performance gain when updating activities.
<b>defaultActivityAllocationRows-Collapsible</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the property AllocationRowsCollapsible of Activity objects.
<b>defaultActivityAllowedBar-DragModes</b>	number (see enum <a href="#">ActivityBarDragModes</a> )	<b>Optional, default: ActivityBarDragModes.Drag-Horizontally</b> – This option holds the default for the attribute AllowedBarDragModes of Activity objects.

Widget Option Name	Type	Description
<b>defaultActivityAllowedRow-DragModes</b>	number (see enum <a href="#">RowDragModes</a> )	<b>Optional, default:</b> <b>RowDragModes.None</b> – This option holds the default for the attribute AllowedRowDragModes of Activity objects
<b>defaultActivityBarDesign</b>	number (see enum <a href="#">BarDesigns</a> )	<b>Optional, default: Default</b> – This option determines the default design for activity bars including or excluding entries, complex shape, symbols, status, constraints, baseline, progress, and text (see property BarDesign of Activity objects).
<b>defaultActivityBarHeight</b>	number ( $\geq 0, \leq 1000$ )	<b>Optional, default: 22</b> – Default height of the activity bars in pixels. See also Activity.BarHeight.
<b>defaultActivityBarSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute BarSelectable of Activity objects.
<b>defaultActivityBarShape</b>	number (see enum <a href="#">ActivityBarShape</a> )	<b>Optional, default: Regular</b> – This option defines which shape should be used by default for the visualization of activity bars.
<b>defaultActivityBarTooltip-TemplateID</b>	string	<b>Optional, default: ""</b> – ID of a TooltipTemplate object that will be used when an activity object has set the property BarTooltipTemplateID to "".
<b>defaultActivityCollapsedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default: 11</b> – This option holds the default for the attribute CollapsedRowDesign of Activity objects.  See explanation for possible values in enumerations chapter.
<b>defaultActivityConstraintSymbolColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Specifies the color used by default for the symbols visualizing the constraint dates (EarliestStart/End, LatestStart/End, MustStart/EndOn).
<b>defaultActivityExpandedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default: 11</b> – This option holds the default for the attribute ExpandedRowDesign of Activity objects.  See explanation for possible values in enumerations chapter.
<b>defaultActivityMinimumRowHeight</b>	number	<b>Optional, default: 42</b> – Default minimum height of the activity rows in pixels. See also Activity.MinimumRowHeight.

Widget Option Name	Type	Description
<b>defaultActivityProgressBackgrounddColor</b>	string (CSS color value)	<b>Optional, default: "transparent"</b> – Color for the background of the progress bar region for activities.
<b>defaultActivityRowCollapsible</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute RowCollapsible of Activity objects.
<b>defaultActivityRowSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute RowSelectable of Activity objects
<b>defaultActivityRowTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a TooltipTemplate object that will be used when an activity object has set the property RowTooltipTemplateID to "".
<b>defaultActivitySnapTargetsForEnd</b>	number (see enum <a href="#">SnapTargets</a> )	<b>Optional, default: 8</b> – This option holds the default for the attribute SnapTargetsForEnd of Activity objects
<b>defaultActivitySnapTargetsForStart</b>	number (see enum <a href="#">SnapTargets</a> )	<b>Optional, default: 8</b> – This option holds the default for the attribute SnapTargetsForStart of Activity objects.
<b>defaultActivityStatusFrameColor</b>	string	<b>Optional, default: "red"</b> – This option holds the default color for the property statusFrameColor of Activity objects.
<b>defaultActivityTableRowDefinitionID</b>	string	<b>Optional, default: null</b> – ID of a TableRowDefinition object that will be used when an activity object has set the property TableRowDefinitionID to "".
<b>defaultAllocationAllowedBarDragModes</b>	number (see enum <a href="#">AllocationBarDragModes</a> )	<b>Optional, default: DragAutoHorOrVer</b> – This option holds the default for the attribute AllowedBarDragModes of Allocation objects. See also callback canDrag.
<b>defaultAllocationAllowedBarDragModesInActivitiesView</b>	number (see enum <a href="#">AllocationBarDragModes</a> )	<b>Optional, default: DragHorizontally</b> – This option holds the default for the attribute AllowedBarDragModesInAllocationView of Allocation objects. See also callback canDrag.
<b>defaultAllocationBarDesign</b>	number (see enum <a href="#">BarDesigns</a> )	<b>Optional, default: Default</b> – This option determines the default design for allocation bars including or excluding entries, complex shape, symbols, status, constraints, progress, and text (see property BarDesign of Allocation objects).
<b>defaultAllocationBarHeight</b>	number ( $\geq 0, \leq 1000$ )	<b>Optional, default: 22</b> – Default height of the allocation bars in pixels. See also Allocation.BarHeight.

Widget Option Name	Type	Description
<b>defaultAllocationBarSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute BarSelectable of Allocation objects.
<b>defaultAllocationBarShape</b>	number (see enum <a href="#">AllocationBarShape</a> )	<b>Optional, default: Regular</b> – This option defines which shape should be used by default for the visualization of allocation bars.
<b>defaultAllocationBarTextFormat</b>	string	<b>Optional, default: undefined</b> – This option holds the default value for the property BarTextFormat of Allocation objects. See there for an explanation of the format.
<b>defaultAllocationBarTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a TooltipTemplate object that will be used when an allocation object has set the property BarTooltipTemplateID to "".
<b>defaultAllocationConstraintSymbolColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Specifies the color used by default for the symbols visualizing the constraint dates (EarliestStart/End, LatestStart/End, MustStart/EndOn).
<b>defaultAllocationMinimumRowHeight</b>	number ( $\geq 0$ )	<b>Optional, default: 42</b> – Default minimum height of the allocation rows in pixels. See also Allocation.MinimumRowHeight.
<b>defaultAllocationProgressBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "transparent"</b> – Color for the background of the progress bar region for allocations.
<b>defaultAllocationRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default: 9</b> – This option holds the default for the attribute RowDesign of Allocation objects.  See explanation for possible values in enumerations chapter.
<b>defaultAllocationRowSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the property RowSelectable of Allocation objects.
<b>defaultAllocationRowTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a TooltipTemplate object that will be used when an allocation object has set the property RowTooltipTemplateID to "".
<b>defaultAllocationSnapTargetsForEnd</b>	number (see enum <a href="#">SnapTargets</a> )	<b>Optional, default: 11</b> – This option holds the default for the attribute SnapTargetsForEnd of Allocation objects.
<b>defaultAllocationSnapTargetsForStart</b>	number (see enum <a href="#">SnapTargets</a> )	<b>Optional, default: 11</b> – This option holds the default for the attribute

Widget Option Name	Type	Description
		SnapTargetsForStart of Allocation objects.
<b>defaultAllocationStatusFrameColor</b>	string	<b>Optional, default: "red"</b> – This option holds the default color for the property statusFrameColor of Allocation objects.
<b>defaultAllocationTableRowDefinitionID</b>	string	<b>Optional, default: one table cell with value of property TableText of the referenced allocation object in it</b> – ID of a TableRowDefinition object that will be used in allocation rows when an allocation object has set the property TableRowDefinitionID to "". See option allocationRowsVisibleInActivitiesView.
defaultAllowedActivityBarDragModes		<b>Deprecated! See renamed option defaultActivityAllowedBarDragModes</b>
defaultAllowedAllocationBarDragModes		<b>Deprecated! See renamed option defaultAllocationAllowedBarDragModes</b>
defaultAllowedEntityRowDragModes		<b>Deprecated! See renamed option defaultEntityAllowedRowDragModes</b>
<b>defaultCalendarID</b>	string	<b>Optional, default: undefined</b> – Specifies a default calendar to be used in the widget. If calendars are defined on activities or resource they will override this calendar. If there is no calendar defined on an activity or a resource and if this default calendar ID is undefined, then the calendar is assumed to be one with constantly non-working time only.
<b>defaultEntityAllowedRowDragModes</b>	number (see enum <a href="#">RowDragModes</a> )	<b>Optional, default: RowDragModes.DragOutside</b> – This option holds the default for the attribute AllowedRowDragModes of Entity objects.
<b>defaultEntityMinimumRowHeight</b>	number	<b>Optional, default: 42</b> – Default minimum height of the entity rows in pixels. See also Entity.MinimumRowHeight.
<b>defaultEntityRowCollapsible</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute RowCollapsible of Entity objects.
<b>defaultEntityRowSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute RowSelectable of Entity objects.















Widget Option Name	Type	Description
<b>defaultEntityRowTooltipTemplateID</b>	string	<b>Optional, default:</b> "" – ID of a TooltipTemplate object that will be used when an entity object has set the property RowTooltipTemplateID to "".
<b>defaultEntityTableRowDefinitionID</b>	string	<b>Optional, default:</b> null – ID of a TableRowDefinition object that will be used when an entity object has set the property TableRowDefinitionID to "".
<b>defaultLinkRoutingType</b>	number (see enum <a href="#">LinkRoutingType</a> )	<b>Option, default</b> <b>LinkRoutingType.Curved</b> – This option holds the default for the attribute RoutingType of Links objects.
<b>defaultLinkSelectable</b>	boolean	<b>Optional, default:</b> false – This option holds the default for the attribute Selectable of Link objects.
<b>defaultLinkTooltipTemplateID</b>	string	<b>Optional, default:</b> "" – ID of a TooltipTemplate object that will be used when a link object has set the property TooltipTemplateID to "".
<b>defaultLoadCurvePaneColor</b>	string (CSS color value)	<b>Optional, default:</b> "rgba(43,86,158,0.2)" – Color for the background of the load curve pane.
<b>defaultLoadCurvePaneHeight</b>	number (> 0)	<b>Optional, default:</b> 50 – Default value for property LoadCurvePaneHeight of Resource objects.
<b>defaultResourceAllocationRowsCollapsible</b>	boolean	<b>Optional, default:</b> true – This option holds the default for the property AllocationRowsCollapsible of Resource objects.
<b>defaultResourceAllowedRowDragModes</b>	number (see enum <a href="#">RowDragModes</a> )	<b>Optional, default:</b> <b>RowDragModes.None</b> – This option holds the default for the attribute AllowedRowDragModes of Resource objects.
<b>defaultResourceCollapsedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default:</b> 11 – This option holds the default for the attribute CollapsedRowDesign of Resource objects.  See explanation for possible values in enumerations chapter.
<b>defaultResourceCurveTooltipTemplateID</b>	string	<b>Optional, default:</b> "" – ID of a TooltipTemplate object that will be used when a resource object has set the property CurveTooltipTemplateID to "".
<b>defaultResourceExpandedRowDesign</b>	number (see enum <a href="#">RowDesigns</a> )	<b>Optional, default:</b> 11 – This option holds the default for the attribute ExpandedRowDesign of Resource objects.



Widget Option Name	Type	Description
		See explanation for possible values in enumerations chapter.
<b>defaultResourceMinimumRowHeight</b>	number	<b>Optional, default: 42</b> – Default minimum height of the resource rows in pixels. See also <code>Resource.MinimumRowHeight</code> .
<b>defaultResourceRowCollapsible</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute <code>RowCollapsible</code> of Resource objects.
<b>defaultResourceRowSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute <code>RowSelectable</code> of Resource objects.
<b>defaultResourceRowTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a <code>TooltipTemplate</code> object that will be used when a resource object has set the property <code>RowTooltipTemplateID</code> to "".
<b>defaultResourceTableRowDefinitionID</b>	string	<b>Optional, default: undefined</b> – ID of a <code>TableRowDefinition</code> object that will be used when a resource object has set the property <code>TableRowDefinitionID</code> to "".
<b>defaultSkilledAllocationBarTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a <code>TooltipTemplate</code> object that will be used when an allocation object has set the property <code>SkilledBarTooltipTemplateID</code> to "".
<b>defaultSkilledAllocationRowTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a <code>TooltipTemplate</code> object that will be used when an allocation object has set the property <code>SkilledRowTooltipTemplateID</code> to "".
<b>defaultSkilledResourceRowTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a <code>TooltipTemplate</code> object that will be used when an resource object has set the property <code>SkilledRowTooltipTemplateID</code> to "".
<b>defaultSkillMinimumRowHeight</b>	number	<b>Optional, default: 42</b> – Default minimum height of the skill rows in pixels. See also <code>Resource.MinimumRowHeight</code> .
<b>defaultSkillRowCollapsible</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute <code>RowCollapsible</code> of Skill objects.
<b>defaultSkillRowSelectable</b>	boolean	<b>Optional, default: true</b> – This option holds the default for the attribute <code>RowSelectable</code> of Skill objects.
<b>defaultSkillRowTooltipTemplateID</b>	string	<b>Optional, default: ""</b> – ID of a <code>TooltipTemplate</code> object that will be



Widget Option Name	Type	Description
		used when a Skill object has set the property RowTooltipTemplateID to "".
<b>defaultSkillTableRowDefinitionID</b>	string	<b>Optional, default: one table cell with value of property TableText of the referenced Skill object in it</b> – ID of a TableRowDefinition object that will be used when a Skill object has set the property TableRowDefinitionID to "".
<b>defaultUpdateMode</b>	number (see enum <a href="#">UpdateModes</a> )	<b>Optional, default: Default</b> – Specifies the default for the parameter updateMode in all update methods.
<b>defaultValuesForActivityEntryProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of ActivityEntry objects.
<b>defaultValuesForActivityProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of Activity objects except ID.
<b>defaultValuesForAllocationEntryProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of AllocationEntry objects.
<b>defaultValuesForAllocationProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of Allocation objects except ID.
<b>defaultValuesForEntityProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of Entity objects except ID.
<b>defaultValuesForLinkProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of Link objects except ID.
<b>defaultValuesForResourceProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of Resource objects except ID.
<b>defaultValuesForSkillProperties</b>	Object	<b>Optional, default: null</b> – Specifies an object with default values used for all properties of Skill objects except ID.
<b>definedAllocationLinksVisibleInActivitiesView</b>	boolean	<b>Optional, default: false</b> – If set to true and the options linksVisibleInActivitiesView and allocationRowsVisibleInActivitiesView are also true, then links that are defined between allocations are shown additionally.
<b>definedAllocationLinksVisibleInResourcesView</b>	boolean	<b>Optional, default: false</b> – If set to true and the option linksVisibleInResourcesView is also true, then links that are defined between allocations are shown

Widget Option Name	Type	Description
		instead of calculated allocation links defined by activity links.
<b>definedAllocationLinksVisibleInSkilledResourcesView</b>	boolean	<b>Optional, default: false</b> – If set to true and the option <code>linksVisibleInSkilledResourcesView</code> is also true, then links that are defined between allocations are shown instead of calculated allocation links defined by activity links.
<b>detailedActivityConstraintSymbolsEnabled</b>	boolean	<b>Optional, default: true</b> – If set to true, there will be shown different symbols for the constraint dates depending on their constraint types: <ul style="list-style-type: none"> <li>• EarliestStart: </li> <li>• LatestStart: </li> <li>• MustStartOn: </li> <li>• EarliestEnd: </li> <li>• LatestEnd: </li> <li>• MustEndOn: </li> </ul> <p>Otherwise, a simple down arrow will be shown: .</p> <p>Please consider to set the option <code>topRowMarginInTimeArea</code> when using detailed symbols.</p>
<b>detailedAllocationConstraintSymbolsEnabled</b>	boolean	<b>Optional, default: true</b> – If set to true, there will be shown different symbols for the constraint dates depending on their constraint types: <ul style="list-style-type: none"> <li>• EarliestStart: </li> <li>• LatestStart: </li> <li>• MustStartOn: </li> <li>• EarliestEnd: </li> <li>• LatestEnd: </li> <li>• MustEndOn: </li> </ul> <p>Otherwise, a simple down arrow will be shown: .</p> <p>Please consider to set the option <code>topRowMarginInTimeArea</code> when using detailed symbols.</p>
<b>dragDatesLimitingInteraction</b>	boolean	<b>Option, default: false</b> – If set to true, then bars cannot be dragged before the value in the property <code>EarliestDragStart</code> and later than <code>LatestDragEnd</code> , respectively.

Widget Option Name	Type	Description
<b>editable</b>	boolean	<b>Optional, default: true</b> – If set to false, nothing can be edited.
<b>end</b>	Date   string	<p><b>Optional, default: actual value of option start plus 4 weeks</b> – End date of the considered time area.</p> <p>When this option is not set on first rendering, then a warning is triggered (see callback onLogWarning).</p> <p>It is strongly recommended to set start and end together in one option call using a literal object as the argument. This way VSW reacts faster.</p>
<b>entitiesTableCellContentTopOffset</b>	number > 0	<b>Optional, default: 21</b> – Top offset for cell content in table cells of entities table. This number is valid for the base line of the first line of text inside the table cell and is only taken into account when it is lower than half of default row height and half of an optional row maximum height.
entitiesTableHeaderBackgroundColor		<b>Deprecated! See option entitiesTableHeaderBackgroundColor.</b>
entitiesTableHeaderColumnSeparatorColor	)	<b>Deprecated! See option entitiesTableHeaderColumnSeparatorColor.</b>
entitiesTableHeaderHighlightingColor		<b>Deprecated! See option entitiesTableHeaderHighlightingColor.</b>
entitiesTableHeaderTextColor		<b>Deprecated! See option entitiesTableHeaderTextColor.</b>
<b>entitiesTableSymbolColumnBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – If set then the symbol column of the entities table will show this color in the background.
<b>entitiesTableSymbolColumnTitleBackgroundColor</b>	string (CSS color value)	<b>Optional, default: value of option entitiesTableSymbolColumnBackgroundColor</b> – If set then the symbol column title of the entities table will show this color in the background when the option entitiesTableSymbolColumnTitleVisible is set to true.
<b>entitiesTableSymbolColumnTitleSymbolsIDs</b>	string[]	<b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the entities table in the title cell of the symbol column. They will only appear when the option

Widget Option Name	Type	Description
		<p>entitiesTableSymbolColumnTitleVisible is set to true.</p> <p>The symbols will be arranged one below the other. However, if the cell is not high enough to hold all symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.</p> <p>An empty string ("" ) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve space for a symbol that may be shown later.</p> <p>Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.</p>
entitiesTableSymbolColumnTitleVisible	boolean	<b>Optional, default: false</b> – If set to true, the symbols specified in the option entitiesTableSymbolColumnTitleSymbolIDs will be displayed in the title cell of the symbol column, provided the option entitiesTableSymbolColumnVisible is also set to true. Otherwise, the title cell will have the same color as defined in the entitiesTableTitleBackgroundColor option.
entitiesTableSymbolColumnVisible	boolean	<b>Optional, default: false</b> – If set to true, a special column at the left of the entities table will be displayed to show the row symbols of the entities.
entitiesTableSymbolColumnWidth	number (≥ 22)	<b>Optional, default: 22</b> – Width of the symbol column in the entities table. If set to a value less than the default, it will be set to the default automatically to ensure that the symbols always remain visible as long as the symbol column is visible.
entitiesTableTitleBackgroundColor	string (CSS color value)	<b>Optional, default: "#646464"</b> – Specifies a color used to color the background of the entities table header.
entitiesTableTitleColumnSeparatorColor	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the column

Widget Option Name	Type	Description
		separators in the entities table header.
<b>entitiesTableTitleHeight</b>	number > 0	<b>Optional, default: 60</b> – Specifies the height of the entities table.
<b>entitiesTableTitleHighlightingColor</b>	string (CSS color value)	<b>Optional, default: "#f7c365"</b> – Specifies the color to be used during the interaction, e.g. to highlight the separation line between two adjacent columns when altering the column widths.
<b>entitiesTableTitleTextColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the text in the entities table header.
<b>entitiesTableTreeViewLineColor</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Determines the color of tree view lines in the entities table. See option <code>entitiesTableTreeViewVisualizationMode</code> .
<b>entitiesTableTreeViewLineDashArray</b>	string (SVG stroke dash array value)	<b>Optional, default: "none"</b> – Pattern of dashes and gaps for drawing the tree view lines in the entities table. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the line is drawn solid. See option <code>entitiesTableTreeViewVisualizationMode</code> .
<b>entitiesTableTreeViewVisualizationMode</b>	number (see enum <a href="#">TreeViewVisualizationMode</a> )	<b>Optional, default: ColoredIndentation</b> – Determines how the tree of objects is visualized in the entities table. See also options <code>entitiesTableTreeViewLineColor</code> and <code>entitiesTableTreeViewLineDashArray</code> .
<b>entitiesTableViewWidth</b>	number	<b>Optional, default: null (means current table width)</b> – This setting defines the width of the entities table view when it becomes visible initially.
<b>entitiesTableVisibleInActivitiesView</b>	boolean	<b>Optional, default: false</b> – This option lets appear/disappear the entities table on the right side in the activities view.
<b>entitiesTableVisibleInResourcesView</b>	boolean	<b>Optional, default: false</b> – This option lets appear/disappear the entities table on the right side in the resources view.

Widget Option Name	Type	Description
<b>entitiesTableVisibleInSkilledResourcesView</b>	boolean	<b>Optional, default: false</b> – This option lets appear/disappear the entities table on the right side in the skilled resources view.
entitiesTableWidth		<b>Deprecated! Renamed to <code>entitiesTableVisibleInResourcesView</code>.</b>
<b>entitiesTitleText</b>	string	<p><b>Optional, default: undefined</b> – This text will be shown in the table header.</p> <p>It will appear only in one the following two cases:</p> <ol style="list-style-type: none"> <li>1. If using the <code>TableRowDefinition</code> objects for defining the table and the property <code>entityTableRowDefinitionIDForTitle</code> is not set.</li> </ol> <p>or</p> <ol style="list-style-type: none"> <li>2. If using the deprecated callback <code>onDetermineColumnDefinitions</code> and there additionally the flag <code>hasColumnTitles</code> is <b>not</b> set in the callback (see there).</li> </ol>
<b>entityHierarchySupplementaryDefinitionID</b>	string	<b>Optional, default: undefined</b> – ID of a <code>HierarchySupplementaryDefinition</code> object that will be used to specify grouping parameters for hierarchy of entity objects.
<b>entityRowSortCodePropertyName</b>	string	<p><b>Optional, default: "SortCode"</b> – Name of a data property to be used as sort criteria while sorting entity rows. The values of the addressed property in the entities can contain strings, numbers, or date values.</p> <p>If using interactive vertical row dragging, the specified data property must contain values of number type.</p> <p>See also option <code>entityRowSortMode</code>.</p>
<b>entityRowSortMode</b>	number (see enum <a href="#">RowSortMode</a> )	<p><b>Optional, default: None</b> – If set to a mode unequal to <code>None</code> allocation rows are sorted ascending or descending.</p> <p>It is a prerequisite to use the ascending mode for dragging entity rows vertically.</p>

Widget Option Name	Type	Description
		See also option <code>entityRowSortCodePropertyName</code> .
<code>entityTableRowDefinitionIDForTitle</code>		<b>Deprecated! See option <code>tableRowDefinitionIDForTitleInEntitiesTable</code>.</b>
<b>firstDayOfWeek</b>	number (allowed values: 0, 1, 2, 3, 4, 5, 6)	<b>Optional, default: undefined</b> – Specifies the first day of a week. 0 = Sunday, 1 = Monday, ... If defined, this option overwrites the settings of the options “weekNumbering” and “locale”, respectively. See also option “weekNumbering”.
<b>fixedTableColumnWidth</b>	number	<b>Optional, default: 30</b> – This setting defines the width of the fixed table column that contains the numeric scale for the curves in each row.
<code>forcedActivityAllowedBar-DragModes</code>		<b>Deprecated! Please use a promise in the callback <code>canDrag</code>.</b>
<code>forcedActivityAllowedRow-DragModes</code>		<b>Deprecated! Please use a promise in the callback <code>canDrag</code>.</b>
<code>forcedAllocationAllowedBar-DragModes</code>		<b>Deprecated! Please use a promise in the callback <code>canDrag</code>.</b>
<code>forcedAllocationAllowed-BarDragModesInActivitiesView</code>		<b>Deprecated! Please use a promise in the callback <code>canDrag</code>.</b>
<code>forcedEntityAllowedRow-DragModes</code>		<b>Deprecated! Please use a promise in the callback <code>canDrag</code>.</b>
<code>forcedResourceAllowedRow-DragModes</code>		<b>Deprecated! Please use a promise in the callback <code>canDrag</code>.</b>
<b>ignoreCalendarOnActivityBarInteractions</b>	boolean	<b>Optional, default: false</b> – If set to true, then the activity calendar is not taken into account when dragging an activity bar.
<b>ignoreCalendarOnAllocationBarInteractions</b>	boolean	<b>Optional, default: false</b> – If set to true, then the resource calendar is not taken into account when dragging an allocation bar.
<b>interactiveActivationOfLogging-Enabled</b>	boolean	<b>Optional, default: false</b> – If set to true, the user can activate the logging by using the keyboard shortcut <code>Shift+Ctrl+Alt+L</code> . The record symbol will appear, the current state of the widget is saved and from then on all calls to the API are recorded. Pressing <code>Shift+Ctrl+Alt+L</code> once again will stop the recording and download a file with the recorded actions. See also option <b>loggingEnabled</b> .

Widget Option Name	Type	Description
<b>interactiveSwitchingOfSortOrder-Enabled</b>	boolean	<p><b>Optional, default: false</b> – If set to true, the user can change the sort order of the rows by clicking or tapping into a column of the table title. When unsorted or sorted by another column, then the sort order is expected to change to ascending. If sorted ascending already, it is expected to change to descending. When sorted descending already, it is expected to change to be unsorted again. The latter mode change can be switched to ascending by using the option <code>rowSortModeNoneEnabledOnInteractiveSwitchingOfSortOrder</code>.</p> <p>The requested mode change triggers the callback <code>onRowSortingChangeRequested</code>. The application will have to change the appropriate sorting options itself to set a realized changed sort mode (see options <code>activity/allocation/entity/resource-RowSortCodePropertyName</code>, and <code>activity/allocation/entity/resource-RowSortMode</code>).</p> <p>The current sorting can be made visible by setting the option <code>sortingIndicatorVisible</code> to true additionally.</p>
<b>intlDateTimeFormatOptionsMap</b>	Map Object	<p><b>Optional, default: null</b> – If set, then the content is used to define options for objects of type <code>Intl.DateTimeFormat</code> (see <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/DateTimeFormat">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/DateTimeFormat</a>).</p> <p>The object itself can be a literal one or a standard Map object. The properties/keys define names for a format, the options of which are defined as value. The value therefore is another object (description see second parameter of constructor function of <code>Intl.DateTimeFormat</code>). The format names can be used in the properties</p>



Widget Option Name	Type	Description
		<p>TableCellDefinition.TextFormat, Activity/Allocation.BarTextFormat, TooltipTemplate.HTMLFormat, and the options defaultActivity/AllocationBarTextFormat. For the case that no format name is defined or it cannot be defined (this is the case for dates shown while dragging or numbers in the scale of curves), you can define a format with the name "default". This is then used in these cases. Every format name must start with a letter and must not contain a dot.</p> <p>The options should not contain a property named timeZone, since this is filled in by the VSW itself using the value of the option "timeZone". Also, the first parameter for the constructor of Intl.DateTimeFormat objects is filled with the value of the option "locale".</p>
intlNumberFormatOptionsMap	Map Object	<p><b>Optional, default: null</b> – If set, then the content is used to define options for objects of type Intl.NumberFormat (see <a href="https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/NumberFormat">https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Intl/NumberFormat</a>).</p> <p>The object itself can be a literal one or a standard Map object. The properties/keys define names for a format, the options of which are defined as value. The value therefore is another object (description see second parameter of constructor function of Intl.NumberFormat). The format names can be used in the properties</p> <p>TableCellDefinition.TextFormat, Activity/Allocation.BarTextFormat, TooltipTemplate.HTMLFormat, and the options defaultActivity/AllocationBarTextFormat. For the case that no format name is defined or it cannot be defined (this is the case for dates shown while dragging or numbers in the scale of curves), you can define a format with the name "default". This is then used</p>


Widget Option Name	Type	Description
		<p>in these cases. Every format name must start with a letter and must not contain a dot.</p> <p>The first parameter for the constructor of Intl.NumberFormat objects is filled with the value of the option "locale".</p>
<b>licenseKey</b>	string	<b>Required</b> – Without a license key, the widget will not work at all. Please contact <a href="#">NETRONIC</a> to get a license. This option must be set at the very beginning of the widget initialization and cannot be changed later at runtime.
<b>linesShownInLoadCurvePanels</b>	boolean	<b>Option, default: false</b> – If this option is set to true, in all load curve panels horizontal auxiliary lines are displayed for each tick mark of the numerical scales in the table. These lines help the user to read the curve values.
<b>linksVisibleInActivitiesView</b>	boolean	<b>Option, default: true</b> – If set to false, the activities view does not show links. When true, it shows at least activity links. See also option defined- <code>AllocationLinksVisibleInActivitiesView</code> .
<b>linksVisibleInResourcesView</b>	boolean	<b>Option, default: false</b> – If set to true, the resources view shows links. See also option defined- <code>AllocationLinksVisibleInResourcesView</code> .
<b>linksVisibleInSkilledResourcesView</b>	boolean	<b>Option, default: false</b> – If set to true, the skilled resources view shows links. See also option defined- <code>AllocationLinksVisibleInSkilledResourcesView</code> .
<b>locale</b>	string (currently possible values: "da" or "da-DK", "de" or "de-DE", "en-GB", "en" or "en-US", "es" or "es-ES", "fi" or "fi-FI", "fr" or "fr-FR", "it" or "it-IT", "ja" or "ja-JP", "nl" or "nl-NL", "no" or "no-NO", "pl" or "pl-PL", "pt-BR", "pt" or "pt-PT",	<p><b>Optional, default: "en-US"</b> – This option will be used for showing the textual parts for date values in the timescale and for formatting date and time values in the timescale and numbers in the numeric scales of curves.</p> <p>You must specify the language at least and can append a country. If the country is not known, then the universal language texts for that locale are used automatically as a fallback. It is also allowed to use both uppercase and lowercase for all letters.</p>

Widget Option Name	Type	Description
	"ru" or "ru-RU", "sv" or "sv-SV", "th" or "th-TH", "zh" or "zh-CN")	
<b>loggingEnabled</b>	boolean	<b>Optional, default: false</b> – If set to true, the record symbol will appear, the current state of the widget is saved and from then on all calls to the API are recorded. Resetting this option to false will stop the recording and download a file with the recorded actions. See also option <b>interactiveActivationOfLoggingEnabled</b> .
<b>loggingVerboseLevel</b>	number	<b>Optional, default: 0</b> – The default verbose level for logging omits loggings of callback triggerings that occur often. If set to 10, then the callbacks canDrag and onShowTooltip are logged additionally. If set to 20, also the callback onDrag is logged. If set to 30, also the callbacks visibilityFilter and compareObjects are logged.
mainViewAreaVisible		<b>Deprecated! Renamed to mainViewAreaVisibleInResourcesView.</b>
<b>mainViewAreaVisibleInActivitiesView</b>	boolean	<b>Optional, default: true</b> – When set to false, then in activities view the main view area is invisible. The main view area contains the rows for activities with ViewArea set to Main. If topViewAreaVisibleInActivitiesView is also false, then the main view area will be visible nevertheless.
<b>mainViewAreaVisibleInLoadsView</b>	boolean	<b>Optional, default: true</b> – When set to false, then in loads view the main view area is invisible. The main view area contains the rows for resources with ViewArea set to Main. If topViewAreaVisibleInLoadsView is also false, then the main view area will be visible nevertheless.
<b>mainViewAreaVisibleInResourcesView</b>	boolean	<b>Optional, default: true</b> – When set to false, then in resources view the main view area is invisible. The main view area contains the rows for resources with ViewArea set to Main. If topViewAreaVisibleInResourcesView

Widget Option Name	Type	Description
		is also false, then the main view area will be visible nevertheless.
<b>maximumSnapDistance</b>	number	<b>Optional, default: 8</b> – Maximum distance in pixels of a currently dragged bar to a snap target, within which a dragged bar will get snapped to the snap target.
<b>maximumTimeResolutionUnit</b>	string (one of "second", "minute", "hour", "day", "week", "month", "quarter", "year") or number (see enumeration <a href="#">TimeUnit</a> )	<p><b>Optional, default: value of option timeStepUnit or "second"</b> – Unit for maximum time resolution in the time area. Used together with option maximumTimeResolutionUnitFactor.</p> <p>Neither interactively nor by using the method setTimeResolutionForView can the time area display a finer time resolution than defined here.</p> <p>When you set this option and do not set the options timeStepUnit/-Factor, this value here also changes the default value of timeStepUnit! This is done for compatibility reasons.</p>
<b>maximumTimeResolutionUnitFactor</b>	number ( $\geq 1$ )	<p><b>Optional, default: value of option timeStepUnitFactor or 1</b> – Number of units for maximum time resolution in the time area. Integer values are recommended. Used together with option maximumTimeResolutionUnit.</p> <p>When you set this option and do not set the options timeStepUnit/-Factor, this value here also changes the default value of timeStepUnitFactor! This is done for compatibility reasons.</p>
<b>maximumTopViewAreaHeightRatio</b>	number ( $-0.8 \leq n \leq 0.8$ )	<b>Optional, default: 0.5</b> – If positive, this value determines the maximum height of the top view area expressed as a fraction of the full view height. If negative, the absolute value instead determines the maximum height of the common view area (so the common view area can be used for unassigned resource allocations alternatively. Vertical scroll bars are shown in both view areas if necessary.
<b>multipleBarDraggingEnabled</b>	boolean	<b>Optional, default: false</b> – If set to true, all selected bars are dragged at once. Also see callback options canDrag, onDragStart, onDrop. Currently, the allocation/activity

Widget Option Name	Type	Description
		properties EarliestDragStart and LatestDragEnd are not supported when dragging multiple bars. The allocation property SuitableResourceIDs is supported. When dragging starts, the allowed drag modes are inherited by default from the allocation/activity that is being dragged directly. This is modifiable by using the callback canDrag.
<b>multipleSelectionEnabled</b>	boolean   number (allowed values: false, true, 2)	<b>Optional, default: true</b> – If set to true or 1 multiple bars can be selected all at once. When using the selection rectangle, a bar is selected even if it is only partially inside the rectangle. Additionally, if set to 2, the behavior when dragging a selection rectangle from left to right is different from that when dragging from right to left. In the first case, a bar is only selected when it is completely inside the rectangle. In the latter case, it will be selected even if it is only partially inside the rectangle.
<b>nonworkingTimesCalendarIDs</b>	string[]	<b>Optional, default: undefined</b> – This option defines the IDs of calendars considered in the calculation of the common non-working time, means that a non-working time will get invisible only, when all calendars contain it. See option nonworkingTimeVisible.
<b>nonworkingTimeVisible</b>	boolean	<b>Optional, default: true</b> – This option defines whether the common non-working time is visible. The common time is calculated by all calendar information that are relevant to the visualization. Therefore, the calendars of visible activities and resources are used or alternatively, the IDs of calendars considered are specified via option nonworkingTimesCalendarIDs.
<b>objectHighlightFlashingEnabled</b>	boolean	<b>Optional, default: true</b> – Specifies whether or not the frame displayed around an object that has been scrolled to by using the method scrollToObject should flash or around objects after using the method highlightObjects.

Widget Option Name	Type	Description
<b>objectHighlightingColor</b>	string (CSS color value)	<b>Optional, default: "#7f0000"</b> – Color of the frame displayed around an object that has been scrolled to by using the method <code>scrollToObject</code> or around objects after using the method <code>highlightObjects</code> .
<b>onCollapseStateChangedTriggeredByUpdateCalls</b>	boolean	<b>Deprecated! Renamed to <code>triggeringOfOnCollapseStateChangedByUpdateCallsEnabled</code>.</b>
<b>pastBackgroundFillColor</b>	string	<b>Optional, default: "rgba(0,0,0,0.2)"</b> – This option defines the color of the darkened area between timescale start and value of the option <code>currentDate</code> .
<b>pastBackgroundLineColor</b>	string	<b>Optional, default: "darkgrey"</b> – This option defines the color of the date line at the value of the option <code>currentDate</code> .
<b>pastBackgroundLineDashArray</b>	string	<b>Optional, default: "1,1"</b> – Pattern of dashes and gaps for drawing the date line at the value of the option <code>currentDate</code> . For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the line is drawn solid.
<b>pastBackgroundLineWidth</b>	number	<b>Optional, default: 1</b> – This option defines the width of the date line at the value of the option <code>currentDate</code> .
<b>preventDefaultOnContextMenuEvents</b>	boolean	<b>Option, default: true</b> – This option determines whether "contextmenu" triggered by the browser's DOM should get a call to <code>preventDefault()</code> . If set to false, then the system default behavior is not prevented (useful for Microsoft Dynamics 365 Finance and Operations).
<b>progressBarHeight</b>	number $\geq 0$	<b>Optional, default: 4</b> – Determines the height of all progress bars within allocation bars and activity bars.

Widget Option Name	Type	Description
<b>progressBarWidthCalculationMode</b>	number (see <a href="#">ProgressBarWidthCalculationMode</a> )	<p><b>Option, default:</b> <b>ProgressBarWidthCalculationMode.ConsiderWorkingTimesOnly</b> – This option determines how the widths of the progress bars are calculated. Possible values:</p> <ul style="list-style-type: none"> <li>ConsiderWorkingTimesOnly – If this value is used, it is assumed that there is no progress during non-working times.</li> <li>ConsiderWorkingAndNonworkingTimes – If this value is used, it is assumed that there is progress during both working and non-working times.</li> </ul>
<b>reducedBarTopOffsetAndHeightScaleFactor</b>	number (> 0, <= 1)	<p><b>Optional, default: 1</b> – This option is used as a scale factor for bars where the flag ReducedHeight in property BarDesign of an Activity or Allocation object is set to true or Allocation object is set to true or when visualizing allocations of other skill in SkilledResources view (see option allocationBarDesignOfOtherSkill).</p>
<b>releaseDueDateConnectionsVisible</b>	boolean	<p><b>Optional, default: false</b> – If set to true and an activity has set both a ReleaseDate and a DueDate, a line will be displayed to visually connect both dates:</p> 
<b>resetValueForDifferentialUpdate</b>	any	<p><b>Optional, default: null</b> – Determines a value that will be replaced by “undefined” for differential updates when using an update method with flag UpdateModes.DifferentialValues set.</p>
<b>resourceHierarchySupplementaryDefinitionID</b>	string	<p><b>Optional, default: undefined</b> – ID of a HierarchySupplementaryDefinition object that will be used to specify grouping parameters for hierarchy of resource objects</p>
<b>resourceHierarchySupplementaryDefinitionIDInLoadsView</b>	string	<p><b>Optional, default: undefined</b> – ID of a HierarchySupplementaryDefinition object that will be used to specify grouping parameters for hierarchy of resource objects in loads view.</p>

Widget Option Name	Type	Description
<b>resourceRowSortCodePropertyName</b>	string	<p><b>Optional, default: "SortCode"</b> – Name of a data property to be used as sort criteria while sorting resource rows. The values of the addressed property in the resources can contain strings, numbers, or date values.</p> <p>If using interactive vertical row dragging, the specified data property must contain values of number type.</p> <p>See also option <code>resourceRowSortMode</code>.</p>
<b>resourceRowSortMode</b>	number (see enum <a href="#">RowSortMode</a> )	<p><b>Optional, default: None</b> – If set to a mode unequal to None resource rows are sorted ascending or descending.</p> <p>It is a prerequisite to use the ascending mode for dragging resource rows vertically.</p> <p>See also option <code>resourceRowSortCodePropertyName</code>.</p>
<b>resourceTableRowDefinitionIDForTitle</b>		<b>Deprecated! See renamed option <code>tableRowDefinitionIDForTitleInResourcesView</code>.</b>
<b>rowSortModeNoneEnabledOn-InteractiveSwitchingOfSortOrder</b>	boolean	<p><b>Optional, default: true</b> – If set to true, then a sort mode can interactively be switched back to unsorted. See option <code>interactiveSwitchingOfSortOrder-Enabled</code>.</p>
<b>scrollOffsetsChangedCallbackTimeDelay</b>	number $\geq 0$	<p><b>Optional, default: 500</b> – This value determines the time delay in milliseconds for triggering the callbacks <code>onVerticalScrollOffsetChanged</code> and <code>onTimeAreaViewParametersChanged</code>.</p>
<b>scrollToObjectAnimation-Enabled</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then scrolling to the target position is animated when using the method <code>scrollToObject</code>.</p>
<b>scrollToObjectHighlightFlashingEnabled</b>		<b>Deprecated! See renamed option <code>objectHighlightFlashingEnabled</code>.</b>
<b>scrollToObjectHighlightingColor</b>		<b>Deprecated! See renamed option <code>objectHighlightingColor</code>.</b>
<b>separationLinesInColoredIndentation</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then vertical separation lines are shown between the colored indentation rectangles at the</p>



Widget Option Name	Type	Description
		beginning of the scrollable part of the tables.
<b>selectionColor</b>	string (CSS color value)	<b>Optional, default: "#ffa000"</b> – Specifies a color used to highlight selected bars, links or table rows.
<b>sortingIndicatorVisible</b>	boolean	<b>Optional, default: false</b> – If set to true, then the current sort mode is visible in the table title by showing a small arrow in the appropriate column: Up arrow for ascending order and down arrow for descending order. See also options <code>interactiveSwitchingOfSortOrder-Enabled</code> , <code>activity/allocation/entity/resourceRowSortCodePropertyName</code> , and <code>activity/allocation/entity/resourceRowSortMode</code> .
<b>splitterHighlightingColor</b>	string (CSS color value)	<b>Optional, default: "#ffa000"</b> – Specifies a color used to highlight the splitters when a splitter is dragged. This refers to the splitters between the table or entities table and the Gantt area.
<b>start</b>	Date   string	<b>Optional, default: beginning of today when widget is instantiated</b> – Start of the considered time area.  When this option is not set on first rendering, then a warning is triggered (see callback <code>onLogWarning</code> ).  It is strongly recommended to set start and end together in one option call using a literal object as the argument. This way VSW reacts faster.
<b>subRowDistanceInTimeArea</b>	number > 0	<b>Optional, default: 5</b> – Vertical distance between two bars in pixels. See also <code>topRowMarginInTimeArea</code> and <code>bottomRowMarginInTimeArea</code> . Please have in mind that symbols are drawn inside this distance.
<b>suitableResourcesOverlayColor</b>	string (CSS color value)	<b>Optional, default: "transparent"</b> – This option determines the color that is added to resource rows that are mentioned in the allocation property <code>SuitableResourceIDs</code> when dragging. See option <code>unsuitableResourcesOverlayColor</code> .

Widget Option Name	Type	Description
<b>symbolColumnBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – If set then the symbol column of the activities/resources table will show this color in the background.
<b>symbolColumnTitleBackgroundColor</b>	string (CSS color value)	<b>Optional, default: value of option symbolColumnBackgroundColor</b> – If set then the symbol column title of the activities/resources table will show this color in the background when the option symbolColumnTitleVisible is set to true.
<b>symbolColumnTitleSymbolIDs</b>	string[]	<p><b>Optional, default: undefined</b> – Array of identifiers of the symbols to be shown in the table in the title cell of the symbol column. They will only appear when the option symbolColumnTitleVisible is set to true and option titleText is not set, so that the table title shows columns.</p> <p>The symbols will be arranged one below the other. However, if the cell is not high enough to hold all symbols, then the remaining symbols are also arranged side-by-side. If this still does not fit, an additional “show more” symbol will be displayed.</p> <p>An empty string ("" ) will cause an “empty” symbol to be displayed. By this placeholder, you can reserve space for a symbol that may be shown later.</p> <p>Please note: Each symbol will be resized to an image with a width and height of 16 pixels each at a zoom level of 100%.</p>
<b>symbolColumnTitleVisible</b>	boolean	<b>Optional, default: false</b> – If set to true, the symbols specified in the option symbolColumnTitleSymbolIDs will be displayed in the title cell of the symbol column, provided the option symbolColumnVisible is also set to true. Otherwise, the title cell will have the same color as defined in the tableTitleBackgroundColor option.
<b>symbolColumnVisible</b>	boolean	<b>Optional, default: false</b> – If set to true, a special column at the left of the table will be displayed to show the

Widget Option Name	Type	Description
		row symbols of the activities in the Activities view and of the resources in the Resources or Loads view.
<b>symbolColumnWidth</b>	number (≥ 22)	<b>Optional, default: 22</b> – Width of the symbol column in the Activities, Resources and Loads view. If set to a value less than the default, it will be set to the default automatically to ensure that the symbols always remain visible as long as the symbol column is visible.
<b>tableCellContentTopOffset</b>	number > 0	<b>Optional, default: 21</b> – Top offset for cell content in table cells of left table. This number is valid for the base line of the first line of text inside the table cell and is only taken into account when it is lower than half of default row height and half of an optional row maximum height.
tableHeaderBackgroundColor		<b>Deprecated! See option tableTitleBackgroundColor.</b>
tableHeaderColumnSeparatorColor		<b>Deprecated! See option tableTitleColumnSeparatorColor.</b>
tableHeaderHighlightingColor		<b>Deprecated! See option tableTitleHighlightingColor.</b>
tableHeaderTextColor		<b>Deprecated! See option tableTitleTextColor.</b>
<b>tableRowDefinitionIDForTitleInActivitiesView</b>	string	<b>Optional, default: value of option defaultActivityTableRowDefinitionID</b> – ID of a TableRowDefinition object that will be used to show the table title in the activities view. In parallel, it is currently only possible to interactively change the column widths for the TableRowDefinition object that is referenced here.
<b>tableRowDefinitionIDForTitleInEntitiesTable</b>	string	<b>Optional, default: value of option defaultEntityTableRowDefinitionID</b> – ID of a TableRowDefinition object that will be used to show the table title in the entities table. In parallel, it is currently only possible to interactively change the column widths for the TableRowDefinition object that is referenced here.
<b>tableRowDefinitionIDForTitleInLoadsView</b>	string	<b>Optional, default: value of option tableRowDefinitionIDForTitleInResourcesView or, when undefined, value of option defaultResourceTableRowDefinitionID</b>

Widget Option Name	Type	Description
		<b>D</b> – ID of a TableRowDefinition object that will be used to show the table title in the resources view. In parallel, it is currently only possible to interactively change the column widths for the TableRowDefinition object that is referenced here.
<b>tableRowDefinitionIDForTitleInResourcesView</b>	string	<b>Optional, default: value of option defaultResourceTableRowDefinitionID</b> – ID of a TableRowDefinition object that will be used to show the table title in the resources view. In parallel, it is currently only possible to interactively change the column widths for the TableRowDefinition object that is referenced here.
<b>tableRowDefinitionIDForTitleInSkilledResourcesView</b>	string	<b>Optional, default: value of option defaultSkilledResourceTableRowDefinitionID</b> – ID of a TableRowDefinition object that will be used to show the table title in the skilled resources view. In parallel, it is currently only possible to interactively change the column widths for the TableRowDefinition object that is referenced here.
<b>tableTitleAndTimescaleHeight</b>	number >= 0	<b>Optional, default: 60</b> – Specifies the height of the left table and the timescale.
<b>tableTitleBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Specifies a color used to color the background of the table header of the Gantt diagram.
<b>tableTitleColumnSeparatorColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the column separators in the table header of the Gantt diagram.
<b>tableTitleHighlightingColor</b>	string (CSS color value)	<b>Optional, default: "#f7c365"</b> – Specifies the color to be used during the interaction, e.g. to highlight the separation line between two adjacent columns when altering the column widths.
<b>tableTitleTextColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the text in the table header of the Gantt diagram.
<b>tableViewWidth</b>	number	<b>Optional, default: null (means table width)</b> – This option defines the width of the table view in all views. A change

Widget Option Name	Type	Description
		to the vertical splitter is not changing this option.
<b>tableViewWidthInActivitiesView</b>	number	<b>Optional, default: null (means table width)</b> – This option defines the width of the table view in activities view. A change to the vertical splitter is not changing this option.
<b>tableViewWidthInLoadsView</b>	number	<b>Optional, default: null (means table width)</b> – This option defines the width of the table view in loads view. A change to the vertical splitter is not changing this option.
<b>tableViewWidthInResourcesView</b>	number	<b>Optional, default: null (means table width)</b> – This option defines the width of the table view in resources view. A change to the vertical splitter is not changing this option.
<b>tableViewWidthInResourcesView</b>	number	<b>Optional, default: null (means table width)</b> – This option defines the width of the table view in resources view. A change to the vertical splitter is not changing this option.
<b>tableViewWidthInSkilledResource sView</b>	number	<b>Optional, default: null (means table width)</b> – This option defines the width of the table view in skilled resources view. A change to the vertical splitter is not changing this option.
<b>tableViewWidthsSynchronized</b>	boolean	<b>Optional, default: true</b> – This option defines whether an interactive change of the table view width sets the view width of all views or not.
tableWidth		<b>Deprecated! Calculated automatically when using TableRowDefinition objects.</b>
<b>timeAreaBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the background of the time area.
<b>timeAreaPanningMode</b>	number (see enum <a href="#">PanningMode</a> )	<b>Optional, default: 3</b> – Specifies, how the widget reacts to user interactions inside the empty space of the time area.  <b>Note:</b> When panning with the mouse, this option is only considered if the <code>multipleSelectionEnabled</code> option is set to false.
<b>timescaleBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "#646464"</b> – Specifies a color used to color the background of the timescale.

Widget Option Name	Type	Description
<b>timescaleHighlightingColor</b>	string (CSS color value)	<b>Optional, default: "#f7c365"</b> – Specifies the color to be used during the interaction on the timescale, e.g. to highlight the time period under the mouse cursor.
<b>timescaleInteractionMode</b>	number (see enum <a href="#">TimescaleInteractionModes</a> )	<b>Optional, default: TimescaleInteractionModes.Default</b> – Specifies which interactions are allowed on the timescale.
timescaleInteractionsEnabled		<b>Deprecated! Use option timescaleInteractionMode instead.</b>
<b>timescaleNavigationMode</b>	number (see <a href="#">TimescaleNavigationMode</a> )	<b>Optional, default: 0</b> – Mode of navigation in the timescale.
<b>timescaleTextColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the text in the timescale.
<b>timescaleTickColor</b>	string (CSS color value)	<b>Optional, default: "white"</b> – Specifies a color used to color the ticks in the timescale.
<b>timescaleWeekendBackgroundColor</b>	string (CSS color value)	<b>Optional, default: "#888888"</b> – Specifies a color used to color the background of the weekend cells of the timescale.
<b>timeStepUnit</b>	string (one of "second", "minute", "hour", "day", "week", "month", "quarter", "year") or number (see enumeration <a href="#">TimeUnit</a> )	<p><b>Optional, default: value of option maximumTimeResolutionUnit or "second"</b> – Unit for time steps on horizontal drag interactions of bars. Used together with option timeStepUnitFactor.</p> <p>When using a time unit "day" or above, the stepping is done day-by-day without concerning nonworking times within the period.</p> <p>When you set this option and do not set the options maximumTimeResolutionUnit/-Factor, the value here also changes the default value of maximumTimeResolutionUnit! This is done for compatibility reasons.</p> <p><b>Attention!</b> Currently, the dates of the bars as well as the dates in the calendar must not be defined finer than this unit together with the option timeStepUnitFactor indicate. Otherwise, unexpected jumps may occur when moving bars.</p>

Widget Option Name	Type	Description
<b>timeStepUnitFactor</b>	number ( $\geq 1$ )	<p><b>Optional, default: value of option maximumTimeResolutionUnitFactor or 1</b> – Number of units for a single time step on horizontal drag interactions of bars. Used together with option timeStepUnit. Integer values are recommended.</p> <p>When you set this option and do not set the options maximumTimeResolutionUnit/-Factor, the value here also changes the default value of maximumTimeResolutionUnitFactor! This is done for compatibility reasons.</p> <p><b>Attention!</b> Currently, the dates of the bars as well as the dates in the calendar must not be defined finer than this factor together with the option timeStepUnit indicate. Otherwise, unexpected jumps may occur when moving bars.</p>
<b>timeZone</b>	string	<p><b>Optional, default: undefined</b> – This option determines the time zone for which dates are shown in the timescale. If set to undefined, then local time zone of the browser is used. When using this option, it is necessary to load the JavaScript libraries Moment.js and Moment Timezone at application startup. The possible values are all the ones that Moment Timezone knows (based on <a href="#">IANA TimeZone database</a>, e.g. "Europe/Berlin", see also <a href="https://en.wikipedia.org/wiki/List_of_tz_database_time_zones">https://en.wikipedia.org/wiki/List_of_tz_database_time_zones</a> for a detailed list of allowed zone names).</p>
<b>titleText</b>	string	<p><b>Optional, default: undefined</b> – This text will be shown in the table header.</p> <p>It will appear only in one the following two cases:</p> <ol style="list-style-type: none"> <li>1. If using the TableRowDefinition objects for defining the table and the property activityTableRowDefinitionIDForTitle or resourceTableRowDefinitionIDForTitle appropriate to the</li> </ol>

Widget Option Name	Type	Description
		<p>corresponding view type is <b>not</b> set.</p> <p>or</p> <p>2. If using the deprecated callback <code>onDetermineComumnDefinitions</code> and there additionally the flag <code>hasColumnTitles</code> is set to <b>false</b> in the callback (see there).</p>
<b>tonedDownOverlayColor</b>	string (CSS color value)	<p><b>Optional, default:</b> <b>"rgba(64,64,64,0.5)"</b> – This option is used as the overlay color for bars where the flag <code>TonedDownColoring</code> in property <code>BarDesign</code> of an Activity or Allocation object is set to true or when visualizing allocations of other skill in <code>SkilledResources</code> view (see option <code>allocationBarDesignOfOtherSkill</code>).</p>
<b>tooltipDelay</b>	number	<p><b>Optional, default: 500</b> – This option determines delay in milliseconds until a tooltip gets visible.</p>
<b>topBarSymbolsVisible</b>	boolean	<p><b>Optional, default: true</b> – If set to false, then no symbols are shown at the top left and top right of allocation bars and activity bars.</p>
<b>topRowMarginInTimeArea</b>	number > 0	<p><b>Optional, default: 10</b> – Height of the margin between the top row border and bars in pixels. See also <code>bottomRowMarginInTimeArea</code> and <code>subRowDistanceInTimeArea</code>. Please have in mind that symbols are drawn inside this margin.</p> <p>When one of the options <code>detailedActivity/AllocationConstraintSymbolsEnabled</code> is set to true, then the value here should be set to a value of 15 or above in order to avoid an vertical overlap.</p>
<b>topViewAreaVisible</b>		<p><b>Deprecated! See renamed option <code>topViewAreaVisibleInResourcesView</code>.</b></p>
<b>topViewAreaVisibleInActivitiesView</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then activities in the activities view are shown in a separate top view area, that have the attribute <code>ViewArea</code> set to Top. See also options <code>mainViewAreaVisibleInActivitiesView</code></p>



Widget Option Name	Type	Description
		and maximumTopViewArea-HeightRatio.
<b>topViewAreaVisibleInLoadsView</b>	boolean	<b>Optional, default: false</b> – If set to true, then resources in the loads view are shown in a separate top view area, that have the attribute ViewArea set to Top. See also options mainViewAreaVisibleInLoadsView and maximumTopViewArea-HeightRatio.
<b>topViewAreaVisibleInResourcesView</b>	boolean	<b>Optional, default: false</b> – If set to true, then resources in the resources view are shown in a separate top view area, that have the attribute ViewArea set to Top. See also options mainViewAreaVisibleInResourcesView and maximumTopViewAreaHeightRatio.
<b>treeViewLineColor</b>	string (CSS color value)	<b>Optional, default: "black"</b> – Determines the color of tree view lines in the left table. See option treeVisualizationMode.
<b>treeViewLineDashArray</b>	string (SVG stroke dash array value)	<b>Optional, default: "none"</b> – Pattern of dashes and gaps for drawing the tree view lines in the left table. For further information, please see <a href="https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty">https://www.w3.org/TR/SVG11/painting.html#StrokeDasharrayProperty</a> or <a href="https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray">https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/stroke-dasharray</a> . The value "none" indicates that no dashing is used. In this case, the line is drawn solid. See option treeVisualizationMode.
<b>treeVisualizationMode</b>	number (see enum <a href="#">TreeVisualizationMode</a> )	<b>Optional, default: ColoredIndentation</b> – Determines how the tree of objects is visualized in the left table. See also options treeViewLineColor and treeViewLineDashArray.
<b>triggeringOfOnCollapseStateChangedByUpdateCallsEnabled</b>	boolean	<b>Optional, default: true</b> – Defines whether the callback onCollapseStateChanged is also triggered by calling update methods.
<b>triggeringOfOnClickedInTimeAreaOfRow</b>	boolean	<b>Optional, default: false</b> – When set to true, then the callbacks onClicked and onDoubleClicked are triggered on time

Widget Option Name	Type	Description
		area background of a row and in the curve pane area of a row.
<b>triggeringOfOnShowTooltipFor-EntriesInBarsEnabled</b>	Boolean	<b>Optional, default: false</b> – If set to true, then the callback onShowTooltip is triggered for each entry of an activity bar or an allocation bar.
<b>unsuitableResourcesOverlayColor</b>	string (CSS color value)	<b>Optional, default: "rgba(0,0,0,0.2)"</b> – This option determines the color that is added to resource rows that are NOT mentioned in the allocation property SuitableResourceIDs when dragging. See option suitableResourcesOverlayColor.
<b>version</b>	string	<b>Read only</b> – This option holds the version number of the widget set by NETRONIC. Usually it is formatted using the semantic versioning format "MAJOR.MINOR.PATCH" (see also <a href="https://semver.org/">https://semver.org/</a> ).
<b>viewType</b>	number (see enum <a href="#">ViewType</a> )	<b>Optional, default: ViewType.Activities</b> – This option determines the type of view that is shown: activities view, resources view, or loads view.
<b>visualZoomFactor</b>	number	<b>Optional, default: 1.0</b> – Factor used to zoom in (>1) and out (<1) the whole widget. Values <= 0 will be ignored.
<b>watermarkOpacity</b>	number (>= 0.0, <= 1.0)	<b>Optional, default: 0.2</b> – Opacity of the watermark. See also option watermarkSymbolID.
<b>watermarkSymbolID</b>	string	<b>Optional, default: undefined</b> – Identifier of the symbol to be shown in the time area of the Gantt chart. The symbol is stretched while maintaining the ratio between width and height of the symbol so that it is as large as possible. See also option watermarkOpacity.
<b>weekNumbering</b>	string null (currently possible values: "ISO8601", "USA")	<b>Optional, default: undefined</b> – This option determines the week numbering scheme (ISO8601: January 4 must be in the first week of the year, USA: January 1 must be in the first week of the year).  This option determines also the first day of the week (ISO8601: Monday, USA: Sunday). If set to null, then the implicit setting of the option "locale"

Widget Option Name	Type	Description
		is used. And that setting can also be overwritten by the option "firstDayOfWeek" (see there for more details).
<b>workDate</b>	Date   string   null	<p><b>Optional, default: null</b> – Date on which the work date line will be displayed. If outside of the time range between start and end of the time area, then no line will be visible.</p> <p>Please note: The work date line is a simple line only. There are no further properties like color, line width, or line pattern to be set. If such properties are needed, then a <a href="#">DateLine</a> object should be used.</p>
<b>workDateLineCaption</b>	string	<b>Optional, default: ""</b> – Text to be displayed at the work date line.
<b>worldViewExtent</b>	number	<b>Optional, default: 150</b> – Defines the extent of the world view in pixels.
<b>worldViewPosition</b>	number (see enum <a href="#">WorldViewPosition</a> )	<b>Optional, default: Bottom</b> – Defines the position of the world view within the widget.
<b>worldViewVisible</b>	boolean	<p><b>Optional, default: false</b> – If set to true, then a world view is visible at the bottom of the Gantt chart. Only the table row background colors and bar colors are shown. Also date lines and separation lines between left table, timescale, top view area are shown. Additionally, selections are shown and frames for the visible parts shown in the widget (separately for table and time area). These frames can also be dragged to modify the visible parts.</p>

## 4.2 Callbacks

For simplicity reasons, we have implemented callbacks instead of events. They can be set in the same way as all other “regular” options.

When we speak of a Promise object within of the callbacks, you can use a standard Promise object or a jQuery Promise (see <http://api.jquery.com/promise/>).

Callback Name	Type	Description
<b>canDrag</b>	Function	<b>Optional, default: undefined</b> – This function is called when the user is moving the mouse

Callback Name	Type	Description
		<p>cursor over an activity/allocation/entity or touches an activity/allocation/entity with a finger.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object,   "visualType" : <a href="#">VisualType</a>,   "entry"<sup>1</sup> : AllocationEntry,   "entryIndex"<sup>1</sup> : number,   "allowedDragModes" : <a href="#">ActivityBarDragModes</a>   <a href="#">AllocationBarDragModes</a>   <a href="#">RowDragModes</a>, //[in/out]   "promise" : Promise undefined, //[out]   "selectedObjects" : Object[] undefined,   "startPropertyName" : string<sup>2</sup>,   "endPropertyName" : string<sup>2</sup>,   "event" : Event // DOM interface }</pre> <p>If the application returns a Promise object in args.promise, then the allowedDragModes will be updated, when the Promise is resolved or rejected. Rejection is the same as resolving with None. When resolving the promise, the application must provide an argument as following:</p> <pre>{   "allowedDragModes" : <a href="#">ActivityBarDragModes</a>   <a href="#">AllocationBarDragModes</a>   <a href="#">RowDragModes</a> }</pre> <p>As an alternative to the promise, the same was formerly possible by setting the options forcedActivity/AllocationAllowedBarDragModes or forcedActivity/Entity/Resource-AllowedRowDragModes, resp., to None.</p> <p>If the option multipleBarDraggingEnabled is set to true and more than one bar is selected, the property selectedObjects will contains all selected objects, so that the application can determine the value for allowedDragModes.</p> <p>If the mouse touches a date symbol or bar of an activity, then the properties start/endPropertyName contain the name of the property to be modified when a dragging or dropping the symbol or bar, resp.</p>

<sup>1</sup> Available only if objectType == ObjectType.Allocation or if visualType == VisualType.PeriodHighlighter.

<sup>2</sup> Only set if touching/dragging an activity bar.

Callback Name	Type	Description
		This callback is called only once every time when the mouse enters the visual representation of the object (bar).
<b>canSelect</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when the user moves the mouse cursor onto the graphical representation of an object.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object,   "visualType" : <a href="#">VisualType</a>,   "otherSelectedObjects" : Object[],   "event" : Event, // DOM interface   "cancel" : boolean [out] }</pre>
<b>compareActivities</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when an activity is added or when its parent is changed during its update. Currently, only objects that appear as table rows can be sorted using this callback. The comparison is always performed only between siblings of same object type. The result will determine the sorting of the rows in the view.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>.Activity,   "objectA" : Object,   "objectB" : Object,   "viewType" : ViewType,   "hierarchySupplementaryDefinitionID"<sup>3</sup> : string,   "hierarchyLevelSupplementaryDefinitionIndex"<sup>Fehler! Textmarke nicht definiert.</sup> : number,   "groupingLevelDefinitionIndex"<sup>Fehler! Textmarke nicht definiert.</sup> : number,   "groupingCodeA"<sup>Fehler! Textmarke nicht definiert.</sup> : string,   "groupingCodeB"<sup>Fehler! Textmarke nicht definiert.</sup> : string,   "isALowerThanB" : boolean //[in/out] }</pre> <p>The function should compare objectA and objectB and write the result into isALowerThanB: true, when A is lower than B and false, when A is greater than B.</p> <p>In case of grouping rows objectA and objectB are null and instead groupingCodeA and groupingCodeB together with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex are set.</p>

<sup>3</sup> Only set when this callback is referencing grouping rows. The properties objectA and objectB then always are null.

Callback Name	Type	Description
<b>compareAllocations</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when an allocation is added or when its referenced activity row or resource row is changed during its update and allocation rows are visible. Currently, only objects that appear as table rows can be sorted using this callback. The comparison is always performed only between siblings of same object type. The result will determine the sorting of the rows in the view.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>.Allocation,   "objectA" : Object,   "objectB" : Object,   "viewType" : ViewType,   "isALowerThanB": boolean //[in/out] }</pre> <p>The function should compare objectA and objectB and write the result into isALowerThanB: true, when A is lower than B and false, when A is greater than B.</p>
<b>compareEntities</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when an entity is added or when its parent is changed during its update. The comparison is always performed only between siblings of same object type. The result will determine the sorting of the rows in the view.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>.Entity,   "objectA" : Object,   "objectB" : Object,   "viewType" : ViewType,   "hierarchySupplementaryDefinitionID"<sup>4</sup> : string,   "hierarchyLevelSupplementaryDefinitionIndex"<sup>Fehler! Textmarke nicht definiert.</sup> : number,   "groupingLevelDefinitionIndex"<sup>Fehler! Textmarke nicht definiert.</sup> : number,   "groupingCodeA"<sup>Fehler! Textmarke nicht definiert.</sup> : string,   "groupingCodeB"<sup>Fehler! Textmarke nicht definiert.</sup> : string,   "isALowerThanB": boolean //[in/out] }</pre> <p>The function should compare objectA and objectB and write the result into isALowerThanB: true, when A is lower than B and false, when A is greater than B.</p>

<sup>4</sup> Only set when this callback is referencing grouping rows. The properties objectA and objectB then always are null.

Callback Name	Type	Description
		In case of grouping rows objectA and objectB are null and instead groupingCodeA and groupingCodeB together with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex are set.
compareObjects		<b>Deprecated! For performance reasons; please see options compareActivities/Allocations/Entities/Resources.</b>
compareResources	Function	<p><b>Optional, default: undefined</b> – This function is called when a resource is added or when its parent is changed during its update. The comparison is always performed only between siblings of same object type. The result will determine the sorting of the rows in the view.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>.Resource,   "objectA" : Object,   "objectB" : Object,   "viewType" : ViewType,   "hierarchySupplementaryDefinitionID"<sup>5</sup> : string,   "hierarchyLevelSupplementaryDefinitionIndex"<sup>Fehler! Textmarke nicht definiert.</sup> : number,   "groupingLevelDefinitionIndex"<sup>Fehler! Textmarke nicht definiert.</sup> : number,   "groupingCodeA"<sup>Fehler! Textmarke nicht definiert.</sup> : string,   "groupingCodeB"<sup>Fehler! Textmarke nicht definiert.</sup> : string,   "isALowerThanB": boolean //[in/out] }</pre> <p>The function should compare objectA and objectB and write the result into isALowerThanB: true, when A is lower than B and false, when A is greater than B.</p> <p>In case of grouping rows objectA and objectB are null and instead groupingCodeA and groupingCodeB together with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex are set.</p>
determineGroupingCode	Function	<p><b>Optional, default: undefined</b> – This function is called to determine grouping information like grouping code and long text.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,</pre>

<sup>5</sup> Only set when this callback is referencing grouping rows. The properties objectA and objectB then always are null.

Callback Name	Type	Description
		<pre> "object" : Object, "hierarchySupplementaryDefinitionID" : string, "hierarchyLevelSupplementaryDefinitionIndex" : number, "groupingLevelDefinition" : Object, "groupingLevelDefinitionIndex" : number, "code" : string // [in/out], "text" : string // [out] } </pre>
<b>onClicked</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when an object is clicked by the user.</p> <p><b>Profile:</b></p> <pre> function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object null,   "visualType" : <a href="#">VisualType</a>,   "date" : Date undefined, // date at mouse   cursor   "entry"<sup>1</sup> : AllocationEntry ActivityEntry      PeriodHighlighterEntry,   "entryIndex"<sup>1</sup> : number,   "curve" : Curve, // Only available when     clicked on a curve; the "object" parameter     will then hold the corresponding resource   "periodHighlighter" : PeriodHighlighter,     // Only available when clicked on a     PeriodHighlighterEntry; the "object"     parameter will then hold the corresponding     resource/activity   "cellIndex" : number, // Only available when     clicked on a table cell; zero-based index     of the cell.   "symbolIndex" : number, // Only available when     clicked on a symbol; zero-based index     of the symbol.}   "hierarchySupplementaryDefinitionID"<sup>6</sup> : string,   "hierarchyLevelSupplementaryDefinitionIndex"<sup>6</sup> : number,   "groupingLevelDefinitionIndex"<sup>6</sup> : number,   "groupingCodes"<sup>6</sup> : string[],   "skillID" : string undefined,   "event" : Event // DOM interface } </pre> <p>In case of a grouping row, object is null and instead groupingCodes together with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex are set.</p> <p>In skilled resources view the property skillID is set when clicking on a resource row, allocation row, or allocation bar.</p>
<b>onCloseContextMenu</b>	Function	<p><b>Optional, default: undefined</b> – When a context menu is visible in the application and the user starts a new action elsewhere in the widget, the widget sends this event in order to close the open context menu.</p>

<sup>6</sup> Only set when this callback is referencing a grouping row. The object property then always is null.



Callback Name	Type	Description
<b>onCollapseStateChanged</b>	Function	<p><b>Profile:</b> function ()</p> <p><b>Optional, default: undefined</b> – This function is called when a group was expanded or collapsed either in the table of the Gantt diagram or of the entities table. This callback can be triggered:</p> <ul style="list-style-type: none"> <li>by the user clicking on the appropriate symbol in the resource, activity or entity row</li> <li>by automatic row expansion when dragging objects</li> <li>by using the method scrollToObject</li> <li>by setting the attribute CollapseState on a resource, an activity, or entity object and option triggeringOfOnCollapseStateChangedByUpdateCallsEnabled is not set to false.</li> </ul> <p><b>Profile:</b> function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object null,   "newCollapseState" : <a href="#">CollapseState</a>,   "interactively" : boolean,   "isForAllocationRows" : boolean,   "hierarchySupplementaryDefinitionID"<sup>8</sup> : string,   "hierarchyLevelSupplementaryDefinitionIndex"<sup>8</sup> : number,   "groupingLevelDefinitionIndex"<sup>8</sup> : number,   "groupingCodes"<sup>8</sup> : string[],   "skillID" : string undefined,   "promise" : Promise undefined [out] }</p> <p>If the application sets the promise attribute, then the update of the DOM is delayed until the promise is resolved.</p> <p>In skilled resources view the property skillID is set when clicking on a resource row, allocation row, or allocation bar.</p>
<b>onCurveCollapseStateChanged</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when a curves pane was expanded or collapsed table of the Gantt diagram. This callback is triggered by the user clicking on the appropriate symbol in the resource or activity row.</p> <p><b>Profile:</b> function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object,   "newCollapseState" : <a href="#">CollapseState</a>,   "resource" : Resource undefined,</p>

Callback Name	Type	Description
		<pre>"skillID" : string undefined, "promise" : Promise undefined [out] }</pre> <p>The property "resource" is only set, when the object is not the resource itself.</p> <p>The application can update the property CurveCollapseState of the object if needed.</p> <p>If the application sets the promise attribute, then the update of the DOM is delayed until the promise is resolved.</p> <p>In skilled resources view the property skillID is set when clicking on a resource row, allocation row, or allocation bar.</p>
onDetermineColumnDefinitions		<b>Deprecated! Please use TableRowDefinition objects.</b>
onDoubleClicked	Function	<p><b>Optional, default: undefined</b> – This function is called when an object is double-clicked by the user.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : ObjectType,   "object" : Object null,   "visualType" : VisualType,   "date" : Date undefined, // date at mouse cursor   "entry"<sup>1</sup> : AllocationEntry ActivityEntry PeriodHighlighterEntry,   "entryIndex"<sup>1</sup> : number,   "periodHighlighter" : PeriodHighlighter,   // Only available when clicked on a PeriodHighlighterEntry; the "object" parameter will then hold the corresponding resource/activity   "cellIndex" : number, // Only available when clicked on a table cell; zero-based index of the cell.   "symbolIndex" : number, // Only available when clicked on a symbol; zero-based index of the symbol.   "hierarchySupplementaryDefinitionID"<sup>6</sup> : string,   "hierarchyLevelSupplementaryDefinitionIndex"<sup>6</sup> : number,   "groupingLevelDefinitionIndex"<sup>6</sup> : number,   "skillID" : string undefined,   "groupingCodes"<sup>6</sup> : string[] }</pre> <p>In case of a grouping row, object is null and instead groupingCodes together with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex are set.</p>

Callback Name	Type	Description
		In skilled resources view the property skillID is set when clicking on a resource row, allocation row, or allocation bar.
<b>onDrag</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when the user drags an activity, allocation, allocation entry, or entity (called anew on every new move of the mouse/finger). If args.dropAllowed is set to false on return of the callback, then a forbidden cursor is shown within the widget and a drop will be ignored.</p> <p>If args.cancel is set to true, then the drag action will be canceled.</p> <p>If an allocation is dragged, then the additional property newRowObjectIsSuitableResource gives the information whether the dragged object is over a suitable resource. Then the application can transfer the value to the property dropAllowed if wishful.</p> <p><b>Profile:</b></p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object,   "visualType" : <a href="#">VisualType</a>,   "entry"<sup>1</sup> : AllocationEntry,   "entryIndex"<sup>1</sup> : number,   "dragMode" : <a href="#">ActivityBarDragModes</a>                  <a href="#">AllocationBarDragModes</a>                  <a href="#">RowDragModes</a>,   "rowInsertionMode" : <a href="#">RowInsertionMode</a>,   "newRowObjectType" : <a href="#">ObjectType</a>,   "newRowObject" : Object,   "newRowObjectIsSuitableActivity" : boolean,   "newRowObjectIsSuitableResource" : boolean,   "newStart" : Date, // not for date lines   "newEnd" : Date, // not for date lines   "newDate" : Date, // only for date lines,   "startPropertyName" : string<sup>2</sup>,   "endPropertyName" : string<sup>2</sup>,   "event" : Event, // DOM interface   "dropAllowed" : boolean [out],   "cancel" : boolean [out] }</pre> <p>If a date symbol or bar of an activity is dragged, then the properties start/endPropertyName contain the name of the property to be modified when a dragging or dropping the symbol or bar, resp.</p> <p>If a row is dragged, the property rowInsertionMode tells about the current insertion mode relative to the object in property newRowObject.</p>

Callback Name	Type	Description
<b>onDragEnd</b>		<p><b>Optional, default: undefined</b> – This function is called when the user ends dragging an activity, allocation, allocation entry, or entity (please check args.objectType!) even when dropping is not allowed on the new row.</p> <p><b>Profile:</b>  function (args)  args = {    "objectType" : <a href="#">ObjectType</a>,    "object" : Object,    "visualType" : <a href="#">VisualType</a>,    "entry"<sup>1</sup> : AllocationEntry,    "entryIndex"<sup>1</sup> : number,    "dragMode" : <a href="#">ActivityBarDragModes</a>                  <a href="#">AllocationBarDragModes</a>                  <a href="#">RowDragModes</a>,    "startPropertyName" : string<sup>2</sup>,    "endPropertyName" : string<sup>2</sup>,    "event" : Event // DOM interface  }</p> <p>If a date symbol or bar of an activity was dragged, then the properties start/endPropertyName contain the name of the property to be modified when a dragging or dropping the symbol or bar, resp.</p>
<b>onDragStart</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when the user starts to drag an activity, allocation, allocation entry, or entity (please check args.objectType!). If args.cancel is set to true, then the drag action will be canceled.</p> <p><b>Profile:</b>  function (args)  args = {    "objectType" : <a href="#">ObjectType</a>,    "object" : Object,    "visualType" : <a href="#">VisualType</a>,    "entry"<sup>1</sup> : AllocationEntry,    "entryIndex"<sup>1</sup> : number,    "dragMode" : <a href="#">ActivityBarDragModes</a>                  <a href="#">AllocationBarDragModes</a>                  <a href="#">RowDragModes</a>,    "startPropertyName" : string<sup>2</sup>,    "endPropertyName" : string<sup>2</sup>,    "event" : Event, // DOM interface    "cancel" : boolean [out]  }</p> <p>If a date symbol or bar of an activity will be dragged, then the properties start/endPropertyName contain the name of the property to be modified when a dragging or dropping the symbol or bar, resp.</p>
<b>onDrop</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when an activity/allocation/entity is dropped by the user after dragging it (but only when dropping was allowed by the last</p>

Callback Name	Type	Description
		<p>triggered onDrag callback). When the function sets a Promise object into args.promise, then the widget disables dragging of the dropped bar until the promise is resolved or rejected. It is also possible to cancel the interaction.</p> <p><b>Profile:</b></p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object,   "visualType" : <a href="#">VisualType</a>,   "entry"<sup>1</sup> : AllocationEntry,   "entryIndex"<sup>1</sup> : number,   "dragMode" : <a href="#">ActivityBarDragModes</a>   <a href="#">AllocationBarDragModes</a>   <a href="#">RowDragModes</a>,   "rowInsertionMode": <a href="#">RowInsertionMode</a>,   "newRowObjectType" : <a href="#">ObjectType</a>,   "newRowObject" : Object,   "newStart" : Date,   "newEnd" : Date,   "newSortCode": number,   "startPropertyName" : string<sup>2</sup>,   "endPropertyName" : string<sup>2</sup>,   "event" : Event, // DOM interface   "cancel" : boolean, // [out]   "promise" : Promise undefined, // [out]   "workingTimeDistance" : number, /*in milliseconds*/   "coupledObjects": Allocation[] Activity[],   "startsAndEndsOfCoupledObjects" : Object[],   "otherNewSortCodesForSiblingObjects": Object[] }</pre> <p>If the promise is resolved, then it is possible to call it with an arguments object, which offers cancel the interaction at last:</p> <pre>args = {   "cancel" : boolean }</pre> <p>When using a promise, then the application should ensure that it will be resolved/rejected later in any way, since the drag action lasts active until then. Maybe there should be a timer for time out.</p> <p>If the option multipleBarDraggingEnabled is set to true and more than one object has been dragged, then the properties coupledObjects and startsAndEndsOfCoupledObjects are set. The latter one contains objects of the form:</p> <pre>{   object : Allocation Activity,   newStart : Date,   newEnd : Date }</pre> <p><b>Remark:</b> If one of the properties newStart or newEnd hat a value of null, then the user</p>

Callback Name	Type	Description
		<p>dragged this object outside of the visible time area and there is no working time in the calendar to calculate the appropriate date.</p> <p>If a date symbol or bar of an activity is dropped, then the properties start/endPropertyName contain the name of the property to be modified when a dragging or dropping the symbol or bar, resp.</p> <p>If the visualType is Row, then the property rowInsertionMode is set. If the sort mode for this type of row object is set to Ascending (see options activity/entity/resourceSortMode), then also the properties newSortCode and otherNewSortCodesForSiblingObjects is set. The latter is an array of objects of the following form:</p> <pre>{   object : Activity Entity Resource,   newSortCode : number }</pre> <p>This array contains all sibling objects for which is new sort code is needed.</p> <p>The code after a vertical row drag&amp;drop could look like this:</p>
<pre> <b>if</b> (callbackArgs.rowInsertionMode !== RowInsertionMode.None) {   <b>// determine sort code property name</b>   let sortCodePropertyName = {     [ObjectType.Activity]: "activityRowSortCodePropertyName",     [ObjectType.Entity]: "entityRowSortCodePropertyName",     [ObjectType.Resource]: "resourceRowSortCodePropertyName"   }[callbackArgs.objectType];   let sortCodePropertyName = vsw.option(sortCodePropertyName)    "SortCode";    let objectsToUpdate = [];    <b>// prepare update of parent ID and sort code of dropped object</b>   let objectToUpdate = Object.assign({}, callbackArgs.object); <b>// clone object</b>   <b>switch</b> (callbackArgs.rowInsertionMode) {     <b>case</b> RowInsertionMode.InsertAsChild:       objectToUpdate.ParentID = callbackArgs.newRowObject.ID;       <b>break</b>;      <b>case</b> RowInsertionMode.InsertAsPreviousSibling:     <b>case</b> RowInsertionMode.InsertAsNextSibling:       objectToUpdate.ParentID = callbackArgs.newRowObject.ParentID;       <b>break</b>;   }   objectToUpdate[sortCodePropertyName] = callbackArgs.newSortCode;    <b>// only for other object types than entities</b>   <b>if</b> (callbackArgs.objectType !== ObjectType.Entity)     objectToUpdate.ViewArea = callbackArgs.newRowObject.ViewArea;    <b>// prepare update of sort code of other objects</b>   objectsToUpdate.push(objectToUpdate);   <b>if</b> (callbackArgs.otherNewSortCodesForSiblingObjects) {     callbackArgs.otherNewSortCodesForSiblingObjects.forEach(newSortCodeAndRowObject =&gt; {       <b>// clone object</b>       let otherObjectToUpdate = Object.assign({}, newSortCodeAndRowObject.object);        otherObjectToUpdate[sortCodePropertyName] = newSortCodeAndRowObject.newSortCode;       objectsToUpdate.push(otherObjectToUpdate);     });   } } </pre>		

Callback Name	Type	Description
<pre>     });   }    // update objects and render   let updateObjectsMethodName = {     [ObjectType.Activity]: "updateActivities",     [ObjectType.Entity]: "updateEntities",     [ObjectType.Resource]: "updateResources"   }[callbackArgs.objectType];   vsw[updateObjectsMethodName] (objectsToUpdate);   vsw.render(); } </pre>		
<b>onLogError</b>	Function	<p><b>Optional, default: undefined</b> – If set, then this function is called when an exception occurs on any method described below that is called on the widget or on setting an option. By default, the exception is re-thrown afterwards.</p> <p>Profile:</p> <pre> function (args) args = {   "commandName" : string,   "commandCounter": number,   "error" : Error,   "rethrow" : boolean /*out*/ } </pre> <p>The commandName property of the argument object contains one of the method or callback names.</p> <p>The commandCounter can be used to bundle errors of the same command.</p> <p>You can use this in your application to send the error from the client to the application server and make it persistent there. Normally no error exception should be triggered at all. See also callback onLogWarning.</p> <p>If the property "rethrow" is set to false, then the exception will not be re-thrown.</p>
<b>onLogWarning</b>	Function	<p><b>Optional, default: undefined</b> – If set then this function is triggered when data is inconsistent among other incidents.</p> <p>Profile:</p> <pre> function (args) args = {   "commandName" : string,   "commandCounter": number,   "code": <a href="#">WarningCode</a>,   "description": string } </pre> <p>The commandName property of the argument object contains the pure current method name. The description contains an English text like</p>

Callback Name	Type	Description
		<p>“Option “xyz” is unknown”, “Object ID empty”, or “Object with ID “xyz” not unique”.</p> <p>The commandCounter can be used to bundle errors of the same command.</p> <p>You can use this in your application to debug your application or to send the warning from the client to the application server and make it persistent there. Normally no warning should be triggered at all. See also callback onLogError.</p>
<b>onRowSortingChangeRequested</b>	Function	<p><b>Optional, default: undefined</b> – This function is triggered when the user clicks or taps on a column in the table title and the option interactiveSwitchingOfSortOrderEnabled is set to true.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "sortMode" : <a href="#">RowSortMode</a>, // [in/out]   "sortCodeSource" : string, // [in/out]   "cancel" : boolean, // [in/out]   "event" : Event }</pre> <p>The application then can change the sort mode or the sort code source property name when necessary. Alternatively, it is possible to abort a change by setting the property "cancel" to true.</p>
<b>onSaveAsPDFProgress</b>	Function	<p><b>Optional, default: undefined</b> – This function is called constantly during the execution of the saveAsPDF method. Especially when saving a diagram to many pages, this callback is helpful for the application to be continuously informed about the progress of the processing.</p> <p>If a promise is returned by the application in the corresponding attribute, then VSW will wait for resolution before continuing the process. This serves to have the chance to show an updated progress dialog.</p> <p>Profile:</p> <pre>function (args) args = {   "pageCount" : number,   "currentPageNumber" : number,   "promise" : Promise undefined /*out*/ }</pre>
<b>onSelectionChanged</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when the user selects/deselects an object solely or in addition. The property "selectedObjects" holds the new selection</p>



Callback Name	Type	Description
		<p>completely and can be changed by the application, while the previously selected objects (if any) are contained in the property "previouslySelectedObjects".</p> <p>The application can also decide whether to accept a selection change by validating the reason property eventually together with the causing object (e.g., clicking on an object or the background, showing a context menu, or starting a drag action).</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a> 0,   "object" : Object null,   "selectedObjects" : Object[], //[in/out]   "visualType" : <a href="#">VisualType</a>,   "previouslySelectedObjects" : Object[] null,   "previouslySelectedObjectsType" : <a href="#">ObjectType</a>    null,   "reason": <a href="#">SelectionChangedReason</a>,   "reasonObject": Object undefined,   "reasonObjectType": <a href="#">ObjectType</a> undefined,   "event" : Event, // DOM interface   "cancel": Boolean /* [in/out],                         Default: false */ }</pre> <p>The property selectedObjects contains the new selection on input. This can be modified and will then determine the actual objects to select. Here it is allowed to return not only the objects that are registered by add or update methods, but also literal objects that contain the properties ID and eventually SkillID. The latter one is used in skilled resources view to select resources or allocations only below the addressed skill row.</p>
onShowContextMenu	Function	<p><b>Optional, default: undefined</b> – This function is called when a context menu can appear. If the function sets a Promise object at args.promise, then the widget will internally hold the state of a context menu being open until the promise is resolved or rejected. Possible items are resources, activities, allocations, allocation entries (only when shown as separate bars instead of allocation bars), links, timescale, empty time area, and period highlighters.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object null,   "visualType" : <a href="#">VisualType</a>,   "date" : Date undefined,</pre>

Callback Name	Type	Description
		<pre> "event" : Event, // DOM interface "entry"<sup>1</sup> : AllocationEntry ActivityEntry  PeriodHighlighterEntry, "entryIndex"<sup>1</sup> : number, "hierarchySupplementaryDefinitionID"<sup>6</sup> : string, "hierarchyLevelSupplementaryDefinitionIndex"<sup>6</sup> : number, "groupingLevelDefinitionIndex"<sup>6</sup> : number, "groupingCodes"<sup>6</sup> : string[], "skillID" : string undefined, "promise" : Promise undefined //[out], "timePeriodStart" : Date undefined, "timePeriodEnd" : Date undefined } </pre> <p>In case of a grouping row, the object is null and instead groupingCodes are set along with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex.</p> <p>In skilled resources view the property skillID is set when clicking on a resource row, allocation row, or allocation bar.</p> <p>In the case of the time scale, the start and end dates of the associated time period are supplied in addition to the date actually hit (see args.timePeriodStart and args.timePeriodEnd).</p>
onShowTooltip	Function	<p><b>Optional, default: undefined</b> – This function is called when a tooltip can appear (i.e. when the mouse cursor hovers over an object). The tooltip itself is to be shown by the application. Possible objects are resources, activities, allocations, links, and period highlighters.</p> <p><b>Profile:</b></p> <pre> function (args) args = {   "objectType" : <a href="#">ObjectType</a>,   "object" : Object null,   "visualType" : <a href="#">VisualType</a>,   "visualSubtype" : <a href="#">VisualSubtype</a>,   "cellIndex" : number undefined,     // Only available when on a table cell;     // zero-based index of the cell.   "event" : Event, // DOM interface   "date"<sup>7</sup> : Date, // date at mouse cursor   "capacity"<sup>8</sup> : number,   "load"<sup>8</sup> : number,   "singleLoads"<sup>8</sup> : Object,   "entry"<sup>1</sup> : AllocationEntry ActivityEntry  </pre>

<sup>7</sup> Available only if objectType == ObjectType.Resource and the mouse cursor hovers over a curve area.

<sup>8</sup> Available only if objectType == ObjectType.Resource and the mouse cursor hovers over a curve area. This object has properties where the names are the IDs of the underlying curves of a curve stack and the values represent the current values of these curves at the current date.

Callback Name	Type	Description
		<p>PeriodHighlighterEntry,</p> <pre> "entryIndex"<sup>1</sup> : number, "periodHighlighter" : PeriodHighlighter, // Only available when clicked on a PeriodHighlighterEntry; the "object" parameter will then hold the corresponding resource/activity "hierarchySupplementaryDefinitionID"<sup>6</sup> : string, "hierarchyLevelSupplementaryDefinitionIndex"<sup>6</sup> : number, "groupingLevelDefinitionIndex"<sup>6</sup> : number, "groupingCodes"<sup>6</sup> : string[], "innerHTML"<sup>9</sup> : string // [in/out] "tooltipTemplateID" : string, // [in/out] "promise" : Promise undefined // [out] } </pre> <p>If you want to avoid showing a tooltip, you will have set the properties innerHTML and tooltipTemplateID to "" or null.</p> <p>In case of a grouping row object is null and instead groupingCodes together with the properties hierarchySupplementaryDefinitionID, hierarchyLevelSupplementaryDefinitionIndex, and groupingLevelDefinitionIndex are set.</p> <p>When the application sets a Promise object into args.promise, then the widget will re-new the text now provided as first parameter in the call to Promise.resolve.</p> <p>In order to triggering the callback for each allocation bar entry or activity bar entry, you must set the option triggeringOfOnShowTooltip-ForEntriesInBarsEnabled. If set to true, the properties entry and entryIndex contains the appropriate value.</p>
<b>onTableCellDefinitionWidthChanged</b>	Function	<p><b>Optional, default: undefined</b> – If set, then this function is called when the user has changed the width of a table column. This callback will only work, when the table columns were defined by TableRowDefinition objects. You then are able to update the cell definition inside of the appropriate TableRowDefinition object e.g. for gaining persistency inside the application.</p>

<sup>9</sup> Text to be displayed inside a tooltip window. This text has to be formatted compliant to the formatting rules for the contents of HTML <div> elements. **Line breaks** can be inserted by adding a <br> tag to the text. Embracing substrings by <b> and </b> tags will show **bold texts**. The same way you can use the <table> and the corresponding <tr> and <td> tags to **tabulate** the tooltip contents. If your original text contains the symbols "<" or ">" - i.e. those symbols should be displayed as they are and must not be interpreted as parts of HTML tag – then you have to replace the symbols by escape sequence codes (replace "<" by "&lt;" and ">" by "&gt;").

Callback Name	Type	Description
		<b>Profile:</b> <pre>function (args) args = {   "tableType" : TableType,   "tableRowDefinition" : Object,   "cellIndex": number,   "newWidth": number,   "oldWidth" : number }</pre>
<b>onTimeAreaViewParametersChanged</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when the visible time area changes either by changing the visible start or by changing the resolution. There is an internal delay that is defined by option <code>scrollOffsetsChangedCallbackTimeDelay</code>.</p> <p><b>Profile:</b>  <pre>function (args) args = {   "scrollOffset" : number, /* in pixels */   "width" : number, /* in pixels */   "start" : Date,   "end" : Date,   "timeResolutionUnit" : string (possible     values: "seconds"/"minutes"/"hours"/     "days"),   "timeResolutionUnitCount" : number,   "tableViewWidth" : number, /* current width,     not to be confused with the option     tableViewWidth */   "entitiesTableViewWidth" : number /* current     width, not to be confused with the option     entitiesTableViewWidth */ }</pre> </p> <p>The values of the properties “start” and “end” can be used in the method <code>fitTimeAreaIntoView</code> to restore the current view at a later time. Alternatively the values of the properties “timeResolutionUnit” and “timeResolutionUnitCount” can be used for the method <code>setTimeResolutionForView</code>.</p>
<b>onVerticalScrollOffsetChanged</b>	Function	<p><b>Optional, default: undefined</b> – This function is called when the visible area is scrolled vertically or when the row object visible at top has changed. There is an internal delay that is defined by option <code>scrollOffsetsChangedCallbackTimeDelay</code>.</p> <p><b>Profile:</b>  <pre>function (args) args = {   "tableType" : TableType,   "scrollOffset" : number, /* in pixels */   "rowObjectTypeAtTop" : ObjectType,   "rowObjectAtTop" : Object,   "topViewScrollOffset" : number, /* in pixels */   "topViewRowObjectAtTop" : Object }</pre> </p>
<b>visibilityFilter</b>		<p><b>Deprecated! For performance reasons; please see options <code>visibilityFilterFor...</code> below.</b></p>

Callback Name	Type	Description
<b>visibilityFilterForActivities</b>	Function	<p><b>Optional, default: undefined</b> – This function is called to hide objects. The result has to be set in the property named "result": true means visible and false means invisible. Setting the option with the same or another value again, triggers the visibility check for all activities immediately.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType.Activity</a>,   "object" : Object,   "result" : boolean /* [out], Default: true */ }</pre>
<b>visibilityFilterForAllocations</b>	Function	<p><b>Optional, default: undefined</b> – This function is called to hide objects. The result has to be set in the property named "result": true means visible and false means invisible. Setting the option with the same or another value again, triggers the visibility check for all allocations immediately.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType.Allocation</a>,   "object" : Object,   "result" : boolean /* [out], Default: true */ }</pre>
<b>visibilityFilterForEntities</b>	Function	<p><b>Optional, default: undefined</b> – This function is called to hide objects. The result has to be set in the property named "result": true means visible and false means invisible. Setting the option with the same or another value again, triggers the visibility check for all entities immediately.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType.Entity</a>,   "object" : Object,   "result" : boolean /* [out], Default: true */ }</pre>
<b>visibilityFilterForResources</b>	Function	<p><b>Optional, default: undefined</b> – This function is called to hide objects. The result has to be set in the property named "result": true means visible and false means invisible. Setting the option with the same or another value again, triggers the visibility check for all resources immediately.</p> <p>Profile:</p> <pre>function (args) args = {   "objectType" : <a href="#">ObjectType.Activity</a>,   "object" : Object,   "result" : boolean /* [out], Default: true */ }</pre>

## 4.3 Methods

The following methods are callable in two ways:

- `$("#gantDiv").nXYZWidget("methodName", param1, param2, ...)`
- `$("#gantDiv").nXYZWidget("instance").methodName(param1, param2, ...)`

The first way is the classical one for jQuery UI Widgets. The second way is more object-oriented and faster, when the instance object is hold in its own variable within the application.

Method Name : Result Type	Parameters	Description
<b>about</b>	-	Opens a popup dialog that shows the licenses of all libraries used. This dialog can also be made visible directly by the user by pressing Shift+Ctrl+Alt+F12.
<b>addActivities</b>	activities : <a href="#">Activity</a> []	Adds activities. <sup>10</sup>
<b>addAllocations</b>	allocations : <a href="#">Allocation</a> []	Adds allocations. <sup>10</sup>
<b>addCalendars</b>	calendars : <a href="#">Calendar</a> []	Adds calendars. <sup>10</sup>
<b>addCurves</b>	curves : <a href="#">Curve</a> []	Adds curves. <sup>10</sup>
<b>addDateLines</b>	dateLines : <a href="#">DateLine</a> []	Adds date lines. <sup>10</sup>
<b>addEntities</b>	entities : <a href="#">Entity</a> []	Adds entities. <sup>10</sup>
<b>addHierarchy-Supplementary-Definitions</b>	hierarchySupplementaryDefinitions : <a href="#">HierarchySupplementaryDefinition</a> []	Adds hierarchy supplementary definitions. <sup>10</sup>
<b>addLinks</b>	links : <a href="#">Link</a> []	Adds links. <sup>10</sup>
<b>addPeriod-Highlighters</b>	periodHighlighters : <a href="#">PeriodHighlighter</a> []	Adds period highlighters. <sup>10</sup>
<b>addResources</b>	resources : <a href="#">Resource</a> []	Adds resources. <sup>10</sup>
<b>addSkills</b>	symbols : <a href="#">Skill</a> []	Adds skills. <sup>10</sup>
<b>addSymbols</b>	symbols : <a href="#">Symbol</a> []	Adds symbols. <sup>10</sup>
<b>addTableRow-Definitions</b>	tableRowDefinitions : <a href="#">TableRowDefinition</a> []	Adds table row definitions. <sup>10</sup>
<b>addTooltip-Templates</b>	tooltipTemplates : <a href="#">TooltipTemplate</a> []	Adds tooltip templates. <sup>10</sup>
<b>addWorkingTime</b> : Date	calendarID : number, start : Date string, workingTime : number	Add a working time given in milliseconds to a date and returns a new date object with the calculated date.
<b>calculateWorking-Time</b> : number	calendarID : number, start : Date string, end : Date string	Calculates the working time of a time period given by a start and an end date. The working time returned is given in milliseconds.
<b>cancelSaveAsPDF</b>	-	Cancels the execution of the saveAsPDF method.

<sup>10</sup> After changing the data model, the changes will not become visible until the method "render" is called. These calls should be made after all changes are made once. If forgotten, there is a timer which calls the method "render" automatically, but this eventually leads to flickering within the Widget's visualization.

Method Name : Result Type	Parameters	Description
<b>determineObject-ByPage-Coordinates</b> : Object undefined	pageX : number, pageY : number	<p>Determines the object the DOM representation of which is hit by the given page coordinates. The returned object resembles callbackArgs of some interaction callbacks like onClicked. Hereby an application can implement its own interactions if necessary.</p> <p>If an object is returned, it has the following properties:</p> <ul style="list-style-type: none"> <li>• "objectType" : ObjectType,</li> <li>• "object" : Object null,</li> <li>• "visualType" : VisualType,</li> <li>• "cellIndex" : number, /*1</li> <li>• "date": Date, /*2</li> <li>• "capacity": number, /*3</li> <li>• "load": number, /*3</li> <li>• "singleLoads": Object, /*3</li> <li>• "entry": ActivityEntry AllocationEntry PeriodHighlighterEntry, /*4</li> <li>• "entryIndex": number, /*4</li> <li>• "curve": Curve, /*3</li> <li>• "periodHighlighter": PeriodHighlighter, /*5</li> <li>• "hierarchySupplementaryDefinitionID": string, /*6</li> <li>• "hierarchyLevelSupplementaryDefinitionIndex": number, /*6</li> <li>• "groupingLevelDefinitionIndex": number, /*6</li> <li>• "groupingCodes": string[] /*6</li> </ul> <p>The objectType property can only be Activity, Allocation, Resource, Link, Entity. If a curve or a period highlighter is hit, then the visualType property is set accordingly.</p> <p>*1: Only available when the table is hit.            *2: Only available when the time area is hit.            *3: Only available when a curve area inside the time area is hit.            *4: Only available when an activity bar, an allocation bar or a period highlighter entry is hit.            *5: Only available when a period highlighter is hit.            6: Only available when a group is hit.</p>
<b>fitTimeAreaInto-View</b> : Promise	start : Date undefined, end : Date undefined	Fits the time area into the visible area. If start and/or end dates are given, then only the time between these are fitted into the visible area. Not given dates are internally replaced by start and end date of the complete time area.
<b>getSelectedObjects</b> : Object	-	<p>Gets all currently selected objects. The result is an object with the following properties:</p> <pre>{   objects : Object[],   objectType : ObjectType undefined,   visualType : VisualType undefined }</pre>

Method Name : Result Type	Parameters	Description
		<p>When no objects are currently selected, then the array is empty and the type properties are set to undefined.</p> <p>See also selectObjects method.</p>
<b>highlightObjects</b>	<p>objectType : <a href="#">ObjectType</a>,  objectsOrIDs : string[]   object[],  visualType : <a href="#">VisualType</a></p>	<p>Highlights the given objects or the objects addressed by the given IDs. In the activities view only activities and allocations can be highlighted. In the resources view only resources and allocations can be highlighted. In the loads view only resources can be highlighted.</p> <p>When objects are provided, these can be the original objects that are registered by one of the add or update methods, or they can be new (literal) objects, since only the properties ID and SkillID (in skilled resources view for resources and allocations) are read on them. When skilled resources view is visible, it is allowed to provide a SkillID property to highlight a specific graphical object in the view.</p> <p>The parameter visualType is only required for objects of type Activity or Allocation. In this case you can define whether the rows (VisualType.Row) or the bars (VisualType.Bar) should be highlighted.</p> <p><b>Note:</b> In the resources view, VisualType.Row can be applied only to allocations that reside in separate rows (allocation rows), but not to allocations inside resource rows.</p> <p>The highlighting is shown by a flashing frame around the objects. The highlighting ends automatically when changing the data or with any user interaction or by using the method scrollToObject.</p> <p>See also methods selectObjects and scrollToObject. See also options objectHighlightingColor, objectHighlightFlashingEnabled.</p>
<b>option</b>	<p>key : Object   string,  [value] : any</p>	<p>Gets the value of an option by using option(key). Sets the value of a single option by using option(key, value). Sets the values of several options by using option({ key1: value1, key2: value2, ... }). When using the last syntax, then it is guaranteed that internal rendering that eventually is necessary only is called once to gain performance. Please do not overwrite the default values in the prototype. Since this method is derived from jQueryUI widget, it is also described here:</p>



Method Name : Result Type	Parameters	Description
		<a href="https://api.jqueryui.com/jquery.widget/#method-option">https://api.jqueryui.com/jquery.widget/#method-option</a> .  If the option value is an object, please do not change inner properties without setting the option explicitly, because the widget will not be informed about the changes implicitly.
<b>removeActivities</b>	activitiesOrIDs : string[]   <a href="#">Activity</a> []	Removes activities. <sup>10</sup>
<b>removeAll</b>	objectType : ObjectType   undefined	Removes all objects or just all objects of the given object type. <sup>10</sup>
<b>removeAllocations</b>	allocationsOrIDs : string[]   <a href="#">Allocation</a> []	Removes allocations. <sup>10</sup>
<b>removeCalendars</b>	calendarsOrIDs : string[]   <a href="#">Calendar</a> []	Removes calendars. <sup>10</sup>
<b>removeCurves</b>	curvesOrIDs : string[]   <a href="#">Curve</a> []	Removes curves. Resources have to be unused to be removable. <sup>10</sup>
<b>removeDateLines</b>	dateLinesOrIDs : string[]   <a href="#">DateLine</a> []	Removes date lines. <sup>10</sup>
<b>removeEntities</b>	entitiesOrIDs : string[]   <a href="#">Entity</a> []	Removes entities. <sup>10</sup>
<b>removeHierarchy-Supplementary-Definitions</b>	hierarchySupplementaryDefinitionsOrIDs : string[]   <a href="#">HierarchySupplementaryDefinition</a> []	Removes hierarchy supplementary definitions. <sup>10</sup>
<b>removeLinks</b>	linksOrIDs : string[]   <a href="#">Link</a> []	Removes links. <sup>10</sup>
<b>removePeriod-Highlighters</b>	periodHighlightersOrIDs : <a href="#">PeriodHighlighter</a> []   string[]	Removes period highlighters. <sup>10</sup>
<b>removeResources</b>	resourcesOrIDs : string[]   <a href="#">Resource</a> []	Removes resources. <sup>10</sup>
<b>removeSkills</b>	skillsOrIDs : string[]   <a href="#">Skill</a> []	Removes skills. <sup>10</sup>
<b>removeSymbols</b>	symbolsOrIDs : string[]   <a href="#">Symbol</a> []	Removes symbols. <sup>10</sup>
<b>removeTableRow-Definitions</b>	tableRowDefinitionsOrIDs : string[]   <a href="#">TableRowDefinition</a> []	Removes table row definitions. <sup>10</sup>
<b>removeTooltip-Templates</b>	tooltipTemplatesOrIDs : string[]   <a href="#">TooltipTemplate</a> []	Removes tooltip templates. <sup>10</sup>
<b>render</b>	-	Refreshes the view after changes to data objects. When the application forgets to call this method,

Method Name : Result Type	Parameters	Description
		then it is called automatically when the application goes idle.
<b>saveAsPDF</b> : Promise	fileName : string, options : Object	<p>Saves the entire chart into a PDF document that is downloaded after creation. Possibly the browser asks whether to wait for completion or not.</p> <p>Additional libraries are needed: PDFKit, SVG-to-PDFKit, and blob-stream. For the properties bottomHTML and topHTML html2canvas is needed additionally. See chapter 2.2!</p> <p>The method returns a Promise object that the application can use, for instance, to react to the finish of the processing (e.g., to make a waiting screen disappear).</p> <p>The optional file name has to be pure (without any path information), and the file will be saved to the downloads folder of the browser by default. If no file name is specified, a new one is generated automatically.</p> <p>The optional options object can be used to specify additional properties for the export. The following attributes are allowed:</p> <ul style="list-style-type: none"> <li>• "author" : string (default: undefined)</li> <li>• "bottomHTML" : string (default: undefined)</li> <li>• "bottomPageMargin" : number ( &gt;= 0; default 10; in millimeters)</li> <li>• "bottomText" : string (default: undefined)</li> <li>• "bottomTimescaleVisible" : boolean (default: false)</li> <li>• "cutMarksVisible" : boolean (default: false)</li> <li>• "horPageCountLimit" : number (default: 0=not active, if "zoomFactorInPercent" is 0, then 1) *</li> <li>• "keywords" : string (default: undefined)</li> <li>• "leftPageMargin" : number ( &gt;= 0; default 10; in millimeters)</li> <li>• "ownerPassword" : string (default: undefined; if defined, you can edit the document in an appropriate application by entering this password)</li> <li>• "pageFormat" : string (default: "A4"; possible values "A0"/"A1"/"A2"/"A3"/"A4"/"A5"/"A6"/"Legal"/"Letter" or "w*h" with width and height in millimeters)</li> <li>• "pageOrientation" : number (default: Portrait; see enum <a href="#">PageOrientation</a>)</li> </ul>

Method Name : Result Type	Parameters	Description
		<ul style="list-style-type: none"> <li>• "permissionToAnnotate" : boolean (default: true; if false, then it will not be possible to annotate text in the document)</li> <li>• "permissionToAssembleDocument" : boolean (default: true; if false, then it will not be possible to combine the document with others)</li> <li>• "permissionToCopy" : boolean (default: true; if false, then it will not be possible to copy text using the clipboard)</li> <li>• "permissionToCopyForContentAccessibility" : boolean (default: true; if false, then it will not be possible to copy content for accessibility)</li> <li>• "permissionToModify" : boolean (default: true; if false, then the PDF document can only be changed by the owner)</li> <li>• "permissionToPrint" : string (default: "highResolution"; possible values are "lowResolution", "highResolution", "none"; if not set to "lowResolution" or "highResolution", then it will not be possible to print the document)</li> <li>• "printingMode" : number (default: Cutting, see enum <a href="#">PrintingMode</a>)</li> <li>• "rightPageMargin" : number (&gt;= 0; default 10; in millimeters)</li> <li>• "subject" : string (default: undefined)</li> <li>• "title" : string (default: undefined)</li> <li>• "topHTML" : string (default: undefined)</li> <li>• "topPageMargin" : number (&gt;= 0; default 10; in millimeters)</li> <li>• "topText" : string (default: undefined)</li> <li>• "userPassword" : string (default: undefined; if given, then it is possible to read the PDF document only by entering the password in an appropriate viewer application)</li> <li>• "verPageCountLimit" : number (default: 0=not active, if "zoomFactorInPercent" is 0, then 1) *</li> <li>• "watermarkSymbolID" : string (default: undefined)</li> <li>• "zoomFactorInPercent" : number (default: 0=not active, else &gt; 0) *</li> </ul> <p>In printing mode <b>Single</b>, the widget content is placed in one single page (zoomFactorInPercent and hor/verPageCountLimit not respected). In printing mode <b>Paging</b>, table and timescale are repeated on each page. In printing mode <b>Cutting</b>, the pages are filled that way you can cut the pages and glue them.</p>

Method Name : Result Type	Parameters	Description
		<p>* If the zoomFactorInPercent is 0 and at least one of hor/verPageCountLimit are 0/undefined, then this limit value(s) will be set to 1. This way, only a minimum of parameters has to be set to get the expected output. Without setting any parameter, you will get a single page as output.</p> <p>The properties bottom/left/right/topPageMargin define the margins that are left blank on each page of the PDF document. This serves for a proper layout for printing it later.</p> <p>The properties bottomText/topText allow to specify additional texts for top/bottom frame area. The keywords {{#PageNo}}, {{#PageCount}}, {{#Date}} maybe used as placeholders. These texts are only usable if the corresponding properties topHTML/bottomHTML are not specified. The text is shown using the font family inherited from the div element of the widget and a font size of 10px.</p> <p>The properties bottomHTML/topHTML allow to specify additional HTML content for top/bottom frame area. The keywords {{#PageNo}}, {{#PageCount}}, {{#Date}} maybe used as placeholders. Also the keyword {{@symbolID}} is a placeholder for a defined symbol and can be used in &lt;image src="..."&gt; to show a symbol if needed (other URLs to external images are also possible). If topHTML or bottomHTML is used, then topText and bottomText are not usable, respectively. Text is shown using the font family and size inherited from the div element of the widget if the style is not modified within the HTML. Please ensure that the given HTML is valid.</p> <p>The property bottomTimescaleVisible determines showing an additional timescale at the bottom of the chart in the PDF document.</p> <p>The property cutMarksVisible determines showing marks at the four corners of each page in order to make it possible to cut the empty margins of printed pages and put the pages together. This only makes sense in printing mode Cutting.</p> <p>The properties horPageCountLimit and verPageCountLimit determine a zoom factor for the chart indirectly by setting the limits of page count. It is possible to set one of the limits only or to leave them both zero.</p>

Method Name : Result Type	Parameters	Description
		<p>The properties pageFormat and pageOrientation determine the size and orientation of each page in the PDF document.</p> <p>The property watermarkSymbolID allows to put a watermark on each page.</p> <p>The property zoomFactorInPercent determines the zoom factor for the chart, when not left zero.</p> <p>The properties author, keywords, subject, title, permissionTo... are put into the PDF document properties.</p> <p>See also method cancelSaveAsPDF and callback option onSaveAsPDFProgress.</p>
<b>scrollToDate</b> : Promise	date: Date   string, offset : string   undefined	<p>Scrolls to the given date. If the parameter offset is set, the view will be scrolled back by the given offset to get a distance between the left margin of the time area view and the given date. The offset can be a string with</p> <ul style="list-style-type: none"> <li>• a number that specifies a number of pixels (e.g. "50px").</li> <li>• a percentage string that specifies the size of the offset as a percentage of the time area view width (e.g. "10%").</li> </ul>
<b>scrollToObject</b> : Promise	objectType : <a href="#">ObjectType</a> , objectOrID : Object   string, targetPositionInView : <a href="#">TargetPositions</a> , highlightingEnabled : boolean	<p>Scrolls to the object (activity/allocation/entity/resource). If the object is not visible because of being a hidden row or being within a hidden row, the corresponding rows are expanded automatically.</p> <p>When an object is provided, this can be the original object that is registered by one of the add or update methods, or it can be a new (literal) object, since only the properties ID and SkillID (in skilled resources view for resources and allocations) is read on it. When skilled resources view is visible, it is allowed to provide a SkillID property to highlight a scroll to a specific graphical object in the view. When the SkillID is not given, the allocation bar in the assigned skill is addressed or the resource row of the first mentioned skill in property SkillID, resp.</p> <p>The third and the fourth parameter are optional.</p> <p>targetPositionInView (default is Necessary) determines the position of the object in the view after scrolling to it. Value Necessary means that the</p>

Method Name : Result Type	Parameters	Description
		<p>object will be made visible using the only necessary scrolling.</p> <p>If highlightingEnabled is set to true (default), then a (eventually blinking) frame is shown until another method is used or a user interaction takes place.</p> <p>See also options objectHighlightingColor, object-HighlightFlashingEnabled, scrollToObjectAnimation-Enabled.</p>
<b>scrollViewArea- Horizontally</b>	viewArea : <a href="#">HorizontallyScrollable ViewArea</a> , scrollPosition : <a href="#">HorizontalScrollPositi on</a>	Scrolls the specified view area horizontally to the left or right.
<b>scrollViewArea- Vertically</b>	viewArea : <a href="#">VerticallyScrollableVi ewArea</a> , scrollPosition : <a href="#">VerticalScrollPosition</a>	Scrolls the specified view area vertically to the top or bottom.
<b>selectObjects</b>	objectType : <a href="#">ObjectType</a> , objectsOrIDs : string[]   object[], visualType : <a href="#">VisualType</a>	<p>Selects the given objects or the objects addressed by the given IDs. When objects are provided, these can be the original objects that are registered by one of the add or update methods, or they can be new (literal) objects, since only the properties ID and SkillID (in skilled resources view for resources and allocations, see below) are read on them.</p> <p>In the activity mode only activities and links can be selected. In the resource mode only resources and allocations can be selected.</p> <p>The parameter visualType is only required in the activity mode if objects of type Activity are to be selected. In this case you can define whether the activity rows (VisualType.Row) or the activity bars (VisualType.Bar) should be selected.</p> <p>It is possible to select objects that are hidden in the collapsed parent object. The selectionChanged callback (see options) is not called by the widget.</p> <p>When skilled resources view is visible, it is allowed to provide a SkillID property to select a specific graphical object in the view.</p> <p>See also getSelectedObjects method.</p>
<b>setResourceProper tiesForSkill</b>	properties : Object[]	Sets some additional graphical attributes and states for resources referenced by their ID shown below a referenced skill.

Method Name : Result Type	Parameters	Description
		<p>Objects in the array have the following profile:</p> <pre>{   ID : string,           //required   SkillID : string,      //required   AllocationRowsCollapseState : number undefined,   CurveCollapseState : number undefined,   RowSymbolIDs : string[] undefined,   TableColor : string undefined,   TableRowDefinitionID : string undefined }</pre> <p>If a property is set, then this setting will be used only for the resource row below the referenced skill. If a property is not set, then the property value with the same name within the resource referenced by the ID will be used.</p> <p>This method can be used define some graphical attributes or states that are different between the resource rows below different skill rows.</p>
<b>setTimeResolutionForView</b>	unit : string ("seconds", "minutes", "hours", "days", "weeks", "months", "quarters", "years")   <a href="#">TimeUnit</a> , unitCount : number undefined, start : Date undefined	<p>Sets the resolution in the time area view. If unitCount is undefined, then 1 is used. If start is undefined, then the current visible start is used.</p> <p>The time resolution cannot be set finer than the maximum time resolution defined by the options maximumTimeResolutionUnit and -Factor!</p>
<b>updateActivities</b>	activities : <a href="#">Activity</a> [], updateMode : <a href="#">UpdateModes</a>	<p>Update activities. Allowed changes are modification of all attributes besides ID.<sup>10</sup></p> <p>updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.</p>
<b>updateAllocations</b>	allocations : <a href="#">Allocation</a> [], updateMode : <a href="#">UpdateModes</a>	<p>Updates allocations. Allowed changes are modification of all attributes besides ID.<sup>10</sup></p> <p>updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.</p>
<b>updateCalendars</b>	calendars : <a href="#">Calendar</a> [], updateMode : <a href="#">UpdateModes</a>	<p>Updates calendars visually. Allowed changes are modification of all attributes besides ID.<sup>10</sup></p> <p>updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter of API description for details.</p>
<b>updateCurves</b>	curves : <a href="#">Curve</a> [], updateMode : <a href="#">UpdateModes</a>	<p>Updates curves. Allowed changes are modification of all attributes but ID and Type.<sup>10</sup></p> <p>updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.</p>

Method Name : Result Type	Parameters	Description
<b>updateDateLines</b>	dateLines : <a href="#">DateLine</a> [], updateMode : <a href="#">UpdateModes</a>	Updates date lines. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateEntities</b>	entities : <a href="#">Entity</a> [], updateMode : <a href="#">UpdateModes</a>	Update entities. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateHierarchySupplementDefinitions</b>	hierarchySupplementDefinitions : <a href="#">HierarchySupplementDefinition</a> [], updateMode : <a href="#">UpdateModes</a>	Updates hierarchy supplementary definitions. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateLinks</b>	links : <a href="#">Link</a> [], updateMode : <a href="#">UpdateModes</a>	Updates links. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updatePeriod-Highlighters</b>	periodHighlighters : <a href="#">PeriodHighlighter</a> [], updateMode : <a href="#">UpdateModes</a>	Updates period highlighters. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateResources</b>	resources : <a href="#">Resource</a> [], updateMode : <a href="#">UpdateModes</a>	Updates resources. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateSkills</b>	skills : <a href="#">Skill</a> [], updateMode : <a href="#">UpdateModes</a>	Updates skills. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateSymbols</b>	symbols : <a href="#">Symbol</a> [], updateMode : <a href="#">UpdateModes</a>	Updates symbols. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateTableRow-Definitions</b>	tableRowDefinitions: <a href="#">TableRowDefinition</a> [], , updateMode : <a href="#">UpdateModes</a>	Updates table row definitions. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.
<b>updateTooltip-Templates</b>	tooltipTemplates: <a href="#">TooltipTemplate</a> [], updateMode : <a href="#">UpdateModes</a>	Updates tooltip templates. Allowed changes are modification of all attributes besides ID. <sup>10</sup>  updateMode is optional. See enum <a href="#">UpdateModes</a> in the Enumerations chapter for details.



## 5 Enumerations

The following enumerations are provided:

### 5.1 ActivityBarDragModes

```
netronic.nVSW.ActivityBarDragModes = {

    // Note: Values are flags,
    //       i.e. they can be combined by using bitwise OR operators.

    None: 0,
    DragStart: 1,
    DragEnd: 2,
    DragHorizontally: 4,
    DragVertically: 8,
    DragAutoHorOrVer: 16,
};
```

### 5.2 ActivityBarShape

```
netronic.nVSW.ActivityBarShape = {
    Regular: 0,
    Summary: 1,
    Diamond: 2,
    Rectangle: 3,
    Symbol: 4
};
```

When the shapes Regular, Summary, or Rectangle are used, then the bar will span the time range defined by the allocation entries or defined by the assigned activity object when there is no allocation entry defined. When the shapes Diamond or Symbol are used the shape is set to the start date of the same time span.

When the shapes Diamond or Symbol are used, then the bar text is shown only when the bar is visible within an expanded row, otherwise the bar text is shown unconditionally.

### 5.3 AllocationBarDragModes

```
netronic.nVSW.AllocationBarDragModes = {

    // Note: Values are flags,
    //       i.e. they can be combined by using bitwise OR operators.

    None: 0,
    DragStart: 1,
    DragEnd: 2,
    DragHorizontally: 4,
    DragVertically: 8,
    DragAutoHorOrVer: 16,
};
```

## 5.4 AllocationBarShape

```
netronic.nVSW.AllocationBarShape = {
  Regular: 0,
  Summary: 1,
  Diamond: 2,
  Rectangle: 3,
  Symbol: 4
};
```

The shapes Regular and Rectangle will show the allocation entries within the figure.

When the shapes Regular, Summary, or Rectangle are used, then the bar will span the time range defined by the allocation entries or defined by the assigned activity object when there is no allocation entry defined. When the shapes Diamond or Symbol are used the shape is set to the start date of the same time span.

When the shapes Diamond or Symbol are used, then the bar text is shown only when the bar is visible within an allocation row, otherwise the bar text is shown unconditionally.

## 5.5 BarDesigns

```
netronic.nVSW.BarDesigns = {

  Simple: 0,
  Default: 65535,           // this value leaves some bits reserved for future extensions
  DefaultReduced: 16711936, // this value leaves some bits reserved for future extensions

  // Note: Values are flags,
  //       i.e. they can be combined by using bitwise OR operators.

  Entries: 1,
  ComplexShape: 2,           // actually only Regular is changed to Rectangle when unset
  Symbols: 4,
  Status: 8,
  Constraints: 16,
  ReleaseAndDueDateSymbols: 32, // ignored on allocations
  Baseline: 64,              // ignored on allocations
  ProgressAndPredictedEnd: 128,
  Text: 256,
  TonedDownColoring: 65536    // see option tonedDownOverlayColor
  ReducedHeight: 131072       // see option reducedBarTopOffsetAndHeightScaleFactor
};
```

## 5.6 BarSortMode

```
netronic.nVSW.BarSortMode = {
  StartAndEnd: 0,
  ByCompareObjects: 1,
  ByCompareObjectsOnSameStart: 2
};
```

## 5.7 CollapseState

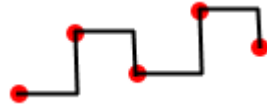
```
netronic.nVSW.CollapseState = {
  Unchanged: -1,
  Expanded: 0,
  Collapsed: 1
};
```

```
};
```

## 5.8 CurveInterpolationType

```
netronic.nVSW.CurveInterpolationType = {
```

```
StepAfter: 0,
```



```
Linear: 1,
```



```
};
```

## 5.9 CurveType

```
netronic.nVSW.CurveType = {
```

```
PointCurve: 0,
```

```
CurveStack: 3,
```

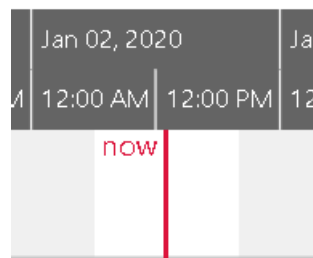
```
CurveList: 4
```

```
};
```

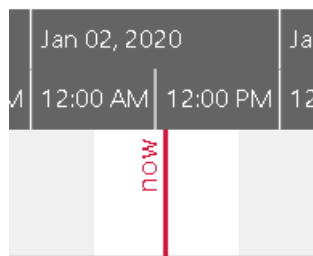
## 5.10 DateLineCaptionOrientation

```
netronic.nVSW.DateLineCaptionOrientation = {
```

```
Horizontal: 1,
```



```
Vertical: 2
```

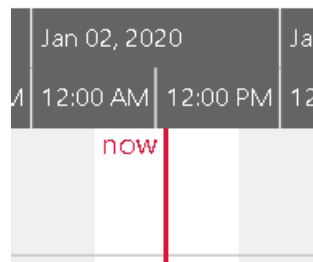


```
};
```

## 5.11 DateLineCaptionPosition

```
netronic.nVSW.DateLineCaptionPosition = {
```

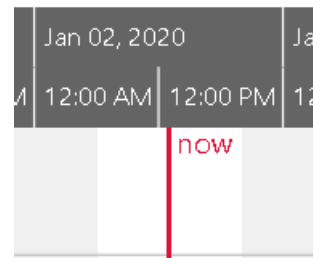
Left: 1,



Center: 2,

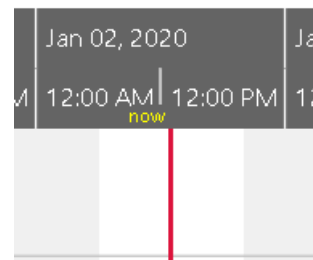


Right: 4,



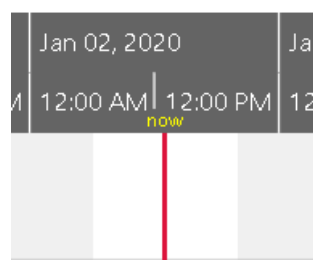
TopLeft: 9,

```
// inside
// timescale
// area
```



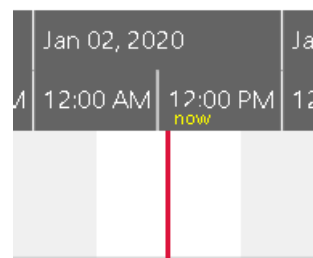
TopCenter: 10,

```
// inside
// timescale
// area
```



TopRight: 12

```
// inside
// timescale
// area
```



};

## 5.12 DateLineGridModes

```
netronic.nVSW.DateLineGridModes = {  
    None: 0,  
    Auto: 1,  
    Weekly: 2,  
    Daily: 4  
};
```

## 5.13 HorizontalAlignment

```
netronic.nVSW.HorizontalAlignment = {  
    Left: 0,  
    Center: 1,  
    Right: 2  
};
```

## 5.14 HorizontallyScrollableViewArea

```
netronic.nVSW.HorizontallyScrollableViewArea = {  
    Table: 0,  
    TimeArea: 1,  
    EntitiesTable: 2  
}
```

## 5.15 HorizontalScrollPosition

```
netronic.nVSW.HorizontalScrollPosition = {  
    Left: 1,  
    Right: 2  
}
```

## 5.16 LinkMarker

```
netronic.nVSW.LinkMarker = {  
    None: 0,  
    FilledArrow: 1  
};
```

## 5.17 LinkRoutingType

```
netronic.nVSW.LinkRoutingType = {  
    Curved: 1,  
    Orthogonal: 2  
};
```

## 5.18 ObjectType

```
netronic.nVSW.ObjectType = {  
    TimeArea: -2,  
    Timescale: -1,  
    Activity: 1,  
    Allocation: 2,  
    Resource: 5,  
    Link: 6,  
    Curve: 7,
```

```

    Entity: 13,
    PeriodHighlighter: 14,
    Symbol: 15,
    DateLine: 16,
    TooltipTemplate: 17,
    TableRowDefinition: 18,
    Calendar: 20,
    HierarchySupplementaryDefinition: 21,
    Skill: 22
};

```

## 5.19 PageOrientation

```

netronic.nVSW.PageOrientation = {
    Portrait: 0,
    Landscape: 1
};

```

## 5.20 PanningMode

```

netronic.nVSW.PanningMode = {
    None: 0,
    HorizontallyOnly: 1,
    VerticallyOnly: 2,
    HorAndVer: 3,
    AutoHorOrVer: 4,
};

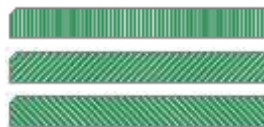
```

## 5.21 PatternType

```

netronic.nVSW.PatternType = {
    None: -1,
    VerticalHatch: 0,
    ForwardHatch: 1,
    BackwardHatch: 2
};

```



## 5.22 PrintingMode

**Note:** See also [here](#) for an overview of the different modes.

```

netronic.nVSW.PrintingMode = {
    Single: 0,
    Cutting: 1,
    Paging: 2
};

```

## 5.23 ProgressBarWidthCalculationMode

```

netronic.nVSW.ProgressBarWidthCalculationMode = {
    ConsiderWorkingTimesOnly: 0,
    ConsiderWorkingAndNonworkingTimes: 1,
};

```

```
};
```

## 5.24 RelationType

```
netronic.nVSW.RelationType = {
    FinishToStart: 0,
    FinishToFinish: 1,
    StartToStart: 2,
    StartToFinish: 3,
    SourceDateToStart: 4,
    SourceDateToFinish: 5,
    FinishToTargetDate: 8,
    StartToTargetDate: 10,
    SourceDateToTargetDate: 12
};
```

## 5.25 RowDesigns

```
netronic.nVSW.RowDesigns = {

    // Note: flags!
    // These values can be combined by using bitwise OR operators.

    Empty: 0,
    Bars: 1,                // Shows bars assigned to row object directly
    Optimized: 2,           // Shows all bars without horizontal overlapping
    BarsInHiddenDescendantRows: 4, // Shows bars of other hidden descendant rows
    CalendarGrid: 8         // Shows calendar grid of row object
};
```

Further explanation:

- “Bars”: In the activities view, the bar in an activity row represents the same object as the row itself. In the resources view, the bars represent the allocations that are assigned to the resource represented by the row. In the loads view, no bars are displayed as a matter of principle. For allocation rows in the activities view or resources view the bars represent the same object as the row itself. If this flag is set then the bars are shown, else they remain invisible.
- “BarsInHiddenDescendantRows”: This flag is only effective when the row is shown in collapsed state but does no harm when the row is shown in expanded state. Bars of hidden rows below the collapsed row are projected into the row.
- “Optimized”: When bars are shown in this row then they will be shown stacked vertically so that they do not overlap graphically when they allocate common time ranges. When bars are stacked the row is getting higher. If the flag is not set, the bars will graphically overlap and the row height is stable.
- “CalendarGrid”: If this flag is set then the calendar assigned to the object represented by the row is made visible through a so-called calendar grid.

## 5.26 RowDragModes

```
netronic.nVSW.RowDragModes = {

    // Note: Values are flags,
    //         i.e. they can be combined by using bitwise OR operators.
```

```
    None: 0,  
    DragVertically: 8,  
    DragOutside: 32,  
    DragOnSameLevelOnly: 64  
};
```

## 5.27 RowInsertionMode

```
netronic.nVSW.RowInsertionMode = {  
    None: 0,  
    InsertAsChild: 1,  
    InsertAsNextSibling: 2,  
    InsertAsPreviousSibling: 3  
};
```

## 5.28 RowSortMode

```
netronic.nVSW.RowSortMode = {  
    None: 0,  
    Ascending: 1,  
    Descending: 2  
};
```

## 5.29 SelectionChangedReason

```
netronic.nVSW.SelectionChangedReason = {  
    Click: 0,  
    BackgroundClick: 1,  
    ContextMenu: 2,  
    DragStart: 3  
};
```

## 5.30 SnapTargets

```
netronic.nVSW.SnapTargets = {  
  
    // Note: Values are flags,  
    //       i.e. they can be combined by using bitwise OR operators.  
  
    None: 0,  
    Start: 1, // only valid for bars representing allocations  
    End: 2, // only valid for bars representing allocations  
    DateLines: 4,  
    CalendarGrids: 8,  
    DateLineGrids: 16  
};
```

## 5.31 TableType

```
netronic.nVSW.TableType = {  
    Gantt: 0,  
    Entities: 1  
};
```



## 5.32 TargetPositions

```
netronic.nVSW.TargetPositions = {

    // Note: Values are flags,
    //       i.e. they can be combined by using bitwise OR operators.

    Necessary: 0,
    Left: 1,
    HCenter: 2,
    Right: 4,
    Top: 8,
    VCenter: 16,
    Bottom: 32
};
```

## 5.33 TextWrapMode

```
netronic.nVSW.TextWrapMode = {
    None: 0, // no wrapping at all
    Line: 1, // text is wrapped at \n
};
```

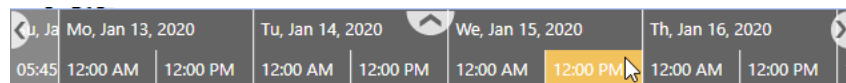
## 5.34 TimescaleInteractionModes

```
netronic.nVSW.TimescaleInteractionModes = {
    // Note: Values are flags,
    //       i.e. they can be combined by using bitwise OR operators.

    None: 0, // no interactions at all
    ScrollingByButtons: 1, // horizontal scrolling by using scrolling buttons
    Rescaling: 2, // rescaling by time period selection, up-button or
                  // mouse wheel
    Default: 3 // all interactions
};
```

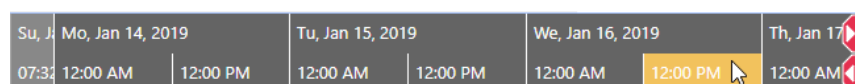
## 5.35 TimescaleNavigationMode

```
netronic.nVSW.TimescaleNavigationMode = {
    Latest: 0, // use the latest version of the timescale navigation
```



- A click onto the left and right button scrolls the chart sideward by the width of the view.
- A click onto the up button reduces the timescale resolution.
- A click onto a highlighted period (see orange area) fits this period completely into the view.
- Use the mouse wheel for increasing and reducing the timescale resolution.

LegacyVersion1: 1,



- A click onto the left and right button scrolls the chart sideward by the widths of one unit in the upper timescale ribbon
- A click onto a highlighted period (see orange area) fits this period completely into the view.
- Use the mouse wheel for increasing and reducing the timescale resolution.

```
};
```

### 5.36 TimeType

```
netronic.nVSW.TimeType = {
    WorkingTime: 1,
    NonworkingTime: 2
};
```

### 5.37 TimeUnit

```
netronic.nVSW.TimeUnit = {
    Seconds: 0,
    Minutes: 1,
    Hours: 2,
    Days: 3,
    Weeks: 4,
    Months: 5,
    Quarters: 6,
    Years: 7
};
```

### 5.38 TreeVisualizationMode

```
netronic.nVSW.TreeVisualizationMode = {
    ColoredIndentation: 0,
    TreeViewLines: 1
};
```

### 5.39 UpdateModes

```
netronic.nVSW.UpdateModes = {
    // Note: Values are flags,
    //       i.e. they can be combined by using bitwise OR operators.

    UpdateOnly: 0,           // deprecated, use Default
    Default: 0,
    ImplicitAddObjects: 1, // If an object to be updated does not exist,
                          // it will be added automatically.
    DifferentialValues: 2 // If set, then the object data given in the update
                          // method can contain only changed property values. If a
                          // property is omitted, it will be supplemented by the
                          // value of the property in the current object. If a
                          // property value shall be set to undefined explicitly,
                          // please use a null value instead or another value that
                          // can be set by the option
                          // resetValueForDifferentialUpdate.*
};
```

\* When using this flag, the application will have to use simple objects in the update methods, because the VSW will modify them and supplement missing property values. Also, these objects have to be different to the ones given in former calls to the appropriate add or update method.

Please be aware that sub objects in arrays like entries must be fully defined despite of the active update mode.

## 5.40 VerticallyScrollableViewArea

```
netronic.nVSW.VerticallyScrollableViewArea = {  
  Top: -1,  
  Main: 0,  
  EntitiesTable: 2  
}
```

## 5.41 VerticalScrollPosition

```
netronic.nVSW.VerticalScrollPosition = {  
  Top: 1,  
  Bottom: 2  
}
```

## 5.42 ViewArea

```
netronic.nVSW.ViewArea = {  
  Top: -1,  
  Main: 0,  
  Default: 0 // for compatibility reasons  
};
```

## 5.43 ViewType

```
netronic.nVSW.ViewType = {  
  Activities: 0,  
  Resources: 1,  
  Loads: 2,  
  SkilledResources: 3  
};
```

## 5.44 VisualSubtype

```
netronic.nVSW.VisualSubtype = {  
  ActivityDueDateSymbol: 1,  
  ActivityReleaseDateSymbol: 2  
};
```

## 5.45 VisualType

```
netronic.nVSW.VisualType = {  
  Background: -1,  
  Bar: 0,  
  Row: 1,  
  Curve: 2,
```

```

    Link: 3,
    PeriodHighlighter: 4,
    DateLine: 5
  };

```

## 5.46 WarningCode

netronic.nVSW.WarningCode =

Key	Value	Explanation
<b>EmptyIDatIndex</b>	W1000	When adding, updating, or removing objects, an empty ID was detected (the index in array of object references is mentioned in the description text). The object reference will be ignored.
<b>ExistingIDatIndex</b>	W1001	When adding objects, an ID was detected that already exists (the key is mentioned in the description text). The object reference will be ignored.
<b>DuplicateIDatIndex</b>	W1002	When adding or updating objects, an ID was detected that is duplicate within the given array (the key is mentioned in the description text). When adding objects, the second object reference will be ignored. When updating objects, the first object reference will be ignored.
<b>CyclicByParentID</b>	W1003	When adding or updating activity, resource, or entity objects, one object has a given ParentID that leads to a cycle (the key is mentioned in the description text). Additionally, an exception will then be thrown after processing the whole given array of objects, because the VSW cannot work with such a cycle in object relations.
<b>UnknownID</b>	W1021	When using the method scrollToObject, then the given object ID is unknown.
<b>ObjectNotVisibleInView</b>	W1022	When using the method scrollToObject, then the object referenced by the given object ID has no presentation in the current view.
<b>UnknownOptionName</b>	W1100	The option name given when setting or getting options is unknown to the VSW.
<b>RequiredOptionUnset</b>	W1101	A required option (namely "start" or "end") is unset by the application. The internally calculated value is mentioned in the description text.
<b>HTMLCanvasFailedOnSaveAsPDF</b>	W3000	When saving a PDF and at least one of the options topHTML or bottomHTML is used, then it was not possible to create a canvas from the HTML content, because of an image URL that references a file on the disk, which is not allowed by the browser security settings. Please replace the URL by another one. The PDF will be created without a top or bottom text.

## 5.47 WorldViewPosition

```

netronic.nVSW.WorldViewPosition = {
  Left: 1,
  Right: 2,
  Top: 3,
}

```

```
    Bottom: 4  
}
```

## APPENDIX

### Information Material on Specific Topics

- [Hello Gantt World - Build your first HTML5/JavaScript Gantt Chart within few minutes with the VSW](#) (video)
- [Update 3 of the Model for Resource Planning HTML5 Gantt Charts](#) (blog post)
- [What is an \*\*activities\*\* and a \*\*resources view\*\*?](#) (blog post)
- [Calendars – Individual resource working times in a HTML5 Gantt chart](#) (blog post)
- [Links as the visualization of \*\*dependencies\*\*](#) (blog post)
- [The art of \*\*designing bars\*\* to map semantics to Gantt charts](#) (video)
- [From awful to awesome \*\*progress visualization\*\* in Gantt charts](#) (blog post)
- [Backlogs in HTML5 Gantt charts](#) (blog post)
- [Hierarchy vs. \*\*grouping\*\*](#) (blog post)
- [CSS Custom Properties for Coloring the Visual Scheduling Widget](#) (blog post)

### Past Release Notes

[Version 6.4](#)  
[Version 6.3](#)  
[Version 6.2](#)  
[Version 6.1](#)  
[Version 6.0](#)  
[Version 5.3](#)  
[Version 5.2](#)  
[Version 5.0](#)  
[Version 4.0](#)  
[Version 3.2](#)  
[Version 3.1](#)