Drawing Gantt Charts in LaTeX with Ti*k*Z

The pgfgantt package*

Wolfgang Skala†

April 18th, 2011

The **pgfgantt** package provides the ganttchart environment, which draws a Gantt chart within a Ti*k*Z picture. The user may add various elements to the chart, namely titles (\ganttitle, \ganttitlelist), bars (\ganttbar), milestones (\ganttmilestone), groups (\ganttgroup) and different links between these elements (\ganttlink). Furthermore, the appearance of the chart elements is highly customizable, owing to a number of keys.

Contents

*This document describes version v1.1, dated 2011/04/18.

†Division of Structural Biology, Department of Molecular Biology, University of Salzburg, Austria; Wolfgang.Skala@stud.sbg.ac.at

1 Introduction

The `pgfgantt` package allows you to draw Gantt charts in LaTeX. Thus, you can describe simple project schedules without having to include images produced by external programs. Similar to Martin Kumm's `gantt` package[1] (which inspired `pgfgantt`'s fundamental aspects), `pgfgantt` bases upon the Ti*k*Z frontend of PGF[2]. Besides, it provides a comprehensive (and portable) alternative to `pst-gantt`[3].

`pgfgantt` requires a *current* PGF installation. **Note that the version number must at least be 2.10, dated October 25th, 2010.** If you get a lot of errors and LaTeX complains that `\pgfkeysdefnargs` is undefined, your PGF installation is most liekly too old.

To load the package, simply put

```
\usepackage{pgfgantt}
```

into the document preamble.

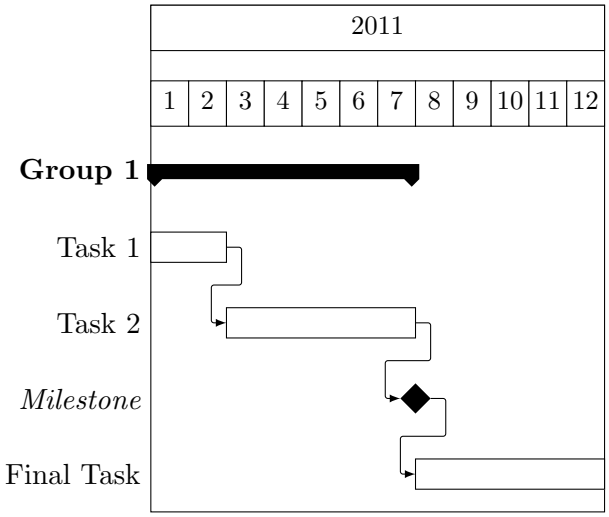
2 User Guide

2.1 Overview

Compare the following code, which demonstrates some commands provided by `pgfgantt`, to the output it produces:

[1] `http://www.martin-kumm.de/tex_gantt_package.php`
[2] `http://ctan.org/tex-archive/graphics/pgf/`
[3] `http://ctan.org/tex-archive/graphics/pstricks/contrib/pst-gantt/`

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}{12}
    \ganttitle{2011}{12} \\
    \ganttitlelist{1,...,12}{1} \\
    \ganttgroup{Group 1}{1}{7} \\
    \ganttbar{Task 1}{1}{2} \\
    \ganttlinkedbar{Task 2}{3}{7} \ganttnewline
    \ganttmilestone{Milestone}{7} \ganttnewline
    \ganttbar{Final Task}{8}{12}
    \ganttlink[b-m]{7}{5}{7}{6}
    \ganttlink[m-b]{7}{6}{8}{7}
  \end{ganttchart}
\end{tikzpicture}
```



2.2 Specifying Keys

Keys (sometimes called *options*) modify the output from `pgfgantt`'s commands. You may specify a key in two ways: (1) Pass it to the optional argument present in each command, e. g.

```
\ganttbar[bar label font=\bfseries]{Task 1}{1}{2}
```

This locally changes a key for the element(s) drawn by that command. (2) Alternatively, specify a key by the `\ganttset{`⟨*key=value list*⟩`}` macro, which sets its keys globally (or rather within the current TEX group):

`\ganttset`

```
\ganttset{bar label font=\bfseries}
```

Since `pgfgantt` uses the `pgfkeys` package for key management, all its keys reside in the `/pgfgantt/` path. However, if you set your keys by one of the methods explained above, this path is automatically prepended to each key.

2.3 The Canvas

Let us have a look at the basic anatomy of a Gantt chart and define some common terms. Each *chart* consists of several *elements*, such as titles, bars and connections between bars. Commands that start with `\gantt...` draw these elements. When specifying start and end *coordinates* for these commands, we use the dimensionless *chart coordinate system*, whose origin lies in the top left corner. Along the x-axis, one unit corresponds to one *time slot*; along the y-axis, one unit equals one *line*.

The `ganttchart` environment groups several of the element-drawing macros into a single chart: `ganttchart`

```
\begin{ganttchart}[⟨options⟩]{⟨number of time slots⟩}
  ...
\end{ganttchart}
```

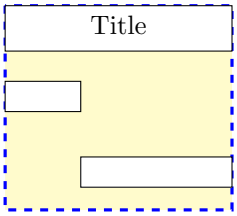
The environment has one optional and one mandatory argument. The former specifies the ⟨*options*⟩ for the chart, the latter indicates the ⟨*number of time slots*⟩.

Each `ganttchart` must be surrounded by a `tikzpicture` environment, whose x-vector/y-vector ratio should approximate 1 : 2 (for example, `x=.5cm, y=1cm` as above). Other ratios are well possible, but you might have to change several spacing-related keys in order to obtain a pleasing chart.

`/pgfgantt/canvas=`⟨*style*⟩ initial value: `fill=white`

The `canvas` key changes the appearance of the canvas. ⟨*style*⟩ is a list of Ti*k*Z keys such as `fill`, `draw` or `dashed`. By default, the canvas is a white rectangle with a black frame.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [canvas={fill=yellow!25, draw=blue, dashed, very thick}]{6}
    \ganttitle{Title}{6} \\
    \ganttbar{}{1}{2} \\
    \ganttbar{}{3}{6}
  \end{ganttchart}
\end{tikzpicture}
```



`/pgfgantt/hgrid[=false/true/`⟨*style*⟩`]` `false`

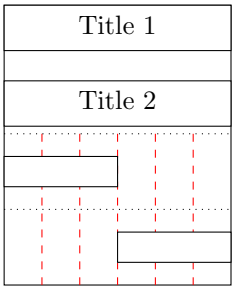
`/pgfgantt/hgrid style=`⟨*style*⟩ `dotted`

/pgfgantt/vgrid[=false/true/⟨*style*⟩] false
/pgfgantt/vgrid style=⟨*style*⟩ dotted
hgrid draws a horizontal grid which starts immediately below the last title element. The key can be specified in four different ways: Firstly, hgrid=false eliminates the horizontal grid. You may omit this declaration, since it is the default. Secondly, both hgrid and hgrid=true activate the horizontal grid, which is then drawn in the default style dotted. Finally, hgrid=⟨*style*⟩ draws the horizontal grid in the given ⟨*style*⟩.

You must change the style of the horizontal grid explicitly with hgrid style if you only wish to draw selected grid lines with \ganttnewline[grid] (see section 2.4). Actually, hgrid=⟨*style*⟩ is just a shortcut for hgrid=true, hgrid style=⟨*style*⟩.

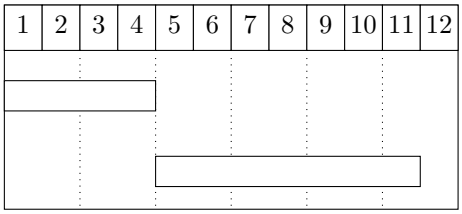
The vgrid key governs the vertical grid; otherwise, it is similar to hgrid.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[hgrid=true, vgrid={draw=red, dashed}]{6}
    \ganttttitle{Title 1}{6} \\
    \ganttttitle{Title 2}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/vgrid lines list=⟨*pgffor list*⟩ 2,3,...,\value{gtt@width}
If the chart contains many time slots, drawing vertical grid lines between all of them will lead to a confusing appearance. In such a case, you can change the vgrid lines list key in order to draw every second grid line, for example.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, vgrid lines list={3,5,...,\value{gtt@width}}]{12}
    \ganttttitlelist{1,...,12}{1} \\
    \ganttbar{}{1}{4} \\
    \ganttbar{}{5}{11}
  \end{ganttchart}
\end{tikzpicture}
```



`/pgfgantt/hgrid shift=`⟨*factor*⟩ `-0.3`

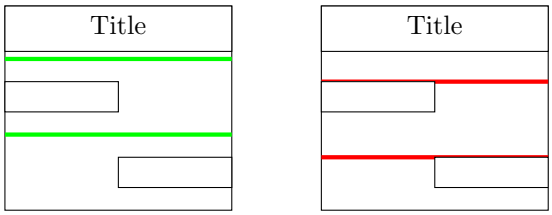
With default space parameters, the upper edge of a bar has an integral y-coordinate. Consequently, horizontal grid lines should be shifted upwards (i. e., along the negative y-axis) from their standard positions (see chart on the left). Otherwise, they will clash with the top of the bars (see chart on the right).

```
% Correct hgrid position

\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [hgrid={green, ultra thick}]{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```

```
% Wrong hgrid position

\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [hgrid={red, ultra thick},
      hgrid shift=0]{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```



`/pgfgantt/last line height=`⟨*factor*⟩ `0.7`

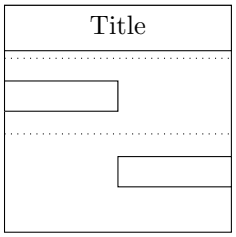
With the default space parameters, the last line would appear too high and the element it contains would appear vertically displaced (see chart on the left). Therefore, the height of the bottommost line is by default decreased to 70 % of its native size (see chart on the right).

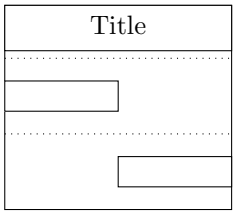
```
% Last line too high

\begin{tikzpicture}%
    [x=.5cm, y=1cm, baseline]
  \begin{ganttchart}%
      [hgrid,
      last line height=1]{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```

```
% Last line correct

\begin{tikzpicture}%
    [x=.5cm, y=1cm, baseline]
  \begin{ganttchart}%
      [hgrid,
      last line height=0.7]{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```



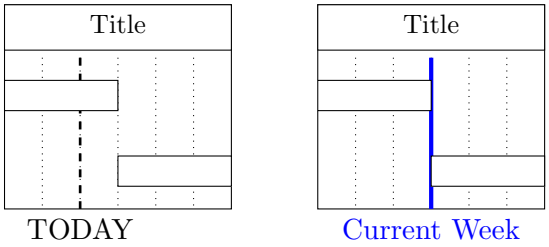


/pgfgantt/today=⟨*time slot*⟩ none
/pgfgantt/today rule=⟨*style*⟩ dashed, line width=1pt
/pgfgantt/today label=⟨*text*⟩ TODAY

Sometimes, you may wish to indicate the current day, month or the like on a Gantt chart. In order to do so, pass an integer value to the `today` key, which draws a vertical rule at the corresponding ⟨*time slot*⟩. This rule appears in the ⟨*style*⟩ denoted by `today rule`, while `today label` contains the ⟨*text*⟩ below the rule.

```
\begin{tikzpicture}%
    [x=.5cm, y=1cm, baseline]
  \begin{ganttchart}%
      [vgrid, today=2]{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```

```
\begin{tikzpicture}%
    [x=.5cm, y=1cm, baseline]
  \begin{ganttchart}%
      [vgrid, today=3,
      today label=\textcolor{blue}%
        {Current Week},
      today rule={blue, ultra thick}]{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```



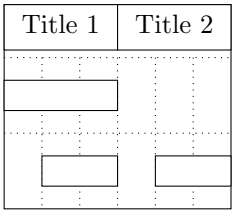
2.4 Line Breaks between Chart Elements

`pgfgantt` does not automatically begin a new line after finishing a chart element. Instead, you must insert an explicit line break with `\ganttnewline`. Within a `ganttchart` environment, `\\` is defined as a shortcut for `\ganttnewline`, so that the syntax is reminiscent of LaTeX's `tabular` enviroment.

`\ganttnewline`

`\\`

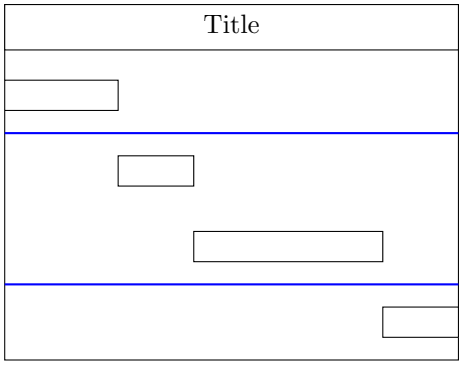
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[hgrid, vgrid]{6}
    \ganttitle{Title 1}{3}
    \ganttitle{Title 2}{3} \\
    \ganttbar{}{1}{3} \ganttnewline
    \ganttbar{}{2}{3}
    \ganttbar{}{5}{6}
  \end{ganttchart}
\end{tikzpicture}
```



Even if you prefer a canvas without a horizontal grid, you may nevertheless want to separate certain lines by a grid rule. For this purpose, specify the optional argument `[grid]` for `\ganttnewline` (or `\\`), which draws a grid rule between the current and the new line.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[hgrid style={thick, blue}]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{}{1}{3} \ganttnewline[grid]
    \ganttbar{}{4}{5} \\
    \ganttbar{}{6}{10} \\[grid]
    \ganttbar{}{11}{12}
```

```
  \end{ganttchart}
\end{tikzpicture}
```



2.5 Titles

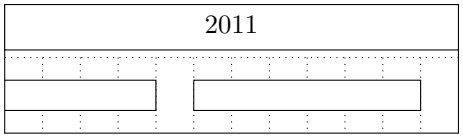
A *title* (comprising one or more lines) at the top of a Gantt chart usually indicates the period of time covered by that chart. For example, the first line could span twelve time slots and display the current year, while the second line could contain twelve elements, each of which corresponds to one month. For these purposes, **pgfgantt** implements two titling commands.

`\gantttitle` draws a single title element: `\gantttitle`

```
\gantttitle[⟨options⟩]{⟨label⟩}{⟨number of time slots⟩}
```

The ⟨*label*⟩ appears in the center of the title element, which covers the ⟨*number of time slots*⟩ starting from the right end of the last title element (or from the beginning of the line, if the title element is the first element in this line). Mostly, you will employ `\gantttitle` for titles that span several time slots.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[hgrid, vgrid]{12}
    \gantttitle{2011}{12} \\
    \ganttbar{}{1}{4}
    \ganttbar{}{6}{11}
  \end{ganttchart}
\end{tikzpicture}
```



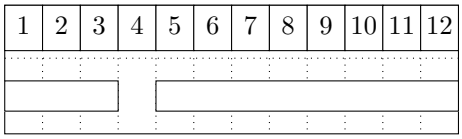
Whenever you want to draw a larger number of title elements that are equal in size and follow a common enumeration scheme, the `\ganttitlelist` macro provides a fast solution: `\ganttitlelist`

`\ganttttitlelist[`⟨*options*⟩`]{`⟨*pgffor list*⟩`}{`⟨*length of each element*⟩`}`

This macro generates one title element for each member of the ⟨*pgffor list*⟩. The second mandatory argument specifies the ⟨*length of each element*⟩. The Ti*k*Z manual describes the syntax for the ⟨*pgffor list*⟩ in more detail, but we will mention two of the most common applications:

1. In order to draw twelve title elements that contain the numbers from 1 to 12 (indicating the months of a year), enter `1,...,12` as the ⟨*pgffor*⟩ list.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[hgrid, vgrid]{12}
    \ganttttitlelist{1,...,12}{1} \\
    \ganttbar{}{1}{3}
    \ganttbar{}{5}{12}
  \end{ganttchart}
\end{tikzpicture}
```



Note that we would have obtained the same result if we had written

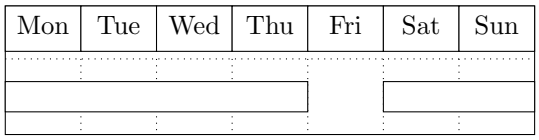
```
\ganttttitle{1}{1} \ganttttitle{2}{1} ... \ganttttitle{12}{1} \\
```

2. In order to draw seven title elements containing the names of the weekdays (e. g., “Mon” to “Sun”), we have to change the `title list options` key:

`/pgfgantt/title list options=`⟨*pgffor options*⟩ `var=\x, evaluate=\x`
This key changes the ⟨*pgffor options*⟩ of the `\foreach` command called by `\ganttttitlelist`. Again, the Ti*k*Z manual is the definitive reference on possible ⟨*pgffor options*⟩. There is just one thing to keep in mind: The macro that yields the labels to be printed by `\ganttttitlelist` must be called `\x`.
The following example shows how you can implement a title line enumerating the days of the week:

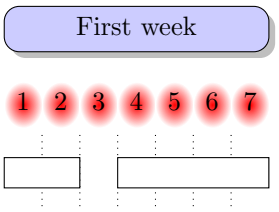
```
\usepackage{pgfcalendar}
  ...
\begin{tikzpicture}[x=1cm, y=1cm]
  \begin{ganttchart}[hgrid, vgrid]{7}
    \ganttttitlelist[title list options={%
        var=\y, evaluate=\y as \x%
        using "\pgfcalendarweekdayshortname{\y}"%
```

```
      }]{0,...,6}{1} \\
    \ganttbar{}{1}{4}
    \ganttbar{}{6}{7}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/title=⟨*style*⟩ `fill=white`
Sets the appearance of a title element.

```
\usetikzlibrary{shadows}
\usetikzlibrary{shadings}
  ...
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, canvas={draw=none},
      title={fill=blue!20, rounded corners=2mm, drop shadow}]{7}
    \ganttitle{First week}{7} \\
    \ganttitlelist[title={draw=none, inner color=red}]{1,...,7}{1} \\
    \ganttbar{}{1}{2}
    \ganttbar{}{4}{7}
  \end{ganttchart}
\end{tikzpicture}
```



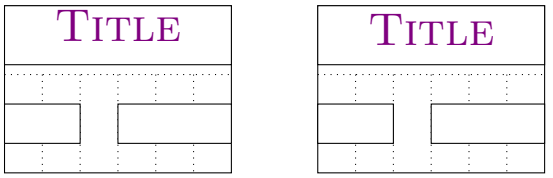
/pgfgantt/title label font=⟨*font commands*⟩ `\small`
Selects the font of the text inside a title element. In most cases, you can include font format commands directly in the first mandatory argument of `\ganttitle`. However, you *must* use the `title label font` key if you intend to change the font size. Otherwise, the vertical alignment of the title label will be incorrect with the standard anchor.

```
% Wrong alignment

\begin{tikzpicture}[x=.5cm, y=1.3cm]
  \begin{ganttchart}%
      [vgrid, hgrid]{6}
    \ganttitle{%
      \LARGE\color{violet}%
      \scshape Title}{6} \\
    \ganttbar{}{1}{2}
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```

```
% Correct alignment

\begin{tikzpicture}[x=.5cm, y=1.3cm]
  \begin{ganttchart}%
      [vgrid, hgrid,
      title label font={\LARGE,
      \color{violet}, \scshape}]{6}
    \ganttitle{Title}{6} \\
    \ganttbar{}{1}{2}
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/title label anchor=⟨*anchor*⟩ anchor=mid

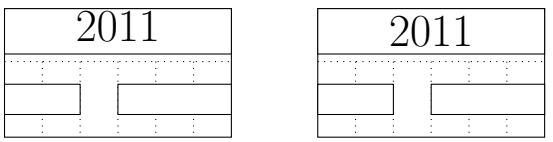
By default, title labels are vertically centered at half their x-height. This yields a good alignment for labels whose letters have equal amounts of ascenders and descenders (e. g., lowercase numbers). However, when the letters contain mostly ascenders (e. g., uppercase numbers), the label position will appear too high. In this case, you should change the anchor:

```
% Badly centered label

\begin{tikzpicture}%
    [x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid,
      title label font={\LARGE}%
      ]{6}
    \ganttitle{2011}{6} \\
    \ganttbar{}{1}{2}
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```

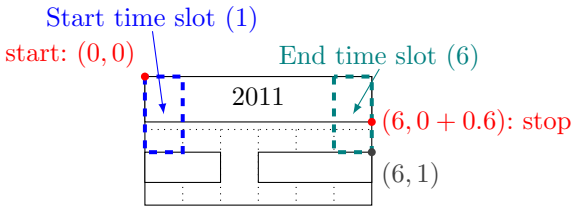
```
% Nicely centered label

\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid,
      title label font={\LARGE},
      title label anchor={below=-1.5ex}%
      ]{6}
    \ganttitle{2011}{6} \\
    \ganttbar{}{1}{2}
    \ganttbar{}{4}{6}
  \end{ganttchart}
\end{tikzpicture}
```

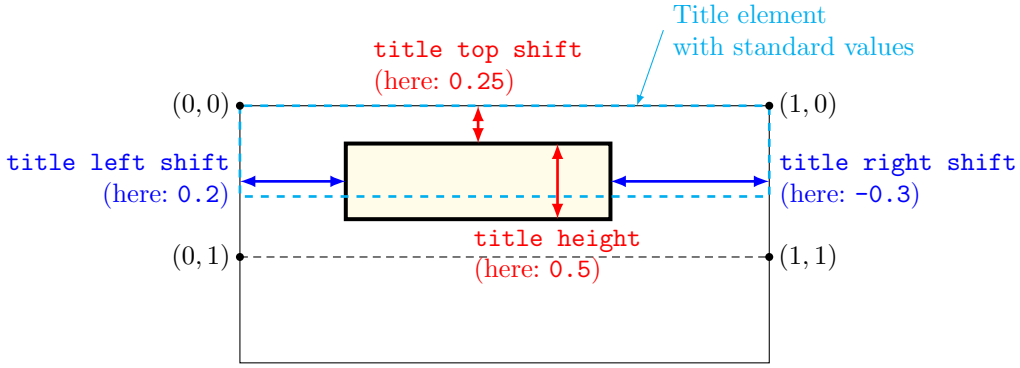


/pgfgantt/title left shift=⟨*factor*⟩ 0
/pgfgantt/title right shift=⟨*factor*⟩ 0
/pgfgantt/title top shift=⟨*factor*⟩ 0
/pgfgantt/title height=⟨*factor*⟩ 0.6

The first three keys shift the coordinates of a title element's borders (or rather of its corners), while `title height` changes its height. By default, the left upper corner of a title element coincides with the origin of the start time slot; its right lower corner touches the right border of the end time slot 0.6 units below the upper line border:

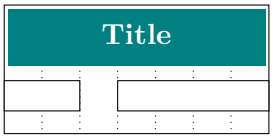


The figure below shows a Gantt chart with two lines and one (large) time slot and indicates the distances modified by these keys.



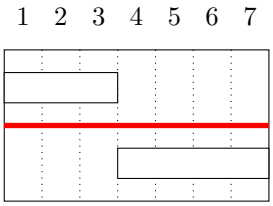
For example, you might devise a layout where the title element does not touch the borders of the start and end time slot.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, title={fill=teal, draw=none},
      title label font=\color{white}\bfseries,
      title left shift=.1, title right shift=-.1,
      title top shift=.05, title height=.75]{7}
    \ganttitle{Title}{7} \\
    \ganttbar{}{1}{2}
    \ganttbar{}{4}{7}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/include title in canvas=false/true true
The canvas normally comprises all lines of the chart. However, you may wish that your title elements only consist of text lacking any frame or background. In this case, the canvas probably should exclude all lines containing title elements, which you achieve by include title in canvas=false.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [hgrid={draw=red, line width=2pt}, vgrid,
      title={draw=none, fill=none}, include title in canvas=false]{7}
    \ganttitlelist{1,...,7}{1} \\
    \ganttbar{}{1}{3} \\
    \ganttbar{}{4}{7}
  \end{ganttchart}
\end{tikzpicture}
```



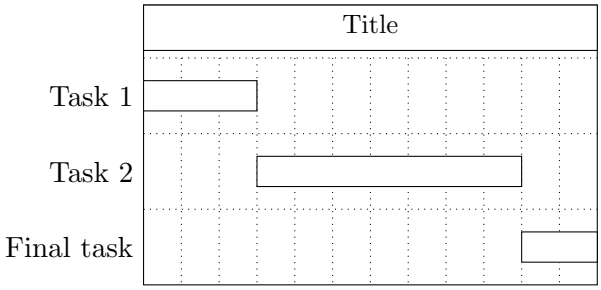
2.6 Bars

On a Gantt chart, a *bar* indicates the duration of a task or one of its parts.

```
\ganttbar[⟨options⟩]{⟨label⟩}{⟨start time slot⟩}{⟨end time slot⟩}
```

The \ganttbar macro draws a bar from the ⟨*start time slot*⟩ to the ⟨*end time slot*⟩ and adds a ⟨*label*⟩ at the left of the chart. \ganttbar

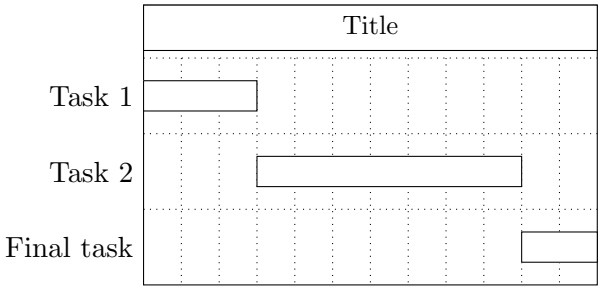
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{1}{3} \\
    \ganttbar{Task 2}{4}{10} \\
    \ganttbar{Final task}{11}{12}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/time slot modifier=⟨*factor*⟩ -1

Note that a bar usually touches the left border of the ⟨*start time slot*⟩ and not the right, as it would if the ⟨*start time slot*⟩ were strictly interpreted as an x-coordinate. However, you may prefer to work with "real" x-coordinates instead of time slots. In this case, just set the time slot modifier key to zero. This will essentially eliminate the semi-intelligent behavior of pgfgantt with respect to the conversion of x-coordinates. This feature may prove useful if you decide to use real numbers for some time slots.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, time slot modifier=0]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{0}{3} \\
    \ganttbar{Task 2}{3}{10} \\
    \ganttbar{Final task}{10}{12}
  \end{ganttchart}
\end{tikzpicture}
```

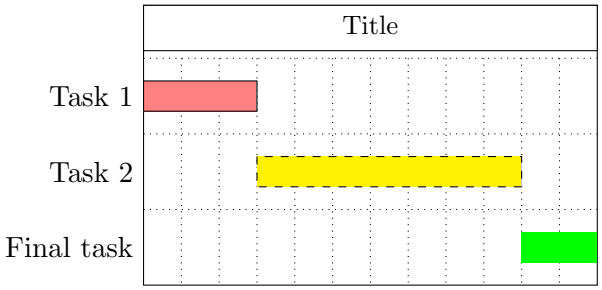


/pgfgantt/bar=⟨*style*⟩ fill=white

Determines the appearance of the bar.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, bar={fill=red!50}]{12}
    \ganttitle{Title}{12} \\
```

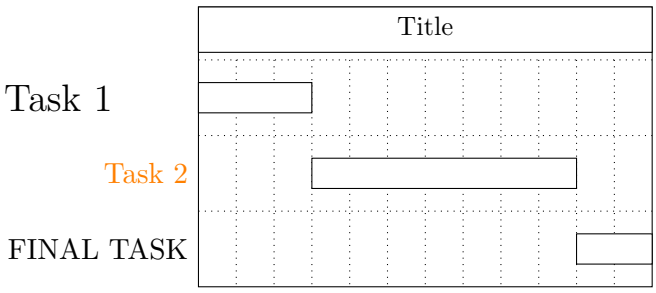
```
    \ganttbar{Task 1}{1}{3} \\
    \ganttbar[bar={fill=yellow, dashed}]{Task 2}{4}{10} \\
    \ganttbar[bar={fill=green, draw=none}]{Final task}{11}{12}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/bar label font=⟨*font commands*⟩ `\normalsize`
/pgfgantt/bar label anchor=⟨*anchor*⟩ `anchor=east`

The `bar label font` key selects the font for the bar label, `bar label anchor` determines its anchor. The last control sequence in ⟨*font commands*⟩ may take a single argument (like `\textit`).

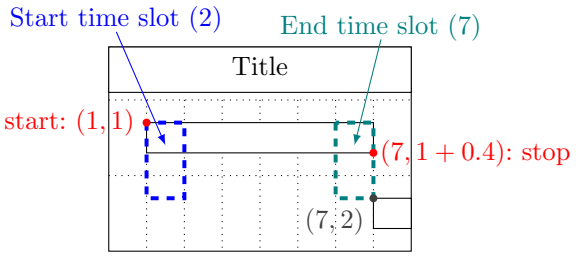
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, bar label font=\Large]{12}
    \ganttitle{Title}{12} \\
    \ganttbar[bar label anchor={left=1cm}]{Task 1}{1}{3} \\
    \ganttbar[bar label font=\color{orange}]{Task 2}{4}{10} \\
    \ganttbar[bar label font=\MakeUppercase]{Final task}{11}{12}
  \end{ganttchart}
\end{tikzpicture}
```



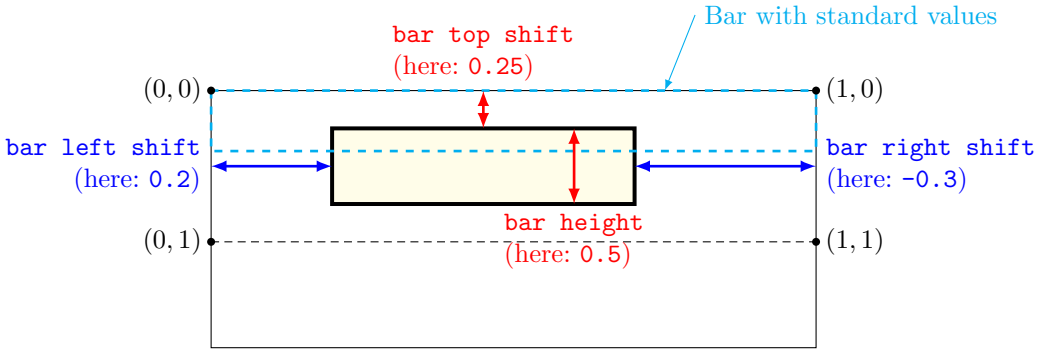
/pgfgantt/bar left shift=⟨*factor*⟩ 0
/pgfgantt/bar right shift=⟨*factor*⟩ 0
/pgfgantt/bar top shift=⟨*factor*⟩ 0

/pgfgantt/bar height=⟨*factor*⟩ 0.4

The first three keys shift the coordinates of a bar's borders (or rather of its corners), while bar height changes its height. By default, the left upper corner of a bar coincides with the origin of the start time slot; its right lower corner touches the right border of the end time slot 0.4 units below the upper line border:

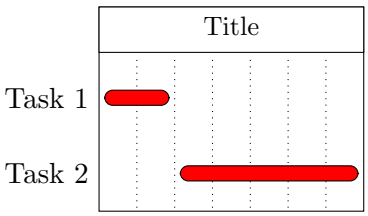


The figure below shows a Gantt chart with two lines and one (large) time slot and indicates the distances modified by these keys.



For example, you might devise a layout with small, rounded bars that do not touch the borders of their start and end time slots.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, bar={fill=red, rounded corners=3pt},
      bar left shift=.15, bar right shift=-.15,
      bar top shift=.1, bar height=.2]{7}
    \ganttitle{Title}{7} \\
    \ganttbar{Task 1}{1}{2} \\
    \ganttbar{Task 2}{3}{7}
  \end{ganttchart}
\end{tikzpicture}
```



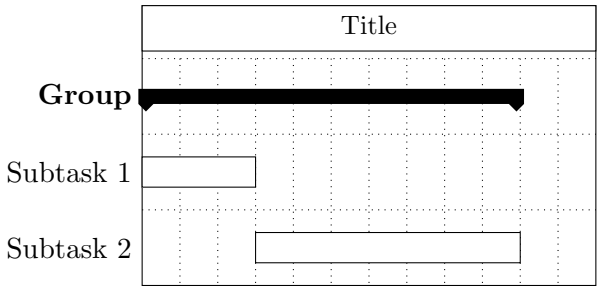
2.7 Groups

Groups subsume several subtasks (represented by bars) into a single task.

`\ganttgroup[`⟨*options*⟩`]{`⟨*label*⟩`}{`⟨*start time slot*⟩`}{`⟨*end time slot*⟩`}`

The `\ganttgroup` macro draws a group from the ⟨*start time slot*⟩ to the ⟨*end time slot*⟩ and adds a ⟨*label*⟩ at the left of the chart. Note that a group will start at the left border of the ⟨*start time slot*⟩ (and not at the right, as it would if the ⟨*start time slot*⟩ were strictly interpreted as an x-coordinate). `\ganttgroup`

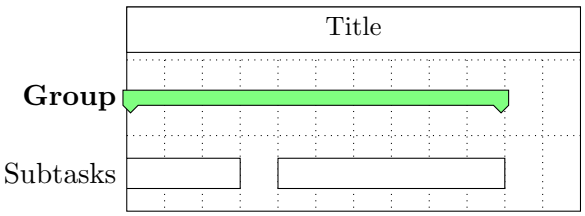
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid]{12}
    \ganttitle{Title}{12} \\
    \ganttgroup{Group}{1}{10} \\
    \ganttbar{Subtask 1}{1}{3} \\
    \ganttbar{Subtask 2}{4}{10}
  \end{ganttchart}
\end{tikzpicture}
```



`/pgfgantt/group=`⟨*style*⟩ `fill=black`

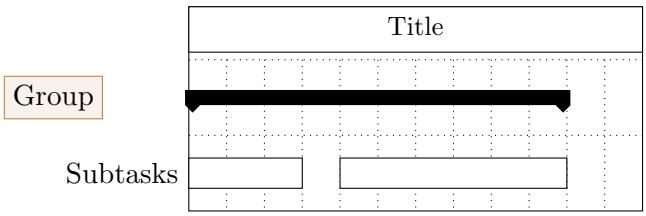
Changes the appearance of a group.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, group={draw=black, fill=green!50}]{12}
    \ganttitle{Title}{12} \\
    \ganttgroup{Group}{1}{10} \\
    \ganttbar{Subtasks}{1}{3}
    \ganttbar{}{5}{10}
  \end{ganttchart}
\end{tikzpicture}
```

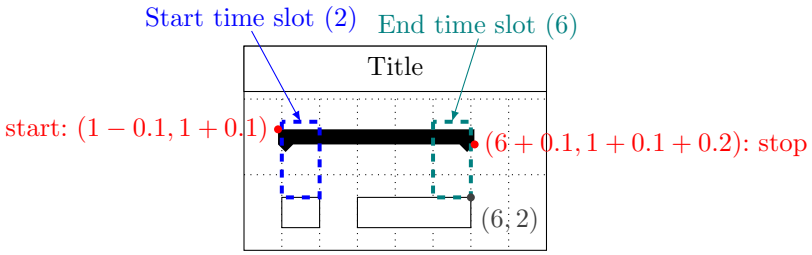


/pgfgantt/group label font=⟨*font commands*⟩ `\normalsize\bfseries`
/pgfgantt/group label anchor=⟨*anchor*⟩ `anchor=east`
The `group label font` key selects the font of the group label, `group label anchor` determines its anchor. The last control sequence in ⟨*font commands*⟩ may take a single argument (like `\textit`).

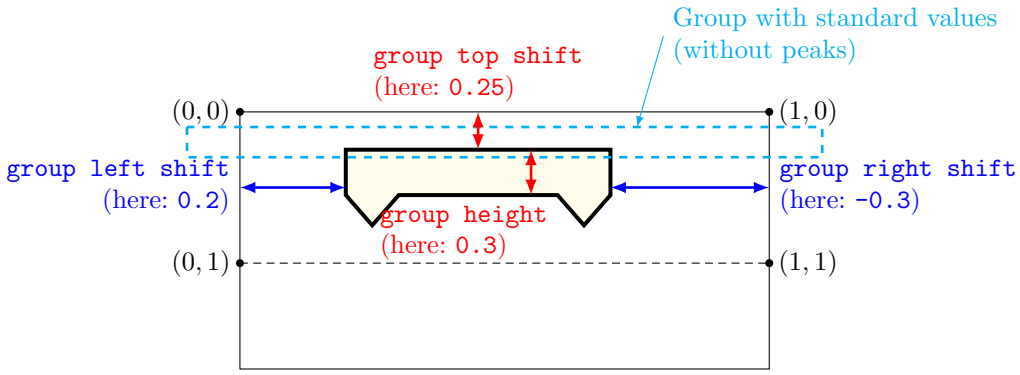
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid,
      group label font={\fcolorbox{brown}{brown!10}},
      group label anchor={left=1cm}]{12}
    \gantttitle{Title}{12} \\
    \ganttgroup{Group}{1}{10} \\
    \ganttbar{Subtasks}{1}{3}
    \ganttbar{}{5}{10}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/group left shift=⟨*factor*⟩ `-0.1`
/pgfgantt/group right shift=⟨*factor*⟩ `0.1`
/pgfgantt/group top shift=⟨*factor*⟩ `0.1`
/pgfgantt/group height=⟨*factor*⟩ `0.2`
The first three keys shift the coordinates of a group's borders (or rather of its corners), while `group height` changes its height. By default, the left upper corner of a group is 0.1 units left of and 0.1 units below the start time slot origin; its right lower corner (not counting the peak) lies 0.1 units right of and 0.3 units below the right border of the end time slot:



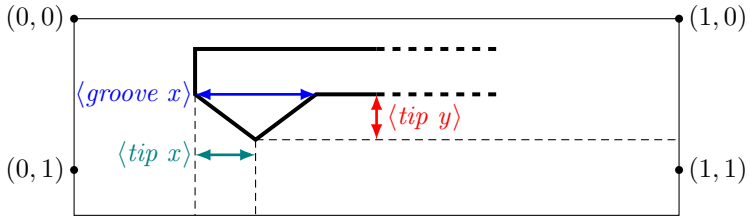
The figure below shows a Gantt chart with two lines and one (large) time slot and indicates the distances modified by these keys.



/pgfgantt/group left peak={⟨*tip x*⟩}{⟨*groove x*⟩}{⟨*tip y*⟩}
/pgfgantt/group right peak={⟨*tip x*⟩}{⟨*groove x*⟩}{⟨*tip y*⟩}
/pgfgantt/group peaks={⟨*tip x*⟩}{⟨*groove x*⟩}{⟨*tip y*⟩} 0.2 0.4 0.1

These keys govern the appearance of the peaks at both ends of a group. By default, the tip of each peak lies 0.2 units inward from a group's bottom corner and 0.1 units beneath, while the groove lies 0.4 units inward. While `group left peak` applies only to the left peak and `group right peak` affects only the right peak, `group peaks` sets the dimensions for both peaks simultaneously. You always have to specify three arguments for these keys. However, if you leave one of them blank, the corresponding space parameter retains its current value.

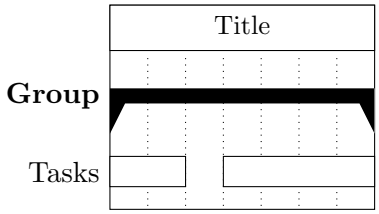
The figure below exemplifies the space parameters as they apply to the left peak.



For example, you might prefer that your groups stay within the start and end time slot, and that the peaks are more acute:

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, group left shift=0, group right shift=0,
      group peaks={0}{}{.4}]{7}
    \ganttitle{Title}{7} \\
    \ganttgroup{Group}{1}{7} \\
    \ganttbar{Tasks}{1}{2}
    \ganttbar{}{4}{7}
```

```
  \end{ganttchart}
\end{tikzpicture}
```



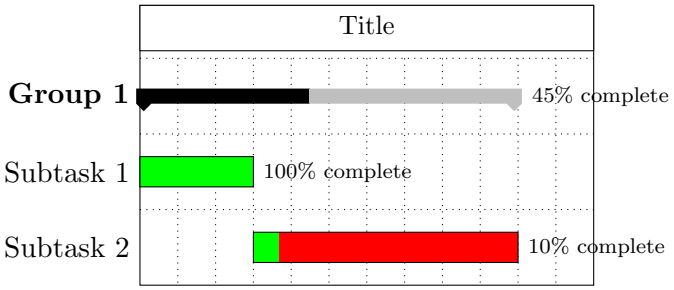
2.8 Progress Bars and Progress Groups

Progress bars and *progress groups* illustrate the extent to which a (sub-)task has been completed. In order to draw a progress element, you simply specify the `progress` key in the optional argument to the respective standard macro.

`/pgfgantt/progress=`none/⟨*number*⟩ `none`
`/pgfgantt/bar incomplete=`⟨*style*⟩
`/pgfgantt/group incomplete=`⟨*style*⟩
`/pgfgantt/incomplete=`⟨*style*⟩ `fill=black!25`

The `progress` key specifies that a task (represented by a bar) or a group thereof is ⟨*number*⟩ percent complete. Starting from the left, ⟨*number*⟩ percent of the element's area appear in the basic style (i. e., `bar` or `group`), while the `bar incomplete` and `group incomplete` keys, respectively, determine the appearance of the remainder. For convenience, the `incomplete` key simultaneously sets the incomplete style for bars and groups.

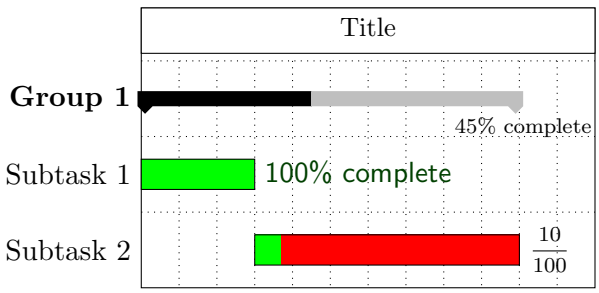
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, bar={fill=green}]{12}
    \ganttitle{Title}{12} \\
    \ganttgroup[progress=45]{Group 1}{1}{10} \\
    \ganttbar[progress=100]{Subtask 1}{1}{3} \\
    \ganttbar[progress=10, bar incomplete={fill=red}]{Subtask 2}{4}{10}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/progress label text=⟨*text*⟩ `#1\% complete`
/pgfgantt/progress label font=⟨*font commands*⟩ `\scriptsize`
/pgfgantt/progress label anchor=⟨*anchor*⟩ `anchor=west`

The `progress label text` key sets the ⟨*text*⟩ that appears beside each progress element in order to indicate its completeness. This key may contain a single parameter token (`#1`), which is replaced by the value of `progress`. The label is typeset in the `progress label font`. In addition, `progress label anchor` governs its placement. By changing the default value, you may prevent the label from overlapping with other elements of your chart.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, bar={fill=green}]{12}
    \ganttitle{Title}{12} \\
    \ganttgroup%
        [progress=45, progress label anchor={below=3pt}]%
      {Group 1}{1}{10} \\
    \ganttbar%
        [progress=100, progress label font=\color{green!25!black}\textsf]%
      {Subtask 1}{1}{3} \\
    \ganttbar%
        [progress=10, incomplete={fill=red},
        progress label text={$\displaystyle\frac{#1}{100}$}]%
      {Subtask 2}{4}{10}
  \end{ganttchart}
\end{tikzpicture}
```



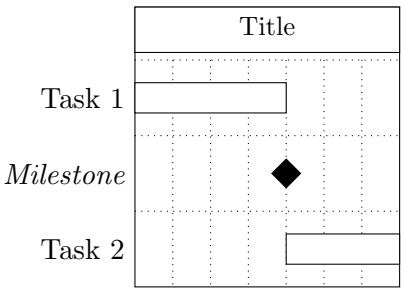
2.9 Milestones

A *milestone* signifies that an important task has been completed or that a crucial goal has been reached.

```
\ganttmilestone[⟨options⟩]{⟨label⟩}{⟨time slot⟩}
```

The `\ganttmilestone` macro draws a milestone at the given ⟨*time slot*⟩ and adds a ⟨*label*⟩ at the left of the chart. `\ganttmilestone`

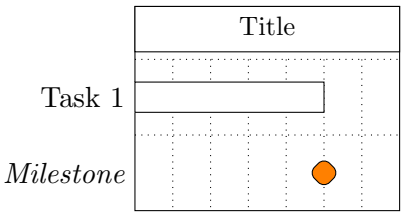
```
  \begin{ganttchart}[vgrid, hgrid]{7}
    \ganttitle{Title}{7} \\
    \ganttbar{Task 1}{1}{4} \\
    \ganttmilestone{Milestone}{4} \\
    \ganttbar{Task 2}{5}{7}
  \end{ganttchart}
\end{tikzpicture}
```



Note that the milestone is usually centered on the vertical grid line between its ⟨*time slot*⟩ and the following one.

/pgfgantt/milestone=⟨*style*⟩ `fill=black`
Determines the appearance of the milestone.

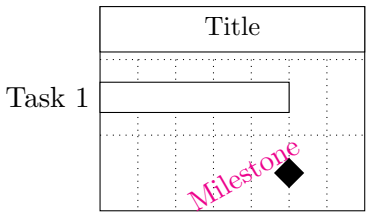
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid,
     milestone={fill=orange, draw=black, rounded corners=3pt}]{7}
    \ganttitle{Title}{7} \\
    \ganttbar{Task 1}{1}{5} \\
    \ganttmilestone{Milestone}{5}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/milestone label font=⟨*font commands*⟩ `\normalsize\itshape`
/pgfgantt/milestone label anchor=⟨*anchor*⟩ `anchor=east`

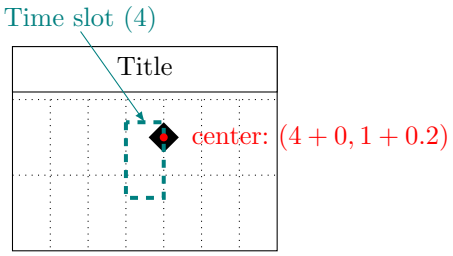
The former key sets the font of the milestone label, while the latter determines its placement. The last macro in ⟨*font commands*⟩ may take a single argument, as we show in the following (somewhat silly) example.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid,
      milestone label font=\color{magenta}\rotatebox{30},
      milestone label anchor={right=1cm}]{7}
    \ganttitle{Title}{7} \\
    \ganttbar{Task 1}{1}{5} \\
    \ganttmilestone{Milestone}{5}
  \end{ganttchart}
\end{tikzpicture}
```

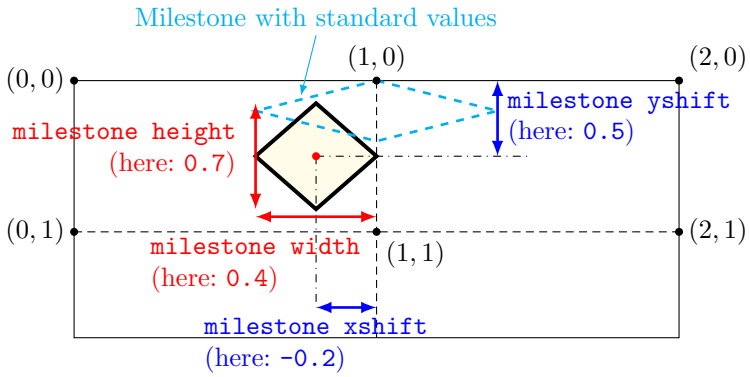


/pgfgantt/milestone width=⟨*factor*⟩	0.8
/pgfgantt/milestone height=⟨*factor*⟩	0.4
/pgfgantt/milestone xshift=⟨*factor*⟩	0
/pgfgantt/milestone yshift=⟨*factor*⟩	0.2

These keys set the width and height of a milestone and shift the coordinates of its center. By default, a milestone is 0.8 units wide and 0.4 units high. Since the ideal x-vector/y-vector ratio of the parent `tikzpicture` is 1 : 2, the milestone appears square with these settings. Its center lies on the right border and 0.2 units below the top border of its time slot.



The figure below shows a Gantt chart with a single milestone and two (large) time slots; it indicates the distances modified by the four keys explained above.



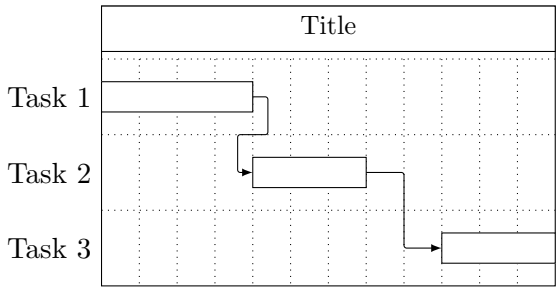
2.10 Links

So far, we have drawn charts whose elements were quite independent of each other. However, relations or *links* between these elements frequently appear on real Gantt charts. For example, a task may only start if a previous one has been completed, or finishing a task may constitute a milestone.

```
\ganttlink[⟨options⟩]{⟨start time slot⟩}{⟨start line⟩}{⟨end time slot⟩}{⟨end line⟩}
```

The \ganttlink macro connects two elements. The first element is in the ⟨*start line*⟩ and ends at the ⟨*start time slot*⟩, while the second element resides in the ⟨*end line*⟩ and starts at the ⟨*end time slot*⟩. \ganttlink

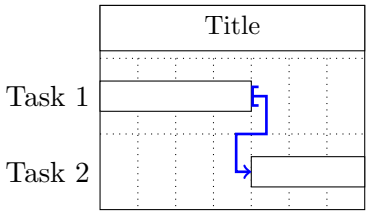
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{1}{4} \\
    \ganttbar{Task 2}{5}{7} \\
    \ganttbar{Task 3}{10}{12}
    \ganttlink{4}{2}{5}{3}
    \ganttlink{7}{3}{10}{4}
  \end{ganttchart}
\end{tikzpicture}
```



/pgfgantt/link=⟨*style*⟩ `-latex, rounded corners=1pt`

Sets the appearance of the link.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, link={[-to, line width=1pt, blue}]{7}
    \ganttitle{Title}{7} \\
    \ganttbar{Task 1}{1}{4} \\
    \ganttbar{Task 2}{5}{7}
    \ganttlink{4}{2}{5}{3}
  \end{ganttchart}
\end{tikzpicture}
```



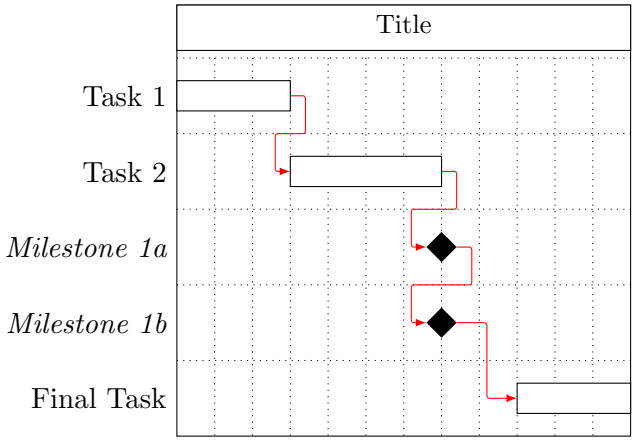
`pgfgantt` calculates the actual start and stop coordinates for each link, but in order to succeed, it has to know which types of elements it should connect. Consequently, the optional argument of `\ganttlink` must contain a *link type key*.

The syntax of a link type key is similar to the syntax for specifying arrow tips in Ti*k*Z: Each such key is composed of two letters separated by a hyphen.

Link types fall into two categories:

1. *Arrow-like links* may contain the letters `b` (for bar) and `m` (for milestone). Thus, `b-b` denotes a link between two bars, while `b-m` denotes a link from a bar to a milestone. Note that `b-b` is the default link type, so you do not have to specify it for connecting two bars.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid, link={-latex, red, rounded corners=1pt}]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{1}{3} \\
    \ganttbar{Task 2}{4}{7} \\
    \ganttmilestone{Milestone 1a}{7} \\
    \ganttmilestone{Milestone 1b}{7} \\
    \ganttbar{Final Task}{10}{12}
    \ganttlink{3}{2}{4}{3}
    \ganttlink[b-m]{7}{3}{7}{4}
    \ganttlink[m-m]{7}{4}{7}{5}
    \ganttlink[m-b]{7}{5}{10}{6}
  \end{ganttchart}
\end{tikzpicture}
```



As you can see from this graph, arrow-like links consist of three segments (two horizontal, one vertical) if their start and end time slots are sufficiently separated. Otherwise, they comprise five segments (three horizontal, two vertical). Three keys further modify the appearance of arrow-like links:

`/pgfgantt/link mid`=⟨*factor*⟩ `0.5`

The `link mid` key changes the position of the single vertical segment (in three-part links) or of the middle horizontal segment (in five-part links). By default, these segments are horizontally centered between the left and the right vertical segment, or vertically centered between the upper and the lower horizontal segment, respectively.

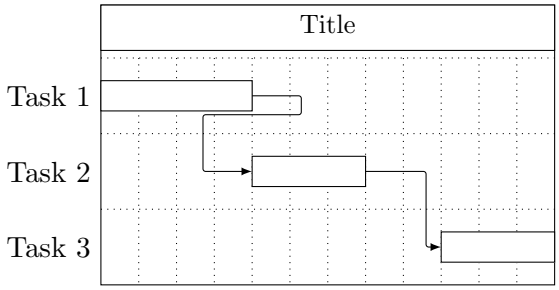
`/pgfgantt/link bulge`=⟨*factor*⟩ `0.4`

In five-part links, the upper and lower vertical segments are shifted along the x-axis by +`link bulge` and −`link bulge`, respectively.

`/pgfgantt/link tolerance`=⟨*factor*⟩ `0.6`

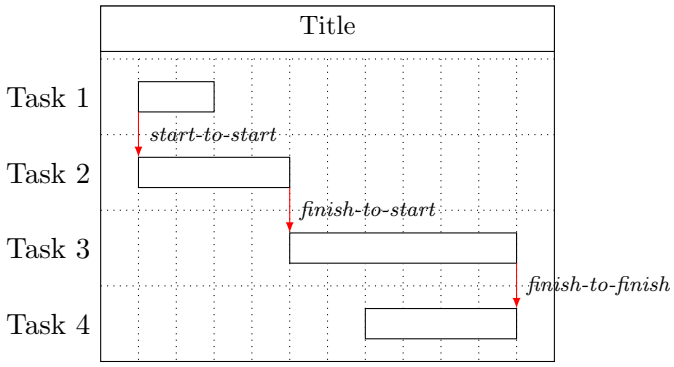
This key governs whether **pgfgantt** draws a five- or a three-part link. If the true x-coordinates of the link start and end differ by at least `link tolerance` (this is the case for the second link in the example below), the package draws a five-part link.

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, link mid=.25, link bulge=1.3]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{1}{4} \\
    \ganttbar{Task 2}{5}{7} \\
    \ganttbar{Task 3}{10}{12}
    \ganttlink{4}{2}{5}{3}
    \ganttlink[link mid=.8]{7}{3}{10}{4}
  \end{ganttchart}
\end{tikzpicture}
```



2. *Straight links* may contain the letters `s` (for start) and `f` (for finish). They are only meant for connecting two bars in order to establish start-to-finish relations (`s-f`), start-to-start relations (`s-s`) etc.

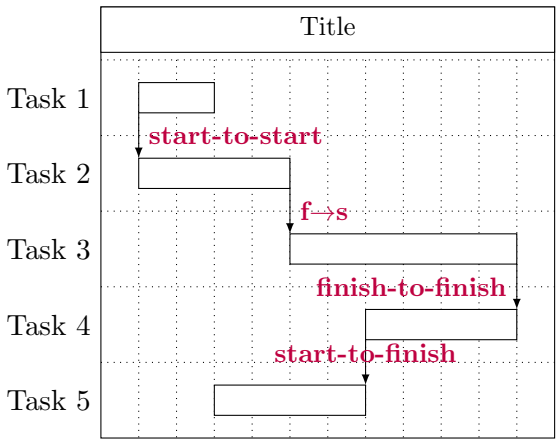
```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid, link={-latex, red}]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{2}{3} \\
    \ganttbar{Task 2}{2}{5} \\
    \ganttbar{Task 3}{6}{11} \\
    \ganttbar{Task 4}{8}{11}
    \ganttlink[s-s]{2}{2}{2}{3}
    \ganttlink[f-s]{5}{3}{6}{4}
    \ganttlink[f-f]{11}{4}{11}{5}
  \end{ganttchart}
\end{tikzpicture}
```



`/pgfgantt/link label={`⟨*link type key*⟩`}{`⟨*label text*⟩`}` (miscellaneous)
`/pgfgantt/link label font=`⟨*font commands*⟩ `\scriptsize\itshape`
`/pgfgantt/link label anchor=`⟨*anchor*⟩ `anchor=west`
Since straight links all look the same, a label indicates the respective relationship. You can redefine these labels with the `link label` key, which changes the ⟨*label text*⟩ for a ⟨*link type key*⟩.

The `link label font` key specifies the font for the label, `link label anchor` determines its placement (by default, the label appears to the right of the straight link's center).

```
\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}[vgrid, hgrid,
      link label font=\small\color{purple}\textbf,
      link label={f-s}{f$\to$s}]{12}
    \ganttitle{Title}{12} \\
    \ganttbar{Task 1}{2}{3} \\
    \ganttbar{Task 2}{2}{5} \\
    \ganttbar{Task 3}{6}{11} \\
    \ganttbar{Task 4}{8}{11} \\
    \ganttbar{Task 5}{4}{7}
    \ganttlink[s-s]{2}{2}{2}{3}
    \ganttlink[f-s]{5}{3}{6}{4}
    \ganttlink[f-f, link label anchor={anchor=east}]{11}{4}{11}{5}
    \ganttlink[s-f, link label anchor={anchor=base}]{8}{5}{7}{6}
  \end{ganttchart}
\end{tikzpicture}
```



2.11 Linked Bars and Linked Milestones

Since you'll most likely draw a lot of arrow-like links between bars and milestones, `pgfgantt` provides two convenient shortcuts for these tasks:

```
\ganttlinkedbar[⟨options⟩]{⟨label⟩}{⟨start time slot⟩}{⟨end time slot⟩}
\ganttlinkedmilestone[⟨options⟩]{⟨label⟩}{⟨time slot⟩}
```

These macros work exactly like the standard versions, but they additionally draw a link from the previous element to the bar or milestone. In the following example, the code on the left is equivalent to the code on the right.

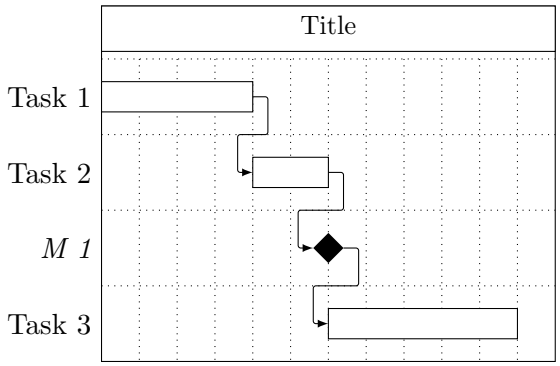
`\ganttlinkedbar`
`\ganttlinkedmilestone`

```
% Short version

\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid]{12}
    \ganttttitle{Title}{12} \\
    \ganttbar{Task 1}{1}{4} \\
    \ganttlinkedbar{Task 2}{5}{6} \\
    \ganttlinkedmilestone{M 1}{6} \\
    \ganttlinkedbar%
      [m-b]{Task 3}{7}{11}
  \end{ganttchart}
\end{tikzpicture}
```

```
% Long version

\begin{tikzpicture}[x=.5cm, y=1cm]
  \begin{ganttchart}%
      [vgrid, hgrid]{12}
    \ganttttitle{Title}{12} \\
    \ganttbar{Task 1}{1}{4} \\
    \ganttbar{Task 2}{5}{6} \\
    \ganttmilestone{M 1}{6} \\
    \ganttbar{Task 3}{7}{11}
    \ganttlink{4}{2}{5}{3}
    \ganttlink[b-m]{6}{3}{6}{4}
    \ganttlink[m-b]{6}{4}{7}{5}
  \end{ganttchart}
\end{tikzpicture}
```

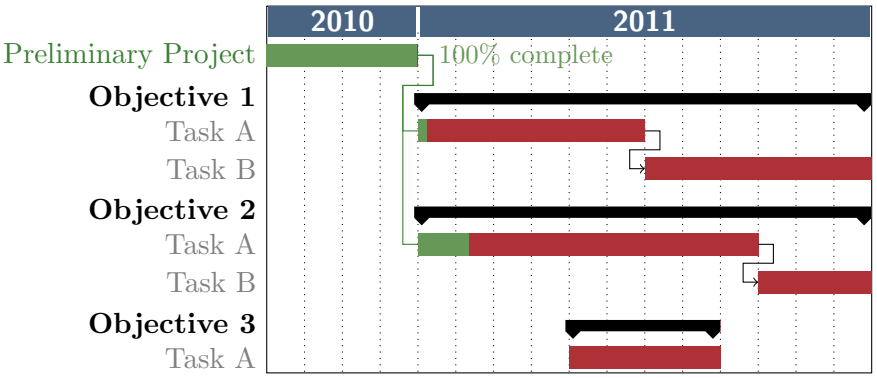


2.12 Style Examples

The first example plays around with colors and notably uses equal x- and y-vectors.

```
\begin{tikzpicture}[x=0.5cm,y=0.5cm]
  \begin{ganttchart}%
      [vgrid,
      title={draw=none, fill=RoyalBlue!50!black},
      title label font=\sffamily\bfseries\color{white},
      title label anchor={below=-1.6ex},
      title left shift=.05,
      title right shift=-.05,
      title height=.8,
      bar={draw=none, fill=OliveGreen!75},
      bar height=.6,
      bar label font=\normalsize\color{black!50},
      group right shift=0,
      group top shift=.3,
```

```
      group height=.3,
      group peaks={}{}{.2},
      incomplete={fill=Maroon},
      link={OliveGreen}]{16}
    \ganttttitle{2010}{4}
    \ganttttitle{2011}{12} \\
    \ganttbar%
        [progress=100, progress label font=\small\color{OliveGreen!75},
        progress label anchor={right=4pt},
        bar label font=\normalsize\color{OliveGreen}]%
      {Preliminary Project}{1}{4} \\
    \ganttlink[link mid=.4]{4}{2}{5}{4}
    \ganttlink[link mid=.159]{4}{2}{5}{7}
    \ganttset{progress label text={}, link={black, -to}}
    \ganttgroup{Objective 1}{5}{16} \\
    \ganttbar[progress=4]{Task A}{5}{10} \\
    \ganttlinkedbar[progress=0]{Task B}{11}{16} \\
    \ganttgroup{Objective 2}{5}{16} \\
    \ganttbar[progress=15]{Task A}{5}{13} \\
    \ganttlinkedbar[progress=0]{Task B}{14}{16} \\
    \ganttgroup{Objective 3}{9}{12} \\
    \ganttbar[progress=0]{Task A}{9}{12}
  \end{ganttchart}
\end{tikzpicture}
```

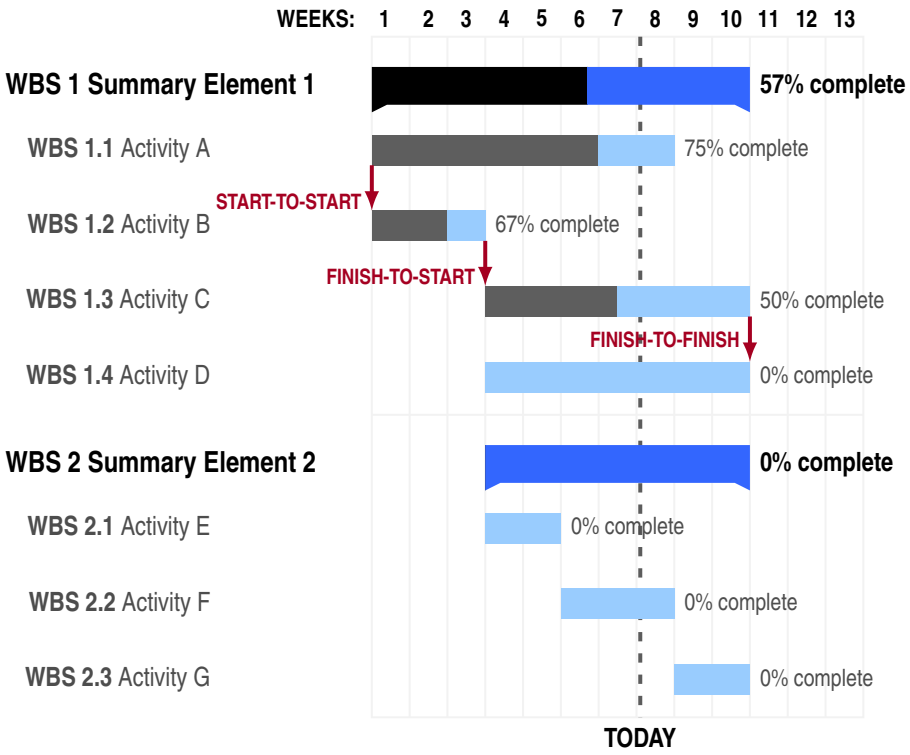


The second example demonstrates that `pgfgantt` is really flexible: Even an appearance quite different from the standard layout is possible. (More precisely, the code below tries to reproduce the Gantt chart from the English Wikipedia site, see `http://en.wikipedia.org/wiki/Gantt_chart`.)

```
\definecolor{barblue}{RGB}{153,204,254}
\definecolor{groupblue}{RGB}{51,102,254}
\definecolor{linkred}{RGB}{165,0,33}
```

```
\begin{tikzpicture}[x=0.5cm,y=1cm]
  \renewcommand\sfdefault{phv}
  \renewcommand\mddefault{mc}
  \renewcommand\bfdefault{bc}
  \sffamily
  \begin{ganttchart}%
      [canvas={fill=none, draw=black!5, line width=.75pt},
      hgrid style={draw=black!5, line width=.75pt},
      vgrid={draw=black!5, line width=.75pt},
      today=7.1,
      today rule={draw=black!64,
        dash pattern=on 3.5pt off 4.5pt, line width=1.5pt},
      today label={\small\bfseries TODAY},
      title={draw=none, fill=none},
      title label font=\bfseries\footnotesize,
      title label anchor={below=-2pt},
      include title in canvas=false,
      bar label font=\mdseries\small\color{black!70},
      bar label anchor={left=2cm},
      bar={draw=none, fill=black!63},
      bar incomplete={fill=barblue},
      progress label font=\mdseries\footnotesize\color{black!70},
      group incomplete={fill=groupblue},
      group left shift=0,
      group right shift=0,
      group height=.5,
      group peaks={0}{}{},
      group label anchor={left=.6cm},
      link={-latex, line width=1.5pt, linkred},
      link label font=\scriptsize\bfseries\color{linkred}\MakeUppercase,
      link label anchor={below left=-2pt and 0pt}
      ]{13}
    \ganttitle[title label anchor={below left=-2pt and -3pt}]%
      {WEEKS:\quad1}{1}
    \ganttitlelist{2,...,13}{1} \\
    \ganttgroup[progress=57, progress label font=\bfseries\small]%
      {WBS 1 Summary Element 1}{1}{10} \\
    \ganttbar[progress=75]{\textbf{WBS 1.1} Activity A}{1}{8} \\
    \ganttbar[progress=67]{\textbf{WBS 1.2} Activity B}{1}{3} \\
    \ganttbar[progress=50]{\textbf{WBS 1.3} Activity C}{4}{10} \\
    \ganttbar[progress=0]{\textbf{WBS 1.4} Activity D}{4}{10} \\[grid]
    \ganttgroup[progress=0, progress label font=\bfseries\small]%
      {WBS 2 Summary Element 2}{4}{10} \\
    \ganttbar[progress=0]{\textbf{WBS 2.1} Activity E}{4}{5} \\
    \ganttbar[progress=0]{\textbf{WBS 2.2} Activity F}{6}{8} \\
    \ganttbar[progress=0]{\textbf{WBS 2.3} Activity G}{9}{10}
    \ganttlink[s-s]{1}{3}{1}{4}
    \ganttlink[f-s]{3}{4}{4}{5}
    \ganttlink[f-f, link label anchor={left}]{10}{5}{10}{6}
```

```
  \end{ganttchart}
\end{tikzpicture}
```



3 Implementation

3.1 Packages

`pgfgantt` is modest in terms of dependencies: It only requires the Ti*k*Z package and some of its libraries.

```
\RequirePackage{tikz}
  \usetikzlibrary{arrows,backgrounds,calc,patterns,positioning}

```

3.2 Global Counters and Booleans

We define a number of global counters: `gtt@width` equals the number of time slots. `gtt@currentline` holds the current line; it starts from 0 and decreases. `gtt@lastline` is the line of the last element drawn (used by `\ganttlinkedbar`). `gtt@lasttitleline` equals the line of the title element drawn last. Furthermore, `gtt@lasttitleslot` corresponds to the x-coordinate of its right border.

The `\gtt@lastslot` macro contains the x-coordinate of the right border of the last drawn element. We use a macro instead of a counter in order to allow fractional coordinates.

The boolean `\ifgtt@intitle` is true at the start of a `ganttchart` environment and set to false as soon as the first non-title element is encountered.

```
\newcounter{gtt@width}
\newcounter{gtt@currentline}
\newcounter{gtt@lastline}
\newcounter{gtt@lasttitleline}
\newcounter{gtt@lasttitleslot}
\def\gtt@lastslot{0}
\newif\ifgtt@intitle

```

3.3 Macros for Key Management

`\ganttset` changes the current key path to `/pgfgantt/` and then executes the keys in its mandatory argument. `\ganttset`

```
\def\ganttset#1{\pgfqkeys{/pgfgantt}{#1}}

```

The following four auxiliary macros save us some code when we devise keys later on. Firstly, `\@gtt@keydef{`⟨*key*⟩`}` declares the key `/pgfgantt/`⟨*key*⟩, which saves its value in the macro `gtt@`⟨*key*⟩. `\@gtt@keydef`

```
\def\@gtt@keydef#1{%
  \pgfkeysdef{/pgfgantt/#1}{%
    \expandafter\def\csname gtt@#1\endcsname{##1}%
  }%
}
```

Secondly, `\@gtt@get{`⟨*key*⟩`}` retrieves the value stored by a ⟨*key*⟩. `\@gtt@get`

```
\def\@gtt@get#1{\csname gtt@#1\endcsname}
```

Thirdly, `\@gtt@stylekeydef{`⟨*key*⟩`}` declares a ⟨*key*⟩ which saves its value as the internal style key `/pgfgantt/`⟨*key*⟩`@style`. `\@gtt@stylekeydef`

```
\def\@gtt@stylekeydef#1{%
  \pgfkeysdef{/pgfgantt/#1}{%
    \pgfkeys{/pgfgantt/#1@style/.style={##1}}%
  }%
}
```

Finally, `\@gtt@linkkeydef{`⟨*link type*⟩`}{`⟨*start type*⟩`}{`⟨*end type*⟩`}{`⟨*label*⟩`}` declares the key `/pgfgantt/`⟨*start type*⟩`-`⟨*end type*⟩. Such keys change the appearance of a `\ganttlink`, since they store the ⟨*start type*⟩ (`b`, `m`, `s` or `f`) in `\gtt@link@starttype`, the ⟨*end type*⟩ in `\gtt@link@endtype` and the ⟨*link type*⟩ (either `0` for arrow-like `\@gtt@linkkeydef`

links or `1` for straight links) in `\gtt@link@type`. If ⟨*label*⟩ differs from `\relax`, it is saved for the respective link.

```
\def\@gtt@linkkeydef#1#2#3#4{%
  \pgfkeysdef{/pgfgantt/#2-#3}{%
    \def\gtt@link@type{#1}%
    \def\gtt@link@starttype{#2}%
    \def\gtt@link@endtype{#3}%
  }%
  \def\@tempa{#4}%
  \ifx\@tempa\relax\else\ganttset{link label={#2-#3}{#4}}\fi%
}
```

3.4 Option Declarations

`hgrid` checks whether its value is `false` and sets the boolean `\ifgtt@hgrid` accordingly. If the value is `true` or missing, the `hgrid style` should draw `dotted` lines.

hgrid
hgrid style

```
\@gtt@stylekeydef{hgrid style}
\newif\ifgtt@hgrid
\pgfkeysdef{/pgfgantt/hgrid}{%
  \def\@tempa{#1}%
  \def\@tempb{false}%
  \ifx\@tempa\@tempb%
    \gtt@hgridfalse%
  \else%
    \gtt@hgridtrue%
    \def\@tempb{true}%
    \ifx\@tempa\@tempb%
      \pgfkeys{/pgfgantt/hgrid style={dotted}}%
    \else%
      \pgfkeys{/pgfgantt/hgrid style={#1}}%
    \fi%
  \fi%
}
\pgfkeys{/pgfgantt/hgrid/.default=dotted}
```

Analogously, we declare `vgrid` and `vgridstyle`. `vgrid lines list` specifies a `\foreach` list that gives the vertical grid lines to be drawn.

vgrid
vgrid style
vgrid lines list

```
\@gtt@stylekeydef{vgrid style}
\newif\ifgtt@vgrid
\pgfkeysdef{/pgfgantt/vgrid}{%
  \def\@tempa{#1}%
  \def\@tempb{false}%
  \ifx\@tempa\@tempb%
    \gtt@vgridfalse%
```

```
  \else%
    \gtt@vgridtrue%
    \def\@tempb{true}%
    \ifx\@tempa\@tempb%
      \pgfkeys{/pgfgantt/vgrid style={dotted}}%
    \else%
      \pgfkeys{/pgfgantt/vgrid style={#1}}%
    \fi%
  \fi%
}
\pgfkeys{/pgfgantt/vgrid/.default=dotted}
\pgfkeysdef{/pgfgantt/vgrid lines list}{%
  \def\gtt@vgridlineslist{#1}%
}

```

Here is a set of keys related to the canvas ... canvas today today rule today label hgrid shift last line height

```
\@gtt@stylekeydef{canvas}
\@gtt@keydef{today}
\@gtt@stylekeydef{today rule}
\@gtt@keydef{today label}
\@gtt@keydef{hgrid shift}
\@gtt@keydef{last line height}

```

... and of keys that influence the title. Note that `\@gtt@keydef` cannot define `title list options`, since `\@gtt@titlelistoptions` is expanded after a `\foreach` statement, where `\@gtt@get` will not work.

title title label font title label anchor title list options title left shift title right shift title top shift title height

```
\@gtt@stylekeydef{title}
\@gtt@keydef{title label font}
\@gtt@stylekeydef{title label anchor}
\pgfkeysdef{/pgfgantt/title list options}{%
  \def\gtt@titlelistoptions{[#1]}%
}
\@gtt@keydef{title left shift}
\@gtt@keydef{title right shift}
\@gtt@keydef{title top shift}
\@gtt@keydef{title height}

```

`include title in canvas` is the only true boolean key in the package. include title in canvas

```
\newif\ifgtt@includetitle
\pgfkeys{/pgfgantt/include title in canvas/.is if=gtt@includetitle}

```

The `time slot modifier` option controls the semi-intelligent behaviour of the package regarding the conversion of title slots to x-coordinates. A value of `0` essentially means "interpret all end time slots as x-coordinates". time slot modifier

```
\@gtt@keydef{time slot modifier}
```

Some standard key declarations for bars ... bar

```
\@gtt@stylekeydef{bar}
\@gtt@keydef{bar label font}
\@gtt@stylekeydef{bar label anchor}
\@gtt@keydef{bar left shift}
\@gtt@keydef{bar right shift}
\@gtt@keydef{bar top shift}
\@gtt@keydef{bar height}
```

bar label font
bar label anchor
bar left shift
bar right shift
bar top shift
bar height

... and groups. group

```
\@gtt@stylekeydef{group}
\@gtt@keydef{group label font}
\@gtt@stylekeydef{group label anchor}
\@gtt@keydef{group left shift}
\@gtt@keydef{group right shift}
\@gtt@keydef{group top shift}
\@gtt@keydef{group height}
```

group label font
group label anchor
group left shift
group right shift
group top shift
group height

`gantt left peak` checks for each of its three values whether it is non-empty and only then changes the corresponding length macro. group left peak

```
\pgfkeysdefnargs{/pgfgantt/group left peak}{3}{%
  \def\@tempa{#1}%
  \ifx\@tempa\@empty\else\def\gtt@groupleftpeakmidx{#1}\fi%
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else\def\gtt@groupleftpeakinnerx{#2}\fi%
  \def\@tempa{#3}%
  \ifx\@tempa\@empty\else\def\gtt@groupleftpeaky{#3}\fi%
}
```

`group right peak` works similar, but a `-` also counts as an empty value (the reason for this will soon become apparent). group right peak

```
\pgfkeysdefnargs{/pgfgantt/group right peak}{3}{%
  \def\@tempa{#1}%
  \def\@tempb{-}%
  \ifx\@tempa\@empty\else%
    \ifx\@tempa\@tempb\else\def\gtt@grouprightpeakmidx{#1}\fi%
  \fi%
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else%
    \ifx\@tempa\@tempb\else\def\gtt@grouprightpeakinnerx{#2}\fi%
  \fi%
  \def\@tempa{#3}%
  \ifx\@tempa\@empty\else\def\gtt@grouprightpeaky{#3}\fi%
}
```

`group peaks` simultaneously sets `group left peak` and `group right peak`. In order to preserve the symmetry of the peaks, the key adds a negative sign (i. e., a group peaks

hyphen in the source code) to ⟨*groove x*⟩ and ⟨*inner x*⟩ of `group right peak`. Therefore, the latter key must interpret its first and second value as "empty" even if they contain a single hyphen.

```
\pgfkeysdefnargs{/pgfgantt/group peaks}{3}{%
  \ganttset{group left peak={#1}{#2}{#3}, group right peak={-#1}{-#2}{#3}}%
}
```

The keys below manage the progress elements. Note the way in which we declare `progress label text`, so that a `#1` in its value is replaced by the argument of `\gtt@progresslabeltext`.

progress
bar incomplete
group incomplete
incomplete
progress label lext
progress label font
progress label anchor

```
\@gtt@keydef{progress}
\@gtt@stylekeydef{bar incomplete}
\@gtt@stylekeydef{group incomplete}
\pgfkeysdef{/pgfgantt/incomplete}{%
  \ganttset{bar incomplete={#1}, group incomplete={#1}}%
}
\pgfkeysdef{/pgfgantt/progress label text}{%
  \def\gtt@progresslabeltext##1{#1}%
}
\@gtt@keydef{progress label font}
\@gtt@stylekeydef{progress label anchor}
```

Here are the declarations of the milestone-related keys.

milestone
milestone label font
milestone label anchor
milestone width
milestone height
milestone xshift
milestone yshift

```
\@gtt@stylekeydef{milestone}
\@gtt@keydef{milestone label font}
\@gtt@stylekeydef{milestone label anchor}
\@gtt@keydef{milestone width}
\@gtt@keydef{milestone height}
\@gtt@keydef{milestone xshift}
\@gtt@keydef{milestone yshift}
```

Next, we declare the keys that modify links.

link
link label font
link label anchor
link label
link mid
link bulge
link tolerance

```
\@gtt@stylekeydef{link}
\@gtt@keydef{link label font}
\@gtt@stylekeydef{link label anchor}
\pgfkeysdefnargs{/pgfgantt/link label}{2}{%
  \expandafter\def\csname gtt@link@#1@labeltext\endcsname{#2}%
}
\@gtt@keydef{link mid}
\@gtt@keydef{link bulge}
\@gtt@keydef{link tolerance}
```

The definitions of the eight link types follow.

b-b
b-m
m-b
m-m
s-s
s-f
f-s
f-f

```
\@gtt@linkkeydef 0 b b \relax
\@gtt@linkkeydef 0 b m \relax
```

```
\@gtt@linkkeydef 0 m b \relax
\@gtt@linkkeydef 0 m m \relax
\@gtt@linkkeydef 1 s s {start-to-start}
\@gtt@linkkeydef 1 s f {start-to-finish}
\@gtt@linkkeydef 1 f s {finish-to-start}
\@gtt@linkkeydef 1 f f {finish-to-finish}

```

Finally, we initialize all keys in order to define the commands that save their values.

```
\ganttset{
  canvas={fill=white},
  hgrid style=dotted,
  vgrid style=dotted,
  vgrid lines list={2,3,...,\value{gtt@width}},
  today=none,
  today rule={dashed, line width=1pt},
  today label=TODAY,
  hgrid shift=-.3,
  last line height=.7,
  title={fill=white},
  title label font=\small,
  title label anchor={anchor=mid},
  title left shift=0,
  title right shift=0,
  title top shift=0,
  title height=.6,
  title list options={var=\x, evaluate=\x},
  include title in canvas,
  time slot modifier=-1,
  bar={fill=white},
  bar label font=\normalsize,
  bar label anchor={anchor=east},
  bar left shift=0,
  bar right shift=0,
  bar top shift=0,
  bar height=.4,
  group={fill=black},
  group label font=\normalsize\bfseries,
  group label anchor={anchor=east},
  group left shift=-.1,
  group right shift=.1,
  group top shift=.1,
  group height=.2,
  group peaks={.2}{.4}{.1},
  progress=none,
  incomplete={fill=black!25},
  progress label text={#1\% complete},
  progress label font=\scriptsize,
  progress label anchor={anchor=west},
```

```
  milestone={fill=black},
  milestone label font=\normalsize\itshape,
  milestone label anchor={anchor=east},
  milestone width=.8,
  milestone height=.4,
  milestone xshift=0,
  milestone yshift=.2,
  link={-latex, rounded corners=1pt},
  link label font=\scriptsize\itshape,
  link label anchor={anchor=west},
  b-b,
  link mid=.5,
  link bulge=.4,
  link tolerance=0.6,
}
```

3.5 The Main Environment

At the beginning of a `ganttchart` environment, the keys in its optional argument are executed. `gtt@width` saves the environment's mandatory argument (i. e., the number of time slots). All line counters are set to 0. Since we expect a chart to start with at least one title element, `\ifgtt@intitle` is true. Within the environment, the control symbol `\\` is equivalent to `\ganttnewline` (similar to the syntax of a LaTeX table).

ganttchart
\\

```
\newenvironment{ganttchart}[2][]{%
  \ganttset{#1}%
  \setcounter{gtt@width}{#2}%
  \setcounter{gtt@currentline}{0}%
  \setcounter{gtt@lastline}{0}%
  \setcounter{gtt@lasttitleline}{0}%
  \gtt@intitletrue%
  \let\\\ganttnewline%
}{%
```

After the contents of the environment have been drawn, we add the canvas to the background layer. The `ganttchart` environment and all `\gantt...` macros save their x- and y-coordinates in local internal macros called `\x@left`, `\x@right`, `\y@upper` and `\y@lower`. The upper y-coordinate of the canvas is either zero or excludes the title lines if `include title in canvas` is false. The lower y-coordinate must take `last line height` into account.

\x@left
\x@right
\y@upper
\y@lower

```
  \begin{scope}[on background layer]%
    \ifgtt@includetitle%
      \def\y@upper{0}%
    \else%
      \def\y@upper{\value{gtt@lasttitleline}-\@gtt@get{hgrid shift}}%
    \fi%
```

```
    \def\y@lower{\value{gtt@currentline}-\@gtt@get{last line height}}%
    \draw[/pgfgantt/canvas@style]
      (0, \y@upper) rectangle
      (\value{gtt@width}, \y@lower);%
```

A `\foreach` loop iterates over all time slots given by the `vgrid lines list` key and adds vertical grid lines between them, considering `hgrid shift` for the upper y-coordinate.

```
    \ifgtt@vgrid%
      \foreach \t [evaluate=\t using \t-1]
          in \gtt@vgridlineslist {%
        \draw[/pgfgantt/vgrid style@style]
          (\t, \value{gtt@lasttitleline}-\@gtt@get{hgrid shift}) --
          (\t, \y@lower);%
      }%
    \fi%
```

Now, we draw the horizontal grid. If we exclude the title from the canvas, we omit the uppermost horizontal grid line since it would coincide with the canvas border.

```
    \ifgtt@hgrid%
      \ifgtt@includetitle\else\addtocounter{gtt@lasttitleline}{-1}\fi%
      \foreach \t [evaluate=\t]
          in {\value{gtt@lasttitleline},...,\value{gtt@currentline}} {%
        \draw[/pgfgantt/hgrid style@style]
          (0, \t-\@gtt@get{hgrid shift}) --
          (\value{gtt@width}, \t-\@gtt@get{hgrid shift});%
      }%
      \ifgtt@includetitle\else\stepcounter{gtt@lasttitleline}\fi%
    \fi%
```

The last task of `ganttchart` is to apply the `today` key if its value differs from `none`.

```
    \def\@tempa{none}%
    \ifx\gtt@today\@tempa\else%
      \draw[/pgfgantt/today rule@style]
        (\@gtt@get{today},
          \value{gtt@lasttitleline}-\@gtt@get{hgrid shift}) --
        (\@gtt@get{today}, \y@lower);%
      \node at (\@gtt@get{today}, \y@lower)
        [anchor=north] {\@gtt@get{today label}};%
    \fi%
  \end{scope}%
}

```

3.6 Starting a New Line

If the optional argument of `\ganttnewline` contains `grid`, this macro adds a horizontal grid rule between the current and the new line. Anyway, `\ganttnewline` `\ganttnewline`

decreases `gtt@currentline` and, if we are still in the title, `gtt@lasttitleline`. Since the new line starts at time slot zero, `gtt@lasttitleslot` is reset.

```
\newcommand\ganttnewline[1][]{%
  \def\@tempa{#1}%
  \def\@tempb{grid}%
  \ifx\@tempa\@tempb%
    \draw[/pgfgantt/hgrid style@style]
      (0, \value{gtt@currentline}-1-\@gtt@get{hgrid shift}) --
      (\value{gtt@width},
        \value{gtt@currentline}-1-\@gtt@get{hgrid shift});%
  \fi%
  \addtocounter{gtt@currentline}{-1}%
  \ifgtt@intitle\addtocounter{gtt@lasttitleline}{-1}\fi%
  \setcounter{gtt@lasttitleslot}{0}%
}
```

3.7 Title Elements

`\ganttttitle` draws a title element (i. e., a rectangle with a single node at its center). For reasons that will become clear below, the rectangle essentially starts at the x-coordinate stored in `gtt@lasttitleslot`. At the end of the macro, `gtt@lasttitleslot` and `gtt@lastline` are updated accordingly. `\ganttttitle`

Note that in order to keep key changes local, all macros that draw chart elements set the keys specified as their optional argument within a group.

```
\newcommand\ganttttitle[3][]{%
  \begingroup%
  \ganttset{#1}%
  \def\x@left{\value{gtt@lasttitleslot}+\@gtt@get{title left shift}}%
  \def\x@right{\value{gtt@lasttitleslot}+#3+\@gtt@get{title right shift}}%
  \def\y@upper{\value{gtt@currentline}-\@gtt@get{title top shift}}%
  \def\y@lower{\value{gtt@currentline}-\@gtt@get{title top shift}%
    -\@gtt@get{title height}}%
  \draw[/pgfgantt/title@style]
    (\x@left, \y@upper) rectangle
    (\x@right, \y@lower);%
  \@gtt@get{title label font}%
  \node at ($(\x@left,\y@upper)!.5!(\x@right,\y@lower)$)
    [/pgfgantt/title label anchor@style] {#2};%
  \addtocounter{gtt@lasttitleslot}{#3}%
  \setcounter{gtt@lastline}{\value{gtt@currentline}}%
  \endgroup%
}
```

`\ganttttitlelist` generates title elements by repeatedly calling `\ganttttitle`. Since the latter always starts after the last time slot occupied by the previous element, `\ganttttitlelist`

`\gantttitlelist` does not have to calculate the respective x-coordinates explicitly.

```
\newcommand\gantttitlelist[3][]{%
  \begingroup%
  \ganttset{#1}%
  \expandafter\foreach\gtt@titlelistoptions in {#2} {\gantttitle{\x}{#3}}%
  \endgroup%
}
```

3.8 Bars

`\ganttbar` begins by defining the usual coordinate macros. `\ganttbar`

```
\newcommand\ganttbar[4][]{%
  \begingroup%
  \ganttset{#1}%
  \def\x@left{#3+\@gtt@get{time slot modifier}+\@gtt@get{bar left shift}}%
  \def\x@right{#4+\@gtt@get{bar right shift}}%
  \def\y@upper{\value{gtt@currentline}-\@gtt@get{bar top shift}}%
  \def\y@lower{\value{gtt@currentline}-\@gtt@get{bar top shift}%
    -\@gtt@get{bar height}}%
```

If the first mandatory argument of `\ganttbar` is not empty, we print a label with its anchor at the left canvas border halfway between the upper and lower y-coordinate of the bar.

```
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else%
    \node at ($(0, \y@upper)!.5!(0, \y@lower)$)
      [/pgfgantt/bar label anchor@style] {\@gtt@get{bar label font}{#2}};%
  \fi%
```

`\gtt@pl@draw` saves the commands that will produce the progress label. This macro does nothing unless (a) the `progress` key differs from `none` and (b) `progress label text` differs from `\relax`. Otherwise, it creates a vertically centered node to the right of the bar. `\gtt@pl@draw`

```
  \def\@tempa{none}%
  \ifx\gtt@progress\@tempa%
    \def\gtt@progress{100}%
    \let\gtt@pl@draw\relax%
  \else
    \expandafter\ifx\gtt@progresslabeltext\relax\relax%
      \let\gtt@pl@draw\relax%
    \else%
      \def\gtt@pl@draw{%
        \node at ($(\x@right, \y@upper)!.5!(\x@right, \y@lower)$)
          [/pgfgantt/progress label anchor@style] {%
            \@gtt@get{progress label font}{%
              \@gtt@get{progresslabeltext}{\@gtt@get{progress}}%
```

```
            }%
          };%
      }%
    \fi%
  \fi%
```

In order to draw the left (complete) and right (incomplete) part of a progress bar, we clip the corresponding rectangles depending on the value of `progress`. Note that we turn off the border of these rectangles and draw it with an additional, third command.

```
  \begin{scope}%
    \clip (\x@left,\y@upper) rectangle
      ($(\x@left, \y@lower)!\gtt@progress/100!(\x@right, \y@lower)$);%
    \draw[/pgfgantt/bar@style, draw=none] (\x@left, \y@upper)
      rectangle (\x@right, \y@lower);%
  \end{scope}%
  \begin{scope}%
    \clip ($(\x@left,\y@upper)!\gtt@progress/100!(\x@right,\y@upper)$)
      rectangle (\x@right,\y@lower);%
    \draw[/pgfgantt/bar incomplete@style, draw=none]
      (\x@left, \y@upper) rectangle (\x@right, \y@lower);%
  \end{scope}%
  \draw[/pgfgantt/bar@style, fill=none]
    (\x@left, \y@upper) rectangle (\x@right, \y@lower);%
  \gtt@pl@draw%
```

Since the first bar clearly appears after the last line containing a title element, we set the boolean `\ifgtt@intitle` to false.

```
  \gdef\gtt@lastslot{#4}%
  \setcounter{gtt@lastline}{\value{gtt@currentline}}%
  \global\gtt@intitlefalse%
  \endgroup%
}
```

The shortcut version `\ganttlinkedbar` calls both `\ganttbar` and `\ganttlink`, taking care of the correct coordinates for the link. `\ganttlinkedbar`

```
\newcommand\ganttlinkedbar[4][]{%
  \begingroup%
  \ganttset{#1}%
  \ganttlink{\gtt@lastslot}{-\value{gtt@lastline}-1}%
    {#3}{-\value{gtt@currentline}-1}%
  \ganttbar{#2}{#3}{#4}%
  \endgroup%
}
```

3.9 Links

\ganttlink takes two completely different approaches to drawing links, depending on the link type defined by \@gtt@linkkeydef. \ganttlink

```
\newcommand\ganttlink[5][]{%
  \begingroup%
  \ganttset{#1}%
  \ifcase\gtt@link@type%
```

Link type 0 (arrow-like): The first and last coordinate of the link should touch the preceding or following element at the center of its right or left border, respectively. Therefore, we have to correct these coordinates if the link starts or ends at a milestone.

```
    \def\x@left{#2+\@gtt@get{bar right shift}}%
    \def\x@right{%
      #4+\@gtt@get{time slot modifier}+\@gtt@get{bar left shift}%
    }%
    \def\y@upper{-#3+1-\@gtt@get{bar top shift}/2-\@gtt@get{bar height}/2}%
    \def\y@lower{-#5+1-\@gtt@get{bar top shift}/2-\@gtt@get{bar height}/2}%
    \def\@tempa{m}
    \ifx\gtt@link@starttype\@tempa
      \def\x@left{%
        #2+\@gtt@get{milestone xshift}+\@gtt@get{milestone width}/2%
       }%
      \def\y@upper{-#3+1-\@gtt@get{milestone yshift}}%
    \fi
    \ifx\gtt@link@endtype\@tempa
      \def\x@right{%
        #4+\@gtt@get{milestone xshift}-\@gtt@get{milestone width}/2%
      }%
      \def\y@lower{-#5+1-\@gtt@get{milestone yshift}}%
    \fi
```

Now we check if the connected elements lie in the same row or not. In the latter case, \pgfmathparse yields 0.

```
    \pgfmathparse{#3==#5}%
    \ifcase\pgfmathresult%
```

Once again, two possibilities arise: Either the elements to be connected are at least separated by `link tolerance` time slots, in which case we draw a three-part arrow. Alternatively, the elements lie in adjacent time slots or even overlap, in which case we draw a five-part arrow.

```
      \pgfmathparse{%
        (#4+\@gtt@get{time slot modifier}-#2)>=\@gtt@get{link tolerance}%
      }%
      \ifcase\pgfmathresult%
        \draw[/pgfgantt/link@style]
          (\x@left, \y@upper) --
```

```
          (\x@left+\@gtt@get{link bulge}, \y@upper) --
          ($(\x@left+\@gtt@get{link bulge},\y@upper)!%
            \@gtt@get{link mid}!%
            (\x@left+\@gtt@get{link bulge},\y@lower)$) --
          ($(\x@right-\@gtt@get{link bulge},\y@upper)!%
            \@gtt@get{link mid}!%
            (\x@right-\@gtt@get{link bulge},\y@lower)$) --
          (\x@right-\@gtt@get{link bulge}, \y@lower) --
          (\x@right, \y@lower);%
      \else%
        \draw[/pgfgantt/link@style]
          (\x@left, \y@upper) --
          ($(\x@left,\y@upper)!\@gtt@get{link mid}!(\x@right,\y@upper)$) --
          ($(\x@left,\y@lower)!\@gtt@get{link mid}!(\x@right,\y@lower)$) --
          (\x@right, \y@lower);%
      \fi%
```

For elements that lie in the same row, we only draw an arrow if they are separated by at least one time slot.

```
    \else%
      \pgfmathparse{(#4+\@gtt@get{time slot modifier})==#2}%
      \ifcase\pgfmathresult%
        \draw[/pgfgantt/link@style]
          (\x@left, \y@upper) -- (\x@right, \y@lower);%
      \fi%
    \fi%
```

Link type `1` (straight): We calculate the start and end coordinates accordingly. Due to our conventions for specifying time slots instead of "real" x-coordinates, some subtleties arise.

```
  \else%
    \def\@tempa{f}%
    \ifx\gtt@link@starttype\@tempa%
      \def\x@left{#2+\@gtt@get{bar right shift}}%
    \else
      \def\@tempa{s}
      \ifx\gtt@link@starttype\@tempa
        \def\x@left{%
          #2+\@gtt@get{time slot modifier}+\@gtt@get{bar right shift}%
        }%
      \fi%
    \fi%
    \def\@tempa{f}%
    \ifx\gtt@link@endtype\@tempa%
      \def\x@right{#4+\@gtt@get{bar left shift}}%
    \else
      \def\@tempa{s}
      \ifx\gtt@link@endtype\@tempa
        \def\x@right{%
```

```
          #4+\@gtt@get{time slot modifier}+\@gtt@get{bar left shift}%
        }%
      \fi%
    \fi%
    \def\y@upper{-#3+1-\@gtt@get{bar height}}%
    \def\y@lower{-#5+1-\@gtt@get{bar top shift}}%
```

The remainder is straightforward: A vertical line connects the two elements; the label is anchored halfway between the start and end coordinates.

```
    \draw[/pgfgantt/link@style]
      (\x@left, \y@upper) --
      (\x@right, \y@lower);%
    \node at ($(\x@left, \y@upper)!.5!(\x@right, \y@lower)$)
      [/pgfgantt/link label anchor@style] {%
        \@gtt@get{link label font}{%
          \csname gtt@link@\gtt@link@starttype-\gtt@link@endtype%
            @labeltext\endcsname%
        }%
      };%
  \fi%
  \endgroup%
}
```

3.10 Groups

Groups and bars are quite similar. First, we define the usual coordinate macros. \ganttgroup

```
\newcommand\ganttgroup[4][]{%
  \begingroup%
  \ganttset{#1}%
  \def\x@left{%
    #3+\@gtt@get{time slot modifier}+\@gtt@get{group left shift}%
  }%
  \def\x@right{#4+\@gtt@get{group right shift}}%
  \def\y@upper{\value{gtt@currentline}-\@gtt@get{group top shift}}%
  \def\y@lower{\value{gtt@currentline}-\@gtt@get{group top shift}%
    -\@gtt@get{group height}}%
```

If the first mandatory argument of \ganttgroup is not empty, we print a label with its anchor at the left canvas border halfway between the upper and lower y-coordinate of the group.

```
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else%
    \node at ($(0, \y@upper)!.5!(0, \y@lower)$)
      [/pgfgantt/group label anchor@style]
      {\@gtt@get{group label font}{#2}};%
  \fi%
```

`\gtt@pl@draw` saves the commands that will produce the progress label. This macro does nothing unless (a) the `progress` key differs from none and (b) `progress label text` differs from `\relax`. Otherwise, it creates a vertically centered node to the right of the group.

```
\def\@tempa{none}%
\ifx\gtt@progress\@tempa%
  \def\gtt@progress{100}%
  \let\gtt@pl@draw\relax%
\else
  \expandafter\ifx\gtt@progresslabeltext\relax\relax%
    \let\gtt@pl@draw\relax%
  \else%
    \def\gtt@pl@draw{%
      \node at ($(\x@right, \y@upper)!.5!(\x@right, \y@lower)$)
        [/pgfgantt/progress label anchor@style] {%
          \@gtt@get{progress label font}{%
            \@gtt@get{progresslabeltext}{\@gtt@get{progress}}%
          }%
        };%
    }%
  \fi%
\fi%
```

In order to draw the left (complete) and right (incomplete) part of a progress group, we clip the corresponding polygons depending on the value of `progress`. Note that we turn off the border of these polygons and draw it with an additional, third command. The clipped area must include the highest peak, so we determine its height and store it in `\@maxpeak`.

`\@maxpeak`

```
\pgfmathsetmacro\@maxpeak{%
  \gtt@grouprightpeaky>\gtt@groupleftpeaky?%
  \gtt@grouprightpeaky:\gtt@groupleftpeaky%
}%
\begin{scope}%
  \clip (\x@left,\y@upper) rectangle
    ($(\x@left,\y@lower-\@maxpeak)!%
      \gtt@progress/100!%
      (\x@right,\y@lower-\@maxpeak)$);%
  \path[/pgfgantt/group@style,draw=none]
    (\x@left, \y@upper) --
    (\x@right, \y@upper) --
    (\x@right, \y@lower) --
    (\x@right+\gtt@grouprightpeakmidx, \y@lower-\gtt@grouprightpeaky) --
    (\x@right+\gtt@grouprightpeakinnerx, \y@lower) --
    (\x@left+\gtt@groupleftpeakinnerx, \y@lower) --
    (\x@left+\gtt@groupleftpeakmidx, \y@lower-\gtt@groupleftpeaky) --
    (\x@left, \y@lower) --
    cycle;%
\end{scope}%
```

```
  \begin{scope}%
    \clip ($(\x@left,\y@upper)!%
        \gtt@progress/100!%
        (\x@right,\y@upper)$)
      rectangle (\x@right,\y@lower-\@maxpeak);
    \path[/pgfgantt/group incomplete@style]
      (\x@left, \y@upper) --
      (\x@right, \y@upper) --
      (\x@right, \y@lower) --
      (\x@right+\gtt@grouprightpeakmidx, \y@lower-\gtt@grouprightpeaky) --
      (\x@right+\gtt@grouprightpeakinnerx, \y@lower) --
      (\x@left+\gtt@groupleftpeakinnerx, \y@lower) --
      (\x@left+\gtt@groupleftpeakmidx, \y@lower-\gtt@groupleftpeaky) --
      (\x@left, \y@lower) --
      cycle;%
  \end{scope}%
  \path[/pgfgantt/group@style,fill=none]
    (\x@left, \y@upper) --
    (\x@right, \y@upper) --
    (\x@right, \y@lower) --
    (\x@right+\gtt@grouprightpeakmidx, \y@lower-\gtt@grouprightpeaky) --
    (\x@right+\gtt@grouprightpeakinnerx, \y@lower) --
    (\x@left+\gtt@groupleftpeakinnerx, \y@lower) --
    (\x@left+\gtt@groupleftpeakmidx, \y@lower-\gtt@groupleftpeaky) --
    (\x@left, \y@lower) --
    cycle;%
  \gtt@pl@draw%
  \global\gtt@intitlefalse%
  \endgroup%
}

```

3.11 Milestones

`\ganttmilestone` has to calculate a single pair of coordinates, namely its center. `\ganttmilestone`

```
\newcommand\ganttmilestone[3][]{%
  \begingroup%
  \ganttset{#1}%
  \def\x@mid{#3+\@gtt@get{milestone xshift}}%
  \def\y@mid{\value{gtt@currentline}-\@gtt@get{milestone yshift}}%
```

If the first mandatory argument of `\ganttmilestone` is not empty, we print a label whose anchor lies on the left canvas border at the height of the milestone's center.

```
  \def\@tempa{#2}%
  \ifx\@tempa\@empty\else%
    \node at (0, \y@mid)
      [/pgfgantt/milestone label anchor@style]
      {\@gtt@get{milestone label font}{#2}};%
```

```
570   \fi%
```

Drawing the milestone itself is quite simple, since the `progress` key is irrelevant.

```
571   \path[/pgfgantt/milestone@style]
572     (\x@mid-\@gtt@get{milestone width}/2, \y@mid) --
573     (\x@mid, \y@mid-\@gtt@get{milestone height}/2) --
574     (\x@mid+\@gtt@get{milestone width}/2, \y@mid) --
575     (\x@mid, \y@mid+\@gtt@get{milestone height}/2) --
576     cycle;%
577   \gdef\gtt@lastslot{#3}%
578   \setcounter{gtt@lastline}{\value{gtt@currentline}}%
579   \global\gtt@intitlefalse%
580   \endgroup%
581 }
582
```

The shortcut version `\ganttlinkedmilestone` calls both `\ganttmilestone` and `\ganttlink`, taking care of the correct coordinates for the link. `\ganttlinkedmilestone`

```
583 \newcommand\ganttlinkedmilestone[3][]{%
584   \begingroup%
585   \ganttset{b-m,#1}%
586   \ganttlink{\gtt@lastslot}{-\value{gtt@lastline}-1}%
587     {#3}{-\value{gtt@currentline}-1}%
588   \ganttmilestone{#2}{#3}%
589   \endgroup%
590 }
```

4 Index

Numbers written in bold refer to the page where the corresponding entry is described; numbers in italic refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

O

P

S

T

5 Change History