Manual for org-gantt
v0.1

Bernhard Schmitz

June 1, 2021

Contents

1 Introduction

org-gantt defines a custom dynamic block for org mode that can create gantt charts from org files, using headlines and their schedules, deadlines, effort and clock values. The latex package pgfgantt is used to create the gantt charts, thus they are only visible on export into a pdf file. This manual is about org-gantt. Many options for creating the gantt charts that are specific to pgfgantt are explained in its manual an not further explained here.

2 Installation and usage

Open org-gantt.el and run eval-buffer. Put the following line near the top of your org mode file:

```
#+LATEX_HEADER: \usepackage{pgfgantt}
```

3 User Guide

The org-gantt block is delimited by the following lines:

```
#+BEGIN: org-gantt-chart
#+END:
```

Using these delimiters without any parameters will create a custom dynamic block spanning the entire current file. As this document is written in org mode, all examples will use an :id parameter to limit the gantt chart to a specific subtree.

Customizable variables can be set in the standard form of org custom dynamic block variables, i.e. with `:variable-name variable value` directly after the `#+BEGIN: org-gantt chart`. Those variables will only affect the block they are set. In almost all cases it is possible to customize the default value of those variables by customizing the emacs variable `org-gantt-default-variable-name`, which can be found in the customize-group `org-gantt`. For the remainder of this manual, the possibility of setting the default value is not explicitly mentioned for each variable. An overview of customizable variables can be found in section 4.1.

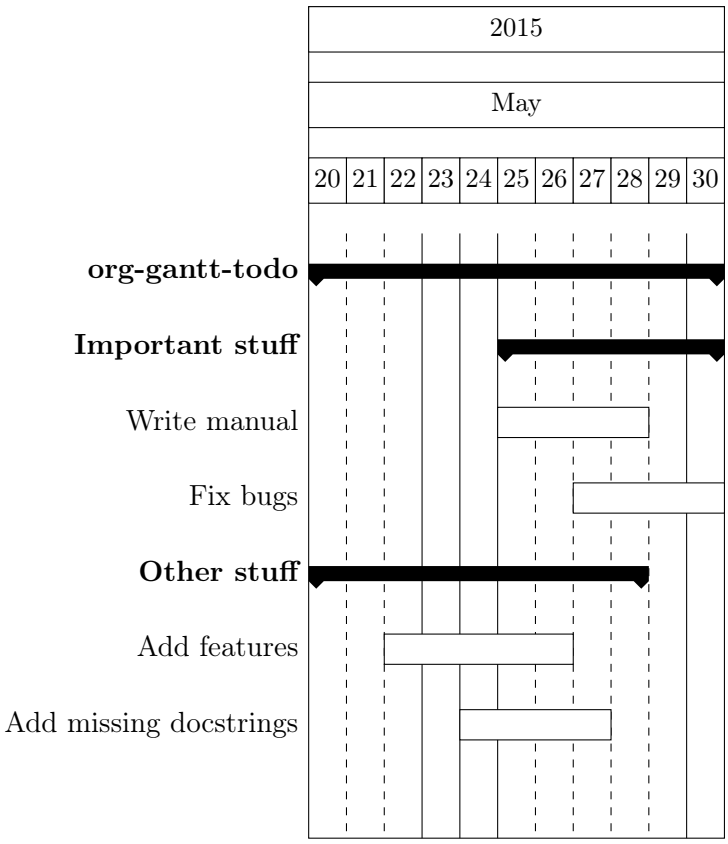
3.1 Deadlines and Schedules

The simplest way of creating a gantt chart is to use hardcoded deadlines and schedules, which are created via *org-deadline* (C-c C-d) and *org-schedule* (C-c C-s)

```
* org-gantt-todo
 :PROPERTIES:
 :ID:       todo-deadlines-schedules
 :END:
** Important stuff
*** Write manual
    SCHEDULED: <2015-05-25 Mo> DEADLINE: <2015-05-28 Do>
*** Fix bugs
   SCHEDULED: <2015-05-27 Mi> DEADLINE: <2015-05-30 Sa>
** Other stuff
   SCHEDULED: <2015-05-20 Mi> DEADLINE: <2015-05-28 Do>
*** Add features
    SCHEDULED: <2015-05-22 Fr> DEADLINE: <2015-05-26 Di>
*** Add missing docstrings
    DEADLINE: <2015-05-27 Mi> SCHEDULED: <2015-05-24 So>

#+BEGIN: org-gantt-chart :id "todo-deadlines-schedules"
#+END
```

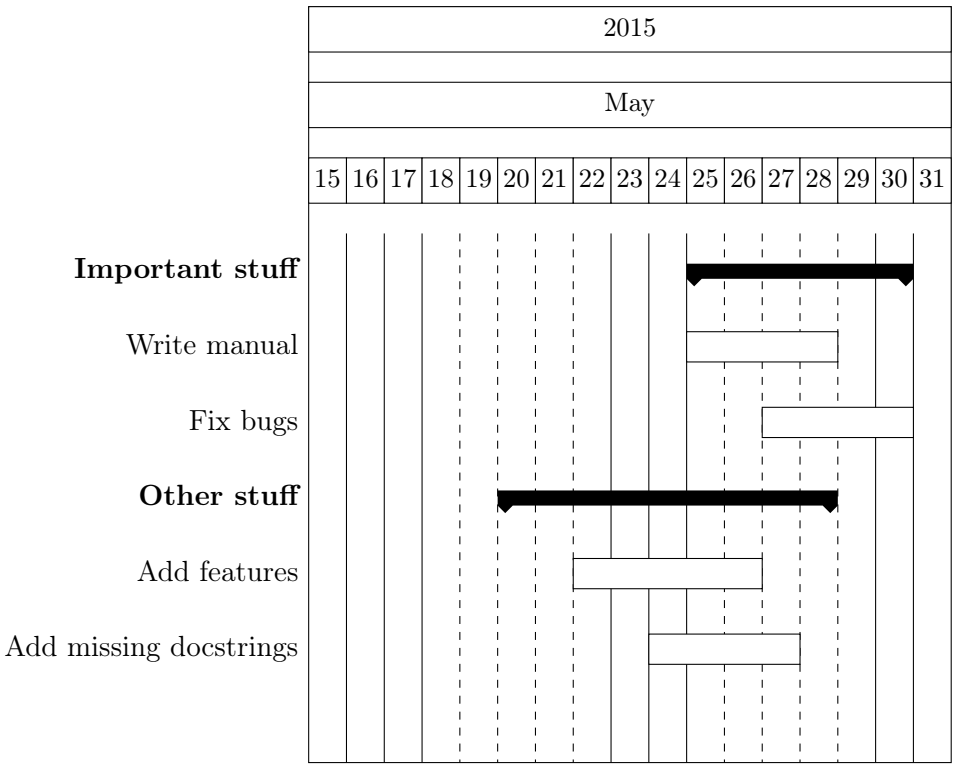
Updating the dynamic block will result in the following gantt chart:



As you can see, org-gantt automatically calculates the deadlines and schedules for superheadlines that do not already have them. Existing dates are not overwritten. Additionally, org-gantt automatically calculates the start and end date for the chart. This can easily be overwritten by using the parameters `:start-date` and `:end-date`. In the previous example, the gantt chart included the top headline, for which the id was given. This is not always desirable, as it results in a single gantt group spanning the entire chart. Set the parameter `:use-id-subheadlines` to `t` in order to use only the subheadlines of the given id. Thus, the dynamic block

```
#+BEGIN: org-gantt-chart :id "todo-deadlines-schedules" \
 :use-id-subheadlines t \
 :start-date "2015-05-15" :end-date "2015-05-31"
#+END
```

creates the following gantt chart:



3.2 Effort

Instead of directly writing down schedules and deadlines, you can use effort estimates. Whenever an effort estimate is present, either the schedule or the deadline can be omitted, and will be automatically calculated from the given effort.

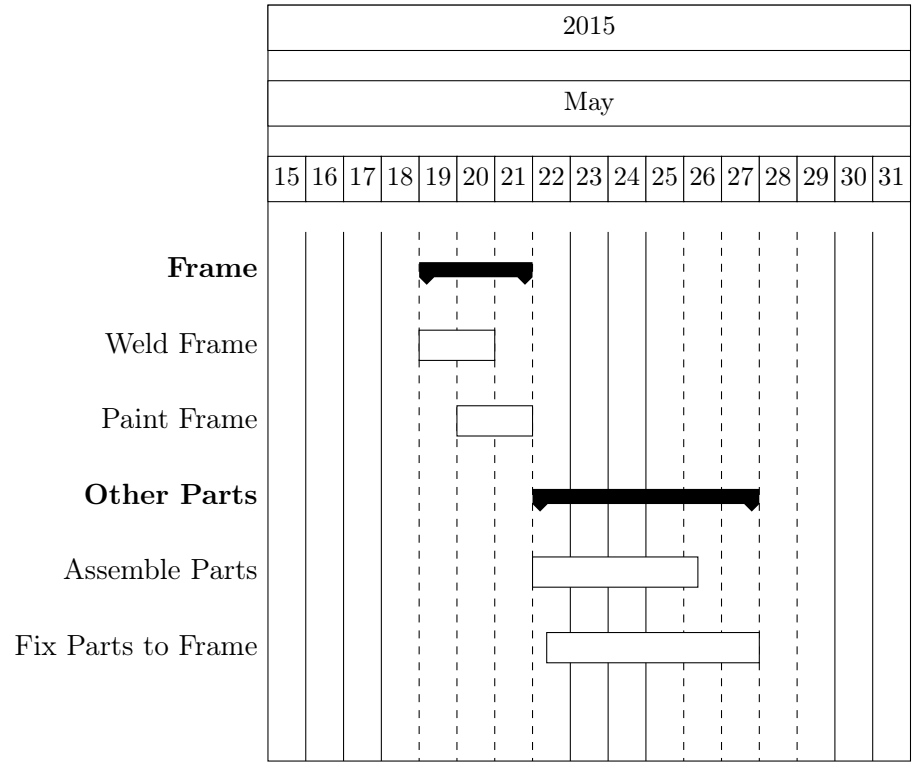
3.2.1 Simple Effort Estimates

Deadline and schedule calculation via effort estimates is done in a sophisticated manner: It takes into account the hours of work per day, and work-free weekends. This works in both directions, meaning that if `:hours-per-day` is `8` (the default value), an effort estimate of `2d` is equivalent to `16:00` and both will result in a full two days on the gantt chart:

```
* COMMENT Build Bike
   :PROPERTIES:
   :ID:       effort-src
   :END:
** Frame
*** Weld Frame
   SCHEDULED: <2015-05-19 Di>
    :PROPERTIES:
    :Effort:   16:00
    :END:
*** Paint Frame
```

```
      DEADLINE: <2015-05-21 Do>
      :PROPERTIES:
      :Effort:   2d
      :END:
  ** Other Parts
  *** Assemble Parts
      SCHEDULED: <2015-05-22 Fr>
      :PROPERTIES:
      :Effort:   19:00
      :END:
  *** Fix Parts to Frame
      DEADLINE: <2015-05-27 Mi>
      :PROPERTIES:
      :Effort:   3d 5:00
      :END:
#+BEGIN: org-gantt-chart :id "effort-src" :use-id-subheadlines t \
:start-date "2015-05-15" :end-date "2015-05-31"
#+END
```

The resulting gantt chart:



Note that org-gantt correctly displays efforts that are not full days. Additionally, weekend days are not counted as workdays. When calculating deadlines and schedules, weekend days are spanned in addition to the estimated effort. Days that are counted as work free (weekends by default) can be set by the variable `:work-free-days`, which is a list integers between 0 (sunday) 6 (saturday).

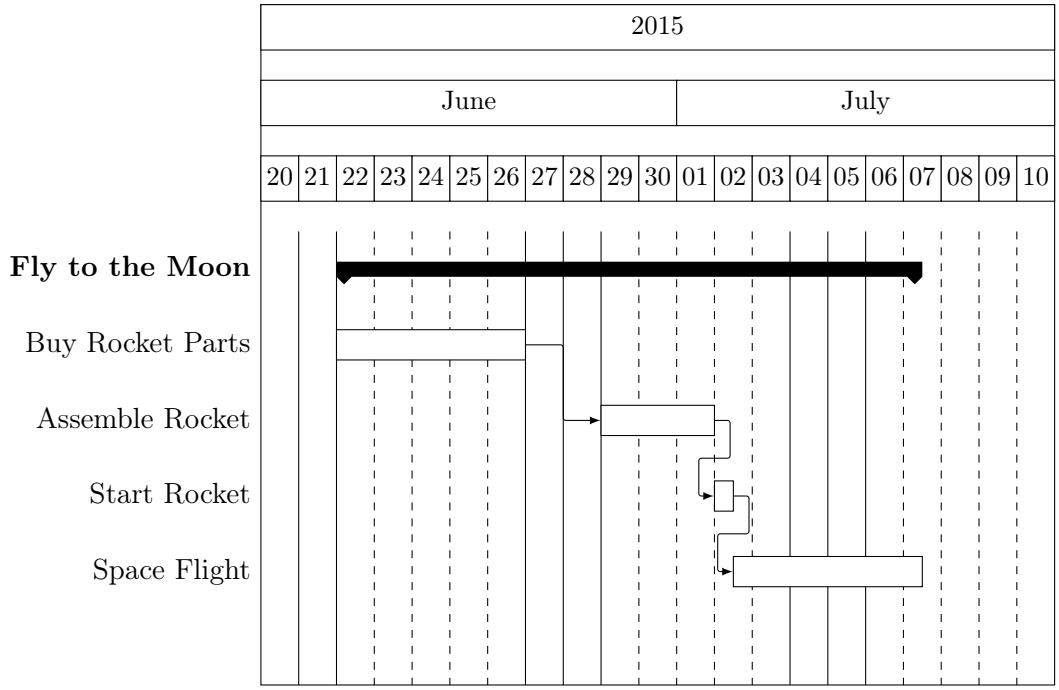
3.2.2 Effort Estimates and Ordered Headlines

Headlines can be marked as ordered, meaning that a subtask can only be started once the previous subtask has been finished. Due to this dependency even more deadlines and schedules can be calculated automatically: If every subtask has an effort estimate, a single deadline or schedule is enough to calculate all other times. The deadline or schedule can be attached to the supertask or to any of the subtasks.

```
  * COMMENT Ordered Task
    :PROPERTIES:
    :ID: ordered-space-src
    :ORDERED:  t
    :END:
  ** Fly to the Moon
     SCHEDULED: <2015-06-22 Mo>
     :PROPERTIES:
     :ORDERED:  t
     :END:
  *** Buy Rocket Parts
      :PROPERTIES:
      :Effort:   5d
      :END:
  *** Assemble Rocket
      :PROPERTIES:
      :Effort:   3d
      :END:
  *** Start Rocket
      :PROPERTIES:
      :Effort:   4:00
      :END:
  *** Space Flight
      :PROPERTIES:
      :Effort:   3d
      :END:

#+BEGIN: org-gantt-chart :id "ordered-space-src" :use-id-subheadlines t  \
:start-date "<2015-06-20 Sa>" :end-date "<2015-07-10 Fr>"
#+END
```

Results in the following gantt chart:

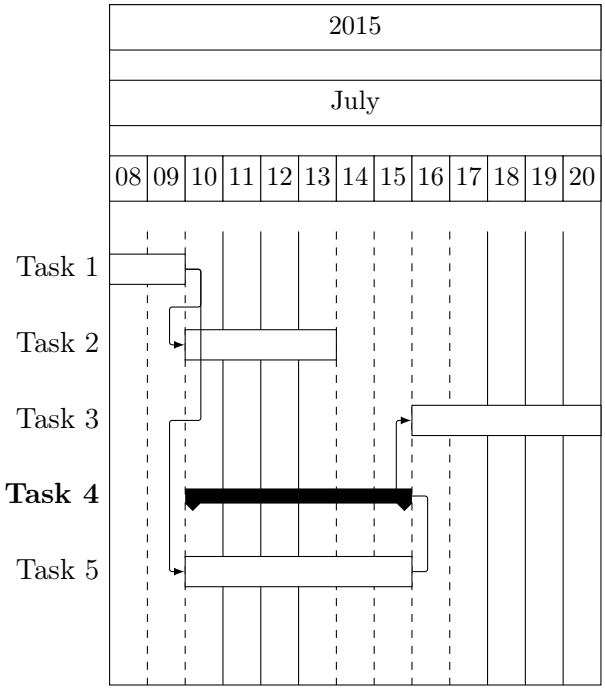


This chart demonstrates that the start of a follow-up task is shifted to the following Monday, if the previous task ends exactly at the start of the weekend. Additionally, it demonstrates how follow-up days start on the same day, if `hours-per-day` still leaves time during that day, but start on the next day, if the previous task takes the entirety of the previous day.

3.3 Manually Linking Headlines

In addition to linking headlines via the `:ORDERED:` property, headlines can be linked manually. In contrast to ordered subheadlines, which will link direct sibling headlines, manual links can be created to arbitrary headlines.

In order to generate a manual link, create a property with a key that is in `:linked-to-property-keys:` (or in the default variable `org-gantt-default-linked-to-property-keys`, which is set to `:linked-to` by default). Any values of these property keys are interpreted as a comma separated list of ids. Links between the headline with the property and headlines with the given ids are created, and the start time of the linked headlines are computed from the end time of the original headline, if the linked-to headlines do not already have start times set.



3.4 Progress

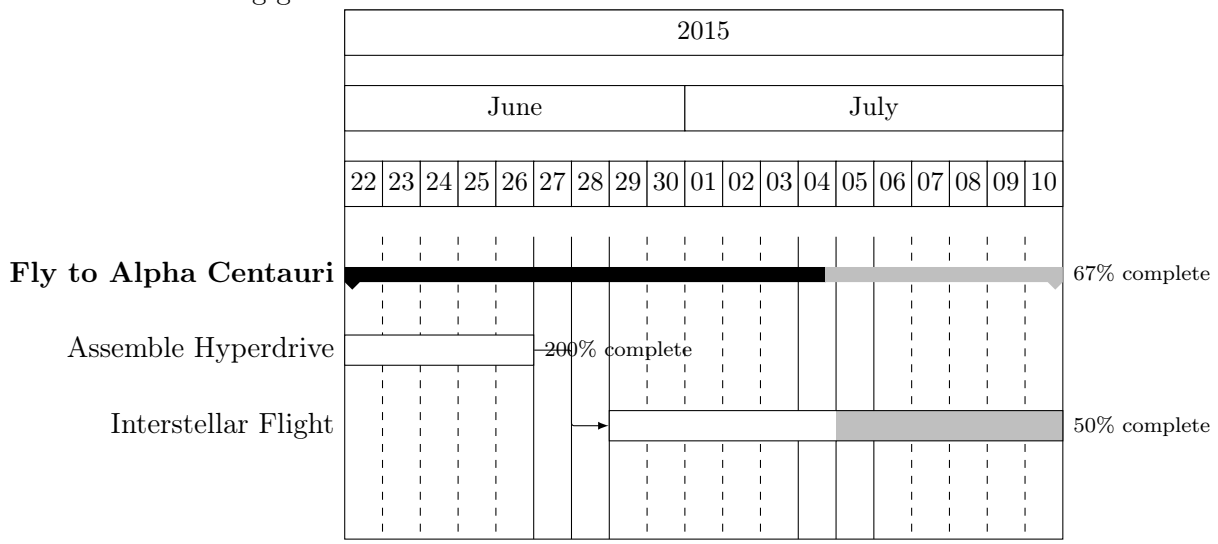
org-gantt can use to calculate the progress on each item. If clocking is used, the progress is currently simply calculated as the ratio of clocked time to estimated time, and thus does not constitute a realistic estimation of the real progress of a specific task. Nevertheless, it can be used to visualize progress on specific tasks. If progress cookies are used, their value is translated directly into a progress value, regardless of whether the cookie uses percentage or fractional display. To show progress, use the parameter `:show-progress`. Setting it to `always` will show the progress on all tasks. Setting it to `if-exists` will show progress only for those tasks with a clocksum, i.e. tasks that have been clocked, or that have subtasks that have been clocked.

```
* COMMENT Using Effort
   :PROPERTIES:
   :ID: clock-space-src
   :END:
** Fly to Alpha Centauri
   SCHEDULED: <2015-06-22 Mo>
   :PROPERTIES:
   :ORDERED:  t
   :END:
*** Assemble Hyperdrive
    CLOCK: [2015-06-22 Mo 08:00]--[2015-06-25 Do 16:00] => 80:00
    :PROPERTIES:
    :Effort:   5d
    :END:
```

```
  *** Interstellar Flight
      CLOCK: [2015-07-02 Do 10:00]--[2015-07-04 Sa 02:00] => 40:00
      :PROPERTIES:
      :Effort:   10d
      :END:

#+BEGIN: org-gantt-chart :id "clock-space-src" :use-id-subheadlines t \
:show-progress if-exists
#+END
```

Creates the following gantt chart:



This chart demonstrates that the progress calculation of org-gantt does not use progress of larger than 100% on subtasks for the calculation of the progress of supertasks, as this could lead to the impression that a supertask is (almost) finished, even if the user took too long on a single subtask, whereas other subtasks are left unfinished. To override this behaviour, set the parameter `:calc-progress` to `use-larger-100`. In this case, supertasks will use the full clocked time of each subtask for the calculation of its progress value:

```
#+BEGIN: org-gantt-chart :id "clock-space-src" :use-id-subheadlines t \
:show-progress if-clocksum :calc-progress use-larger-100
#+END
```

Creates the following gantt chart:

The variable `:progress-source` can influence whether progress cookies or clocksum values are used for calculating the progress. `cookie-clocksum` (the default), prefers cookies, but also uses clocksums. `clocksum-cookie` prefers clocksums, but also uses cookies. `clocksum` and `cookie` will make org-gantt use only clocksums or cookies, respectively.

3.5 Styling

You can use all the styling parameters available in pgf-gantt (see the pgf-gantt manual for more information) by using the parameter `:parameters`. The content of this parameter is

pasted unchanged into the ganttchart parameter list. Remember that you have to escape backslashes in order for them to work. [1] The exception to this are the pgf-gantt parameters `time slot format` and `vgrid`. While `time slot format` is always set to `isodate` in order for org-gantt to work correctly, the `vgrid` parameter is used to emphasize the difference between weekend and work days. The default line style for those (settable via the custom variables `org-gantt-default-weekend-style` and `org-gant-default-workday-style`) can be overwritten using the parameters `:weekend-style` and `workday-style`.

3.5.1 Title Calendar

The title calendar, represented by the pgfgantt command `\pgftitlecalendar`, can be set with the parameter `title-calendar`. Refer to the pgfgantt manual for all possible options.

3.5.2 Compressing Charts

pgfgantt offers an option for compressing charts that span a long time, so that instead of each day occupying one slot, each slot represents an entire month. org-gantt allows to activate the compression by setting the parameter `:compress` to non-nil. If compression is activated, you can use the variable `compressed-title-calendar` to style the title calendar. The distinction for title calendars is necessary, so that reasonable defaults can be supplied for both cases.

3.5.3 Styling Example

It therefore becomes possible to create styles such as the following (shamelessly stolen from the pgf gantt manual) - look at the source in org-gantt-manual.org for the full list of parameters:

4 Reference

4.1 Parameters

General parameters:

`:id` The scope of the gantt chart. If `nil`, use the current document. If it starts with `file:\`, use the given document. Otherwise, use the headline with the given id property.

`:maxlevel` The maximum subheadline level for which the gantt chart is generated. The default value for this parameter can be set via the custom variable *org-gantt-default-maxlevel*.

`:use-id-subheadlines` Setting this parameter to t will make the gantt chart ignore the headline of the given id

Calculation parameters:

`:calc-progress` Setting this parameter to `use-larger-100` will make the progress calculation use values of larger 100 for overclocked subtasks. See section 3.4.

[1] Due to a bug in pgfgantt, it is advisable not to use the parameter `today offset`. This parameter unintentionally influences progress rendering.

`:progress-source` Determines the source that is used for progress calculation. `clocksum` will calculate from clocksums only. `cookie` will calculate from progress cookies in the headlines only. Both `[%]` and `[/]` cookies work. `clocksum-cookie` will use both, but preference clocksums, if both are available. `cookie-clocksum` will preference cookies. The default value for this parameter can be set via the custom variable *org-gantt-default-progress-source*.

`:hours-per-day` Sets the number of work hours in a work day. The default value for this parameter can be set via the custom variable *org-gantt-default-hours-per-day*.

`:work-free-days` The days on which no work is done, normally weekends. Is a list of integers, where each integer denotes a weekday, from 0 (sunday) to 6 (monday). The default value for this parameter can be set via the custom variable *org-gantt-default-work-free-days*.

`:incomplete-date-headlines` Determines the treatment for that have either deadline or schedule (also computed), but not both. `keep` will place the headline normally The default value for this parameter can be set via the custom variable *org-gantt-default-incomplete-date-headlines*.

`:no-date-headlines` Determines the treatment for that have neither deadline nor schedule. `keep` will place the headline normally The default value for this parameter can be set via the custom variable *org-gantt-default-no-date-headlines*.

`:tag-style-effect` Defines, where tag styles should be applied. If `current`, a tag style is only applied to headlines with the appropriate tag. If `subheadlines`, a tag style is applied to the headline and all its subheadlines. The default value for this parameter can be set via the custom variable *org-gantt-default-tag-style-effect*.

`:use-tags` A list of tags. Only headlines with these tags (and their subheadlines) will be printed. If `nil` The default value for this parameter can be set via the custom variable *org-gantt-default-use-tags*.

`:ignore-tags` A list of tags. Headlines with these tags (and all their subheadlines) will not be printed. The default value for this parameter can be set via the custom variable *org-gantt-default-ignore-tags*.

`:milestone-tags` A list of tags. Headlines with these tags will be printed as milesontes The default value for this parameter can be set via the custom variable *org-gantt-default-milestone-tags*.

`:linked-to-property-keys` A list of property keys. The values of these properties in headlines are interpreted as comma-seperated list, which indicates the ids of other headlines. Those other headlines will be visually linked to the headline with the property, and have its end time calculated, unless it already has an end time. The default value for this parameter can be set via the custom variable *org-gantt-default-linked-to-property-keys*.

Style parameters:

`:end-date` The end date of the chart. By default the date will be calculated as the latest date in the gantt chart.

`:parameters` Additional parameters added to the parameter list of the `\begin{ganttchart}` command. Any parameters allowed by pgfgantt can be used

`:show-progress` Setting this parameter to it to `always` will show the progress on all tasks. Setting it to `if-exists` will show progress only for those tasks with a clocksum, i.e. tasks that have been clocked, or that have subtasks that have been clocked. `nil` (the default) will not show progress on any tasks. The default value for this parameter can be set via the custom variable *org-gantt-default-show-progress*.

`:start-date` The start date of the chart. By default the date will be calculated as the earliest date in the gantt chart.

`:title-calendar` Sets the title calendar, that is the content of the pgfgantt command `\ganttitlecalendar{con` The default value for this parameter can be set via the custom variable *org-gantt-default-title-calendar*.

`:compress` If non-nil, the chart will be compressed according to pgfgantt. In compressed mode, only months are shown and not the days of each month.

`:compressed-title-calendar` Sets the title calendar, that is used if the chart is compressed. The default value for this parameter can be set via the custom variable *org-gantt-default-compressed-title-calendar*.

`:today` If set to t, the current date is highlighted as today in the gantt chart. If set to a timestamp, the given date is highlighted as today. If not set, no today value is used.

`:weekend-style` The style used for delimiting weekend days. The default value for this parameter can be set via the custom variable *org-gantt-default-weekend-style*.

`:workday-style` The style used for delimiting workday days. The default value for this parameter can be set via the custom variable *org-gantt-default-workday-style*.

`:inactive-bar-style` styles for bars that are considered inactive by `:incomplete-date-headlines` or `:no-date-headlines`. The default value for this parameter can be set via the custom variable *org-gantt-default-inactive-bar-style*.

`:inactive-group-style` styles for groups that are considered inactive by `:incomplete-date-headlines` or `:no-date-headlines`. The default value for this parameter can be set via the custom variable *org-gantt-default-inactive-group-style*.

`:tags-bar-style` An alist that associates tags to styles for bars in the form `(tag . style)` The default value for this parameter can be set via the custom variable *org-gantt-default-tags-bar-style*.

`:tags-group-style` An alist that associates tags to styles for groups in the form `(tag . style)` The default value for this parameter can be set via the custom variable *org-gantt-default-tags-group-style*.