

KDGantt Reference Manual

Generated by Doxygen 1.3.2

Fri Mar 11 13:48:25 2005

Contents

1	KDGantt Hierarchical Index	1
1.1	KDGantt Class Hierarchy	1
2	KDGantt Compound Index	3
2.1	KDGantt Compound List	3
3	KDGantt Page Index	5
3.1	KDGantt Related Pages	5
4	KDGantt Class Documentation	7
4.1	KDGanttMinimizeSplitter Class Reference	7
4.2	KDGanttView Class Reference	10
4.3	KDGanttViewItem Class Reference	54
4.4	KDGanttViewItem Class Reference	57
4.5	KDGanttViewItemDrag Class Reference	78
4.6	KDGanttViewSummaryItem Class Reference	80
4.7	KDGanttViewTaskItem Class Reference	84
4.8	KDGanttViewTaskLink Class Reference	87
4.9	KDGanttViewTaskLinkGroup Class Reference	93
5	KDGantt Page Documentation	99
5.1	Deprecated List	99

Chapter 1

KDGantt Hierarchical Index

1.1 KDGantt Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

KDGanttMinimizeSplitter	7
KDGanttView	10
KDGanttViewItem	57
KDGanttViewItemEventItem	54
KDGanttViewItemSummaryItem	80
KDGanttViewItemTaskItem	84
KDGanttViewItemDrag	78
KDGanttViewTaskLink	87
KDGanttViewTaskLinkGroup	93

Chapter 2

KDGantt Compound Index

2.1 KDGantt Compound List

Here are the classes, structs, unions and interfaces with brief descriptions:

KDGanttMinimizeSplitter (Implements a splitter widget with minimize buttons)	7
KDGanttView	10
KDGanttViewItemEventItem	54
KDGanttViewItem	57
KDGanttViewItemDrag (Drag and drop of KD Gantt items)	78
KDGanttViewSummaryItem	80
KDGanttViewTaskItem	84
KDGanttViewTaskLink	87
KDGanttViewTaskLinkGroup	93

Chapter 3

KDGantt Page Index

3.1 KDGantt Related Pages

Here is a list of all related documentation pages:

Deprecated List	99
---------------------------	----

Chapter 4

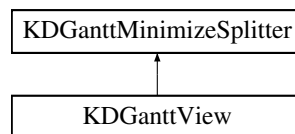
KDGantt Class Documentation

4.1 KDGanttMinimizeSplitter Class Reference

The KDGanttMinimizeSplitter class implements a splitter widget with minimize buttons.

```
#include <KDGanttMinimizeSplitter.h>
```

Inheritance diagram for KDGanttMinimizeSplitter::



Public Types

- enum [ResizeMode](#) { **Stretch**, **KeepSize**, **FollowSizeHint** }
- enum [Direction](#) { **Left**, **Right**, **Up**, **Down** }

Public Member Functions

- [KDGanttMinimizeSplitter](#) (QWidget *parent=0, const char *name=0)
- [KDGanttMinimizeSplitter](#) (Orientation, QWidget *parent=0, const char *name=0)
- [~KDGanttMinimizeSplitter](#) ()
- virtual void [setOrientation](#) (Orientation)
the orientation of the splitter
- Orientation [orientation](#) () const
- void [setMinimizeDirection](#) (Direction)
- Direction [minimizeDirection](#) () const

4.1.1 Detailed Description

The KDGanttMinimizeSplitter class implements a splitter widget with minimize buttons.

This class (and its documentation) is largely a copy of Qt's QSplitter; the copying was necessary because QSplitter is not extensible at all. QSplitter and its documentation are licensed according to the GPL and the Qt Professional License (if you hold such a license) and are (C) Trolltech AS.

A splitter lets the user control the size of child widgets by dragging the boundary between the children. Any number of widgets may be controlled.

To show a QListBox, a QListView and a QTextEdit side by side:

```
KDGanttMinimizeSplitter *split = new KDGanttMinimizeSplitter( parent );
QListBox *lb = new QListBox( split );
QListView *lv = new QListView( split );
QTextEdit *ed = new QTextEdit( split );
```

In KDGanttMinimizeSplitter, the boundary can be either horizontal or vertical. The default is horizontal (the children are side by side) but you can use setOrientation(QSplitter::Vertical) to set it to vertical.

Use setSizeMode() to specify that a widget should keep its size when the splitter is resized.

Although KDGanttMinimizeSplitter normally resizes the children only at the end of a resize operation, if you call setOpaqueResize(TRUE) the widgets are resized as often as possible.

The initial distribution of size between the widgets is determined by the initial size of each widget. You can also use setSizes() to set the sizes of all the widgets. The function sizes() returns the sizes set by the user.

If you hide() a child, its space will be distributed among the other children. It will be reinstated when you show() it again. It is also possible to reorder the widgets within the splitter using moveToFirst() and moveToLast().

4.1.2 Member Enumeration Documentation

4.1.2.1 enum KDGanttMinimizeSplitter::Direction

The values of this enumeration describe into which direction the splitter will collapse its child widgets. By extension, it also specifies the orientation of the splitter; collapsing to the left or to the right results in a horizontal splitter, collapsing to the top or bottom in a vertical splitter.

4.1.2.2 enum KDGanttMinimizeSplitter::SizeMode

This enum type describes how KDGanttMinimizeSplitter will resize each of its child widgets. The currently defined values are:

Stretch: the widget will be resized when the splitter itself is resized.

KeepSize: KDGanttMinimizeSplitter will try to keep this widget's size unchanged.

FollowSizeHint: KDGanttMinimizeSplitter will resize the widget when the widget's size hint changes.

4.1.3 Constructor & Destructor Documentation

4.1.3.1 KDGanttMinimizeSplitter::KDGanttMinimizeSplitter (QWidget *parent = 0, const char *name = 0)

Constructs a horizontal splitter with the *parent* and *name* arguments being passed on to the QFrame constructor.

4.1.3.2 KDGanttMinimizeSplitter::KDGanttMinimizeSplitter (Orientation *o*, QWidget * *parent* = 0, const char * *name* = 0)

Constructs a splitter with orientation *o* with the *parent* and *name* arguments being passed on to the QFrame constructor.

4.1.3.3 KDGanttMinimizeSplitter::~KDGanttMinimizeSplitter ()

Destroys the splitter and any children.

4.1.4 Member Function Documentation

4.1.4.1 KDGanttMinimizeSplitter::Direction KDGanttMinimizeSplitter::minimizeDirection () const

Returns the direction of the minimize buttons.

4.1.4.2 Orientation KDGanttMinimizeSplitter::orientation () const [inline]

Returns the orientation of the splitter.

4.1.4.3 void KDGanttMinimizeSplitter::setMinimizeDirection (Direction *direction*)

Specifies the direction of the minimize buttons. If the orientation of the splitter is horizontal then with KDGanttMinimizeSplitter::Left or KDGanttMinimizeSplitter::Right should be used, otherwise either KDGanttMinimizeSplitter::Up or KDGanttMinimizeSplitter::Down should be used.

4.1.4.4 void KDGanttMinimizeSplitter::setOrientation (Orientation *o*) [virtual]

the orientation of the splitter

By default the orientation is horizontal (the widgets are side by side). The possible orientations are Qt::Vertical and Qt::Horizontal (the default).

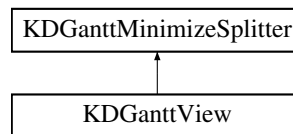
The documentation for this class was generated from the following files:

- KDGanttMinimizeSplitter.h
- KDGanttMinimizeSplitter.cpp

4.2 KDGanttView Class Reference

```
#include <KDGanttView.h>
```

Inheritance diagram for KDGanttView::



Public Types

- enum [Scale](#) {
Minute, Hour, Day, Week,
Month, Auto }
- enum [YearFormat](#) { **FourDigit, TwoDigit, TwoDigitApostrophe, NoDate** }
- enum [HourFormat](#) { **Hour_24, Hour_12, Hour_24_FourDigit** }
- enum [RepaintMode](#) { **No, Medium, Always** }

Public Slots

- void [editItem](#) ([KDGanttViewItem](#) *)
- void [zoomToSelection](#) (const QDateTime &start, const QDateTime &end)
- void [zoomToFit](#) ()
- void [setShowLegend](#) (bool show)
- void [setShowListView](#) (bool show)
- void [setListViewWidth](#) (int)
- void [setEditable](#) (bool editable)
- void [center](#) ([KDGanttViewItem](#) *)
- void [centerTimeline](#) (const QDateTime ¢er)
- void [setTimelineToStart](#) ()
- void [setTimelineToEnd](#) ()
- void [setShowTaskLinks](#) (bool show)
- void [setScale](#) ([Scale](#))

Signals

- void [timeIntervallSelected](#) (const QDateTime &start, const QDateTime &end)
- void [timeIntervalSelected](#) (const QDateTime &start, const QDateTime &end)
- void [rescaling](#) ([Scale](#))
- void [itemLeftClicked](#) ([KDGanttViewItem](#) *)
- void [itemMidClicked](#) ([KDGanttViewItem](#) *)
- void [itemRightClicked](#) ([KDGanttViewItem](#) *)
- void [itemDoubleClicked](#) ([KDGanttViewItem](#) *)
- void [gvCurrentChanged](#) ([KDGanttViewItem](#) *)
- void [gvItemLeftClicked](#) ([KDGanttViewItem](#) *)

- void [gvItemMidClicked](#) ([KDGanttViewItem *](#))
- void [gvItemRightClicked](#) ([KDGanttViewItem *](#))
- void [gvMouseButtonClicked](#) (int button, [KDGanttViewItem *](#)item, const [QPoint &pos](#))
- void [gvItemDoubleClicked](#) ([KDGanttViewItem *](#))
- void [gvContextMenuRequested](#) ([KDGanttViewItem *](#)item, const [QPoint &pos](#))
- void [lvCurrentChanged](#) ([KDGanttViewItem *](#))
- void [lvItemRenamed](#) ([KDGanttViewItem *](#), int col, const [QString &text](#))
- void [lvMouseButtonPressed](#) (int button, [KDGanttViewItem *](#)item, const [QPoint &pos](#), int c)
- void [lvItemLeftClicked](#) ([KDGanttViewItem *](#))
- void [lvItemMidClicked](#) ([KDGanttViewItem *](#))
- void [lvItemRightClicked](#) ([KDGanttViewItem *](#))
- void [lvContextMenuRequested](#) ([KDGanttViewItem *](#)item, const [QPoint &pos](#), int col)
- void [lvMouseButtonClicked](#) (int button, [KDGanttViewItem *](#)item, const [QPoint &pos](#), int c)
- void [lvItemDoubleClicked](#) ([KDGanttViewItem *](#))
- void [lvSelectionChanged](#) ([KDGanttViewItem *](#))
- void [itemConfigured](#) ([KDGanttViewItem *](#))
- void [taskLinkLeftClicked](#) ([KDGanttViewTaskLink *](#))
- void [taskLinkMidClicked](#) ([KDGanttViewTaskLink *](#))
- void [taskLinkRightClicked](#) ([KDGanttViewTaskLink *](#))
- void [taskLinkDoubleClicked](#) ([KDGanttViewTaskLink *](#))
- void [dropped](#) ([QDropEvent *e](#), [KDGanttViewItem *](#)droppedItem, [KDGanttViewItem *](#)itemBelow-[Mouse](#))

Public Member Functions

- [KDGanttView](#) ([QWidget *parent=0](#), const char *name=0)
- virtual void [show](#) ()
- virtual bool [close](#) (bool alsoDelete)
- void [setRepaintMode](#) ([RepaintMode mode](#))
- void [setUpdateEnabled](#) (bool enable)
- bool [getUpdateEnabled](#) () const
- void [setGanttMaximumWidth](#) (int w)
- int [ganttMaximumWidth](#) () const
- bool [showLegend](#) () const
- void [setLegendIsDockwindow](#) (bool dock)
- bool [legendIsDockwindow](#) () const
- [QDockWindow *](#) [legendDockwindow](#) () const
- bool [showListView](#) () const
- void [setEditorEnabled](#) (bool enable)
- bool [editorEnabled](#) () const
- int [listViewWidth](#) ()
- bool [editable](#) () const
- void [setCalendarMode](#) (bool mode)
- bool [calendarMode](#) () const
- void [setDisplaySubitemsAsGroup](#) (bool show)
- bool [displaySubitemsAsGroup](#) () const
- void [setDisplayEmptyTasksAsLine](#) (bool show)
- bool [displayEmptyTasksAsLine](#) () const
- void [setHorBackgroundLines](#) (int count=2, [QBrush brush=QBrush\(QColor\(200, 200, 200\), Qt::Dense6Pattern\)](#))

- int `horBackgroundLines` (QBrush &brush)
- bool `saveProject` (QIODevice *)
- bool `loadProject` (QIODevice *)
- void `print` (QPrinter *printer=0, bool printListView=true, bool printTimeLine=true, bool printLegend=false)
- QSize `drawContents` (QPainter *p=0, bool drawListView=true, bool drawTimeLine=true, bool drawLegend=false)
- void `setZoomFactor` (double factor, bool absolute)
- double `zoomFactor` () const
- void `ensureVisible` (KDGanttViewItem *)
- void `centerTimelineAfterShow` (const QDateTime ¢er)
- void `addTicksLeft` (int num=1)
- void `addTicksRight` (int num=1)
- bool `showTaskLinks` () const
- void `setFont` (const QFont &f)
- void `setShowHeaderPopupMenu` (bool show=true, bool showZoom=true, bool showScale=true, bool showTime=true, bool showYear=true, bool showGrid=true, bool showPrint=false)
- bool `showHeaderPopupMenu` () const
- void `setShowTimeTablePopupMenu` (bool)
- bool `showTimeTablePopupMenu` () const
- void `setShapes` (KDGanttViewItem::Type type, KDGanttViewItem::Shape start, KDGanttViewItem::Shape middle, KDGanttViewItem::Shape end, bool overwriteExisting=true)
- bool `shapes` (KDGanttViewItem::Type type, KDGanttViewItem::Shape &start, KDGanttViewItem::Shape &middle, KDGanttViewItem::Shape &end) const
- void `setColors` (KDGanttViewItem::Type type, const QColor &start, const QColor &middle, const QColor &end, bool overwriteExisting=true)
- bool `colors` (KDGanttViewItem::Type type, QColor &start, QColor &middle, QColor &end) const
- void `setDefaultColor` (KDGanttViewItem::Type type, const QColor &, bool overwriteExisting=true)
- QColor `defaultColor` (KDGanttViewItem::Type type) const
- void `setHighlightColors` (KDGanttViewItem::Type type, const QColor &start, const QColor &middle, const QColor &end, bool overwriteExisting=true)
- bool `highlightColors` (KDGanttViewItem::Type type, QColor &start, QColor &middle, QColor &end) const
- void `setDefaultHighlightColor` (KDGanttViewItem::Type type, const QColor &, bool overwriteExisting=true)
- QColor `defaultHighlightColor` (KDGanttViewItem::Type type) const
- void `setTextColor` (const QColor &color)
- QColor `textColor` () const
- void `setNoInformationBrush` (const QBrush &brush)
- QBrush `noInformationBrush` () const
- QList< KDGanttViewTaskLink > `taskLinks` () const
- QList< KDGanttViewTaskLinkGroup > `taskLinkGroups` () const
- void `addLegendItem` (KDGanttViewItem::Shape shape, const QColor &shapeColor, const QString &text)
- void `clearLegend` ()
- void `setHorizonStart` (const QDateTime &start)
- QDateTime `horizonStart` () const
- void `setHorizonEnd` (const QDateTime &start)
- QDateTime `horizonEnd` () const
- Scale `scale` () const
- void `setMaximumScale` (Scale)

- [Scale](#) `maximumScale () const`
- `void setMinimumScale (Scale)`
- [Scale](#) `minimumScale () const`
- `void setAutoScaleMinorTickCount (int count)`
- `int autoScaleMinorTickCount () const`
- `void setMajorScaleCount (int count)`
- `int majorScaleCount () const`
- `void setMinorScaleCount (int count)`
- `int minorScaleCount () const`
- `void setMinimumColumnWidth (int width)`
- `int minimumColumnWidth () const`
- `void setYearFormat (YearFormat format)`
- [YearFormat](#) `yearFormat () const`
- `void setHourFormat (HourFormat format)`
- [HourFormat](#) `hourFormat () const`
- `void setShowMajorTicks (bool)`
- `bool showMajorTicks () const`
- `void setShowMinorTicks (bool)`
- `bool showMinorTicks () const`
- `void setColumnBackgroundColor (const QDateTime &column, const QColor &color, Scale mini=KDGanttView::Minute, Scale maxi=KDGanttView::Month)`
- `void setIntervalBackgroundColor (const QDateTime &start, const QDateTime &end, const QColor &color, Scale mini=KDGanttView::Minute, Scale maxi=KDGanttView::Month)`
- `bool changeBackgroundInterval (const QDateTime &oldstart, const QDateTime &oldend, const QDateTime &newstart, const QDateTime &newend)`
- `bool deleteBackgroundInterval (const QDateTime &start, const QDateTime &end)`
- `void clearBackgroundColor ()`
- `QColor columnBackgroundColor (const QDateTime &column) const`
- `void setWeekendBackgroundColor (const QColor &color)`
- `QColor weekendBackgroundColor () const`
- `void setWeekdayBackgroundColor (const QColor &color, int weekday)`
- `QColor weekdayBackgroundColor (int weekday) const`
- `void setPaletteBackgroundColor (const QColor &col)`
- `void setGvBackgroundColor (const QColor &)`
- `void setLvBackgroundColor (const QColor &)`
- `void setTimeHeaderBackgroundColor (const QColor &)`
- `void setLegendHeaderBackgroundColor (const QColor &)`
- `QColor gvBackgroundColor () const`
- `QColor lvBackgroundColor () const`
- `QColor timeHeaderBackgroundColor () const`
- `QColor legendHeaderBackgroundColor () const`
- `void addUserdefinedLegendHeaderWidget (QWidget *w)`
- `void setWeekendDays (int start, int end)`
- `void weekendDays (int &start, int &end) const`
- `void setHeaderVisible (bool)`
- `bool headerVisible () const`
- `void setShowLegendButton (bool show)`
- `bool showLegendButton () const`
- `virtual int addColumn (const QString &label, int width=-1)`
- `virtual int addColumn (const QIconSet &iconset, const QString &label, int width=-1)`

- virtual void [removeColumn](#) (int index)
- [KDGanttViewItem](#) * [selectedItem](#) () const
- void [setSelected](#) ([KDGanttViewItem](#) *, bool)
- [KDGanttViewItem](#) * [firstChild](#) () const
- [KDGanttViewItem](#) * [lastItem](#) () const
- int [childCount](#) () const
- void [clear](#) ()
- void [setDragEnabled](#) (bool b)
- void [setDropEnabled](#) (bool b)
- void [setDragDropEnabled](#) (bool b)
- bool [dragEnabled](#) () const
- bool [dropEnabled](#) () const
- bool [isDragEnabled](#) () const
- bool [isDropEnabled](#) () const
- virtual bool [lvDropEvent](#) (QDropEvent *e, [KDGanttViewItem](#) *, [KDGanttViewItem](#) *)
- virtual void [lvStartDrag](#) ([KDGanttViewItem](#) *)
- virtual bool [lvDragMoveEvent](#) (QDragMoveEvent *e, [KDGanttViewItem](#) *, [KDGanttViewItem](#) *)
- virtual void [lvDragEnterEvent](#) (QDragEnterEvent *e)
- QSize [sizeHint](#) ()
- [KDGanttViewItem](#) * [getItemByName](#) (const QString &name) const
- QDateTime [getTimeForCoordX](#) (int coordX, bool global=true) const
- [KDGanttViewItem](#) * [getItemByListViewPos](#) (const QPoint &pos) const
- [KDGanttViewItem](#) * [getItemByGanttViewPos](#) (const QPoint &pos) const
- [KDGanttViewItem](#) * [getItemAt](#) (const QPoint &pos, bool global=true) const
- void [setLvVScrollBarMode](#) (QScrollView::ScrollBarMode)
- void [setGvVScrollBarMode](#) (QScrollView::ScrollBarMode)

Static Public Member Functions

- QPixmap [getPixmap](#) ([KDGanttViewItem::Shape](#) shape, const QColor &shapeColor, const QColor &background-color, int itemSize)

Protected Member Functions

- virtual QDragObject * [dragObject](#) ()
- virtual void [startDrag](#) ()

4.2.1 Detailed Description

This class represents a Gantt view with the Gantt chart, the header, an optional listview and an optional legend.

In order to set up a Gantt view, create an object of this class, and populate it with a number of [KDGanttViewItem](#) objects.

If you experience problems with the repainting of the content of the Gantt View after scrolling, call [setRepaintMode\(\)](#).

4.2.2 Member Enumeration Documentation

4.2.2.1 enum [KDGanttView::HourFormat](#)

This enum is used to specify the hour format used in the header.

4.2.2.2 enum [KDGanttView::RepaintMode](#)

Please see [setRepaintMode\(\)](#) for a description of the values of this enumeration.

4.2.2.3 enum [KDGanttView::Scale](#)

This enum is used to specify the units of the scales in the header.

4.2.2.4 enum [KDGanttView::YearFormat](#)

This enum is used to specify the year format used in the header.

4.2.3 Constructor & Destructor Documentation

4.2.3.1 [KDGanttView::KDGanttView](#) ([QWidget](#) * *parent* = 0, const char * *name* = 0)

Constructs an empty KDGanttView.

Parameters:

parent the widget parent

name the internal debugging name

4.2.4 Member Function Documentation

4.2.4.1 int [KDGanttView::addColumn](#) (const [QIconSet](#) & *iconset*, const [QString](#) & *label*, int *width* = -1) [[virtual](#)]

Calls to this method are passed through to the underlying [QListView](#).

4.2.4.2 int [KDGanttView::addColumn](#) (const [QString](#) & *label*, int *width* = -1) [[virtual](#)]

Calls to this method are passed through to the underlying [QListView](#).

4.2.4.3 void [KDGanttView::addLegendItem](#) ([KDGanttViewItem::Shape](#) *shape*, const [QColor](#) & *shapeColor*, const [QString](#) & *text*)

Adds an item to the legend.

Parameters:

shape the shape to display

shapeColor the color in which to display the shape

text the text to display

See also:

[clearLegend\(\)](#)

4.2.4.4 void KDGanttView::addTicksLeft (int *num* = 1)

Add *num* minor ticks of the current scale of the timeline to the start of the timeline. The timeline is not set automatically at the start. Call [setTimelineToStart\(\)](#) to ensure that the timeline is at the start after calling this method.

Parameters:

num the number of minor ticks which should be added

See also:

[addTicksRight\(\)](#), [setTimelineToStart\(\)](#), [setTimelineToEnd\(\)](#)

4.2.4.5 void KDGanttView::addTicksRight (int *num* = 1)

Add *num* minor ticks of the current scale of the timeline to the end of the timeline. The timeline is not set automatically at the end. Call [setTimelineToEnd\(\)](#) to ensure that the timeline is at the end after calling this method.

Parameters:

num the number of minor ticks which should be added

See also:

[addTicksLeft\(\)](#), [setTimelineToStart\(\)](#), [setTimelineToEnd\(\)](#)

4.2.4.6 void KDGanttView::addUserdefinedLegendHeaderWidget (QWidget * *w*)

Adds a widget to the spacer widget above the list view part and below the ShowLegendButton. To assign all the space above the Listview to the spacer widget, hide the ShowLegendButton by calling [setShowLegendButton\(false \)](#). The spacer widget is a QHBox. You may add as many widgets as you want. They are ordered horizontally from left to right. To remove a widget from the spacer widget, call `widget->reparent(newParent,...)` or delete the widget. Since the spacer is a QHBox, the layout of the added widgets is managed by this QHBox.

Parameters:

w A pointer to the widget to be added.

See also:

[setShowLegendButton\(\)](#)

4.2.4.7 int KDGanttView::autoScaleMinorTickCount () const

Returns the absolut number of minor ticks, if scaling is set to Auto

Returns:

the absolut number of minor ticks

See also:

[setAutoScaleMinorTickCount\(\)](#), [setScale\(\)](#), [scale\(\)](#)

4.2.4.8 bool KDGanttView::calendarMode () const

Returns true, if the Gantt view is in calendar mode. See [setCalendarMode\(\)](#) for the meaning of calendar mode.

Returns:

returns true, if the Gantt view is in calendermode

See also:

[setCalendarMode\(\)](#)

4.2.4.9 void KDGanttView::center (KDGanttViewItem * item) [slot]

Makes sure that the specified Gantt item is in the center of the visible Gantt chart (if possible).

4.2.4.10 void KDGanttView::centerTimeline (const QDateTime & center) [slot]

Makes sure that the specified QDateTime is in the center of the visible Gantt chart (if possible). If you want to center the timeline when the KDGanttView is hidden, calling [centerTimelineAfterShow\(\)](#) is the better alternative.

See also:

[center\(\)](#), [centerTimelineAfterShow\(\)](#)

4.2.4.11 void KDGanttView::centerTimelineAfterShow (const QDateTime & center)

Makes sure that the specified QDateTime is in the center of the visible Gantt chart (if possible). If the KDGanttView is currently hidden, this method resets the center once again after the next [show\(\)](#) call. Use this method if you want to center the timeline when the KDGanttView is hidden. After calling [KDGanttView::show\(\)](#), there may be computations of the sizes of the widgets and subwidgets and of the automatically computed start datetime. This method ensures that the center of the timeline is to be properly reset after [show\(\)](#).

See also:

[center\(\)](#), [centerTimeline\(\)](#)

4.2.4.12 **bool KDGanttView::changeBackgroundInterval (const QDateTime & *oldstart*, const QDateTime & *oldend*, const QDateTime & *newstart*, const QDateTime & *newend*)**

Changes the times of an already defined background color interval. The new values *startnew* and *endnew* should not be datetime values of an already defined background color interval. If that is the case, nothing is changed and false is returned.

Parameters:

oldstart the start date and time of the interval to change

oldend the end date and time of the interval to change

newstart the new start date and time

newend the new end date and time

Returns:

true, if there is a backgroundcolor interval with values *start* and *end* found and the new values *startnew* and *endnew* are not datetime values of an already defined background color interval. Returns false otherwise.

See also:

[changeBackgroundInterval\(\)](#), [deleteBackgroundInterval\(\)](#), [columnBackgroundColor\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#)

4.2.4.13 **int KDGanttView::childCount () const**

Returns the number of items in the Gantt view.

Returns:

the number of items in the Gantt view.

4.2.4.14 **void KDGanttView::clear ()**

Removes all items from the Gantt view.

4.2.4.15 **void KDGanttView::clearBackgroundColor ()**

Removes all background color settings set with [setColumnBackgroundColor\(\)](#) and [setIntervalBackgroundColor\(\)](#). Does not affect the settings of [setWeekendBackgroundColor\(\)](#).

See also:

[setColumnBackgroundColor\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#), [setIntervalBackgroundColor\(\)](#)

4.2.4.16 **void KDGanttView::clearLegend ()**

Removes all items from the legend.

See also:

[addLegendItem\(\)](#)

4.2.4.17 bool KDGanttView::close (bool *alsoDelete*) [virtual]

Closes the widget. The closing is rejected, if a repainting is currently being done.

Parameters:

alsoDelete if true, the widget is deleted

Returns:

true, if the widget was closed

4.2.4.18 bool KDGanttView::colors (KDGanttViewItem::Type *type*, QColor & *start*, QColor & *middle*, QColor & *end*) const

Queries the colors for a particular type of Gantt item.

Parameters:

type the type of Gantt items for which to query the colors

start the start color is returned in this parameter

middle the middle color is returned in this parameter

end the end color is returned in this parameter

Returns:

true if there was a general color set for the specified type. If the return value is false, the values of the three color parameters are undefined.

See also:

[setColor\(\)](#), [setDefaultColor\(\)](#), [defaultColor\(\)](#)

4.2.4.19 QColor KDGanttView::columnBackgroundColor (const QDateTime & *column*) const

Returns the background color for the column closest to *column*.

Parameters:

column the column to query the background color for

Returns:

the background color of the specified color

See also:

[setColumnBackgroundColor\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#)

4.2.4.20 QColor KDGanttView::defaultColor (KDGanttViewItem::Type *type*) const

Returns the default color for a particular type of Gantt item that is used for the item if no specific start, middle, or end colors are set.

Parameters:

type the type of Gantt items for which to query the highlight colors

Returns:

color the default color used

See also:

[setDefaultColor\(\)](#), [setColors\(\)](#), [colors\(\)](#)

4.2.4.21 QColor KDGanttView::defaultHighlightColor (KDGanttViewItem::Type type) const

Returns the default highlighting color for a particular type of Gantt item that is used for the item if no specific start, middle, or end colors are set.

Parameters:

type the type of Gantt items for which to query the highlight colors

Returns:

color the default highlighting color used

See also:

[setDefaultHighlightColor\(\)](#), [setHighlightColors\(\)](#), [highlightColors\(\)](#)

4.2.4.22 bool KDGanttView::deleteBackgroundInterval (const QDateTime & start, const QDateTime & end)

Deletes an already defined background color interval.

Parameters:

start start datetime of time interval

end end datetime of time interval

Returns:

true, if there is a backgroundcolor interval with values *start* and *end* found (and hence deleted).

See also:

[changeBackgroundInterval\(\)](#), [columnBackgroundColor\(\)](#)

4.2.4.23 bool KDGanttView::displayEmptyTasksAsLine () const

Returns, whether tasks where the start time and the end time are the same are displayed as a line over the full height of the Gantt view.

Returns:

true, if empty tasks are displayed as line.

4.2.4.24 bool KDGanttView::displaySubitemsAsGroup () const

Returns, whether new items are created with the displayHiddenSubitems property.

Returns:

true, if hidden subitems should be displayed on newly created items.

See also:

[setDisplaySubitemsAsGroup\(\)](#), [KDGanttViewItem::setDisplaySubitemsAsGroup\(\)](#), [KDGanttViewItem::displaySubitemsAsGroup\(\)](#)

4.2.4.25 bool KDGanttView::dragEnabled () const

Deprecated

Use [isDragEnabled\(\)](#) instead

4.2.4.26 QDragObject * KDGanttView::dragObject () [protected, virtual]

Implements a pass-through to the list view.

4.2.4.27 QSize KDGanttView::drawContents (QPainter * p = 0, bool drawListView = true, bool drawTimeLine = true, bool drawLegend = false)

Paints a Gantt view on a QPainter. You can specify, whether the list view, the time line, or the legend is painted. All combinations of these three widgets are allowed. Returns the size of the painted area. Paints the list view in the top-left corner, the time line to the right of the list view, and the legend below the list view. If called with $p = 0$, nothing is painted and only the size of the painted area is computed. This is useful for determining only the painted area and setting the scale of the painter, before calling this method with a painter. In order to get the output fitted to your paper and your printer, call first `QSize size = drawContents(0, printListView, printTimeLine, printLegend);` then compute the scale $dx = \text{paper.width()} / \text{size.width()};$ $dy = \text{paper.height()} / \text{size.height()};$ then make float scale to fit the width or height of the paper if $(dx < dy)$ $scale = dx;$ else $scale = dy;$ then set the scale `p.scale(scale, scale);` and now call `drawContents` with painter `p` `drawContents(&p, printListView, printTimeLine, printLegend);`

For a detailed example, please see the commented source code in [KDGanttView::print\(...\)](#)

Parameters:

p a pointer to the painter to paint on. If p is 0, nothing is painted and only the size of the painted area is computed

drawListView if true, the list view is painted

drawTimeLine if true, the time line is painted

drawLegend if true, the legend is painted

Returns:

the size of the painted area

See also:

[print\(\)](#)

4.2.4.28 bool KDGanttView::dropEnabled () const

Deprecated

Use [isDropEnabled\(\)](#) instead

4.2.4.29 void KDGanttView::dropped (QDropEvent * *e*, KDGanttViewItem * *droppedItem*, KDGanttViewItem * *itemBelowMouse*) [signal]

This signal is emitted whenever a Gantt item is dropped onto the Gantt view. *droppedItem* is 0, if this is a drag operation from another KDGanttView instance. If this drag is an internal drag (i.e. within the KDGanttView), this parameter points to the dropped item. *itemBelowMouse* is a pointer to the item below the dragged item (i.e., below the mouse). The dragged item may be inserted in the KDGanttView as a child of this item. If The value is 0, if there is no item below the dragged item, and the dragged item will be inserted as a root item.

In order to get user-defined behavior for drop events, reimplement [KDGanttView::lvDropEvent\(\)](#)

4.2.4.30 bool KDGanttView::editable () const

Returns whether the Gantt chart is user-editable

Returns:

true if the Gantt chart is user-editable

See also:

[setEditable\(\)](#)

4.2.4.31 void KDGanttView::editItem (KDGanttViewItem * *item*) [slot]

This slot is called when a new item has been added to the Gantt view. It will show the item attribute dialog in case the item is editable. *item* is a pointer to the item that has been created.

4.2.4.32 bool KDGanttView::editorEnabled () const

Returns whether it is possible to edit the appearance of a Gantt item visually in a dialog by double-clicking the item.

Returns:

true if visual editing is enabled, false otherwise

See also:

[setEditorEnabled\(\)](#)

4.2.4.33 void KDGanttView::ensureVisible (KDGanttViewItem * *item*)

Makes sure that the specified Gantt item is visible without scrolling.

See also:

[center\(\)](#), [centerTimelineAfterShow\(\)](#)

4.2.4.34 `KDGanttViewItem * KDGanttView::firstChild () const`

Returns the first item in the Gantt view.

Returns:

the first item in the Gantt view, 0 if there are no items

4.2.4.35 `int KDGanttView::gantMaximumWidth () const`

Returns the maximum width of the Gantt view part widget in pixels. The default maximum width is 32767 pixels.

Returns:

the maximum width of the Gantt view part widget in pixels.

4.2.4.36 `QDateTime KDGanttView::getDateForCoordX (int coordX, bool global = true) const`

Returns the corresponding date and time of the coordinate X in the Gantt view.

Parameters:

coordX the coordinate to search for

global true if coordX is a global position, false otherwise

Returns:

the date and time at coordinate X in the Gantt view.

4.2.4.37 `KDGanttViewItem * KDGanttView::getItemAt (const QPoint & pos, bool global = true) const`

Returns the pointer to the Gantt item at the position *pos* in the list view part of the Gantt view. The position *pos* is a global position if parameter *global* is true. If the vertical part (y coordinate) of *pos* (mapped to local coordinates) is less than 0 or larger than the height of the listview, 0 is returned. The horizontal part (x coordinate) of *pos* is ignored.

Parameters:

pos the position of the Gantt item

global if true, pos is assumed to be global

Returns:

the pointer to the item with position *pos*. 0, if there is no item in the Gantt view at this position.

4.2.4.38 `KDGanttViewItem * KDGanttView::getItemByGanttViewPos (const QPoint & pos) const`

Returns the pointer to the Gantt item at the position *pos* in the Gantt view. The position *pos* is a global position. If no items are found, or the item is disabled, 0 is returned. If there is more than one item with the same position in the Gantt view, the first item found will be returned. This is not necessarily the first item in the listview.

Parameters:

pos the (global) position of the Gantt item

Returns:

the pointer to the item with position *pos*. 0, if there is no item in the Gantt view at this position.

4.2.4.39 KDGanttViewItem * KDGanttView::getItemByListViewPos (const QPoint & pos) const

Returns the pointer to the Gantt item at the position *pos* in the list view. The position *pos* is a global position. If no item is found, 0 is returned.

Parameters:

pos the (global) position of the Gantt item

Returns:

the pointer to the item with position *pos*. 0, if there is no item in the list view at this position.

4.2.4.40 KDGanttViewItem * KDGanttView::getItemByName (const QString & name) const

Returns the pointer to the Gantt item with the name *name*. If no item is found, the return value is 0. If there is more than one item with the same name in the Gantt view, the first item found will be returned. This may not necessarily be the first item in the listview.

Parameters:

name the name of the Gantt item

Returns:

the pointer to the item with name *name*. 0, if there is no item in the Gantt view with this name.

4.2.4.41 QPixmap KDGanttView::getPixmap (KDGanttViewItem::Shape shape, const QColor & shapeColor, const QColor & backgroundColor, int itemSize) [static]

This method returns the pixmap used for a certain shape, in the selected color and size.

Parameters:

shape the shape to generate

shapeColor the foreground color of the shape

backgroundColor the background color of the shape

itemSize the size of the shape

Returns:

the generated shape pixmap

4.2.4.42 bool KDGanttView::getUpdateEnabled () const

Returns whether updating is enabled or not.

Returns:

true, if updating is enabled

See also:

[setUpdateEnabled\(\)](#)

4.2.4.43 QColor KDGanttView::gvBackgroundColor () const

Returns the background color of the Gantt view.

Returns:

the background color of the Gantt view

See also:

[setGvBackgroundColor\(\)](#)

4.2.4.44 void KDGanttView::gvContextMenuRequested (KDGanttViewItem * item, const QPoint & pos) [signal]

This signal is emitted when the user requests a context menu in the Gantt view. Notice that *pos* is the absolute mouse position.

4.2.4.45 void KDGanttView::gvCurrentChanged (KDGanttViewItem * item) [signal]

This signal is emitted whenever the user clicks on the Gantt view *item* parameter is 0, if no item was clicked

4.2.4.46 void KDGanttView::gvItemDoubleClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user double-clicks into the Gantt view.

4.2.4.47 void KDGanttView::gvItemLeftClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user clicks into the Gantt view with the left mouse button.

4.2.4.48 void KDGanttView::gvItemMidClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user clicks into the Gantt view with the middle mouse button.

4.2.4.49 void KDGanttView::gvItemRightClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user clicks into the Gantt view with the right mouse button.

4.2.4.50 void KDGanttView::gvMouseButtonClicked (int *button*, KDGanttViewItem * *item*, const QPoint & *pos*) [signal]

This signal is emitted when the user clicks into the Gantt view with any mouse button. Notice that *pos* is the absolute mouse position.

4.2.4.51 bool KDGanttView::headerVisible () const

Returns whether the listview header is visible.

Returns:

whether the header is visible

4.2.4.52 bool KDGanttView::highlightColors (KDGanttViewItem::Type *type*, QColor & *start*, QColor & *middle*, QColor & *end*) const

Queries the highlight colors for a particular type of Gantt item.

Parameters:

type the type of Gantt items for which to query the highlight colors

start the start highlight color is returned in this parameter

middle the middle highlight color is returned in this parameter

end the end highlight color is returned in this parameter

Returns:

true if there was a general highlight color set for the specified type. If the return value is false, the values of the three highlight color parameters are undefined.

See also:

[setHighlightColors\(\)](#), [setDefaultHighlightColor\(\)](#), [defaultHighlightColor\(\)](#)

4.2.4.53 int KDGanttView::horBackgroundLines (QBrush & *brush*)

Returns the definition of the horizontal background lines of the Gantt chart.

Parameters:

brush the brush of the lines

Returns:

every *n*th line gets a background specified by brush if 0 is returned, no background lines are drawn

4.2.4.54 QDateTime KDGanttView::horizonEnd () const

Returns the end of the horizon of the Gantt chart.

Returns:

the end of the horizon of the Gantt chart

See also:

[setHorizonEnd\(\)](#)

4.2.4.55 QDateTime KDGanttView::horizonStart () const

Returns the start of the horizon of the Gantt chart.

Returns:

the start of the horizon of the Gantt chart

See also:

[setHorizonStart\(\)](#)

4.2.4.56 KDGanttView::HourFormat KDGanttView::hourFormat () const

Returns the format in which to display hours.

Returns:

the hour format

See also:

[setHourFormat\(\)](#), [setYearFormat\(\)](#), [yearFormat\(\)](#)

4.2.4.57 bool KDGanttView::isDragEnabled () const

Returns whether dragging is enabled for this Gantt view.

Returns:

true if dragging is enabled

See also:

[setDragEnabled\(\)](#), [setDragDropEnabled\(\)](#)

4.2.4.58 bool KDGanttView::isDropEnabled () const

Returns whether dropping is enabled for this Gantt view.

Returns:

true if dropping is enabled

See also:

[setDropEnabled\(\)](#), [setDragDropEnabled\(\)](#)

4.2.4.59 void KDGanttView::itemConfigured (KDGanttViewItem *) [signal]

This signal is emitted when the user has configured an item visually.

4.2.4.60 void KDGanttView::itemDoubleClicked (KDGanttViewItem *) [signal]

This signal is emitted when the user double-clicks an item.

4.2.4.61 void KDGanttView::itemLeftClicked (KDGanttViewItem *) [signal]

This signal is emitted when the user clicks on an item with the left mouse button.

4.2.4.62 void KDGanttView::itemMidClicked (KDGanttViewItem *) [signal]

This signal is emitted when the user clicks on an item with the middle mouse button.

4.2.4.63 void KDGanttView::itemRightClicked (KDGanttViewItem *) [signal]

This signal is emitted when the user clicks on an item with the right mouse button.

4.2.4.64 KDGanttViewItem * KDGanttView::lastItem () const

Returns the last item in the Gantt view

Returns:

the last item in the Gantt view, 0 if there are no items

4.2.4.65 QDockWindow * KDGanttView::legendDockwindow () const

Returns the pointer to the legend dock window. DO NOT DELETE THIS POINTER! If the legend is not a dock window, 0 is returned To set the legend as a dock window, call KDGanttView::setLegendIsDockwindow(true);

Returns:

the pointer to the legend dock window 0 is returned, if the legend is no dock window DO NOT DELETE THIS POINTER!

See also:

[setShowLegend\(\)](#), [setLegendIsDockwindow\(\)](#),[legendIsDockwindow\(\)](#)

4.2.4.66 QColor KDGanttView::legendHeaderBackgroundColor () const

Returns the background color of the legend header.

Returns:

the background color of the legend header

See also:

[setLegendHeaderBackgroundColor\(\)](#)

4.2.4.67 bool KDGanttView::legendIsDockwindow () const

Returns whether the legend is shown as a dock window

Returns:

true if the legend is shown as a dock window

See also:

[setShowLegend\(\)](#), [setLegendIsDockwindow\(\)](#), [legendDockwindow\(\)](#)

4.2.4.68 int KDGanttView::listViewWidth ()

Returns the width of the list view.

Returns:

the width of the list view

See also:

[setListViewWidth\(\)](#)

4.2.4.69 bool KDGanttView::loadProject (QIODevice * *device*)

Loads a previously saved state of the Gantt view. All current settings and items are discarded before loading the data.

Parameters:

device a pointer to the IO device from which to load the Gantt view state.

Returns:

true if the file could be read, false if an error occurred

See also:

[saveProject\(\)](#)

4.2.4.70 QColor KDGanttView::lvBackgroundColor () const

Returns the background color of the list view.

Returns:

the background color of the list view

See also:

[setLvBackgroundColor\(\)](#)

4.2.4.71 void KDGanttView::lvContextMenuRequested (KDGanttViewItem * *item*, const QPoint & *pos*, int *col*) [signal]

This signal is emitted when the user requests a context menu in the list view. Notice that *pos* is the absolute mouse position.

4.2.4.72 void KDGanttView::lvCurrentChanged (KDGanttViewItem * *item*) [signal]

This signal is emitted whenever the user clicks on the list view *item* parameter is 0, if no item was clicked

4.2.4.73 void KDGanttView::lvDragEnterEvent (QDragEnterEvent * e) [virtual]

This virtual method specifies whether a drag enter event may be accepted or not. To accept a drag enter event, call `e->accept(true)`; To not accept a drag enter event, call `e->accept(false)`; This method does nothing but accepting the drag enter event, in case decoding is possible. In order to define accepting drops for particular items yourself, subclass `KDGanttView` and reimplement this method.

Parameters:

- e* The `QDragMoveEvent` Note: `e->source()` is a pointer to the `KDGanttView`, the drag started from. I.e., if `e->source() == this`, this drag is an internal drag.

See also:

[lvDropEvent\(\)](#), [lvStartDrag\(\)](#), [lvDragMoveEvent\(\)](#)

4.2.4.74 bool KDGanttView::lvDragMoveEvent (QDragMoveEvent * e, KDGanttViewItem *, KDGanttViewItem *) [virtual]

This virtual method specifies whether a drop event may be accepted or not. To accept a drop event, call `e->accept(true)`; To not accept a drop event, call `e->accept(false)`; This method does nothing but allowing to execute the internal drag move event handling.

In order to specify user-defined drop acceptance for particular items, subclass `KDGanttView` and reimplement this method.

Parameters:

- e* The `QDragMoveEvent` Note: `e->source()` is a pointer to the `KDGanttView`, the drag started from. I.e. if `e->source() == this`, this drag is an internal drag. `draggedItem` 0, if this is a drag operation from another `KDGanttView` instance. If this drag is an internal drag (i.e., within the `KDGanttView`), this parameter points to the dragged item. `itemBelowMouse` a pointer to the item below the dragged item (i.e., below the mouse). If you accept the drop, the dragged item will be inserted in the `KDGanttView` as a child of this item. The value is 0 if there is no item below the dragged item, and the dragged item will be inserted as a root item.

Returns:

false, when the internal drag move event handling should be executed true, when the internal drag move event handling should not be executed; usually you should return true, if you have called `e->accept(true)` before.

See also:

[lvDropEvent\(\)](#), [lvStartDrag\(\)](#)

4.2.4.75 bool KDGanttView::lvDropEvent (QDropEvent * e, KDGanttViewItem * droppedItem, KDGanttViewItem * itemBelowMouse) [virtual]

This virtual method makes it possible to specify user-defined drop handling. The method is called directly before the internal drop handling is executed. Return false to execute internal drop handling. Return true to not execute internal drop handling. In order to specify user-defined drop handling, subclass `KDGanttView` and reimplement this method.

Parameters:

- e* The `QDropEvent` Note: `e->source()` is a pointer to the `KDGanttView` from which the drag started. I.e., if `e->source() == this`, this drag is an internal drag.

droppedItem 0, if this is a drag operation from another KDGanttView instance. If this drag is an internal drag (i.e., within the KDGanttView), this parameter points to the dropped item.

itemBelowMouse a pointer to the item below the dragged item (i.e., below the mouse). If you accept, the dragged item may be inserted in the KDGanttView as a child of this item. The value is 0 if there is no item below the dragged item, and the dragged item will be inserted as a root item.

Returns:

false, when the internal drop handling should be executed true, when the internal drop handling should not be executed

See also:

[lvDropEvent\(\)](#), [lvStartDrag\(\)](#)

4.2.4.76 void KDGanttView::lvItemDoubleClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user double-clicks into the list view.

4.2.4.77 void KDGanttView::lvItemLeftClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user clicks into the list view with the left mouse button.

4.2.4.78 void KDGanttView::lvItemMidClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user clicks into the list view with the middle mouse button.

4.2.4.79 void KDGanttView::lvItemRenamed (KDGanttViewItem *, int col, const QString & text) [signal]

This signal is emitted whenever the user changes the name of an item in the list view using in-place editing. *text* contains the new text in the list view.

4.2.4.80 void KDGanttView::lvItemRightClicked (KDGanttViewItem *) [signal]

This signal is emitted whenever the user clicks into the list view with the right mouse button.

4.2.4.81 void KDGanttView::lvMouseButtonClicked (int button, KDGanttViewItem * item, const QPoint & pos, int col) [signal]

This signal is emitted when the user clicks into the list view with any mouse button . Notice that *pos* is the absolute mouse position.

4.2.4.82 void KDGanttView::lvMouseButtonPressed (int button, KDGanttViewItem * item, const QPoint & pos, int col) [signal]

This signal is emitted when the user presses any mouse button in the list view. Notice that *pos* is the absolute mouse position.

4.2.4.83 void KDGanttView::lvSelectionChanged (KDGanttViewItem *) [signal]

This signal is emitted whenever the user changes the selection in the list view.

4.2.4.84 void KDGanttView::lvStartDrag (KDGanttViewItem * item) [virtual]

This virtual method creates a QDragObject and starts a drag for a [KDGanttViewItem](#). In order to prevent drags of particular items, subclass from KDGanttView and reimplement this method.

Parameters:

item the [KDGanttViewItem](#), which should be dragged

See also:

[lvDropEvent\(\)](#), [lvDragMoveEvent\(\)](#)

4.2.4.85 int KDGanttView::majorScaleCount () const

Returns the number of ticks per unit in the major scale.

Returns:

the number of ticks in the major scale

See also:

[setMajorScaleCount\(\)](#), [setMinorScaleCount\(\)](#), [minorScaleCount\(\)](#)

4.2.4.86 KDGanttView::Scale KDGanttView::maximumScale () const

Returns the maximum allowed time scale of the lower scale of the header.

Returns:

the unit of the lower scale of the header.

See also:

[setScale\(\)](#)

4.2.4.87 int KDGanttView::minimumColumnWidth () const

Returns the minimum width a column needs to have.

Returns:

the column minimum width

See also:

[setMinimumColumnWidth\(\)](#)

4.2.4.88 [KDGanttView::Scale](#) `KDGanttView::minimumScale () const`

Returns the minimum allowed time scale of the lower scale of the header.

Returns:

the unit of the lower scale of the header.

See also:

[setScale\(\)](#)

4.2.4.89 `int KDGanttView::minorScaleCount () const`

Returns the number of ticks per unit in the minor scale.

Returns:

the number of ticks in the minor scale

See also:

[setMinorScaleCount\(\)](#), [setMajorScaleCount\(\)](#), [majorScaleCount\(\)](#)

4.2.4.90 `QBrush KDGanttView::noInformationBrush () const`

Returns the brush of the 'showNoInformation' lines

Returns:

the brush of the 'showNoInformation' lines

See also:

[KDGanttViewItem::showNoInformation\(\)](#), [KDGanttViewItem::setShowNoInformation\(\)](#), [setNoInformationBrush\(\)](#)

4.2.4.91 `void KDGanttView::print (QPrinter * printer = 0, bool printListView = true, bool printTimeLine = true, bool printLegend = false)`

Sends a Gantt view to a printer. The printer should already be set up for printing (by calling `QPrinter::setup()`). If the printer is not set up, `QPrinter::setup()` is called before printing

You can specify, whether the ListView, TimeLine, or Legend will be printed. All combinations of these three widgets are allowed.

Parameters:

printer a pointer to the printer to print on. If *printer* is 0, the method creates a temporary printer and discards it when it is done printing.

printListView if true, the list view is printed

printTimeLine if true, the time line is printed

printLegend if true, the legend is printed

See also:

[drawContents\(\)](#)

4.2.4.92 void KDGanttView::removeColumn (int *index*) [virtual]

Calls to this method are passed through to the underlying *QListView*.

4.2.4.93 void KDGanttView::rescaling (Scale) [signal]

This signal is emitted if another scale is chosen than the specified one: i.e. if the horizon has a very wide range from start to end and as scale is chosen minute it may be that the size of the Gantt widget would become more than 32000 pixels. In this case the scale is automatically changed to Hour and rescaling(Hour) is emitted. If the widget size would be still more than 32000 pixels, the scale is automatically changed to day and rescaling(Day) is emitted. In the new scale, the minortickcount is increased such that the horizon will fit in the maximum size of 32000 pixels.

4.2.4.94 bool KDGanttView::saveProject (QIODevice * *device*)

Saves the state of the Gantt view in an IO device in XML format. The saved data can be reloaded with [loadProject\(\)](#).

Parameters:

device a pointer to the IO device in which to store the Gantt view state.

Returns:

true if the data could be written, false if an error occurred

See also:

[loadProject\(\)](#)

4.2.4.95 KDGanttView::Scale KDGanttView::scale () const

Returns the unit of the lower scale of the header.

Returns:

the unit of the lower scale of the header.

See also:

[setScale\(\)](#)

4.2.4.96 KDGanttViewItem * KDGanttView::selectedItem () const

Calls to this method are passed through to the underlying *QListView*.

4.2.4.97 void KDGanttView::setAutoScaleMinorTickCount (int *count*)

Sets the absolute number of minor ticks, if scaling is set to Auto. If the scale mode is set to Auto, then the actual scale and the minorScaleCount is automatically computed, such that there are count minor ticks

Parameters:

count the number of minor ticks

See also:

[autoScaleMinorTickCount\(\)](#),[setScale\(\)](#),[scale\(\)](#)

4.2.4.98 void KDGanttView::setCalendarMode (bool *mode*)

This method turns calendar mode on and off. In calendar mode, only those items can be opened which have subitems which have subitems. I.e., if an item contains multiple calendars, it can be opened, but not a calendar item itself. If you want to use this GanttView as a calendar view, you have to call `setDisplaySubitemsAsGroup(true)`; to use the root items as calendar items. To create new calendar entries for these root items, create a new [KDGanttViewTaskItem](#) with this root item as a parent. If you want an item with subitems to behave like a calendar (which is possibly empty at startup), please call `setIsCalendar(true)`; for this item.

Parameters:

mode if true, the calendar view mode is turned on if false, the calendar view mode is turned off

See also:

[setDisplaySubitemsAsGroup\(\)](#), [displaySubitemsAsGroup\(\)](#), [calendarMode\(\)](#)

4.2.4.99 void KDGanttView::setColors (KDGanttViewItem::Type *type*, const QColor & *start*, const QColor & *middle*, const QColor & *end*, bool *overwriteExisting* = true)

Sets the colors for a certain type of Gantt item. Not all items use all three colors (e.g., only summary items use the middle color).

This setting overrides any color settings made on individual items. These settings will be taken as initial values of any newly created item of this certain type. See also the description of the [KDGanttViewItem](#) class.

Parameters:

type the type of Gantt items for which to set the colors

start the color to use for the beginning of the item

middle the color to use for the middle of the item

end the color to use for the end of the item

overwriteExisting if true, overwrites existing color settings on individual items

See also:

[colors\(\)](#), [setDefaultColors\(\)](#), [defaultColors\(\)](#)

4.2.4.100 void KDGanttView::setColumnBackgroundColor (const QDateTime & *column*, const QColor & *color*, Scale *mini* = KDGanttView::Minute, Scale *maxi* = KDGanttView::Month)

Sets the background color for the column closest to *column*. It can be specified whether the color should be shown in all scales or only in specific scales. If you want to define the color only for the daily view, specify *mini* and *maxi* as Day. If there is no value specified for *mini* and *maxi*, the color for the column is shown on all scales. Note that it is possible that there are two values for a column in a scale. In this case, the shown color is unspecified.

Parameters:

column the column to set the background color for

color the background color

mini show the colour only in scales greater than this

maxi show the colour only in scales lesser than this

See also:

[columnBackgroundColor\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#)

4.2.4.101 void KDGanttView::setDefaultColor ([KDGanttViewItem::Type](#) *type*, const QColor & *color*, bool *overwriteExisting* = true)

Sets the default color for a particular type of Gantt item that is used for the item if no specific start, middle, or end colors are set.

Parameters:

type the type of Gantt items for which to query the highlight colors

color the default color to use

overwriteExisting if true, existing settings for individual items are overwritten

See also:

[defaultColor\(\)](#), [setColors\(\)](#), [colors\(\)](#)

4.2.4.102 void KDGanttView::setDefaultHighlightColor ([KDGanttViewItem::Type](#) *type*, const QColor & *color*, bool *overwriteExisting* = true)

Sets the default highlighting color for a particular type of Gantt item that is used for the item if no specific start, middle, or end colors are set.

Parameters:

type the type of Gantt items for which to query the highlight colors

color the default highlighting color to use

overwriteExisting if true, existing color settings in individual items are overwritten

See also:

[defaultHighlightColor\(\)](#), [setHighlightColors\(\)](#), [highlightColors\(\)](#)

4.2.4.103 void KDGanttView::setDisplayEmptyTasksAsLine (bool *show*)

This method specifies whether tasks where the start time and the end time are the same are displayed as a line over the full height of the Gantt view.

Parameters:

show if true, tasks with starttime == endtime are displayed as a line

4.2.4.104 void KDGanttView::setDisplaySubitemsAsGroup (bool *show*)

This method specifies whether hidden subitems should be displayed. It iterates over all KDGanttViewItems in this Gantt view and sets their [displaySubitemsAsGroup\(\)](#) property. All newly created items will have this setting by default.

Parameters:

show if true, the hidden subitems are displayed in all items of this Gantt view.

See also:

[KDGanttViewItem::setDisplaySubitemsAsGroup\(\)](#), [KDGanttViewItem::displaySubitemsAsGroup\(\)](#)

4.2.4.105 void KDGanttView::setDragDropEnabled (bool *b*)

Combines [setDragEnabled\(\)](#) and [setDropEnabled\(\)](#) in one convenient method.

Parameters:

b true if dragging and dropping are enabled, false if dragging and dropping are disabled

See also:

[setDragEnabled\(\)](#), [setDropEnabled\(\)](#)

4.2.4.106 void KDGanttView::setDragEnabled (bool *b*)

Specifies whether drag operations are allowed in the Gantt view. Recurses over all items contained in the Gantt view and enables or disabled them for dragging.

Parameters:

b true if dragging is enabled, false if dragging is disabled

See also:

[isDragEnabled\(\)](#), [setDropEnabled\(\)](#), [isDropEnabled\(\)](#), [setDragDropEnabled\(\)](#)

4.2.4.107 void KDGanttView::setDropEnabled (bool *b*)

Specifies whether drop operations are allowed in the Gantt view. Recurses over all items contained in the Gantt view and enables or disabled them for dropping.

Parameters:

b true if dragging is enabled, false if dragging is disabled

See also:

[setDropEnabled\(\)](#), [setDragEnabled\(\)](#), [isDragEnabled\(\)](#), [setDragDropEnabled\(\)](#)

4.2.4.108 void KDGanttView::setEditable (bool *editable*) [slot]

Specifies whether the Gantt chart is user-editable.

Parameters:

editable pass true in order to get a user-editable Gantt chart, pass false in order to get a read-only chart

See also:

[editable\(\)](#)

4.2.4.109 void KDGanttView::setEditorEnabled (bool *enable*)

Specifies whether it should be possible to edit the appearance of a Gantt item visually in a dialog by double-clicking the item.

Parameters:

enable pass true in order to enable the visual editor and false in order to turn it off

See also:

[editorEnabled\(\)](#)

4.2.4.110 void KDGanttView::setFont (const QFont & *font*)

Sets the font in the left list view widget and in the right time header widget. The settings of the fonts in the time table widget are not effected.

Parameters:

font the new font of the widget

4.2.4.111 void KDGanttView::setGanttMaximumWidth (int *w*)

Sets the maximum width of the Gantt view part widget in pixels. The largest allowed width is 32767.

Parameters:

w the maximum width

4.2.4.112 void KDGanttView::setGvBackgroundColor (const QColor & *c*)

Sets the background color of the Gantt view.

Parameters:

c the background color of the Gantt view.

See also:

[gvBackgroundColor\(\)](#)

4.2.4.113 void KDGanttView::setGvVScrollBarMode (QScrollView::ScrollBarMode *m*)

Sets the scrollbar mode of the time table. The default is always on. Possible values are always on and always off. It only makes sense to set this to always off if [setLvVScrollBarMode\(\)](#) is set to always on or auto.

Parameters:

m The scrollbar mode.

See also:

[setLvVScrollBarMode\(\)](#)

4.2.4.114 void KDGanttView::setHeaderVisible (bool *visible*)

Specifies whether the listview header should be visible. By default, it is not visible.

Parameters:

visible true to make the header visible, false to make it invisible

4.2.4.115 void KDGanttView::setHighlightColors (KDGanttViewItem::Type *type*, const QColor & *start*, const QColor & *middle*, const QColor & *end*, bool *overwriteExisting* = true)

Sets the highlight colors for a certain type of Gantt item. Not all items use all three highlight colors (e.g., only summary items use the middle highlight color).

This setting overrides any highlight color settings made on individual items. These settings will be taken as initial values of any newly created item of this certain type. See also the description of the [KDGanttViewItem](#) class.

Parameters:

type the type of Gantt items for which to set the highlight colors

start the highlight color to use for the beginning of the item

middle the highlight color to use for the middle of the item

end the highlight color to use for the end of the item

overwriteExisting if true, overwrites existing color settings in the individual items

See also:

[highlightColors\(\)](#), [setDefaultHighlightColor\(\)](#), [defaultHighlightColor\(\)](#)

4.2.4.116 void KDGanttView::setHorBackgroundLines (int *count* = 2, QBrush *brush* = QBrush(QColor (200,200,200), Qt::Dense6Pattern))

Defines the horizontal background lines of the Gantt chart. Call [setHorBackgroundLines\(\)](#) (equivalent to [setHorBackgroundLines\(2, QBrush\(QColor \(240,240,240 \) \) \)](#)) to draw a light grey horizontal background line for every second Gantt item. Call [setHorBackgroundLines\(0\)](#) in order to not show horizontal background lines. You may specify the number of lines and the brush of the lines.

Parameters:

count for *count* >= 2, every *count* line gets a background specified by *brush* for *count* < 2, no background lines are drawn

brush the brush of the lines

4.2.4.117 void KDGanttView::setHorizonEnd (const QDateTime & end)

Sets the end of the horizon of the Gantt chart. If *end* is null, the horizon end is computed automatically.

Parameters:

end the end of the horizon

See also:

[setHorizonEnd\(\)](#)

4.2.4.118 void KDGanttView::setHorizonStart (const QDateTime & start)

Sets the start of the horizon of the Gantt chart. If *start* is null, the horizon start is computed automatically.

Parameters:

start the start of the horizon

See also:

[horizonStart\(\)](#)

4.2.4.119 void KDGanttView::setHourFormat (HourFormat format)

Specifies the format in which to display hours. If no hours are shown, this method has no effect.

Parameters:

format the hour format

See also:

[hourFormat\(\)](#), [setYearFormat\(\)](#), [yearFormat\(\)](#)

4.2.4.120 void KDGanttView::setIntervalBackgroundColor (const QDateTime & start, const QDateTime & end, const QColor & color, Scale mini = KDGanttView::Minute, Scale maxi = KDGanttView::Month)

Sets the background color for a time interval given by *start* and *end*. *start* may be later than *end*. If there is already a background interval with the same *start* and *end* values defined, the values (i.e. const QColor& color, Scale mini, Scale maxi) of this background interval are changed. Change the times of an already defined interval with [changeBackgroundInterval\(\)](#). Delete an already defined interval with [deleteBackgroundInterval\(\)](#).

It can be defined, whether the color should be shown in all scales or only in specific scales. If you want to define the color only for the daily view, specify *mini* and *maxi* as Day. If there is no value for *mini*/*maxi* specified, the color for the columns is shown in all scales.

Parameters:

start start datetime of the time interval

end end datetime of the time interval

color the background color

mini show the color only in scales greater than this

maxi show the color only in scales lesser than this

See also:

[changeBackgroundInterval\(\)](#), [deleteBackgroundInterval\(\)](#), [columnBackgroundColor\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#)

4.2.4.121 void KDGanttView::setLegendHeaderBackgroundColor (const QColor & c)

Sets the background color of the legend header.

Parameters:

c the background color of the legend header

See also:

[legendHeaderBackgroundColor\(\)](#)

4.2.4.122 void KDGanttView::setLegendIsDockwindow (bool show)

Specifies whether the legend should be shown as a dock window or not.

Parameters:

show if true, show legend as a dock window

See also:

[showLegend\(\)](#), [legendIsDockwindow\(\)](#), [legendDockwindow\(\)](#)

4.2.4.123 void KDGanttView::setListViewWidth (int w) [slot]

Sets the width of the list view. Space will be taken from or given to the Gantt view.

Parameters:

w the width of the list view

See also:

[listViewWidth\(\)](#)

4.2.4.124 void KDGanttView::setLvBackgroundColor (const QColor & c)

Sets the background color of the list view.

Parameters:

c the background color of the list view

See also:

[lvBackgroundColor\(\)](#)

4.2.4.125 void KDGanttView::setLvVScrollBarMode (QScrollView::ScrollBarMode *m*)

Sets the scrollbar mode of the listview. The default is always off. Possible values are always on, always off and auto. It only makes sense to set this to always off if [setGvVScrollBarMode\(\)](#) is set to always on.

Parameters:

m the scrollbar mode.

See also:

[setGvVScrollBarMode\(\)](#)

4.2.4.126 void KDGanttView::setMajorScaleCount (int *count*)

Sets the number of ticks in the major scale.

Parameters:

count the number of ticks in the major scale

See also:

[majorScaleCount\(\)](#), [setMinorScaleCount\(\)](#), [minorScaleCount\(\)](#)

4.2.4.127 void KDGanttView::setMaximumScale (Scale *unit*)

Sets the maximum allowed time scale of the lower scale of the header.

Parameters:

unit the unit of the lower scale of the header.

See also:

[scale\(\)](#)

4.2.4.128 void KDGanttView::setMinimumColumnWidth (int *width*)

Sets the minimum width that a column needs to have. If the size of the Gantt chart and the scale would make it necessary to go below this limit otherwise, the chart will automatically be made less exact.

Parameters:

width the minimum column width

See also:

[minimumColumnWidth\(\)](#)

4.2.4.129 void KDGanttView::setMinimumScale (Scale *unit*)

Sets the minimum allowed time scale of the lower scale of the header.

Parameters:

unit the unit of the lower scale of the header.

See also:

[scale\(\)](#)

4.2.4.130 void KDGanttView::setMinorScaleCount (int *count*)

Sets the number of ticks in the minor scale.

Parameters:

count the number of ticks in the minor scale

See also:

[minorScaleCount](#), [setMajorScaleCount](#), [majorScaleCount\(\)](#)

4.2.4.131 void KDGanttView::setNoInformationBrush (const QBrush & *brush*)

Specifies the brush in which the 'showNoInformation' line of items should be drawn.

Parameters:

brush the brush of the 'showNoInformation' lines

See also:

[KDGanttViewItem::showNoInformation\(\)](#), [KDGanttViewItem::setShowNoInformation\(\)](#), [KDGanttView::noInformationBrush\(\)](#)

4.2.4.132 void KDGanttView::setPaletteBackgroundColor (const QColor & *col*)

This method is overridden for internal purposes.

4.2.4.133 void KDGanttView::setRepaintMode ([RepaintMode](#) *mode*)

Specifies whether the content should be repainted after scrolling or not.

Parameters:

mode If No, there is no repainting after scrolling. This is the fastest mode. If Medium, there is extra repainting after releasing the scrollbar. This provides fast scrolling with updated content after scrolling. Recommended, when repaint problems occur. This is the default value after startup. If Always, there is an extra update after every move of the scrollbar. This entails slow scrolling with updated content at all time.

4.2.4.134 void KDGanttView::setScale ([Scale unit](#)) [[slot](#)]

Configures the unit of the lower scale of the header. The higher unit is computed automatically.

Parameters:

unit the unit of the lower scale of the header.

See also:

[scale\(\)](#)

4.2.4.135 void KDGanttView::setSelected ([KDGanttViewItem](#) * *item*, bool *selected*)

Calls to this method are passed through to the underlying *QListView*.

4.2.4.136 void KDGanttView::setShapes (KDGanttViewItem::Type *type*, KDGanttViewItem::Shape *start*, KDGanttViewItem::Shape *middle*, KDGanttViewItem::Shape *end*, bool *overwriteExisting* = true)

Sets the shapes for a certain type of Gantt item. Not all items use all three shapes (e.g., only summary items use the middle shape).

This setting overrides any shape settings made on individual items. These settings will be taken as initial values of any newly created item of this certain type. See also the documentation of the [KDGanttViewItem](#) class.

Parameters:

- type* the type of Gantt items for which to set the shapes
- start* the shape to use for the beginning of the item
- middle* the shape to use for the middle of the item
- end* the shape to use for the end of the item
- overwriteExisting* if true, overwrites existing shape settings in the individual items

See also:

[shapes\(\)](#)

4.2.4.137 void KDGanttView::setShowHeaderPopupMenu (bool *show* = true, bool *showZoom* = true, bool *showScale* = true, bool *showTime* = true, bool *showYear* = true, bool *showGrid* = true, bool *showPrint* = false)

Specifies whether the configure popup menu should be shown on right click on the time header widget. This menu lets the user quickly change the zoom factor, the scale mode (minute, hour, day, week, month, auto), the time format, the year format, the grid format, and printing. The default setting is not to show the popup menu. This functionality must be enabled explicitly by the application developer. You can disable each submenu of the popmenu.

Parameters:

- show* true in order to show the popup menu, false in order not to. The default is true.
- showZoom* show the zoom submenu, default: true
- showScale* show the scale submenu, default: true
- showTime* show the time format submenu, default: true
- showYear* show the year format submenu, default: true
- showGrid* show the grid submenu, default: true
- showPrint* show the print submenu, default: false

4.2.4.138 void KDGanttView::setShowLegend (bool *show*) [slot]

Specifies whether the legend should be shown or not. Besides setting this programmatically, the user can also show/hide the legend by using the button provided for this purpose.

Parameters:

- show* force legend to be shown

See also:

[showLegend\(\)](#)

4.2.4.139 void KDGanttView::setShowLegendButton (bool *show*)

Specifies whether the legend button should be visible. By default, it is visible.

Parameters:

show true to show the legend button, false to hide it

See also:

[showLegendButton\(\)](#)

4.2.4.140 void KDGanttView::setShowListView (bool *show*) [slot]

Specifies whether the listview of the Gantt view should be shown or not.

Parameters:

show pass true in order to show the listview and false in order to hide it.

See also:

[showListView\(\)](#)

4.2.4.141 void KDGanttView::setShowMajorTicks (bool *show*)

Hides/shows the grid for the major ticks of the time header in the gantt view.

Parameters:

show true in order to show ticks, false in order to hide them. If *show* is true, `setShowMinorTicks(false)` is performed automatically to hide the grid of the minor ticks. In order to show now grid, call `setShowMinorTicks(false)` and `setShowMajorTicks(false)`.

See also:

[showMajorTicks\(\)](#), [setShowMinorTicks\(\)](#), [showMinorTicks\(\)](#)

4.2.4.142 void KDGanttView::setShowMinorTicks (bool *show*)

Hides/shows the grid for the minor ticks of the time header in the gantt view.

Parameters:

show true in order to show ticks, false in order to hide them. If *show* is true, `setShowMajorTicks(false)` is performed automatically to hide the grid of the major ticks. In order to show now grid, call `setShowMinorTicks(false)` and `setShowMajorTicks(false)`.

See also:

[showMinorTicks\(\)](#), [setShowMajorTicks\(\)](#), [showMajorTicks\(\)](#)

4.2.4.143 void KDGanttView::setShowTaskLinks (bool *show*) [slot]

Specifies whether task links should be shown.

Parameters:

show true for showing task links, false for not showing them

See also:

[showTaskLinks\(\)](#), [KDGanttViewTaskLink](#)

4.2.4.144 void KDGanttView::setShowTimeTablePopupMenu (bool *show*)

Specifies whether the add item popup menu should be shown on right click on the time table widget. This menu lets the user quickly add new items to the Gantt view (as root, as child or after an item). It also offers cutting and pasting of items.

The default setting is that the popup menu is not shown. It must be enabled by the program.

Parameters:

show true in order to show popup menu, false in order not to

4.2.4.145 void KDGanttView::setTextColor (const QColor & *color*)

Sets the color used for texts in the Gantt chart. Overrides all individual settings of the Gantt items.

Parameters:

color the text color to use

See also:

[textColor\(\)](#)

4.2.4.146 void KDGanttView::setTimeHeaderBackgroundColor (const QColor & *c*)

Sets the background color of the time header.

Parameters:

c the background color of the time header.

See also:

[timeHeaderBackgroundColor\(\)](#)

4.2.4.147 void KDGanttView::setTimelineToEnd () [slot]

Sets the timeline to the horizon end.

4.2.4.148 void KDGanttView::setTimelineToStart () [slot]

Sets the timeline to the horizon start.

4.2.4.149 void KDGanttView::setUpdateEnabled (bool *enable*)

Enables or disables updating of the content of the Gantt view. To avoid flickering in the Gantt view while inserting large amounts of Gantt items, you should call

```
bool upd = KDGanttView::getUpdateEnabled(); KDGanttView::setUpdateEnabled( false ); ... insert items here ... KDGanttView::setUpdateEnabled( upd );
```

With this code, you avoid unwanted side effects with other parts in your code, where you disable (and re-enable) the update.

When calling `setUpdateEnabled(true)`, all the content is recomputed, resized, and updated.

Before calling `show()` for the first time, updating is disabled. When calling `show()`, updating is automatically enabled.

Parameters:

enable if true, the content of the Gantt view is updated after every insertion of a new item.

See also:

[getUpdateEnabled\(\)](#)

4.2.4.150 void KDGanttView::setWeekdayBackgroundColor (const QColor & *color*, int *weekday*)

Specifies the background color for weekday days. If no individual days are visible on the Gantt chart, this method has no visible effect. The days are specified as an intervals of integer values where 1 means Monday and 7 means Sunday.

Parameters:

color the background color to use for weekend days.

weekday the day of the week (Monday = 1, Sunday = 7)

See also:

[weekendBackgroundColor\(\)](#), [setWeekendDays\(\)](#), [weekendDays\(\)](#)

4.2.4.151 void KDGanttView::setWeekendBackgroundColor (const QColor & *color*)

Specifies the background color for weekend days. If no individual days are visible on the Gantt chart, this method has no visible effect.

Parameters:

color the background color to use for weekend days.

See also:

[weekendBackgroundColor\(\)](#), [setWeekendDays\(\)](#), [weekendDays\(\)](#)

4.2.4.152 void KDGanttView::setWeekendDays (int *start*, int *end*)

Defines which days are considered weekends. The days are specified as an interval of integer values where 1 means Monday and 7 means Sunday. In order to define a weekend from Sunday to Monday, specify (7,1).

Parameters:

start the first day of the weekend

end the last day of the weekend

See also:

[weekendDays\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#)

4.2.4.153 void KDGanttView::setYearFormat (YearFormat format)

Specifies the format in which to display years. If no years are shown, this method has no effect.

Parameters:

format the year format

See also:

[yearFormat\(\)](#), [setHourFormat\(\)](#), [hourFormat\(\)](#)

4.2.4.154 void KDGanttView::setZoomFactor (double factor, bool absolute)

Zooms into the Gantt chart. Values less than 1 mean zooming in, values greater than 1 mean zooming out. A zooming factor of exactly 1.0 means original size.

Parameters:

factor the zoom factor

absolute if true, factor is interpreted absolutely, if false, factor is interpreted relatively to the current zoom factor

See also:

[zoomToFit\(\)](#)
[zoomToSelection\(\)](#)
[zoomFactor\(\)](#)

4.2.4.155 bool KDGanttView::shapes (KDGanttViewItem::Type type, KDGanttViewItem::Shape & start, KDGanttViewItem::Shape & middle, KDGanttViewItem::Shape & end) const

Queries the shapes for a particular type of Gantt item.

Parameters:

type the type of Gantt items for which to query the shapes

start the start shape is returned in this parameter

middle the middle shape is returned in this parameter

end the end shape is returned in this parameter

Returns:

true if there was a general shape set for the specified type. If the return value is false, the values of the three shape parameters are undefined.

See also:

[setShapes\(\)](#)

4.2.4.156 void KDGanttView::show () [virtual]

Updates the content of the GanttView and shows it. Automatically sets `setUpdateEnabled(true)`.

See also:

[setUpdateEnabled\(\)](#)

4.2.4.157 bool KDGanttView::showHeaderPopupMenu () const

Returns whether the configure popup menu should be shown on right click on the time header widget.

Returns:

true if the popup menu should be shown

4.2.4.158 bool KDGanttView::showLegend () const

Returns whether the legend is currently shown. The visibility of the legend can be changed both by [setShowLegend\(\)](#), and interactively by the user.

Returns:

true if the legend is currently visible

See also:

[setShowLegend\(\)](#)

4.2.4.159 bool KDGanttView::showLegendButton () const

Returns whether the legend button is visible.

Returns:

whether the legend button is visible

See also:

[setShowLegendButton\(\)](#)

4.2.4.160 bool KDGanttView::showListView () const

Returns whether the listview of the Gantt view is shown or not.

Returns:

true if the listview is shown

See also:

[setShowListView\(\)](#)

4.2.4.161 bool KDGanttView::showMajorTicks () const

Returns whether the grid is shown on the major scale.

Returns:

true if ticks are shown on the major scale

See also:

[setShowMajorTicks\(\)](#), [setShowMinorTicks\(\)](#), [showMinorTicks\(\)](#)

4.2.4.162 bool KDGanttView::showMinorTicks () const

Returns whether ticks are shown on the minor scale.

Returns:

true if ticks are shown on the minor scale

See also:

[setShowMinorTicks\(\)](#), [setShowMajorTicks\(\)](#), [showMajorTicks\(\)](#)

4.2.4.163 bool KDGanttView::showTaskLinks () const

Returns whether task links should be shown.

Returns:

true if task links are shown, false otherwise

See also:

[setShowTaskLinks\(\)](#), [KDGanttViewTaskLink](#)

4.2.4.164 bool KDGanttView::showTimeTablePopupMenu () const

Returns whether the add item popup menu should be shown on right click on the time table widget.

Returns:

true if the popup menu should be shown

4.2.4.165 QSize KDGanttView::sizeHint ()

Returns a useful size for the view. Returned width: [sizeHint\(\).width\(\)](#) of the list view + width of TimeTable
Returned height: height() of TimeHeader + height() of TimeTable + height() of Legend (if shown)

4.2.4.166 void KDGanttView::startDrag () [protected, virtual]

Implements a pass-through to the list view.

4.2.4.167 void KDGanttView::taskLinkDoubleClicked (KDGanttViewTaskLink *) [signal]

This signal is emitted when the user double-clicks a task link.

4.2.4.168 QPtrList< KDGanttViewTaskLinkGroup > KDGanttView::taskLinkGroups () const

Returns the list of task link groups in the Gantt view.

Returns:

the list of task link groups in the Gantt view

4.2.4.169 void KDGanttView::taskLinkLeftClicked (KDGanttViewTaskLink *) [signal]

This signal is emitted when the user clicks on a task link with the left mouse button.

4.2.4.170 void KDGanttView::taskLinkMidClicked (KDGanttViewTaskLink *) [signal]

This signal is emitted when the user clicks on a task link with the middle mouse button.

4.2.4.171 void KDGanttView::taskLinkRightClicked (KDGanttViewTaskLink *) [signal]

This signal is emitted when the user clicks on a task link with the right mouse button.

4.2.4.172 QPtrList< KDGanttViewTaskLink > KDGanttView::taskLinks () const

Returns the list of task links in the Gantt view.

Returns:

the list of task links in the Gantt view

4.2.4.173 QColor KDGanttView::textColor () const

Returns the color used for texts in the Gantt chart.

Returns:

the color used for texts in the Gantt chart.

See also:

[setTextColor\(\)](#)

4.2.4.174 QColor KDGanttView::timeHeaderBackgroundColor () const

Returns the background color of the time header.

Returns:

the background color of the time header

See also:

[setTimeHeaderBackgroundColor\(\)](#)

4.2.4.175 void `KDGanttView::timeIntervallSelected` (const `QDateTime` & *start*, const `QDateTime` & *end*) [`signal`]

Deprecated

This signal is deprecated, do not use it in new code; use `timeIntervalSelected()` instead. `timeIntervallSelected()` will be removed in future versions.

4.2.4.176 void `KDGanttView::timeIntervalSelected` (const `QDateTime` & *start*, const `QDateTime` & *end*) [`signal`]

This signal is emitted when the user selects a time interval with the mouse on the time header connect this signal to the slot `zoomToSelection(const QDateTime& start, const QDateTime& end)` to obtain automatic zooming.

4.2.4.177 `QColor` `KDGanttView::weekdayBackgroundColor` (int *weekday*) const

Returns the background color for weekday days.

Parameters:

weekday the day of the week (Monday = 1, Sunday = 7)

Returns:

the background color for weekend days

See also:

[setWeekendBackgroundColor\(\)](#), [setWeekendDays\(\)](#), [weekendDays\(\)](#)

4.2.4.178 `QColor` `KDGanttView::weekendBackgroundColor` () const

Returns the background color for weekend days.

Returns:

the background color for weekend days

See also:

[setWeekendBackgroundColor\(\)](#), [setWeekendDays\(\)](#), [weekendDays\(\)](#)

4.2.4.179 void `KDGanttView::weekendDays` (int & *start*, int & *end*) const

Returns which days are considered weekends.

Parameters:

start in this parameter, the first day of the weekend is returned

end in this parameter, the end day of the weekend is returned

See also:

[setWeekendDays\(\)](#), [setWeekendBackgroundColor\(\)](#), [weekendBackgroundColor\(\)](#)

4.2.4.180 KDGanttView::YearFormat KDGanttView::yearFormat () const

Returns the format in which to display years.

Returns:

the year format

See also:

[setYearFormat\(\)](#), [setHourFormat\(\)](#), [hourFormat\(\)](#)

4.2.4.181 double KDGanttView::zoomFactor () const

Returns the current zoom factor.

Returns:

the current zoom factor

See also:

[zoomToFit\(\)](#), [zoomToSelection\(\)](#), [setZoomFactor\(\)](#)

4.2.4.182 void KDGanttView::zoomToFit () [slot]

Zooms such that the Gantt chart is less than the available space of the widget.

See also:

[setZoomFactor\(\)](#)
[zoomFactor\(\)](#)
[zoomToSelection\(\)](#)

4.2.4.183 void KDGanttView::zoomToSelection (const QDateTime & start, const QDateTime & end) [slot]

Zooms so that at least the selected time period is visible after the zoom.

Parameters:

start the new font of the widget

end the new font of the widget

See also:

[setZoomFactor\(\)](#)
[zoomFactor\(\)](#)
[zoomToFit\(\)](#)

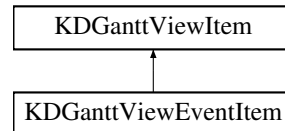
The documentation for this class was generated from the following files:

- KDGanttView.h
- KDGanttView.cpp

4.3 KDGanttViewItem Class Reference

```
#include <KDGanttViewItem.h>
```

Inheritance diagram for KDGanttViewItem::



Public Member Functions

- [KDGanttViewItemEventItem](#) ([KDGanttView](#) *view, const QString &lvtext=QString::null, const QString &name=QString::null)
- [KDGanttViewItemEventItem](#) ([KDGanttViewItem](#) *parent, const QString &lvtext=QString::null, const QString &name=QString::null)
- [KDGanttViewItemEventItem](#) ([KDGanttView](#) *view, [KDGanttViewItem](#) *after, const QString &lvtext=QString::null, const QString &name=QString::null)
- [KDGanttViewItemEventItem](#) ([KDGanttViewItem](#) *parent, [KDGanttViewItem](#) *after, const QString &lvtext=QString::null, const QString &name=QString::null)
- virtual [~KDGanttViewItemEventItem](#) ()
- void [setLeadTime](#) (const QDateTime &leadTimeStart)
- void [setStartTime](#) (const QDateTime &start)
- QDateTime [leadTime](#) () const

4.3.1 Detailed Description

An event item in a Gantt chart.

This class represents event items in Gantt charts.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 KDGanttViewItemEventItem::KDGanttViewItemEventItem ([KDGanttView](#) * view, const QString & lvtext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

view the Gantt view to insert this item into

lvtext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.3.2.2 KDGanttViewItem::KDGanttViewItem (KDGanttViewItem * parent, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

parent a parent item under which this one goes

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.3.2.3 KDGanttViewItem::KDGanttViewItem (KDGanttView * view, KDGanttViewItem * after, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

view the Gantt view to insert this item into

after another item at the same level behind which this one should go

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.3.2.4 KDGanttViewItem::KDGanttViewItem (KDGanttViewItem * parent, KDGanttViewItem * after, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

parent a parent item under which this one goes

after another item at the same level behind which this one should go

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.3.2.5 KDGanttViewItem::~~KDGanttViewItem () [virtual]

The destructor. Delete the datetimes, if created.

4.3.3 Member Function Documentation

4.3.3.1 QDateTime KDGanttViewItem::leadTime () const

Returns whether the event item is shown with a lead time line and if yes, when the lead time starts.

Returns:

if the event item is shown with a lead time line, returns the QDateTime object representing the start of the lead time, otherwise returns an invalid QDateTime object

See also:

[setLeadTime\(\)](#)

4.3.3.2 void KDGanttViewItem::setLeadTime (const QDateTime & leadTimeStart)

Specifies whether the event item should be shown with a lead time line, and if yes, when the lead time starts. If the start time is less than the lead time, the start time is set to this lead time automatically.

Parameters:

leadTimeStart the start time of the lead time; pass an invalid QDateTime object in order to turn the lead time off.

See also:

[leadTime\(\)](#)

4.3.3.3 void KDGanttViewItem::setStartTime (const QDateTime & start) [virtual]

Specifies the start time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. If the start time is less than the lead time, the lead time is set to this start time automatically.

Parameters:

start the start time

See also:

[startTime\(\)](#)

Reimplemented from [KDGanttViewItem](#).

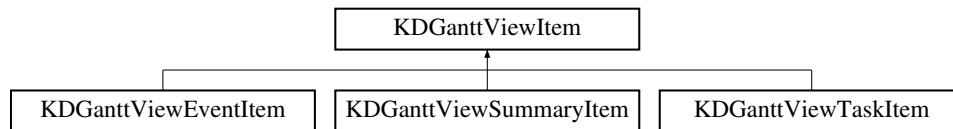
The documentation for this class was generated from the following files:

- KDGanttViewItem.h
- KDGanttViewItem.cpp

4.4 KDGanttViewItem Class Reference

```
#include <KDGanttViewItem.h>
```

Inheritance diagram for KDGanttViewItem::



Public Types

- enum [Type](#) { **Event**, **Task**, **Summary** }
- enum [Shape](#) {
TriangleDown, **TriangleUp**, **Diamond**, **Square**,
Circle }

Public Member Functions

- virtual [~KDGanttViewItem](#) ()
- [Type](#) [type](#) () const
- void [setEnabled](#) (bool on)
- bool [enabled](#) () const
- virtual void [setOpen](#) (bool o)
- void [setItemVisible](#) (bool on)
- bool [itemVisible](#) () const
- void [setEditable](#) (bool editable)
- bool [editable](#) () const
- void [setShowNoInformation](#) (bool show)
- bool [showNoInformation](#) ()
- void [setDisplaySubitemsAsGroup](#) (bool show)
- bool [displaySubitemsAsGroup](#) () const
- void [setPriority](#) (int prio)
- int [priority](#) ()
- virtual void [setStartTime](#) (const QDateTime &start)
- QDateTime [startTime](#) () const
- virtual void [setEndTime](#) (const QDateTime &end)
- QDateTime [endTime](#) () const
- void [setText](#) (const QString &text)
- QString [text](#) () const
- void [setListViewText](#) (const QString &text, int column=0)
- void [setListViewText](#) (int column, const QString &text)
- QString [listViewText](#) (int column=0) const
- void [setFont](#) (const QFont &font)
- QFont [font](#) () const
- void [setTooltipText](#) (const QString &text)
- QString [tooltipText](#) () const

- void [setWhatsThisText](#) (const QString &text)
- QString [whatsThisText](#) () const
- void [setPixmap](#) (int column, const QPixmap &pixmap)
- void [setPixmap](#) (const QPixmap &pixmap)
- const QPixmap * [pixmap](#) (int column=0) const
- void [setHighlight](#) (bool)
- bool [highlight](#) () const
- bool [subitemIsCalendar](#) () const
- void [setShapes](#) (Shape start, Shape middle, Shape end)
- void [shapes](#) (Shape &start, Shape &middle, Shape &end) const
- void [setDefaultColor](#) (const QColor &)
- QColor [defaultColor](#) () const
- void [setColors](#) (const QColor &start, const QColor &middle, const QColor &end)
- void [colors](#) (QColor &start, QColor &middle, QColor &end) const
- void [setDefaultHighlightColor](#) (const QColor &)
- QColor [defaultHighlightColor](#) () const
- void [setHighlightColors](#) (const QColor &start, const QColor &middle, const QColor &end)
- void [highlightColors](#) (QColor &start, QColor &middle, QColor &end) const
- void [setTextColor](#) (const QColor &color)
- QColor [textColor](#) () const
- KDGanttViewItem * [firstChild](#) () const
- KDGanttViewItem * [nextSibling](#) () const
- KDGanttViewItem * [parent](#) () const
- KDGanttViewItem * [itemAbove](#) ()
- KDGanttViewItem * [itemBelow](#) (bool includeDisabled=true)
- KDGanttViewItem * [getChildByName](#) (const QString &name)
- void [createNode](#) (QDomDocument &doc, QDomElement &parentElement)
- QString [name](#) () const

Static Public Member Functions

- KDGanttViewItem * [createFromDomElement](#) (KDGanttView *view, QDomElement &element)
- KDGanttViewItem * [createFromDomElement](#) (KDGanttView *view, KDGanttViewItem *previous, QDomElement &element)
- KDGanttViewItem * [createFromDomElement](#) (KDGanttViewItem *parent, QDomElement &element)
- KDGanttViewItem * [createFromDomElement](#) (KDGanttViewItem *parent, KDGanttViewItem *previous, QDomElement &element)
- KDGanttViewItem * [find](#) (const QString &name)

Protected Member Functions

- KDGanttViewItem (Type type, KDGanttView *view, const QString &lvtext=QString::null, const QString &name=QString::null)
- KDGanttViewItem (Type type, KDGanttViewItem *parent, const QString &lvtext=QString::null, const QString &name=QString::null)
- KDGanttViewItem (Type type, KDGanttView *view, KDGanttViewItem *after, const QString &lvtext=QString::null, const QString &name=QString::null)
- KDGanttViewItem (Type type, KDGanttViewItem *parent, KDGanttViewItem *after, const QString &lvtext=QString::null, const QString &name=QString::null)

- void [updateCanvasItems](#) ()
- int [getCoordY](#) ()
- QDateTime [myChildStartTime](#) ()
- QDateTime [myChildEndTime](#) ()
- QCanvasText * [textcanvas](#) ()
- void [generateAndInsertName](#) (const QString &name)

Protected Attributes

- bool [isVisibleInGanttView](#)
- KDCanvasLine * [startLine](#)
- KDCanvasLine * [endLine](#)
- KDCanvasLine * [startLineBack](#)
- KDCanvasLine * [endLineBack](#)
- KDCanvasLine * [actualEnd](#)
- KDCanvasPolygonItem * [startShape](#)
- KDCanvasPolygonItem * [midShape](#)
- KDCanvasPolygonItem * [endShape](#)
- KDCanvasPolygonItem * [startShapeBack](#)
- KDCanvasPolygonItem * [midShapeBack](#)
- KDCanvasPolygonItem * [endShapeBack](#)
- KDGanttView * [myGanttView](#)
- KDCanvasText * [mTextCanvas](#)
- QString [textCanvasText](#)
- QDateTime [myStartTime](#)
- QDateTime [myEndTime](#)
- bool [isHighlighted](#)
- bool [isEditable](#)
- int [myItemSize](#)
- bool [blockUpdating](#)

4.4.1 Detailed Description

This class represents an item in a Gantt chart.

This class is an abstract base class, it cannot be instantiated directly. Instead, you should create items of one of the subclasses. This class provides methods common to all Gantt items.

The initialization of the shapes/colors of the item works as follows:

Shapes: When a new item is created, the shapes are set to the default values for items of the type of this item, defined in the [KDGanttView](#) class with `void setShapes(KDGanttViewItem::Type type, KDGanttViewItem::Shape start, KDGanttViewItem::Shape middle, KDGanttViewItem::Shape end);` If there is no default value defined for this type, the shapes are set as follows: For TaskViewItems all three shapes are set to Square. For SummaryViewItems all three shapes are set to TriangleDown. For EventViewItems all three shapes are set to Diamond.

Colors: When a new item is created, the colors are set to the default values for items of the type of this item, defined in the [KDGanttView](#) class with `void setColors(KDGanttViewItem::Type type, const QColor& start, const QColor& middle, const QColor& end);` If there is no default value defined for this type, the colors of the shapes are set to the default color for items of this type, defined in the [KDGanttView](#) class with: `void setDefaultColor(KDGanttViewItem::Type type, const QColor&);` The initial default color

in the [KDGanttView](#) class is set to blue for `KDGanttViewItem::Event`, green for `KDGanttViewItem::Task`, cyan for `KDGanttViewItem::Summary`.

Highlight Colors: When a new item is created, the highlight colors are set to the default values for items of the type of this item, defined in the [KDGanttView](#) class with: `void setHighlightColors(KDGanttViewItem::Type type, const QColor& start, const QColor& middle, const QColor& end);` If there is no default value defined for this type, the highlight colors of the shapes are set to the default color for items of this type, defined in the [KDGanttView](#) class with: `void setDefaultHighlightColor(KDGanttViewItem::Type type, const QColor&);` The initial default highlight color in the [KDGanttView](#) class is set to red for all types.

Start/End time: When a new item is created, the start time and the end time is set automatically. The time, which is currently displayed in the middle of the Gantt View, is set as start/end time. At startup of a newly created Gantt view, this is the current time.

The Priority: The priority is set with `setPriority()`. The priority determines which items are painted over which other items. The item with the highest priority is painted on top of all others. The priority for an item can be between 1 and 199. A priority less than 100 means that the item is painted below the grid in the Gantt chart. For Task items, the default priority is 50, for all other items it is 150. This feature only makes sense for an item which is a child of another item, for which `displaySubitemsAsGroup()` property is set to true.

The Display Mode: The display mode is set with `setDisplaySubitemsAsGroup()`. In the normal view mode (set with `setDisplaySubitemsAsGroup(false);`), an item is displayed in the same manner, when its child items are shown or not. In the other mode (set with `setDisplaySubitemsAsGroup(true);`), called "calendar mode", the item is displayed as follows: If the item has no children, it is displayed as usual. If the item is opened (i.e., its children are displayed), the start/end time of this item is computed automatically according to the earliest start time/latest end time of its children. The item and its children are displayed as usual. If the item is closed (i.e., its children are hidden in the left list view), the item itself is hidden, and its children are displayed on the timeline of this item instead. To control the painting of overlapping children, call `setPriority()` for the childs.

Blocking of user interaction to open item: If you want to block users to open items used as parents of calendar items, call `KDGanttView::setCalendarMode(true);`

Example 1, Color: If you create an instance of a [KDGanttView](#) class and add a `SummaryViewItem` without setting any color/shape values, you get an item with three shapes of the form `TriangleDown` in the color magenta. If the item is highlighted, the color will change to the highlight color red.

Example 2, Calendar View: To use a Gantt view as a calendar view, call `KDGanttView::setCalendarMode(true);`; `KDGanttView::setDisplaySubitemsAsGroup(true);`; Insert root items in the Gantt view. Insert items as children of these root item in the Gantt view. You may use any item type as parent and child; there are no limitations. It is, however, recommended to use `KDGanttViewTaskItems` Actually, you may add child items to the children themselves. Such a child behaves then like a parent. Now set the start/end time of the children to specify a time interval for these items.

4.4.2 Member Enumeration Documentation

4.4.2.1 enum [KDGanttViewItem::Shape](#)

This enum is used in order to specify the shapes of a Gantt chart item.

4.4.2.2 enum [KDGanttViewItem::Type](#)

This enum is used in order to return the type of a Gantt chart item.

4.4.3 Constructor & Destructor Documentation

4.4.3.1 KDGanttViewItem::KDGanttViewItem (Type *type*, KDGanttView * *view*, const QString & *lvt* = QString::null, const QString & *name* = QString::null) [protected]

Constructs an empty Gantt item.

Parameters:

type the type of the item to insert

view the Gantt view to insert this item into

lvt the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.4.3.2 KDGanttViewItem::KDGanttViewItem (Type *type*, KDGanttViewItem * *parentItem*, const QString & *lvt* = QString::null, const QString & *name* = QString::null) [protected]

Constructs an empty Gantt item.

Parameters:

type the type of the item to insert

parentItem a parent item under which this one goes

lvt the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.4.3.3 KDGanttViewItem::KDGanttViewItem (Type *type*, KDGanttView * *view*, KDGanttViewItem * *after*, const QString & *lvt* = QString::null, const QString & *name* = QString::null) [protected]

Constructs an empty Gantt item.

Parameters:

type the type of the item to insert

view the Gantt view to insert this item into

after another item at the same level behind which this one should go

lvt the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.4.3.4 KDGanttViewItem::KDGanttViewItem (Type *type*, KDGanttViewItem * *parentItem*, KDGanttViewItem * *after*, const QString & *lvt* = QString::null, const QString & *name* = QString::null) [protected]

Constructs an empty Gantt item.

Parameters:

type the type of the item to insert

parentItem a parent item under which this one goes

after another item at the same level behind which this one should go

lvtext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.4.3.5 `KDGanttViewItem::~~KDGanttViewItem ()` [virtual]

Destroys the object and frees any allocated resources.

4.4.4 Member Function Documentation

4.4.4.1 `void KDGanttViewItem::colors (QColor & start, QColor & middle, QColor & end) const`

Returns the colors used for this item

Parameters:

start returns the start color

middle returns the middle color

end returns the end color

See also:

[setColor\(\)](#), [setShapes\(\)](#), [shapes\(\)](#), [setDefaultColor\(\)](#), [defaultColor\(\)](#)

4.4.4.2 `KDGanttViewItem * KDGanttViewItem::createFromDomElement (KDGanttViewItem * parent, KDGanttViewItem * previous, QDomElement & element)` [static]

Creates a KDGanttViewItem according to the specification in a DOM element.

Parameters:

parent the parent item under which the item will be inserted

previous to item behind this one should appear

element the DOM element from which to read the specification

Returns:

the newly created element

4.4.4.3 `KDGanttViewItem * KDGanttViewItem::createFromDomElement (KDGanttViewItem * parent, QDomElement & element)` [static]

Creates a KDGanttViewItem according to the specification in a DOM element.

Parameters:

parent the parent item under which the item will be inserted

element the DOM element from which to read the specification

Returns:

the newly created element

4.4.4.4 `KDGanttViewItem * KDGanttViewItem::createFromDomElement (KDGanttView * view, KDGanttViewItem * previous, QDomElement & element) [static]`

Creates a KDGanttViewItem according to the specification in a DOM element.

Parameters:

- view* the view in which the item will be inserted
- previous* to item behind this one should appear
- element* the DOM element from which to read the specification

Returns:

the newly created element

4.4.4.5 `KDGanttViewItem * KDGanttViewItem::createFromDomElement (KDGanttView * view, QDomElement & element) [static]`

Creates a KDGanttViewItem according to the specification in a DOM element.

Parameters:

- view* the view in which the item will be inserted
- element* the DOM element from which to read the specification

Returns:

the newly created item

4.4.4.6 `void KDGanttViewItem::createNode (QDomDocument & doc, QDomElement & parentElement)`

Creates a DOM node that describes this item.

Parameters:

- doc* the DOM document to which the node belongs
- parentElement* the element into which to insert this node

4.4.4.7 `QColor KDGanttViewItem::defaultColor () const`

Returns the default color that is used for the item if no specific start, middle, or end colors are set.

Returns:

color the default color used

See also:

[setDefaultColor\(\)](#), [setColors\(\)](#), [colors\(\)](#)

4.4.4.8 QColor KDGanttViewItem::defaultHighlightColor () const

Returns the default highlighting color that is used for the item if no specific start, middle, or end colors are set.

Returns:

color the default highlighting color used

See also:

[setDefaultHighlightColor\(\)](#), [setHighlightColors\(\)](#), [highlightColors\(\)](#)

4.4.4.9 bool KDGanttViewItem::displaySubitemsAsGroup () const

Returns whether this item displays hidden subitems. Initial set to false.

Returns:

true if this item displays hidden subitems, false otherwise

See also:

[setDisplaySubitemsAsGroup\(\)](#)

4.4.4.10 bool KDGanttViewItem::editable () const

Returns whether this item is editable.

Returns:

true if this item is editable, false otherwise

See also:

[setEditable\(\)](#), [KDGanttView::setEditable\(\)](#), [KDGanttView::editable\(\)](#)

4.4.4.11 bool KDGanttViewItem::enabled () const

Returns whether this item is enabled.

Returns:

true if this item is enabled, false otherwise

See also:

[setEnabled\(\)](#)

4.4.4.12 QDateTime KDGanttViewItem::endTime () const

Returns the end time of this item.

Returns:

the end time of this item

See also:

[setEndTime\(\)](#), [setStartTime\(\)](#), [startTime\(\)](#)

4.4.4.13 [KDGanttViewItem](#) * [KDGanttViewItem::find](#) (const QString & *name*) [static]

Returns the item with the specified name.

Parameters:

name the name to search for

Returns:

the item with the specified name; 0 if no group with that name exists

4.4.4.14 [KDGanttViewItem](#) * [KDGanttViewItem::firstChild](#) () const

Returns the first child of this item.

Returns:

the first child of this item, 0 if this item has no children

4.4.4.15 [QFont](#) [KDGanttViewItem::font](#) () const

Returns the font used for the text in this item.

Returns:

the font used for the text in this item

See also:

[setFont\(\)](#)

4.4.4.16 void [KDGanttViewItem::generateAndInsertName](#) (const QString & *name*)
[protected]

Generates a unique name if necessary and inserts it into the item dictionary.

4.4.4.17 [KDGanttViewItem](#) * [KDGanttViewItem::getChildByName](#) (const QString & *name*)

If the name of this item is *name* (i.e., [listViewText\(\)](#) == *name*), the pointer to this item is returned. Otherwise, it looks for an item with name *name* in the set of children and subchildren of this item.

Parameters:

name the name of the item

Returns:

the pointer to the item with name *name*

4.4.4.18 int [KDGanttViewItem::getCoordY](#) () [protected]

Returns the y coordinate of this item.

Returns:

the y coordinate of this item

4.4.4.19 `bool KDGanttViewItem::highlight () const`

Returns whether this item is highlighted, either programmatically with `setHighlight()` or by the user with the mouse.

Returns:

true if the item is highlighted

See also:

[setHighlight\(\)](#)

4.4.4.20 `void KDGanttViewItem::highlightColors (QColor & start, QColor & middle, QColor & end) const`

Returns the highlight colors used for this item

Parameters:

start returns the start highlight color

middle returns the middle highlight color

end returns the end highlight color

See also:

[setHighlightColors\(\)](#), [setShapes\(\)](#), [shapes\(\)](#)

4.4.4.21 `KDGanttViewItem * KDGanttViewItem::itemAbove ()`

Returns the item above this item in the listview

Returns:

the item above this item, 0 if this is the first item

4.4.4.22 `KDGanttViewItem * KDGanttViewItem::itemBelow (bool includeDisabled = true)`

Returns the item below this item in the listview. It can be specified whether the disabled items are taken into account as well.

Parameters:

includeDisabled if true, disabled items are considered as well

Returns:

the item below this item, 0 if this is the last item

4.4.4.23 `bool KDGanttViewItem::itemVisible () const`

Returns whether this item is visible.

Returns:

true if this item is visible, false otherwise

See also:

[setItemVisible\(\)](#)

4.4.4.24 QString KDGanttViewItem::listViewText (int *column* = 0) const

Returns the text to be shown in this item in the list view.

Parameters:

column the column in which the text will be shown

Returns:

the text to be shown in this item

See also:

[setText\(\)](#), [setTextColor\(\)](#), [textColor\(\)](#), [text\(\)](#), [setListViewText\(\)](#)

4.4.4.25 QDateTime KDGanttViewItem::myChildEndTime () [protected]

Returns the end time of the children of this item.

Returns:

the end time of the children of this item

4.4.4.26 QDateTime KDGanttViewItem::myChildStartTime () [protected]

Returns the start time of the children of this item.

Returns:

the start time of the children of this item

4.4.4.27 QString KDGanttViewItem::name () const

Returns the unique name that can identify the item.

Returns:

the unique item name

4.4.4.28 KDGanttViewItem * KDGanttViewItem::nextSibling () const

Returns the next sibling item of this item

Returns:

the next sibling item of this item, 0 if this item has no more siblings

4.4.4.29 KDGanttViewItem * KDGanttViewItem::parent () const

Returns the parent item of this item

Returns:

the parent item of this item, 0 if this item is a top-level item

4.4.4.30 `const QPixmap * KDGanttViewItem::pixmap (int column = 0) const`

Returns a pixmap that is shown in the listview.

Parameters:

column the column for which to query the pixmap

Returns:

a pointer to the pixmap shown

See also:

[setPixmap\(\)](#)

4.4.4.31 `int KDGanttViewItem::priority ()`

Returns the priority of this item.

Returns:

the priority of this item

See also:

[setDisplaySubitemsAsGroup\(\)](#)

4.4.4.32 `void KDGanttViewItem::setColors (const QColor & start, const QColor & middle, const QColor & end)`

Specifies the colors in which to draw the shapes of this item.

It is advisable not to use this method, but rather set the colors for all items of a type with [KDGanttView::setColors\(\)](#) in order to get a uniform Gantt view.

Parameters:

start the color for the start shape

middle the color for the middle shape

end the color for the end shape

See also:

[colors\(\)](#), [setShapes\(\)](#), [shapes\(\)](#), [setDefaultColor\(\)](#), [defaultColor\(\)](#)

4.4.4.33 `void KDGanttViewItem::setDefaultColor (const QColor & color)`

Sets the default color that is used for the item if no specific start, middle, or end colors are set.

It is advisable not to use this method, but rather set the colors for all items of a type with [KDGanttView::setDefaultColor\(\)](#) in order to get a uniform Gantt view.

If the item displays its subitems (children) as a group, ([displaySubitemsAsGroup\(\)](#) == true) all changes apply to all subitems as well.

Parameters:

color the default color to use

See also:

[defaultColor\(\)](#), [setColors\(\)](#), [colors\(\)](#)

4.4.4.34 void KDGanttViewItem::setDefaultHighlightColor (const QColor & *color*)

Sets the default highlighting color that is used for the item if no specific start, middle, or end colors are set.

It is advisable not to use this method, but rather set the colors for all items of a type with [KDGanttView::setDefaultHighlightColor\(\)](#) in order to get a uniform Gantt view.

If the item displays its subitems (children) as a group, ([displaySubitemsAsGroup\(\)](#) == true) all changes apply to all subitems as well.

Parameters:

color the default highlighting color to use

See also:

[defaultHighlightColor\(\)](#), [setHighlightColors\(\)](#), [highlightColors\(\)](#)

4.4.4.35 void KDGanttViewItem::setDisplaySubitemsAsGroup (bool *show*)

Specifies whether this item shows hidden subitems on its timeline. Useful to get a so called "calendar view" with many items in one row. When [displaySubitemsAsGroup\(\)](#) is set to true, this item has a normal view, when it is expanded. If it is not expanded (and has at least one child), the item itself is hidden, and all children are displayed instead. To manage the painting priority of the childs (if overlapping), you may set [priority\(\)](#) of these items.

Parameters:

show pass true to make this item displaying hidden subitems

See also:

[editable\(\)](#), [KDGanttView::setEditable\(\)](#), [KDGanttView::editable\(\)](#), [setPriority\(\)](#)

4.4.4.36 void KDGanttViewItem::setEditable (bool *editable*)

Specifies whether this item is editable. The whole Gantt view needs to be editable as well for this to have any effect.

Parameters:

editable pass true to make this item editable

See also:

[editable\(\)](#), [KDGanttView::setEditable\(\)](#), [KDGanttView::editable\(\)](#)

4.4.4.37 void KDGanttViewItem::setEnabled (bool *on*)

Specifies whether this item is enabled. If disabled, the item stays in the Gantt view and the item is shown in gray to show that the item is disabled. All signals of this item (like [itemLeftClicked\(this \)](#)) are blocked. If the item displays its subitems (childs) as a group, ([displaySubitemsAsGroup\(\)](#) == true) all changes apply to all subitems as well.

Parameters:

on pass true to make this item editable

See also:

[enabled\(\)](#)

4.4.4.38 void KDGanttViewItem::setEndTime (const QDateTime & end) [virtual]

Specifies the end time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. Reimplemented in the subclasses

Parameters:

end the end time

See also:

[endTime\(\)](#), [setStartTime\(\)](#), [startTime\(\)](#)

Reimplemented in [KDGanttViewSummaryItem](#), and [KDGanttViewTaskItem](#).

4.4.4.39 void KDGanttViewItem::setFont (const QFont & font)

Sets the font to be used for the text in this item.

Parameters:

font the font to be shown

See also:

[font\(\)](#)

4.4.4.40 void KDGanttViewItem::setHighlight (bool highlight)

Specifies whether this item should be shown highlighted. The user can also highlight items with the mouse. If the item displays its subitems (children) as a group ([displaySubitemsAsGroup\(\)](#) == true), all changes apply to all subitems as well.

Parameters:

highlight true in order to highlight, false in order to turn highlighting off for this item

See also:

[highlight\(\)](#)

4.4.4.41 void KDGanttViewItem::setHighlightColors (const QColor & start, const QColor & middle, const QColor & end)

Specifies the highlight colors in which to draw the shapes of this item.

It is advisable not to use this method, but rather set the highlight colors for all items of a type with [KDGanttView::setHighlightColors\(\)](#) in order to get a uniform Gantt view.

If the item displays its subitems (children) as a group, ([displaySubitemsAsGroup\(\)](#) == true) all changes apply to all subitems as well.

Parameters:

start the highlight color for the start shape

middle the highlight color for the middle shape

end the highlight color for the end shape

See also:

[highlightColors\(\)](#), [setShapes\(\)](#), [shapes\(\)](#)

4.4.4.42 void KDGanttViewItem::setItemVisible (bool *on*)

Specifies whether this item is visible.

Parameters:

on pass true to make this item visible

See also:

[itemVisible \(\)](#)

4.4.4.43 void KDGanttViewItem::setListViewText (int *column*, const QString & *text*)

Sets the text to be shown in this item in the list view.

Parameters:

column the column in which the text will be shown

text the text to be shown

See also:

[text\(\)](#), [setTextColor\(\)](#), [textColor\(\)](#), [setText\(\)](#), [listViewText\(\)](#)

4.4.4.44 void KDGanttViewItem::setListViewText (const QString & *text*, int *column* = 0)**Deprecated**

Use [setListViewText\(int, const QString& \)](#) instead

4.4.4.45 void KDGanttViewItem::setOpen (bool *open*) [virtual]

This method is reimplemented for internal purposes.

4.4.4.46 void KDGanttViewItem::setPixmap (const QPixmap & *pixmap*)**Deprecated**

use [setPixmap\(int, const QPixmap& \)](#) instead

4.4.4.47 void KDGanttViewItem::setPixmap (int *column*, const QPixmap & *pixmap*)

Sets the pixmap that is shown in the listview.

Parameters:

column the column in which the pixmap is shown

pixmap the pixmap to show

See also:

[pixmap\(\)](#)

4.4.4.48 void KDGanttViewItem::setPriority (int *prio*)

Specifies the priority of this item. Valid values are between 1 and 199. A priority less than 100 means that the item is painted in the Gantt chart below the grid. A priority more than 100 means that the item is painted in the Gantt chart over the grid. For a value of 100, the behavior is unspecified. An item with a higher priority is painted over an item with a lower priority in the Gantt chart. The painting order of items with the same priority is unspecified. For Calendar items, the default priority is 50, for all other items it is 150. This feature makes only sense for an item which is a child of another item, which [displaySubitemsAsGroup\(\)](#) property is set to true.

Parameters:

prio the new priority of this item.

See also:

[priority\(\)](#), [displaySubitemsAsGroup\(\)](#)

4.4.4.49 void KDGanttViewItem::setShapes (Shape *start*, Shape *middle*, Shape *end*)

Specifies the shapes to be used for this item.

It is advisable not to use this method, but rather set the shapes for all items of a type with [KDGanttView::setShapes\(\)](#) in order to get a uniform Gantt view.

Parameters:

start the start shape

middle the middle shape

end the end shape

See also:

[shapes\(\)](#), [setColors\(\)](#), [colors\(\)](#)

4.4.4.50 void KDGanttViewItem::setShowNoInformation (bool *show*)

Specifies whether the 'showNoInformation' line should be shown for this item. The 'showNoInformation' line is drawn over the whole timeline. The height of the line is the height of the item. The brush of the line is specified by [KDGanttView::setNoInformationBrush\(\)](#). (i.e. the same brush for all items of the Gantt view). The default brush is `QBrush(QColor (100,100,100), Qt::FDiagPattern)`;

Parameters:

show if true, the 'showNoInformation' line is shown for this item

See also:

[showNoInformation\(\)](#), [KDGanttView::setNoInformationBrush\(\)](#), [KDGanttView::noInformationBrush\(\)](#)

4.4.4.51 void KDGanttViewItem::setStartTime (const QDateTime & *start*) [virtual]

Specifies the start time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. Reimplemented in the subclasses.

Parameters:

start the start time

See also:

[startTime\(\)](#), [setEndTime\(\)](#), [endTime\(\)](#)

Reimplemented in [KDGanttViewItem](#), [KDGanttViewSummaryItem](#), and [KDGanttViewTaskItem](#).

4.4.4.52 void KDGanttViewItem::setText (const QString & text)

Sets the text to be shown in this item in the Gantt view. For a [KDGanttViewTaskItem](#) with [displaySubitemsAsGroup\(\)](#) == true, the text is shown in the item itself and the text is truncated automatically, if it does not fit in the item. For all other item types, the text is shown to the right of the item.

Parameters:

text the text to be shown

See also:

[text\(\)](#), [setTextColor\(\)](#), [textColor\(\)](#), [setListViewText\(\)](#), [listViewText\(\)](#)

4.4.4.53 void KDGanttViewItem::setTextColor (const QColor & color)

Specifies the color to be used for the text of this item.

It is advisable not to use this method, but rather set the text color for all items with [KDGanttView::setTextColor\(\)](#) in order to get a uniform Gantt view. If the item displays its subitems (children) as a group, ([displaySubitemsAsGroup\(\)](#) == true) all changes apply to all subitems as well.

Parameters:

color the text color

See also:

[textColor\(\)](#), [setText\(\)](#), [text\(\)](#)

4.4.4.54 void KDGanttViewItem::setTooltipText (const QString & text)

Sets the text to show in a tooltip for this item.

Parameters:

text the tooltip text

See also:

[tooltipText\(\)](#)

4.4.4.55 void KDGanttViewItem::setWhatsThisText (const QString & text)

Sets the text to show in a What's This window for this item.

Parameters:

text the what's this text

See also:

[whatsThisText\(\)](#)

4.4.4.56 void KDGanttViewItem::shapes (Shape & start, Shape & middle, Shape & end) const

Returns the shapes used for this item

Parameters:

start returns the start shape

middle returns the middle shape

end returns the end shape

See also:

[setShapes\(\)](#), [setColors\(\)](#), [colors\(\)](#)

4.4.4.57 bool KDGanttViewItem::showNoInformation ()

Returns whether the 'showNoInformation' line should be shown for this item

Returns:

true if showNoInformation line should be shown

See also:

[setShowNoInformation\(\)](#), [KDGanttView::setNoInformationBrush\(\)](#), [KDGanttView::noInformationBrush\(\)](#)

4.4.4.58 QDateTime KDGanttViewItem::startTime () const

Returns the start time of this item.

Returns:

the start time of this item

See also:

[setStartTime\(\)](#), [setEndTime\(\)](#), [endTime\(\)](#)

4.4.4.59 bool KDGanttViewItem::subitemIsCalendar () const

Returns whether this item has at least one subitem that is a calendar. A subitem is a calendar, if that item has at least one subitem or `displaySubitemAsGroup()` is true for that item.

Returns:

true if the item has at least one subitem that is a calendar.

4.4.4.60 QString KDGanttViewItem::text () const

Returns the text to be shown in this item in the Gantt view.

Returns:

the text to be shown in this item

See also:

[setText\(\)](#), [setTextColor\(\)](#), [textColor\(\)](#), [setListViewText\(\)](#), [listViewText\(\)](#)

4.4.4.61 QColor KDGanttViewItem::textColor () const

Returns the color used for the text of this item.

Returns:

the text color

See also:

[setTextColor\(\)](#), [setText\(\)](#), [text\(\)](#)

4.4.4.62 QString KDGanttViewItem::tooltipText () const

Returns the tooltip text of this item

Returns:

the tooltip text

See also:

[setTooltipText\(\)](#)

4.4.4.63 KDGanttViewItem::Type KDGanttViewItem::type () const

Returns the type of the item. This may be Event, Task, Summary.

Returns:

the type of the item

4.4.4.64 void KDGanttViewItem::updateCanvasItems () [protected]

Updates the colors of the item, but not the coordinates.

4.4.4.65 QString KDGanttViewItem::whatsThisText () const

Returns the what's this text of this item

Returns:

the what's this text

See also:

[setWhatsThisText\(\)](#)

4.4.5 Member Data Documentation

4.4.5.1 KDGanttViewItem::actualEnd [protected]

the line at the actual end of the item

4.4.5.2 KDGanttViewItem::blockUpdating [protected]

if true, updates to this item are currently blocked, to reduce flicker or speed up redraws

4.4.5.3 KDGanttViewItem::endLine [protected]

the line at the end of the item

4.4.5.4 KDGanttViewItem::endLineBack [protected]

the background line at the end of the item

4.4.5.5 KDGanttViewItem::endShape [protected]

the shape at the end of the item

4.4.5.6 KDGanttViewItem::endShapeBack [protected]

the background shape at the end of the item

4.4.5.7 KDGanttViewItem::isEditable [protected]

whether this item is currently editable or not

4.4.5.8 KDGanttViewItem::isHighlighted [protected]

whether this item is currently highlighted or not

4.4.5.9 KDGanttViewItem::isVisibleInGanttView [protected]

this instance variable is true if the item is visible in the Gantt view

4.4.5.10 KDGanttViewItem::midShape [protected]

the shape in the middle of the item

4.4.5.11 KDGanttViewItem::midShapeBack [protected]

the background shape in the middle of the item

4.4.5.12 KDGanttViewItem::myEndTime [protected]

the ending time of this item

4.4.5.13 KDGanttViewItem::myGanttView [protected]

a pointer to the [KDGanttView](#) object to which this item belongs

4.4.5.14 KDGanttViewItem::myItemSize [protected]

the current size of this item

4.4.5.15 KDGanttViewItem::myStartTime [protected]

the starting time of this item

4.4.5.16 KDGanttViewItem::startLine [protected]

the line at the beginning of the item

4.4.5.17 KDGanttViewItem::startLineBack [protected]

the background line at the beginning of the item

4.4.5.18 KDGanttViewItem::startShape [protected]

the shape at the beginning of the item

4.4.5.19 KDGanttViewItem::startShapeBack [protected]

the background shape at the beginning of the item

4.4.5.20 KDGanttViewItem::textCanvasText [protected]

the actual string that is displayed in the text object for this item

The documentation for this class was generated from the following files:

- KDGanttViewItem.h
- KDGanttViewItem.cpp

4.5 KDGanttViewItemDrag Class Reference

Drag and drop of KD Gantt items.

```
#include <KDGanttViewItemDrag.h>
```

Public Member Functions

- [KDGanttViewItemDrag](#) ([KDGanttViewItem](#) *item, QWidget *source, const char *name)
- QByteArray [encodedData](#) (const char *c) const
- [KDGanttViewItem](#) * [getItem](#) ()

Static Public Member Functions

- bool [canDecode](#) (const QMimeSource *e)
- bool [decode](#) (const QMimeSource *e, QString &)

4.5.1 Detailed Description

Drag and drop of KD Gantt items.

This class implements drag and drop of KD Gantt items within a Gantt chart. It is mainly used for internal purposes, but made a part of the public API nevertheless, as you may want to subclass it for some specialized functionality.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 KDGanttViewItemDrag::KDGanttViewItemDrag ([KDGanttViewItem](#) * *item*, QWidget * *source*, const char * *name*)

The constructor. Creates a KDGanttViewItemDrag object and initializes the drag data in the form of an XML document.

Parameters:

- item* the item that is dragged
- source* the source widget
- name* the internal object name

4.5.3 Member Function Documentation

4.5.3.1 bool KDGanttViewItemDrag::canDecode (const QMimeSource * *e*) [static]

Returns whether this drag object class can decode the data passed in *e*.

Parameters:

- e* the mime source that has been dragged

Returns:

- true if KDGanttViewItemDrag can decode the data in *e*.

4.5.3.2 `bool KDGanttViewItemDrag::decode (const QMimeSource * e, QString & string)` [static]

Decodes the data passed in *e* into an XML string that is written into *string*.

Parameters:

- e* the data to decode
- string* the resulting XML string

Returns:

- true if the operation succeeded

4.5.3.3 `QByteArray KDGanttViewItemDrag::encodedData (const char * c) const`

Returns the encoded data of the drag object.

Parameters:

- c* the format of the data

Returns:

- the encoded data of the drag object

4.5.3.4 `KDGanttViewItem * KDGanttViewItemDrag::getItem ()`

Returns the dragged item

Returns:

- the dragged item

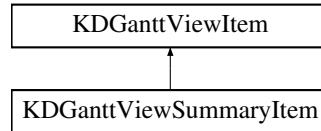
The documentation for this class was generated from the following files:

- KDGanttViewItemDrag.h
- KDGanttViewItemDrag.cpp

4.6 KDGanttViewSummaryItem Class Reference

```
#include <KDGanttViewSummaryItem.h>
```

Inheritance diagram for KDGanttViewSummaryItem::



Public Member Functions

- [KDGanttViewSummaryItem](#) ([KDGanttView](#) *view, const QString &lvtext=QString::null, const QString &name=QString::null)
- [KDGanttViewSummaryItem](#) ([KDGanttViewItem](#) *parent, const QString &lvtext=QString::null, const QString &name=QString::null)
- [KDGanttViewSummaryItem](#) ([KDGanttView](#) *view, [KDGanttViewItem](#) *after, const QString &lvtext=QString::null, const QString &name=QString::null)
- [KDGanttViewSummaryItem](#) ([KDGanttViewItem](#) *parent, [KDGanttViewItem](#) *after, const QString &lvtext=QString::null, const QString &name=QString::null)
- virtual [~KDGanttViewSummaryItem](#) ()
- void [setMiddleTime](#) (const QDateTime &)
- QDateTime [middleTime](#) () const
- void [setActualEndTime](#) (const QDateTime &end)
- QDateTime [actualEndTime](#) () const
- void [setStartTime](#) (const QDateTime &start)
- void [setEndTime](#) (const QDateTime &end)

4.6.1 Detailed Description

A summary item in a Gantt chart.

This class represents summary items in Gantt charts.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 KDGanttViewSummaryItem::KDGanttViewSummaryItem ([KDGanttView](#) * view, const QString & lvtext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

view the Gantt view to insert this item into

lvtext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.6.2.2 KDGanttViewSummaryItem::KDGanttViewSummaryItem (KDGanttViewItem * parent, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

parent a parent item under which this one goes

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.6.2.3 KDGanttViewSummaryItem::KDGanttViewSummaryItem (KDGanttView * view, KDGanttViewItem * after, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

view the Gantt view to insert this item into

after another item at the same level behind which this one should go

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.6.2.4 KDGanttViewSummaryItem::KDGanttViewSummaryItem (KDGanttViewItem * parent, KDGanttViewItem * after, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

parent a parent item under which this one goes

after another item at the same level behind which this one should go

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.6.2.5 KDGanttViewSummaryItem::~~KDGanttViewSummaryItem () [virtual]

The destructor. Delete the datetimes, if created.

4.6.3 Member Function Documentation

4.6.3.1 QDateTime KDGanttViewSummaryItem::actualEndTime () const

Returns the actual end time of this item.

Returns:

the actual end time of this item

See also:

[setActualEndTime\(\)](#)

4.6.3.2 QDateTime KDGanttViewSummaryItem::middleTime () const

Returns the middle time of this summary item. If there is no middle time defined, the start time is returned.

Returns:

the middle time of this summary item. If there is no middle time defined, the start time is returned.

4.6.3.3 void KDGanttViewSummaryItem::setActualEndTime (const QDateTime & end)

Specifies the actual end time of this item. The parameter must be valid and non-null. Items with undefined start or end times lead to undefined visual results.

Parameters:

end the actual end time

See also:

[actualEndTime\(\)](#) [startTime\(\)](#)

4.6.3.4 void KDGanttViewSummaryItem::setEndTime (const QDateTime & end) [virtual]

Specifies the end time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. If the end time is less the mid time, the mid time is set to this end time automatically.

Parameters:

end the end time

See also:

[endTime\(\)](#), [setStartTime\(\)](#), [startTime\(\)](#)

Reimplemented from [KDGanttViewItem](#).

4.6.3.5 void KDGanttViewSummaryItem::setMiddleTime (const QDateTime & dateTime)

Specifies the middle time of this summary item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set.

Parameters:

dateTime the middle time

See also:

[middleTime\(\)](#)

4.6.3.6 void KDGanttViewSummaryItem::setStartTime (const QDateTime & *start*) [virtual]

Specifies the start time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. If the start time is less the mid time, the mid time is set to this start time automatically.

Parameters:

start the start time

See also:

[startTime\(\)](#), [setEndTime\(\)](#), [endTime\(\)](#)

Reimplemented from [KDGanttViewItem](#).

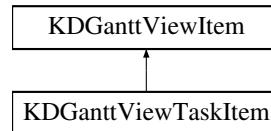
The documentation for this class was generated from the following files:

- KDGanttViewSummaryItem.h
- KDGanttViewSummaryItem.cpp

4.7 KDGanttViewItem Class Reference

```
#include <KDGanttViewItem.h>
```

Inheritance diagram for KDGanttViewItem::



Public Member Functions

- `KDGanttViewItemTaskItem` (`KDGanttView` *view, const `QString` &lvtext=`QString::null`, const `QString` &name=`QString::null`)
- `KDGanttViewItemTaskItem` (`KDGanttViewItem` *parent, const `QString` &lvtext=`QString::null`, const `QString` &name=`QString::null`)
- `KDGanttViewItemTaskItem` (`KDGanttView` *view, `KDGanttViewItem` *after, const `QString` &lvtext=`QString::null`, const `QString` &name=`QString::null`)
- `KDGanttViewItemTaskItem` (`KDGanttViewItem` *parent, `KDGanttViewItem` *after, const `QString` &lvtext=`QString::null`, const `QString` &name=`QString::null`)
- virtual `~KDGanttViewItemTaskItem` ()
- void `setStartTime` (const `QDateTime` &start)
- void `setEndTime` (const `QDateTime` &end)

4.7.1 Detailed Description

This class represents calendar items in Gantt charts.

A calendar item in a Gantt chart has no start/end shape, it is displayed as a rectangle. You can set the colors as usual, where only the first argument of `setColors(col, col, col)` is important. If the start time is equal to the end time, the item is displayed as \emptyset , showing that there is no time interval set.

For a `KDGanttViewItemTaskItem`, the text, setted by `setText()`, is shown in the item itself and the text is truncated automatically to fit in the item. For all other item types, the text is shown right of the item.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 `KDGanttViewItemTaskItem::KDGanttViewItem` (`KDGanttView` * view, const `QString` & lvtext = `QString::null`, const `QString` & name = `QString::null`)

Constructs an empty Gantt item of type event.

Parameters:

view the Gantt view to insert this item into

lvtext the text to show in the listview

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.7.2.2 KDGanttViewItem::KDGanttViewItem (KDGanttViewItem * parent, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

parent a parent item under which this one goes

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.7.2.3 KDGanttViewItem::KDGanttViewItem (KDGanttView * view, KDGanttViewItem * after, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

view the Gantt view to insert this item into

after another item at the same level behind which this one should go

ltext the text to show in the list view

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.7.2.4 KDGanttViewItem::KDGanttViewItem (KDGanttViewItem * parent, KDGanttViewItem * after, const QString & ltext = QString::null, const QString & name = QString::null)

Constructs an empty Gantt item of type event.

Parameters:

parent a parent item under which this one goes

after another item at the same level behind which this one should go

ltext the text to show in the listview

name the name by which the item can be identified. If no name is specified, a unique name will be generated

4.7.2.5 KDGanttViewItem::~~KDGanttViewItem () [virtual]

The destructor.

4.7.3 Member Function Documentation

4.7.3.1 void KDGanttViewItem::setEndTime (const QDateTime & end) [virtual]

Specifies the end time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. If the end time is less the start time, the start time is set to this end time automatically.

Parameters:

end the end time

See also:

[setStartTime\(\)](#), [startTime\(\)](#), [endTime\(\)](#)

Reimplemented from [KDGanttViewItem](#).

4.7.3.2 void KDGanttViewItem::setStartTime (const QDateTime & start) [virtual]

Specifies the start time of this item. The parameter must be valid and non-null. If the parameter is invalid or null, no value is set. If the start time is greater than the end time, the end time is set to this start time automatically.

Parameters:

start the start time

See also:

[startTime\(\)](#), [setEndTime\(\)](#), [endTime\(\)](#)

Reimplemented from [KDGanttViewItem](#).

The documentation for this class was generated from the following files:

- [KDGanttViewItem.h](#)
- [KDGanttViewItem.cpp](#)

4.8 KDGanttViewTaskLink Class Reference

```
#include <KDGanttViewTaskLink.h>
```

Public Member Functions

- [KDGanttViewTaskLink](#) (QPtrList< [KDGanttViewItem](#) > from, QPtrList< [KDGanttViewItem](#) > to)
- [KDGanttViewTaskLink](#) ([KDGanttViewTaskLinkGroup](#) *group, QPtrList< [KDGanttViewItem](#) > from, QPtrList< [KDGanttViewItem](#) > to)
- [KDGanttViewTaskLink](#) ([KDGanttViewTaskLinkGroup](#) *group, [KDGanttViewItem](#) *from, [KDGanttViewItem](#) *to)
- [KDGanttViewTaskLink](#) ([KDGanttViewItem](#) *from, [KDGanttViewItem](#) *to)
- QPtrList< [KDGanttViewItem](#) > from () const
- QPtrList< [KDGanttViewItem](#) > to () const
- void [removeItemFromList](#) ([KDGanttViewItem](#) *)
- void [setVisible](#) (bool)
- bool [isVisible](#) () const
- [KDGanttViewTaskLinkGroup](#) * group ()
- void [setGroup](#) ([KDGanttViewTaskLinkGroup](#) *)
- void [setHighlight](#) (bool highlight)
- bool [highlight](#) () const
- void [setColor](#) (const QColor &color)
- QColor [color](#) () const
- void [setHighlightColor](#) (const QColor &color)
- QColor [highlightColor](#) () const
- void [setTooltipText](#) (const QString &text)
- QString [tooltipText](#) () const
- void [setWhatsThisText](#) (const QString &text)
- QString [whatsThisText](#) () const
- void [createNode](#) (QDomDocument &doc, QDomElement &parentElement)

Static Public Member Functions

- [KDGanttViewTaskLink](#) * [createFromDomElement](#) (QDomElement &)

4.8.1 Detailed Description

This class represents a link between a number of Gantt chart items.

It always connects source items with target items. Task links can be grouped into [KDGanttViewTaskLinkGroup](#) objects. If a Gantt view item is deleted, it is removed from the fromList or from the toList. If one of the lists becomes empty, the complete task link is deleted as well.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 [KDGanttViewTaskLink::KDGanttViewTaskLink](#) (QPtrList< [KDGanttViewItem](#) > from, QPtrList< [KDGanttViewItem](#) > to)

Creates a task link that connects all items in the source item list from to all items in the destination item list to.

Parameters:*from* the source items*to* the target items**4.8.2.2 KDGanttViewTaskLink::KDGanttViewTaskLink (KDGanttViewTaskLinkGroup * group, QPtrList< KDGanttViewItem > from, QPtrList< KDGanttViewItem > to)**

Creates a task link that connects all items in the source item list from to all items in the destination item list to. Inserts the link directly into a link group.

Parameters:*group* the link group to insert this link into*from* the source items*to* the target items**4.8.2.3 KDGanttViewTaskLink::KDGanttViewTaskLink (KDGanttViewTaskLinkGroup * group, KDGanttViewItem * from, KDGanttViewItem * to)**

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Creates a task link that connects two items and inserts the link directly into a link group. Note, that the [from\(\)](#) and [to\(\)](#) functions are returning a list, in this case containing only one item.

Parameters:*group* the link group to insert this link into*from* the source item*to* the target item**4.8.2.4 KDGanttViewTaskLink::KDGanttViewTaskLink (KDGanttViewItem * from, KDGanttViewItem * to)**

This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Creates a task link that connects two items. Note, that the [from\(\)](#) and [to\(\)](#) functions are returning a list, in this case containing only one item.

Parameters:*from* the source item*to* the target item**4.8.3 Member Function Documentation****4.8.3.1 QColor KDGanttViewTaskLink::color () const**

Returns the color in which this task link is drawn.

Returns:

the color in which this task link is drawn

See also:

[setColor\(\)](#)

4.8.3.2 KDGanttViewTaskLink * KDGanttViewTaskLink::createFromDomElement (QDomElement & *element*) [static]

Creates a KDGanttViewTaskLink according to the specification in a DOM element.

Parameters:

element the DOM element from which to read the specification

Returns:

the newly created task link

4.8.3.3 void KDGanttViewTaskLink::createNode (QDomDocument & *doc*, QDomElement & *parentElement*)

Creates a DOM node that describes this task link.

Parameters:

doc the DOM document to which the node belongs

parentElement the element into which to insert this node

4.8.3.4 QPtrList< KDGanttViewItem > KDGanttViewTaskLink::from () const

Returns the list of source item of this task link.

Returns:

the list of source item of this task link

See also:

[to\(\)](#)

4.8.3.5 KDGanttViewTaskLinkGroup * KDGanttViewTaskLink::group ()

Returns the group (if any) to which this task link belongs.

Returns:

the group to which this task link belongs; 0 if it does not belong to any group.

See also:

[KDGanttViewTaskLinkGroup](#)

4.8.3.6 bool KDGanttViewTaskLink::highlight () const

Returns whether this task link is highlighted, either programmatically by [setHighlight\(\)](#) or by the user with the mouse.

Returns:

true if the task link is highlighted

See also:

[setHighlight\(\)](#)

4.8.3.7 QColor KDGanttViewTaskLink::highlightColor () const

Returns the highlight color in which this task link is drawn.

Returns:

the highlight color in which this task link is drawn

See also:

[setHighlightColor\(\)](#)

4.8.3.8 bool KDGanttViewTaskLink::isVisible () const

Returns whether this task link should be visible or not.

Returns:

true if the task link is visible

See also:

[setVisible\(\)](#)

4.8.3.9 void KDGanttViewTaskLink::removeItemFromList (KDGanttViewItem * item)

Removes a [KDGanttViewItem](#) from the lists.

See also:

[to\(\)](#) [from\(\)](#)

4.8.3.10 void KDGanttViewTaskLink::setColor (const QColor & color)

Specifies the color to draw this task link in.

Parameters:

color the color to draw this task link in

See also:

[color\(\)](#)

4.8.3.11 void KDGanttViewTaskLink::setGroup (KDGanttViewTaskLinkGroup * group)

Inserts this task link in a group. If the parameter is 0, the task link is removed from any group

Parameters:

group the group, this task link has to be inserted

See also:

[KDGanttViewTaskLinkGroup](#)

4.8.3.12 void KDGanttViewTaskLink::setHighlight (bool highlight)

Specifies whether this task link should be shown highlighted. The user can also highlight a task link with the mouse.

Parameters:

highlight pass true in order to highlight this task link

See also:

[highlight\(\)](#)

4.8.3.13 void KDGanttViewTaskLink::setHighlightColor (const QColor & color)

Specifies the highlight color to draw this task link in.

Parameters:

color the highlight color to draw this task link in

See also:

[highlightColor\(\)](#)

4.8.3.14 void KDGanttViewTaskLink::setTooltipText (const QString & text)

Specifies the text to be shown as a tooltip for this task link.

Parameters:

text the tooltip text

See also:

[tooltipText\(\)](#)

4.8.3.15 void KDGanttViewTaskLink::setVisible (bool visible)

Specifies whether this task link should be visible or not.

Parameters:

visible pass true to make this task link visible, and false to hide it

See also:

[isVisible\(\)](#)

4.8.3.16 void KDGanttViewTaskLink::setWhatsThisText (const QString & text)

Specifies the text to be shown in a what's this window for this task link.

Parameters:

text the what's this text

See also:

[whatsThisText\(\)](#)

4.8.3.17 QList< KDGanttViewItem > KDGanttViewTaskLink::to () const

Returns the list of target items of this task link.

Returns:

the list of target item of this task link

See also:

[from\(\)](#)

4.8.3.18 QString KDGanttViewTaskLink::tooltipText () const

Returns the tooltip text of this task link.

Returns:

the tooltip text of this task link

See also:

[setTooltipText\(\)](#)

4.8.3.19 QString KDGanttViewTaskLink::whatsThisText () const

Returns the what's this text of this task link.

Returns:

the what's this text of this task link

See also:

[setWhatsThisText\(\)](#)

The documentation for this class was generated from the following files:

- KDGanttViewTaskLink.h
- KDGanttViewTaskLink.cpp

4.9 KDGanttViewTaskLinkGroup Class Reference

```
#include <KDGanttViewTaskLinkGroup.h>
```

Public Member Functions

- [KDGanttViewTaskLinkGroup](#) (const QString &name)
- [KDGanttViewTaskLinkGroup](#) ()
- [~KDGanttViewTaskLinkGroup](#) ()
- void [insert](#) ([KDGanttViewTaskLink](#) *)
- bool [remove](#) ([KDGanttViewTaskLink](#) *)
- void [setVisible](#) (bool show)
- bool [visible](#) () const
- void [setHighlight](#) (bool highlight)
- bool [highlight](#) () const
- void [setColor](#) (const QColor &color)
- QColor [color](#) () const
- void [setHighlightColor](#) (const QColor &color)
- QColor [highlightColor](#) () const
- void [createNode](#) (QDomDocument &doc, QDomElement &parentElement)

Static Public Member Functions

- [KDGanttViewTaskLinkGroup](#) * [find](#) (const QString &name)
- [KDGanttViewTaskLinkGroup](#) * [createFromDomElement](#) (QDomElement &)

4.9.1 Detailed Description

A group of task links.

This class groups a number of task links together in order to manipulate them uniformly.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 KDGanttViewTaskLinkGroup::KDGanttViewTaskLinkGroup (const QString & name)

Constructs an empty task link group and records it under the name *name* so that it can later be found again with [KDGanttViewTaskLinkGroup::find\(\)](#).

Parameters:

name the search name of this task link group

4.9.2.2 KDGanttViewTaskLinkGroup::KDGanttViewTaskLinkGroup ()

Constructs an empty task link group

4.9.2.3 `KDGanttViewTaskLinkGroup::~~KDGanttViewTaskLinkGroup ()`

Destructor Removes this task link group from the list of task link groups in the `KDGanttView` class.

4.9.3 Member Function Documentation

4.9.3.1 `QColor KDGanttViewTaskLinkGroup::color () const`

Returns the color in which the task links in this group are drawn. If task links have been assigned individual colors, the return value of this method is undefined. This method is not particularly useful and is mainly provided for API uniformity reasons.

Returns:

the color in which the task links in this group are drawn

See also:

[setColor\(\)](#)

4.9.3.2 `KDGanttViewTaskLinkGroup * KDGanttViewTaskLinkGroup::createFromDomElement (QDomElement & element) [static]`

Creates a `KDGanttViewTaskLinkGroup` according to the specification in a DOM element.

Parameters:

element the DOM element from which to read the specification

Returns:

the newly created task link group

4.9.3.3 `void KDGanttViewTaskLinkGroup::createNode (QDomDocument & doc, QDomElement & parentElement)`

Creates a DOM node that describes this task link group.

Parameters:

doc the DOM document to which the node belongs

parentElement the element into which to insert this node

4.9.3.4 `KDGanttViewTaskLinkGroup * KDGanttViewTaskLinkGroup::find (const QString & name) [static]`

Returns the task link group with the specified name.

Parameters:

name the name to search for

Returns:

the task link group with the specified name; 0 if no group with that name exists

4.9.3.5 bool KDGanttViewTaskLinkGroup::highlight () const

Returns whether all task links in this group are highlighted, either programmatically by [setHighlight\(\)](#) or by the user with the mouse. This method is not particularly useful and is mainly provided for API uniformity reasons.

Returns:

true if all the task links in this group are highlighted

See also:

[setHighlight\(\)](#)

4.9.3.6 QColor KDGanttViewTaskLinkGroup::highlightColor () const

Returns the highlight color in which the task links in this group are drawn. If task links have been assigned individual highlight colors, the return value of this method is undefined. This method is not particularly useful and is mainly provided for API uniformity reasons.

Returns:

the highlight color in which the task links in this group are drawn

See also:

[setColor\(\)](#)

4.9.3.7 void KDGanttViewTaskLinkGroup::insert (KDGanttViewTaskLink * link)

Adds a task link LINK to this group. If the task link is already a member of another group, it will be removed from it. This function is equivalent to LINK->setGroup(this), where this is a pointer to this TaskLinkGroup.

Parameters:

link a pointer to the task link to add to this task link group visible, and false to hide them

See also:

[remove\(\)](#)

4.9.3.8 bool KDGanttViewTaskLinkGroup::remove (KDGanttViewTaskLink * link)

Removes a task link LINK from this group. You may remove a tasklink LINK from its group with LINK->setGroup(0).

Parameters:

link a pointer to the task link to remove from this task link group

Returns:

true if the task link was a member of this group

See also:

[insert\(\)](#)

4.9.3.9 void KDGanttViewTaskLinkGroup::setColor (const QColor & *color*)

Specifies the color to draw the task links in this group in.

Parameters:

color the color to draw the task links in this group in

See also:

[color\(\)](#)

4.9.3.10 void KDGanttViewTaskLinkGroup::setHighlight (bool *highlight*)

Specifies whether the task links of this group should be shown highlighted. The user can also highlight a task link with the mouse.

Parameters:

highlight pass true in order to highlight the task links in this group

See also:

[highlight\(\)](#)

4.9.3.11 void KDGanttViewTaskLinkGroup::setHighlightColor (const QColor & *color*)

Specifies the highlight color to draw the task links in this group in.

Parameters:

color the highlight color to draw the task links in this group in

See also:

[color\(\)](#)

4.9.3.12 void KDGanttViewTaskLinkGroup::setVisible (bool *show*)

Specifies whether the task links of this group should be visible or not.

Parameters:

show visible pass true to make the task links of this group visible, and false to hide them

See also:

[isVisible\(\)](#)

4.9.3.13 bool KDGanttViewTaskLinkGroup::visible () const

Returns whether the task links of this group should be visible or not.

Returns:

true if the task links of this group are visible

See also:

[setVisible\(\)](#)

The documentation for this class was generated from the following files:

- KDGanttViewTaskLinkGroup.h
- KDGanttViewTaskLinkGroup.cpp

Chapter 5

KDGantt Page Documentation

5.1 Deprecated List

Member **KDGanttView::dragEnabled()** **const** Use `isDragEnabled()` instead

Member **KDGanttView::dropEnabled()** **const** Use `isDropEnabled()` instead

Member **KDGanttView::timeIntervallSelected(const QDateTime &start, const QDateTime &end)**
This signal is deprecated, do not use it in new code; use `timeIntervalSelected()` instead. `timeIntervallSelected()` will be removed in future versions.

Member **KDGanttViewItem::setListViewText(const QString &text, int column=0)** Use `setListViewText(int, const QString&)` instead

Member **KDGanttViewItem::setPixmap(const QPixmap &pixmap)** use `setPixmap(int, const QPixmap&)` instead

Index

- ~KDGanttMinimizeSplitter
 - KDGanttMinimizeSplitter, 9
- ~KDGanttViewItemEventItem
 - KDGanttViewItemEventItem, 55
- ~KDGanttViewItem
 - KDGanttViewItem, 62
- ~KDGanttViewSummaryItem
 - KDGanttViewSummaryItem, 81
- ~KDGanttViewTaskItem
 - KDGanttViewTaskItem, 85
- ~KDGanttViewTaskLinkGroup
 - KDGanttViewTaskLinkGroup, 93
- actualEnd
 - KDGanttViewItem, 75
- actualEndTime
 - KDGanttViewSummaryItem, 81
- addColumn
 - KDGanttView, 15
- addLegendItem
 - KDGanttView, 15
- addTicksLeft
 - KDGanttView, 16
- addTicksRight
 - KDGanttView, 16
- addUserdefinedLegendHeaderWidget
 - KDGanttView, 16
- autoScaleMinorTickCount
 - KDGanttView, 16
- blockUpdating
 - KDGanttViewItem, 75
- calendarMode
 - KDGanttView, 17
- canDecode
 - KDGanttViewItemDrag, 78
- center
 - KDGanttView, 17
- centerTimeline
 - KDGanttView, 17
- centerTimelineAfterShow
 - KDGanttView, 17
- changeBackgroundInterval
 - KDGanttView, 17
- childCount
 - KDGanttView, 18
- clear
 - KDGanttView, 18
- clearBackgroundColor
 - KDGanttView, 18
- clearLegend
 - KDGanttView, 18
- close
 - KDGanttView, 18
- color
 - KDGanttViewTaskLink, 88
 - KDGanttViewTaskLinkGroup, 94
- colors
 - KDGanttView, 19
 - KDGanttViewItem, 62
- columnBackgroundColor
 - KDGanttView, 19
- createFromDomElement
 - KDGanttViewItem, 62, 63
 - KDGanttViewTaskLink, 89
 - KDGanttViewTaskLinkGroup, 94
- createNode
 - KDGanttViewItem, 63
 - KDGanttViewTaskLink, 89
 - KDGanttViewTaskLinkGroup, 94
- decode
 - KDGanttViewItemDrag, 78
- defaultColor
 - KDGanttView, 19
 - KDGanttViewItem, 63
- defaultHighlightColor
 - KDGanttView, 20
 - KDGanttViewItem, 63
- deleteBackgroundInterval
 - KDGanttView, 20
- Direction
 - KDGanttMinimizeSplitter, 8
- displayEmptyTasksAsLine
 - KDGanttView, 20
- displaySubitemsAsGroup
 - KDGanttView, 20
 - KDGanttViewItem, 64
- dragEnabled

- KDGanttView, 21
- dragObject
 - KDGanttView, 21
- drawContents
 - KDGanttView, 21
- dropEnabled
 - KDGanttView, 21
- dropped
 - KDGanttView, 22
- editable
 - KDGanttView, 22
 - KDGanttViewItem, 64
- editItem
 - KDGanttView, 22
- editorEnabled
 - KDGanttView, 22
- enabled
 - KDGanttViewItem, 64
- encodedData
 - KDGanttViewItemDrag, 79
- endLine
 - KDGanttViewItem, 76
- endLineBack
 - KDGanttViewItem, 76
- endShape
 - KDGanttViewItem, 76
- endShapeBack
 - KDGanttViewItem, 76
- endTime
 - KDGanttViewItem, 64
- ensureVisible
 - KDGanttView, 22
- find
 - KDGanttViewItem, 64
 - KDGanttViewTaskLinkGroup, 94
- firstChild
 - KDGanttView, 22
 - KDGanttViewItem, 65
- font
 - KDGanttViewItem, 65
- from
 - KDGanttViewTaskLink, 89
- gantMaximumWidth
 - KDGanttView, 23
- generateAndInsertName
 - KDGanttViewItem, 65
- getChildByName
 - KDGanttViewItem, 65
- getCoordY
 - KDGanttViewItem, 65
- getDateTimeForCoordX
 - KDGanttView, 23
- getItem
 - KDGanttViewItemDrag, 79
- getItemAt
 - KDGanttView, 23
- getItemByGanttViewPos
 - KDGanttView, 23
- getItemByListViewPos
 - KDGanttView, 24
- getItemByName
 - KDGanttView, 24
- getPixmap
 - KDGanttView, 24
- getUpdateEnabled
 - KDGanttView, 24
- group
 - KDGanttViewTaskLink, 89
- gvBackgroundColor
 - KDGanttView, 25
- gvContextMenuRequested
 - KDGanttView, 25
- gvCurrentChanged
 - KDGanttView, 25
- gvItemDoubleClicked
 - KDGanttView, 25
- gvItemLeftClicked
 - KDGanttView, 25
- gvItemMidClicked
 - KDGanttView, 25
- gvItemRightClicked
 - KDGanttView, 25
- gvMouseButtonClicked
 - KDGanttView, 25
- headerVisible
 - KDGanttView, 26
- highlight
 - KDGanttViewItem, 65
 - KDGanttViewTaskLink, 89
 - KDGanttViewTaskLinkGroup, 94
- highlightColor
 - KDGanttViewTaskLink, 90
 - KDGanttViewTaskLinkGroup, 95
- highlightColors
 - KDGanttView, 26
 - KDGanttViewItem, 66
- horBackgroundLines
 - KDGanttView, 26
- horizonEnd
 - KDGanttView, 26
- horizonStart
 - KDGanttView, 26
- HourFormat
 - KDGanttView, 15

- hourFormat
 - KDGanttView, 27
- insert
 - KDGanttViewTaskLinkGroup, 95
- isDragEnabled
 - KDGanttView, 27
- isDropEnabled
 - KDGanttView, 27
- isEditable
 - KDGanttViewItem, 76
- isHighlighted
 - KDGanttViewItem, 76
- isVisible
 - KDGanttViewTaskLink, 90
- isVisibleInGanttView
 - KDGanttViewItem, 76
- itemAbove
 - KDGanttViewItem, 66
- itemBelow
 - KDGanttViewItem, 66
- itemConfigured
 - KDGanttView, 27
- itemDoubleClicked
 - KDGanttView, 27
- itemLeftClicked
 - KDGanttView, 27
- itemMidClicked
 - KDGanttView, 28
- itemRightClicked
 - KDGanttView, 28
- itemVisible
 - KDGanttViewItem, 66
- KDGanttMinimizeSplitter, 7
 - KDGanttMinimizeSplitter, 8
- KDGanttMinimizeSplitter
 - ~KDGanttMinimizeSplitter, 9
 - Direction, 8
 - KDGanttMinimizeSplitter, 8
 - minimizeDirection, 9
 - orientation, 9
 - ResizeMode, 8
 - setMinimizeDirection, 9
 - setOrientation, 9
- KDGanttView, 10
 - KDGanttView, 15
- KDGanttView
 - addColumn, 15
 - addLegendItem, 15
 - addTicksLeft, 16
 - addTicksRight, 16
 - addUserdefinedLegendHeaderWidget, 16
 - autoScaleMinorTickCount, 16
 - calendarMode, 17
 - center, 17
 - centerTimeline, 17
 - centerTimelineAfterShow, 17
 - changeBackgroundInterval, 17
 - childCount, 18
 - clear, 18
 - clearBackgroundColor, 18
 - clearLegend, 18
 - close, 18
 - colors, 19
 - columnBackgroundColor, 19
 - defaultColor, 19
 - defaultHighlightColor, 20
 - deleteBackgroundInterval, 20
 - displayEmptyTasksAsLine, 20
 - displaySubitemsAsGroup, 20
 - dragEnabled, 21
 - dragObject, 21
 - drawContents, 21
 - dropEnabled, 21
 - dropped, 22
 - editable, 22
 - editItem, 22
 - editorEnabled, 22
 - ensureVisible, 22
 - firstChild, 22
 - gantMaximumWidth, 23
 - getDateTimeForCoordX, 23
 - getItemAt, 23
 - getItemByGanttViewPos, 23
 - getItemByListViewPos, 24
 - getItemByName, 24
 - getPixmap, 24
 - getUpdateEnabled, 24
 - gvBackgroundColor, 25
 - gvContextMenuRequested, 25
 - gvCurrentChanged, 25
 - gvItemDoubleClicked, 25
 - gvItemLeftClicked, 25
 - gvItemMidClicked, 25
 - gvItemRightClicked, 25
 - gvMouseButtonClicked, 25
 - headerVisible, 26
 - highlightColors, 26
 - horBackgroundLines, 26
 - horizonEnd, 26
 - horizonStart, 26
 - HourFormat, 15
 - hourFormat, 27
 - isDragEnabled, 27
 - isDropEnabled, 27
 - itemConfigured, 27
 - itemDoubleClicked, 27

- itemLeftClicked, 27
- itemMidClicked, 28
- itemRightClicked, 28
- KDGanttView, 15
- lastItem, 28
- legendDockwindow, 28
- legendHeaderBackgroundColor, 28
- legendIsDockwindow, 28
- listViewWidth, 29
- loadProject, 29
- lvBackgroundColor, 29
- lvContextMenuRequested, 29
- lvCurrentChanged, 29
- lvDragEnterEvent, 29
- lvDragMoveEvent, 30
- lvDropEvent, 30
- lvItemDoubleClicked, 31
- lvItemLeftClicked, 31
- lvItemMidClicked, 31
- lvItemRenamed, 31
- lvItemRightClicked, 31
- lvMouseButtonClicked, 31
- lvMouseButtonPressed, 31
- lvSelectionChanged, 31
- lvStartDrag, 32
- majorScaleCount, 32
- maximumScale, 32
- minimumColumnWidth, 32
- minimumScale, 32
- minorScaleCount, 33
- noInformationBrush, 33
- print, 33
- removeColumn, 33
- RepaintMode, 15
- rescaling, 34
- saveProject, 34
- Scale, 15
- scale, 34
- selectedItem, 34
- setAutoScaleMinorTickCount, 34
- setCalendarMode, 34
- setColors, 35
- setColumnBackgroundColor, 35
- setDefaultColor, 36
- setDefaultHighlightColor, 36
- setDisplayEmptyTasksAsLine, 36
- setDisplaySubitemsAsGroup, 36
- setDragDropEnabled, 37
- setDragEnabled, 37
- setDropEnabled, 37
- setEditable, 37
- setEditorEnabled, 38
- setFont, 38
- setGanttMaximumWidth, 38
- setGvBackgroundColor, 38
- setGvVScrollBarMode, 38
- setHeaderVisible, 39
- setHighlightColors, 39
- setHorBackgroundLines, 39
- setHorizonEnd, 39
- setHorizonStart, 40
- setHourFormat, 40
- setIntervalBackgroundColor, 40
- setLegendHeaderBackgroundColor, 41
- setLegendIsDockwindow, 41
- setListViewWidth, 41
- setLvBackgroundColor, 41
- setLvVScrollBarMode, 41
- setMajorScaleCount, 42
- setMaximumScale, 42
- setMinimumColumnWidth, 42
- setMinimumScale, 42
- setMinorScaleCount, 42
- setNoInformationBrush, 43
- setPaletteBackgroundColor, 43
- setRepaintMode, 43
- setScale, 43
- setSelected, 43
- setShapes, 43
- setShowHeaderPopupMenu, 44
- setShowLegend, 44
- setShowLegendButton, 44
- setShowListView, 45
- setShowMajorTicks, 45
- setShowMinorTicks, 45
- setShowTaskLinks, 45
- setShowTimeTablePopupMenu, 46
- setTextColor, 46
- setTimeHeaderBackgroundColor, 46
- setTimelineToEnd, 46
- setTimelineToStart, 46
- setUpdateEnabled, 46
- setWeekdayBackgroundColor, 47
- setWeekendBackgroundColor, 47
- setWeekendDays, 47
- setYearFormat, 48
- setZoomFactor, 48
- shapes, 48
- show, 48
- showHeaderPopupMenu, 49
- showLegend, 49
- showLegendButton, 49
- showListView, 49
- showMajorTicks, 49
- showMinorTicks, 50
- showTaskLinks, 50
- showTimeTablePopupMenu, 50
- sizeHint, 50

- startDrag, 50
- taskLinkDoubleClicked, 50
- taskLinkGroups, 51
- taskLinkLeftClicked, 51
- taskLinkMidClicked, 51
- taskLinkRightClicked, 51
- taskLinks, 51
- textColor, 51
- timeHeaderBackgroundColor, 51
- timeIntervallSelected, 51
- timeIntervalSelected, 52
- weekdayBackgroundColor, 52
- weekendBackgroundColor, 52
- weekendDays, 52
- YearFormat, 15
- yearFormat, 52
- zoomFactor, 53
- zoomToFit, 53
- zoomToSelection, 53
- KDGanttViewItemEventItem, 54
 - KDGanttViewItemEventItem, 54, 55
- KDGanttViewItemEventItem
 - ~KDGanttViewItemEventItem, 55
 - KDGanttViewItemEventItem, 54, 55
 - leadTime, 55
 - setLeadTime, 56
 - setStartTime, 56
- KDGanttViewItem, 57
 - KDGanttViewItem, 61
- KDGanttViewItem
 - ~KDGanttViewItem, 62
 - actualEnd, 75
 - blockUpdating, 75
 - colors, 62
 - createFromDomElement, 62, 63
 - createNode, 63
 - defaultColor, 63
 - defaultHighlightColor, 63
 - displaySubitemsAsGroup, 64
 - editable, 64
 - enabled, 64
 - endLine, 76
 - endLineBack, 76
 - endShape, 76
 - endShapeBack, 76
 - endTime, 64
 - find, 64
 - firstChild, 65
 - font, 65
 - generateAndInsertName, 65
 - getChildByName, 65
 - getCoordY, 65
 - highlight, 65
 - highlightColors, 66
 - isEditable, 76
 - isHighlighted, 76
 - isVisibleInGanttView, 76
 - itemAbove, 66
 - itemBelow, 66
 - itemVisible, 66
 - KDGanttViewItem, 61
 - listViewText, 66
 - midShape, 76
 - midShapeBack, 76
 - myChildEndTime, 67
 - myChildStartTime, 67
 - myEndTime, 76
 - myGanttView, 76
 - myItemSize, 77
 - myStartTime, 77
 - name, 67
 - nextSibling, 67
 - parent, 67
 - pixmap, 67
 - priority, 68
 - setColors, 68
 - setDefaultColor, 68
 - setDefaultHighlightColor, 68
 - setDisplaySubitemsAsGroup, 69
 - setEditable, 69
 - setEnabled, 69
 - setEndTime, 69
 - setFont, 70
 - setHighlight, 70
 - setHighlightColors, 70
 - setItemVisible, 70
 - setListViewText, 71
 - setOpen, 71
 - setPixmap, 71
 - setPriority, 71
 - setShapes, 72
 - setShowNoInformation, 72
 - setStartTime, 72
 - setText, 73
 - setTextColor, 73
 - setTooltipText, 73
 - setWhatsThisText, 73
 - Shape, 60
 - shapes, 73
 - showNoInformation, 74
 - startLine, 77
 - startLineBack, 77
 - startShape, 77
 - startShapeBack, 77
 - startTime, 74
 - subitemIsCalendar, 74
 - text, 74
 - textCanvasText, 77

- textColor, 74
- tooltipText, 75
- Type, 60
- type, 75
- updateCanvasItems, 75
- whatsThisText, 75
- KDGanttViewItemDrag, 78
 - KDGanttViewItemDrag, 78
- KDGanttViewItemDrag
 - canDecode, 78
 - decode, 78
 - encodedData, 79
 - getItem, 79
 - KDGanttViewItemDrag, 78
- KDGanttViewSummaryItem, 80
 - KDGanttViewSummaryItem, 80, 81
- KDGanttViewSummaryItem
 - ~KDGanttViewSummaryItem, 81
 - actualEndTime, 81
 - KDGanttViewSummaryItem, 80, 81
 - middleTime, 82
 - setActualEndTime, 82
 - setEndTime, 82
 - setMiddleTime, 82
 - setStartTime, 82
- KDGanttViewTaskItem, 84
 - KDGanttViewTaskItem, 84, 85
- KDGanttViewTaskItem
 - ~KDGanttViewTaskItem, 85
 - KDGanttViewTaskItem, 84, 85
 - setEndTime, 85
 - setStartTime, 86
- KDGanttViewTaskLink, 87
 - KDGanttViewTaskLink, 87, 88
- KDGanttViewTaskLink
 - color, 88
 - createFromDomElement, 89
 - createNode, 89
 - from, 89
 - group, 89
 - highlight, 89
 - highlightColor, 90
 - isVisible, 90
 - KDGanttViewTaskLink, 87, 88
 - removeItemFromList, 90
 - setColor, 90
 - setGroup, 90
 - setHighlight, 91
 - setHighlightColor, 91
 - setTooltipText, 91
 - setVisible, 91
 - setWhatsThisText, 91
 - to, 92
 - tooltipText, 92
 - whatsThisText, 92
- KDGanttViewTaskLinkGroup, 93
 - KDGanttViewTaskLinkGroup, 93
- KDGanttViewTaskLinkGroup
 - ~KDGanttViewTaskLinkGroup, 93
 - color, 94
 - createFromDomElement, 94
 - createNode, 94
 - find, 94
 - highlight, 94
 - highlightColor, 95
 - insert, 95
 - KDGanttViewTaskLinkGroup, 93
 - remove, 95
 - setColor, 95
 - setHighlight, 96
 - setHighlightColor, 96
 - setVisible, 96
 - visible, 96
- lastItem
 - KDGanttView, 28
- leadTime
 - KDGanttViewItem, 55
- legendDockwindow
 - KDGanttView, 28
- legendHeaderBackgroundColor
 - KDGanttView, 28
- legendIsDockwindow
 - KDGanttView, 28
- listViewText
 - KDGanttViewItem, 66
- listViewWidth
 - KDGanttView, 29
- loadProject
 - KDGanttView, 29
- lvBackgroundColor
 - KDGanttView, 29
- lvContextMenuRequested
 - KDGanttView, 29
- lvCurrentChanged
 - KDGanttView, 29
- lvDragEnterEvent
 - KDGanttView, 29
- lvDragMoveEvent
 - KDGanttView, 30
- lvDropEvent
 - KDGanttView, 30
- lvItemDoubleClicked
 - KDGanttView, 31
- lvItemLeftClicked
 - KDGanttView, 31
- lvItemMidClicked
 - KDGanttView, 31

- lvItemRenamed
 - KDGanttView, 31
- lvItemRightClicked
 - KDGanttView, 31
- lvMouseButtonClicked
 - KDGanttView, 31
- lvMouseButtonPressed
 - KDGanttView, 31
- lvSelectionChanged
 - KDGanttView, 31
- lvStartDrag
 - KDGanttView, 32
- majorScaleCount
 - KDGanttView, 32
- maximumScale
 - KDGanttView, 32
- middleTime
 - KDGanttViewSummaryItem, 82
- midShape
 - KDGanttViewItem, 76
- midShapeBack
 - KDGanttViewItem, 76
- minimizeDirection
 - KDGanttMinimizeSplitter, 9
- minimumColumnWidth
 - KDGanttView, 32
- minimumScale
 - KDGanttView, 32
- minorScaleCount
 - KDGanttView, 33
- myChildEndTime
 - KDGanttViewItem, 67
- myChildStartTime
 - KDGanttViewItem, 67
- myEndTime
 - KDGanttViewItem, 76
- myGanttView
 - KDGanttViewItem, 76
- myItemSize
 - KDGanttViewItem, 77
- myStartTime
 - KDGanttViewItem, 77
- name
 - KDGanttViewItem, 67
- nextSibling
 - KDGanttViewItem, 67
- noInformationBrush
 - KDGanttView, 33
- orientation
 - KDGanttMinimizeSplitter, 9
- parent
 - KDGanttViewItem, 67
- pixmap
 - KDGanttViewItem, 67
- print
 - KDGanttView, 33
- priority
 - KDGanttViewItem, 68
- remove
 - KDGanttViewTaskLinkGroup, 95
- removeColumn
 - KDGanttView, 33
- removeItemFromList
 - KDGanttViewTaskLink, 90
- RepaintMode
 - KDGanttView, 15
- rescaling
 - KDGanttView, 34
- ResizeMode
 - KDGanttMinimizeSplitter, 8
- saveProject
 - KDGanttView, 34
- Scale
 - KDGanttView, 15
- scale
 - KDGanttView, 34
- selectedItem
 - KDGanttView, 34
- setActualEndTime
 - KDGanttViewSummaryItem, 82
- setAutoScaleMinorTickCount
 - KDGanttView, 34
- setCalendarMode
 - KDGanttView, 34
- setColor
 - KDGanttViewTaskLink, 90
 - KDGanttViewTaskLinkGroup, 95
- setColors
 - KDGanttView, 35
 - KDGanttViewItem, 68
- setColumnBackgroundColor
 - KDGanttView, 35
- setDefaultColor
 - KDGanttView, 36
 - KDGanttViewItem, 68
- setDefaultHighlightColor
 - KDGanttView, 36
 - KDGanttViewItem, 68
- setDisplayEmptyTasksAsLine
 - KDGanttView, 36
- setDisplaySubitemsAsGroup
 - KDGanttView, 36
 - KDGanttViewItem, 69

- setDragDropEnabled
 - KDGanttView, 37
- setDragEnabled
 - KDGanttView, 37
- setDropEnabled
 - KDGanttView, 37
- setEditable
 - KDGanttView, 37
 - KDGanttViewItem, 69
- setEditorEnabled
 - KDGanttView, 38
- setEnabled
 - KDGanttViewItem, 69
- setEndTime
 - KDGanttViewItem, 69
 - KDGanttViewSummaryItem, 82
 - KDGanttViewTaskItem, 85
- setFont
 - KDGanttView, 38
 - KDGanttViewItem, 70
- setGanttMaximumWidth
 - KDGanttView, 38
- setGroup
 - KDGanttViewTaskLink, 90
- setGvBackgroundColor
 - KDGanttView, 38
- setGvVScrollBarMode
 - KDGanttView, 38
- setHeaderVisible
 - KDGanttView, 39
- setHighlight
 - KDGanttViewItem, 70
 - KDGanttViewTaskLink, 91
 - KDGanttViewTaskLinkGroup, 96
- setHighlightColor
 - KDGanttViewTaskLink, 91
 - KDGanttViewTaskLinkGroup, 96
- setHighlightColors
 - KDGanttView, 39
 - KDGanttViewItem, 70
- setHorBackgroundLines
 - KDGanttView, 39
- setHorizonEnd
 - KDGanttView, 39
- setHorizonStart
 - KDGanttView, 40
- setHourFormat
 - KDGanttView, 40
- setIntervalBackgroundColor
 - KDGanttView, 40
- setItemVisible
 - KDGanttViewItem, 70
- setLeadTime
 - KDGanttViewItem, 56
- setLegendHeaderBackgroundColor
 - KDGanttView, 41
- setLegendIsDockwindow
 - KDGanttView, 41
- setListViewText
 - KDGanttViewItem, 71
- setListViewWidth
 - KDGanttView, 41
- setLvBackgroundColor
 - KDGanttView, 41
- setLvVScrollBarMode
 - KDGanttView, 41
- setMajorScaleCount
 - KDGanttView, 42
- setMaximumScale
 - KDGanttView, 42
- setMiddleTime
 - KDGanttViewSummaryItem, 82
- setMinimizeDirection
 - KDGanttMinimizeSplitter, 9
- setMinimumColumnWidth
 - KDGanttView, 42
- setMinimumScale
 - KDGanttView, 42
- setMinorScaleCount
 - KDGanttView, 42
- setNoInformationBrush
 - KDGanttView, 43
- setOpen
 - KDGanttViewItem, 71
- setOrientation
 - KDGanttMinimizeSplitter, 9
- setPaletteBackgroundColor
 - KDGanttView, 43
- setPixmap
 - KDGanttViewItem, 71
- setPriority
 - KDGanttViewItem, 71
- setRepaintMode
 - KDGanttView, 43
- setScale
 - KDGanttView, 43
- setSelected
 - KDGanttView, 43
- setShapes
 - KDGanttView, 43
 - KDGanttViewItem, 72
- setShowHeaderPopupMenu
 - KDGanttView, 44
- setShowLegend
 - KDGanttView, 44
- setShowLegendButton
 - KDGanttView, 44
- setShowListView

- KDGanttView, 45
- setShowMajorTicks
 - KDGanttView, 45
- setShowMinorTicks
 - KDGanttView, 45
- setShowNoInformation
 - KDGanttViewItem, 72
- setShowTaskLinks
 - KDGanttView, 45
- setShowTimeTablePopupMenu
 - KDGanttView, 46
- setStartTime
 - KDGanttViewItemEventItem, 56
 - KDGanttViewItem, 72
 - KDGanttViewSummaryItem, 82
 - KDGanttViewTaskItem, 86
- setText
 - KDGanttViewItem, 73
- setTextColor
 - KDGanttView, 46
 - KDGanttViewItem, 73
- setTimeHeaderBackgroundColor
 - KDGanttView, 46
- setTimelineToEnd
 - KDGanttView, 46
- setTimelineToStart
 - KDGanttView, 46
- setTooltipText
 - KDGanttViewItem, 73
 - KDGanttViewTaskLink, 91
- setUpdateEnabled
 - KDGanttView, 46
- setVisible
 - KDGanttViewTaskLink, 91
 - KDGanttViewTaskLinkGroup, 96
- setWeekdayBackgroundColor
 - KDGanttView, 47
- setWeekendBackgroundColor
 - KDGanttView, 47
- setWeekendDays
 - KDGanttView, 47
- setWhatsThisText
 - KDGanttViewItem, 73
 - KDGanttViewTaskLink, 91
- setYearFormat
 - KDGanttView, 48
- setZoomFactor
 - KDGanttView, 48
- Shape
 - KDGanttViewItem, 60
- shapes
 - KDGanttView, 48
 - KDGanttViewItem, 73
- show
 - KDGanttView, 48
- showHeaderPopupMenu
 - KDGanttView, 49
- showLegend
 - KDGanttView, 49
- showLegendButton
 - KDGanttView, 49
- showListView
 - KDGanttView, 49
- showMajorTicks
 - KDGanttView, 49
- showMinorTicks
 - KDGanttView, 50
- showNoInformation
 - KDGanttViewItem, 74
- showTaskLinks
 - KDGanttView, 50
- showTimeTablePopupMenu
 - KDGanttView, 50
- sizeHint
 - KDGanttView, 50
- startDrag
 - KDGanttView, 50
- startLine
 - KDGanttViewItem, 77
- startLineBack
 - KDGanttViewItem, 77
- startShape
 - KDGanttViewItem, 77
- startShapeBack
 - KDGanttViewItem, 77
- startTime
 - KDGanttViewItem, 74
- subitemIsCalendar
 - KDGanttViewItem, 74
- taskLinkDoubleClicked
 - KDGanttView, 50
- taskLinkGroups
 - KDGanttView, 51
- taskLinkLeftClicked
 - KDGanttView, 51
- taskLinkMidClicked
 - KDGanttView, 51
- taskLinkRightClicked
 - KDGanttView, 51
- taskLinks
 - KDGanttView, 51
- text
 - KDGanttViewItem, 74
- textCanvasText
 - KDGanttViewItem, 77
- textColor
 - KDGanttView, 51

- KDGanttViewItem, [74](#)
- timeHeaderBackgroundColor
 - KDGanttView, [51](#)
- timeIntervallSelected
 - KDGanttView, [51](#)
- timeIntervalSelected
 - KDGanttView, [52](#)
- to
 - KDGanttViewTaskLink, [92](#)
- tooltipText
 - KDGanttViewItem, [75](#)
 - KDGanttViewTaskLink, [92](#)
- Type
 - KDGanttViewItem, [60](#)
- type
 - KDGanttViewItem, [75](#)
- updateCanvasItems
 - KDGanttViewItem, [75](#)
- visible
 - KDGanttViewTaskLinkGroup, [96](#)
- weekdayBackgroundColor
 - KDGanttView, [52](#)
- weekendBackgroundColor
 - KDGanttView, [52](#)
- weekendDays
 - KDGanttView, [52](#)
- whatsThisText
 - KDGanttViewItem, [75](#)
 - KDGanttViewTaskLink, [92](#)
- YearFormat
 - KDGanttView, [15](#)
- yearFormat
 - KDGanttView, [52](#)
- zoomFactor
 - KDGanttView, [53](#)
- zoomToFit
 - KDGanttView, [53](#)
- zoomToSelection
 - KDGanttView, [53](#)