

## **KS-Gantt**

# **Gantt Chart Control for DotNet Winforms**

This document refers to

[http://www.kroll-software.ch/products/ksgantt\\_e.asp](http://www.kroll-software.ch/products/ksgantt_e.asp)

## ***Manual***

### **Introduction**

This document is targeted to a developer with at least medium experience. Basic knowledge about how to use controls in general, use the Integrated Development Environment (IDE) and your programming language is required.

### **Installation**

Unpack KS.Gantt.dll anywhere on your computer.

In the MS-VisualStudio IDE, add the control to your toolbox by browsing for the component file.

### **Drop the controls on a form**

Once available in the toolbox you can drag and drop the Gantt-Control onto a form in the IDE's form designer.

You will see another control called GanttDataGrid.

The Gantt Control can be used standalone or optionally in combination with the GanttDataGrid.

Best way is to put them both in a SplitContainer or a similar kind of panel manager, so that both controls have the same height.

### **Assigning the Gantt to the DataGrid**

Assign the Gantt to the GanttDataGrid by selecting the GanttControl-Property for the GanttDataGrid. That's all to do to let them work together.

## Setting some colors and styles

Now you can start to set some colors and styles in the property explorer. We prefer to set them by code right after the Form's initialization, so that you don't need to select them again when you have to re-insert the control for some reasons. But that's up to you.

Setting column properties for the Gantt-DataGrid as well as adding custom columns must be done in code. This enhances the loading time of the control and prevents upcoming problems, when the properties may change in future versions.

## Setting Date Granularity

Version 3 supports different granularities for the DateTime values. You can select between *Day, half day, quarter day, hour, half hour, quarter hour, 10 minutes, 5 minutes and minute.*

*MinPixelsPerDay* allows you to control how far the user can zoom out.

StartDate Values are always rounded down to the next floor value, EndDate values are always rounded up to the next valid value (depending on granularity) minus 1 millisecond.

## Other Display Options

The control offers some options for display:

TodayLineVisible

TodayLineStyle

MinMaxLineVisible

MinMaxLineStyle

ShowWeeks

ShowWeekends

WeekPrefix

NewTaskName

Different HeaderStyles: [ FormatStyles for every Header Line ]

HeaderBottomLine

BarStyle

RelationLineArrowWidth

RelationLineStyle

BarHeight

RowHeight

PercentBarDeflate (distance between bar-border and the progress bar within the bar)

BarTooltips

GridLines  
GridLineStyle

## Setting default styles for the Gantt bars

DefaultBarFormatStyle  
DefaultGroupLineStyle  
ResourceBarStyle  
ResourceBarStyleMode

These default styles and settings are used when new items are added. You can later set different styles for each individual item.

## Adding Tasks, Groups and Resources

Please see the sample application on how to add some items.  
The GanttControl offers some basic functions to add items:

AddTask(..)  
AddMilestone(..)  
AddGroup(..)  
AddItem2Group(..)  
AddResource(..)

These functions return the created object just to keep a reference or to modify some properties later. Beside these basic functions you can always create the objects by yourself and add them directly to the collections, discussed later.

## The Data Object Model

Now it's time to understand the Gantt's data object model.

The GanttControl holds three main collections of objects:

Tasks : List<TaskItem>  
Groups : List<GroupItem>  
Resources : List<ResourceItem>

SelectedItems holds the items that are selected, and is still valid after editing operations.

## The Object Classes

TaskItem, GroupItem and ResourceItem all inherit from GanttItem, which is a base class and is never used. Some Events return a GanttItem that can be casted to the desired object.

### **GanttItem:**

StartDate  
EndDate  
ParentItem  
Text  
Description  
RowIndex  
Tag : free to use, not used internally  
Selected  
Visible  
Modified

#### *Read-Only:*

ItemType : returns the type of an Item. You can check this before casting to a inherited type  
Key : A unique key is created with the item  
Duration

You can add more and any data with

SetProperty(string key, object value)  
GetProperty(string key)

This data stored with SetProperty can be displayed and edited in the GanttDataGrid by setting up a grid column with the same key.

### **TaskItem:**

TaskItem inherits from GanttItem and so it has all of it's properties and methods plus..

PercentDone  
EarliestStartDate  
IsMilestone  
Priority  
Status  
Icons  
Dependencies (to other TaskItems)  
Resources (that are assigned to this item)  
IsResourceHeaderItem : this task is a resource header shown in the Resource-View

#### *Read-Only:*

IsOverDue

Methods:

AddDependence(..)

RemoveDependence(..)

AddResource(..)

RemoveResource(..)

### **GroupItem:**

GroupItem inherits from GanttItem and so it has all of it's properties and methods plus..

StartDate : is read-only and is calculated from the containing tasks (ChildItems)

EndDate : same as above

ChildItems : the TaskItems or GroupItems that belong to this group, read-only array

*Read-only:*

ChildCount

ChildCountVisible

IsExpanded

Methods:

AddChild()

RemoveChild()

ClearChildItems()

Expand()

Collapse()

RecalculateDates()

### **ResourceItem:**

ResourceItem inherits from **GroupItem** and so it has all of it's properties and methods plus..

StartDate, EndDate for ResourceItem are used internally and have no special meaning outside

Role

CostPerDay

### ***Relations between Items***

#### **TaskRelation:**

does not inherit from GanttItem

Task

TaskRelationMode

DelayDays

#### **TaskRelationCollection:**

A List<TaskRelation> that holds references to the related tasks

### **ResourceRelation:**

does not inherit from GanttItem

Resource

Unit : percent (0..1) how this resource is involved with the task

Responsible

### **ResourceRelationCollection:**

A List<ResourceRelation> that holds references to the related resources

## **Main Item Collections**

The GanttControl holds three different Collections to the items:

Tasks

Groups

Resources

These collections are created on initialization of the GanttControl.

They contain methods for adding and removing items and to access some cumulated properties  
The GanttControl itself has some redundant high-level functions to add items.

Some general methods that these collections offer are

FindItemByKey(string key)

Modified : Iterates to all items in the collection

ResetModified()

ResetSelected()

FinalizeDeSerialization(Gantt gantt)

## **References of Items and Collections to the GanttControl**

Every Item and the item-collections holds a reference to the GanttControl that is created on runtime.  
The GanttControl adds this reference by itself whenever an item is added to a collection, so you never need to bother about it.

All properties that you can access will keep track of the GanttControl's state. E.g. setting the 'selected' property for an item will update the SelectedItems collection of the GanttControl automatically. Changing properties and calling some functions will cause the GanttControl to invalidate (redraw) or to update it's state in some other way.

## ***The GanttControl at Runtime***

### **Editing**

The GanttControl supports interactive editing of items by the user.

**Enable/Disable editing** with

AllowEdit : Editing in general, includes the following..

AllowItemMove

AllowItemResize

AllowItemDelete (when the user pressed Del-Key)

AllowItemAddNew (when the user double-clicks into an empty area)

### **ViewMode : Switch to Resource-View**

The control starts up in the Task-View.

You can switch to the Resource-View with

ViewMode = ResourceView

The ResourceView is a calculated view where you can see the load of each resource in different colors. By default, red shows that the resource is overloaded. You can change these colors with

ResourceStandardColor

ResourceOverloadColor

ResourceUnderloadColor

### **Undo/Redo**

The GanttControl supports multi-level undo and redo

CanUndo

CanRedo

Undo()

Redo()

When you want to take a snapshot of the current state by code, just call  
HistorySetUndoPoint()

## Serialization/DeSerialization to file

The GanttItem and inherited objects and all FormatStyles support serialization.

You can save and load the data and bar-styles to a binary file with

`SerializeData(string fileName)`

`DeSerializeData(string fileName)`

However it is no guaranteed that this format can be read (is compatible) with future versions.

You should use these functions only for temporarily saving and restoring states of the data.



## GanttControl Events

The Gantt throws some events to control the user interaction.

Some events provide the corresponding GanttItem, some allow to cancel the action.

ItemClick

ItemDoubleClick

ItemHover

ItemSelected

SelectionChanged

BeforeDelete

DeletingItem

AfterDelete

BeforeAdd

AfterAdd

AfterDrawHeader

BarAreaTopChanged

ScrollChanged

BarHeightChanged

RowHeightChanged

ViewModeChanged

UndoRedoCalled

IconSettingsChanged

ItemsChanged

ItemsMoved

ItemsResized

AfterAutoMoveItems

AfterAutoArrangeRows

## Auto Arranging of Items

After adding or removing items, and after editing operations, the GanttControl must arrange the items in rows and in the time-line to keep a consistent state.

Reordering the rows after such operations is crucial so that the data can be displayed in the corresponding GanttDataGrid control.

Moving the items on the time-line fulfills the consistent state of dependencies (relations) between tasks and their rules.

**These functions are called internally whenever needed.**

AutoArrangeRows() re-arranges the rows order considering the groups  
AutoMoveItems() shifts the items on the time-line

You can disable this behavior with  
AutoArrangeRowsOnUpdate  
AutoMoveItemsOnUpdate

You should never disable AutoArrangeRowsOnUpdate when the GanttControl is bound to a GanttDataGrid.

If you want so, you can use the GanttControl without all of it's internal logic by disabling these features.

SkipWeekendsOnAutoMove : controls, whether items can start on weekends after moving.

## ***Other Functions***

ZoomToFit()  
Zooms to show all items, if possible

CenterNow()  
Centers to today

ItemByRowIndex(int RowIndex)  
returns the item found at the given RowIndex or null

FindItem(int x, int y)  
Returns an item at the given coordinate

Modified  
checks if any item was modified

ResetModified()  
resets all items to modified = false

BarAreaTop  
the top coordinate of the bar area

BarAreaHeight  
Height of the bar area

ScrollOffsetY  
Offset of the scroll position

MaxRowIndex  
the maximum row index

CalcParams()  
Calculates the MinDate, MaxDate and MaxRowIndex after items are changed in TaskView. This is called internally whenever items are changed in Task-View.

CreateResourceBars()

Re-Creates the resource bars in Resource-View. This is called internally whenever items are changed in Resource-View.

## **Suspending item layout on bulk operations**

SuspendItemLayout()

ResumeItemLayout()

When changing many items at once, you should call this to prevent unnecessary repaints and so to speed up operation.

You must ensure that ResumeItemLayout is reliably called after SuspendItemLayout.

## ***Calculation of Project Cost***

Call CalcProjectCost() to calculate the total cost for the resources depending on  
ResourceItem.CostPerDay and  
ResourceRelation.Unit

## **The GanttDataGrid Control**

The GanttDataGrid can be bound to the GanttControl by setting the property 'GanttControl'. Both controls must have the same top and height.

Setting up formats and styles is very equivalent as for the GanttControl.

The GanttDataGrid provides two collections to Columnsettings

TaskColumns

ResourceColumns

both of type GanttDataColumnCollection.

With these properties you can access and change the columns of the grid.

### **GanttDataColumn:**

Position

Key

Caption

Visible

Width

EditType

CanEdit  
AllowResize  
AutoMinWidth  
ValueList  
DisplayFormatString  
EditFormatString  
TextAlignment  
Show/Edit-ValueForTaskItem  
Show/Edit-ValueForGroupItem  
Show/Edit-ValueForResourceItem  
Show/Edit-ValueForMilestone

Methods:  
AddValueListItem(..)

Columns can be edited with the following edit types:

Text  
Date  
Time  
DropDownList  
DropDownCombo  
CheckBox  
NumericUpDown

The DropDownList and DropDownCombo are filled with the values from ValueList.

## **GanttDataGrid Events**

BeforeEdit  
UpdatingItem  
ItemUpdated

ItemDoubleClick  
ItemRightClick

SelectedRowIndexChanged

## Custom Drawing Events

There are two events where you can do your own custom drawings:

*BeforeDrawHeader* is called after the background is erased with the background color and just before any header painting was started.

*AfterDrawHeader* is called after all headers were drawn, just before the gantt-bars are drawn.

----- End of document -----

Copyright © by Kroll-Software, Höfligasse 11, CH - 6460 Altdorf  
Phone +41-41-5351767 | Email [info@kroll-software.ch](mailto:info@kroll-software.ch)