

Plugin Gantt Chart

Introduced in Tiki19

Use this wiki plugin to help you manage projects within Tiki, where project data is stored in Tiki trackers. You will be able to draw Gantt chart diagrams, and edit them easily to have your gantt chart updated while the project life evolves changing project details, internal deadlines, etc. Please see: [jQuery Gantt editor](#).

Parameters

Create and display a gantt graphic using tracker data
*Introduced in Tiki 19. Required parameters are in **bold**.*

Go to the source code

Preferences required: `wikiplugin_ganttchart`

Parameters	Accepted Values	Description	Default	Since
begin		Permanent name of the field to use for event beginning		
end		Permanent name of the field to use for event ending		19
depends	word	Permanent name of the field to use for row dependency		19
level	word	Permanent name of the field to use for row level		19
name	word	Permanent name of the field to use for row name		19
order	word	Permanent name of the field to use for row number order		19
status	word	Permanent name of the field to use for row status		19
trackerId	int	Tracker to search from	0	19
code	word	Permanent name of the field to use for row code		19
canDelete	none	Flag field that permit to delete or not tasks		19
description	word	Permanent name of the field to use for row description		19
effort	word	Permanent name of the field to use for effort ending, values in milliseconds		19
endIsMilestone	none	Flag field to end is milestone		19
ganttId	word	Gantt chart ID value to filter		23
ganttIdField	word	Permanent name of the field to use to filters the ganttId		23
canWriteOnParent	none	Flag field to write or not in parent tasks		19
progress	word	Permanent name of the field to use for row progress, values between 0-100		19
resourceId	word	Permanent name of the field to use for resources ending		19
roleId	word	Permanent name of the field to use for roles ending		19
startIsMilestone	none	Flag field to start is milestone		19

canWrite	none	Flag field to write or not in tasks		19
canDuplicate	y n	Flag field to allow duplicate tasks	y	23

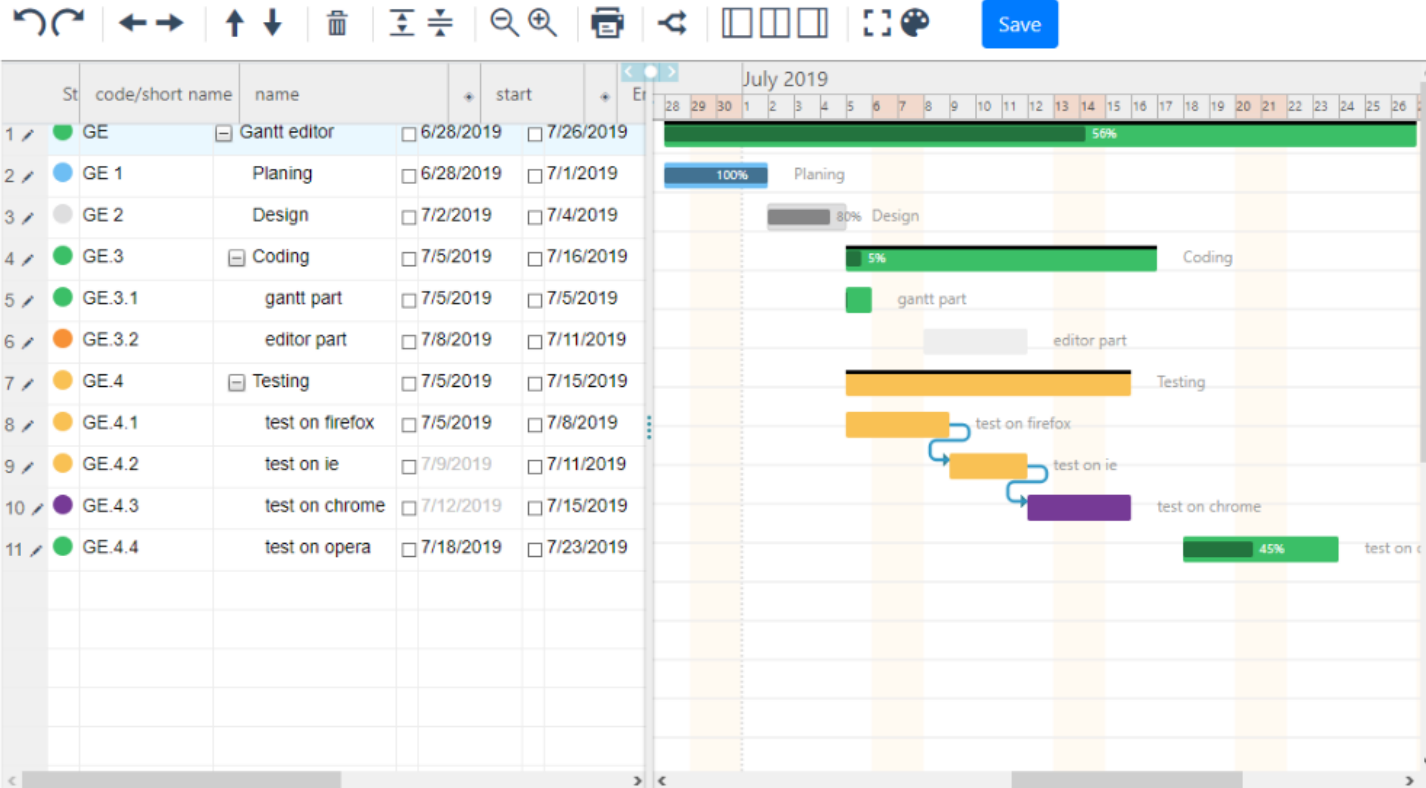
Get Started

As mentioned above, the project data used to create the Gantt charts are stored in Tiki trackers. Create the Tracker with all the specific fields or use this Profile as a template, adjusting it to your own needs. Start with the Plugin Gantt Chart by adding the code to a Wiki page once you have created a tracker.

```
{ganttchart trackerId="1" name="trackerganttTaskName" begin="trackerganttStartDate"
end="trackerganttEndDate" resourceId="trackerganttResponsible" roleId="trackerganttRole"
status="trackerganttStatus" progress="trackerganttProgress" canWrite="true" canDelete="true"
canWriteOnParent="true" description="trackerganttDescription" code="trackerganttCode"
effort="trackerganttEffort" resourceId="trackerganttResourceId" order="trackerganttOrder"
level="trackerganttLevel" depends="trackerganttDependencies"}
```

Overview

The previous code will give the result this overview:



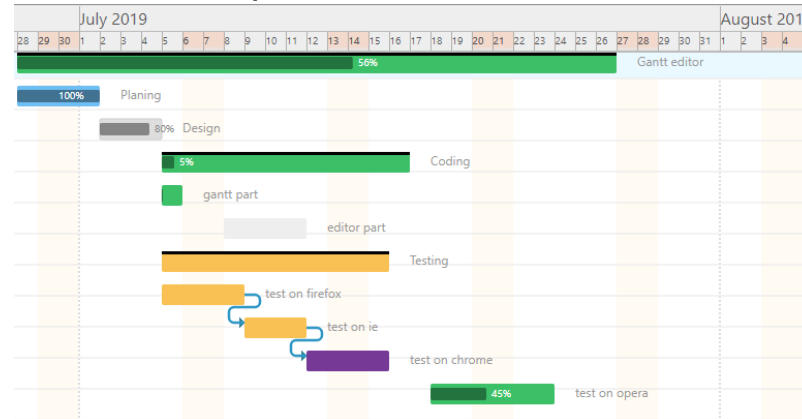
Click to expand

Key Features

The Gantt Chart Plugin allows you to manage a complete project of any size, to divide it into tasks, to associate resources with each one with its role. When managing the project, you can create a task hierarchy, create dependencies between tasks, determine milestones, highlight the critical path, modify resources, change time scheduling, etc.

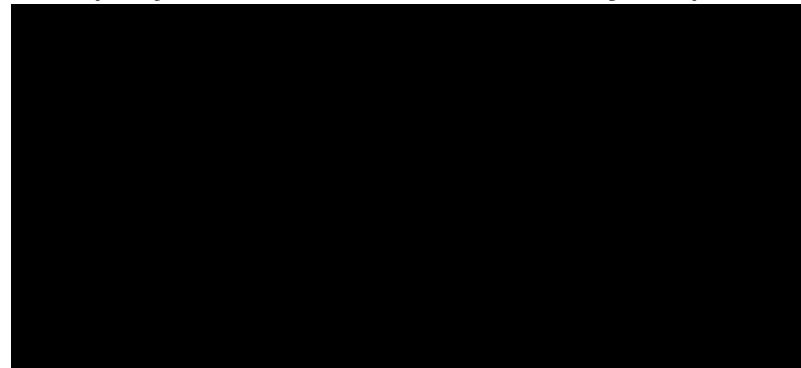
To quickly build Gantt charts, use the undo, redo, un-indent task, indent task, move up, move down, delete, EXPAND_ALL, COLLAPSE_ALL, zoom out, zoom in, Print, CRITICAL_PATH, control options to handle the different task display modes, fullscreen, task status color control and the Save button.

With a simple click, set the critical path; remove a task; switch to fullscreen mode, etc.



Click to expand

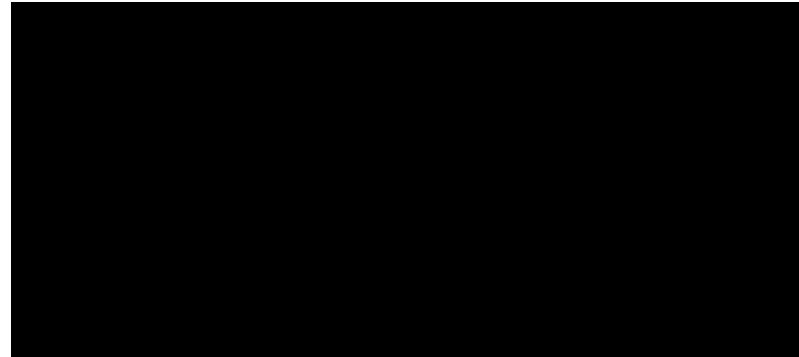
Organise projects into tasks linked by dependencies



Click to expand

Explore Plugin Gantt Chart

Changing a task



Click to expand

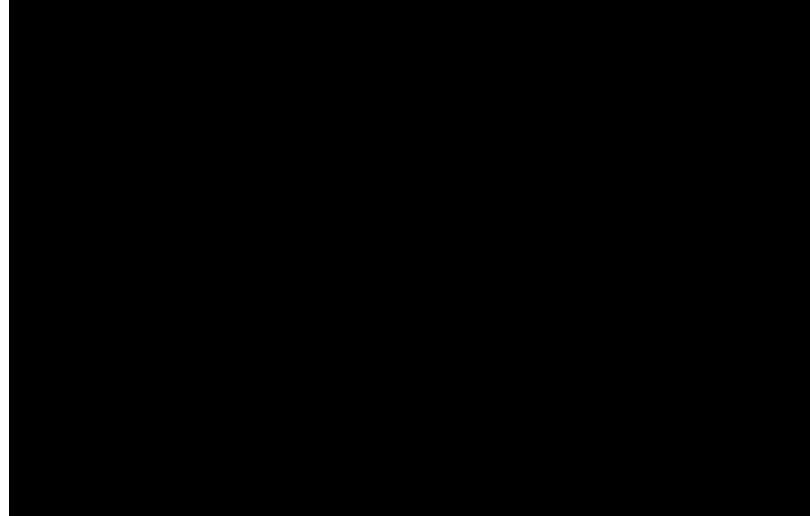
a few clicks change the status, code, name, start date, end date, duration or progress of the task just by clicking on the element you need to update.

code/short name	name	
<input type="text" value="GE"/>	<input type="text" value="Gantt editor"/>	
start <input type="checkbox"/> is milestone	End <input type="checkbox"/> is milestone	Days
<input type="text" value="6/28/2019"/>	<input type="text" value="7/26/2019"/>	<input type="text" value="21"/>
status	progress	
<input type="text" value="active"/>	<input type="text" value="56"/>	
Description		
<input type="text" value="The person in charge to get things done by all team members"/>		

Click to expand

*When you want to modify a task, click on the edit icon or on the diagram that represents a task, a window will open **Task Editor**, change the code, the name of the task, define milestones, add a description of the task, etc.*

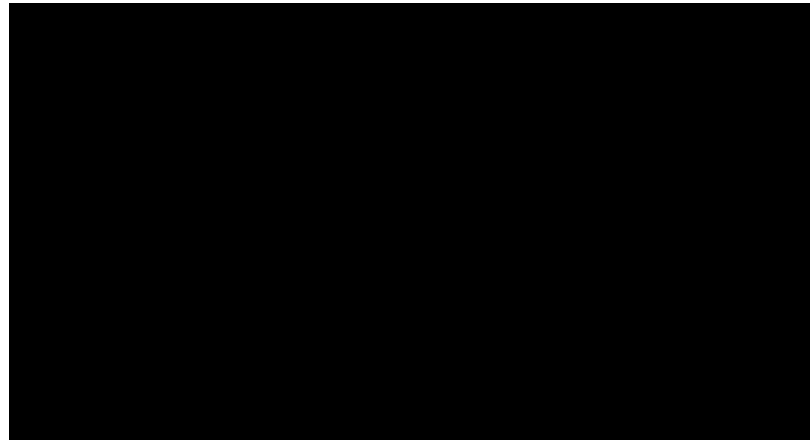
Moving a task



Click to expand

*After editing a task, use the «**move up**» and «**move down**» controls to move a task from bottom to top or from top to bottom.*

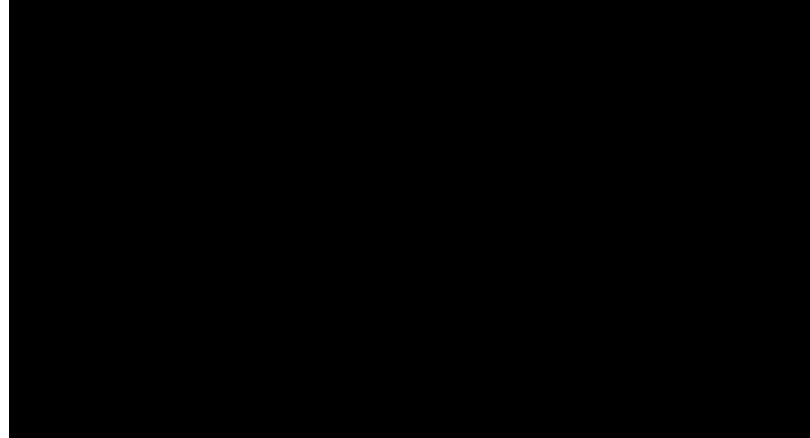
Creating dependencies



Click to expand

Very easy to create dependencies between two or more tasks by simply dragging the mouse. Place the mouse over the graph, once the two points have been shown, select one and dragging it to another task.

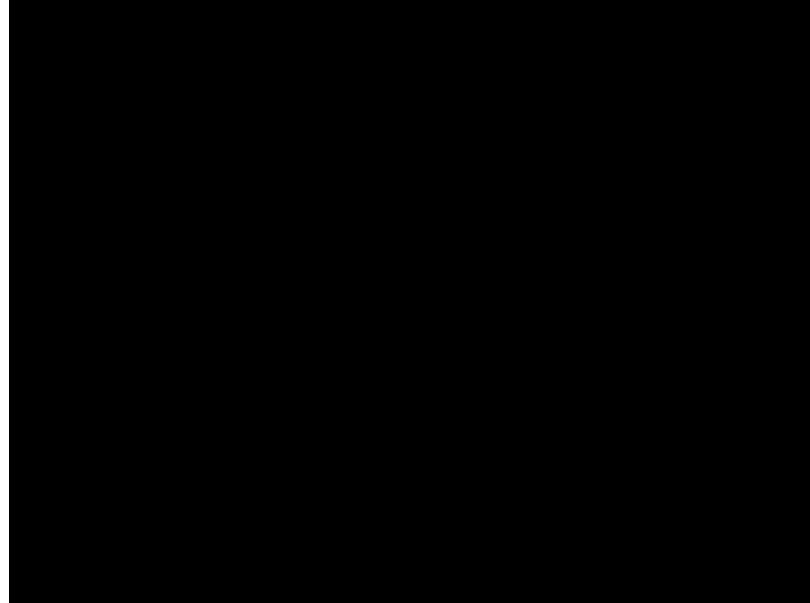
Changing task duration



Click to expand

Easily change the duration of tasks with just a simple drag of the mouse.

Adjusting the chart preview



Click to expand

Adjust the zoom option to view the charts according to day, week, month, semester, and year.

Some extra details on some fields

- **id:** used to synch data with the server. If the id is supplied by the server it will be untouched. Tasks created client side will acquire a temporary id starting with "tmp_"
- **level:** it is the depth (the indentation) of a task in the Gantt structure. The root task is at level 0 (zero), its children at level 1 (one) and so on. Levels must be consistent with the project structure: you can't have a task of level $n+1$ if you don't have a task of level n above on the list
- **start, end:** are expressed in milliseconds. "start" is set to the first millisecond of the day, "end" is set to the last millisecond of the day.
- **duration:** is always in working days. If there are holidays on the calendar the end date will take it into account. Actually the end date is always recomputed using "start" and "duration", and it is supplied for comfort
- **startIsMilestone, endIsMilestone:** booleans. Once set to true, task' start/end can't move accidentally. You always can change dates directly on the task, but not by acting on children or predecessors.
- **depends:** a string comma delimited containing indexes of tasks on which this task depends. Multiple dependencies are supported. Only the finish-to-start dependency type is supported (other types can be workarounded by introducing intermediary brother tasks or children). It is possible to specify a "lag" in days by using a ":". E.g.: 7:3,8 means that the task will start 3 days after task 7 is completed and task 8 is

completed.

- **status:** this is a string representing the status of the task. Allowed statuses are: `STATUS_ACTIVE`, `STATUS_DONE`, `STATUS_FAILED`, `STATUS_SUSPENDED`, `STATUS_UNDEFINED`. As stated before, task statuses allow to use your project like a sort of workflow: e.g.: if “task b” depends on “task a”, “task b” will remain in “`STATUS_SUSPENDED`” until “task a” will pass from “`STATUS_ACTIVE`” to “`STATUS_DONE`”. For the complete status transition rules see below.
- **progress:** a number that specifies progress: 0% none 50% half way and so on. In some cases 123% can be a meaningful value for progress, so there are no constraints.

Status transition rules (aka project workflow)

Task statuses are a key feature in this plugin.

When you close some tasks (change status to “completed”) you will see how dependent tasks/children will change their status according to the following rules:

- **any status -> STATUS_DONE:** may activate dependent tasks, both suspended and undefined. Will set to done all descendants.
- **STATUS_FAILED -> STATUS_DONE:** do nothing if not forced by hand.
- **STATUS_UNDEFINED -> STATUS_ACTIVE:** all children become active, if they have no dependencies.
- **STATUS_SUSPENDED -> STATUS_ACTIVE :** sets to active all children and their

descendants that have no inhibiting dependencies.

- **STATUS_DONE -> STATUS_ACTIVE**: all those that have dependencies must be set to suspended.
- **STATUS_FAILED -> STATUS_ACTIVE**: nothing happens: child statuses must be reset by hand.
- **any status-> STATUS_SUSPENDED**: all active children and their active descendants become suspended. when not failed or forced
- **any status-> STATUS_UNDEFINED**: all active children and their active descendants become suspended. when not failed or forced.
- **any status-> STATUS_FAILED**: children and dependants are set to failed.

Example

- See Sample Gantt Chart

Improvements in Tiki23

Added gantt items duplication within same tracker since Tiki23:

- Use ganttId to indicate the value to filter items;
- User ganttIdField to indicate the field perm_name that holds the value above;
- Use canDuplicate to indicate if it can duplicate or not (both ganttId and ganttIdField need to have values)
- When duplicating items, a new ganttId is chosen and also the option to update the existing wikiplugin_ganttchart to display the new duplicated items.
- If user does not have permissions to create new tracker items, the duplication is not available.

Related pages

- [Trackers](#)
- [jQuery Gantt editor](#)

Related Profile

- GanttChart

Aliases

- Gantt Chart
- Gantt
- PluginGantt
- Plugin Gantt
- Plugin GanttChart
- Plugin Gantt Chart