Drawing Gantt Charts in LaTeX with Ti*k*Z

The `pgfgantt` Package

Wolfgang Skala[*]

v5.0
2018/01/10

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, for example, titles, bars, groups, milestones and different links between these elements. The appearance of the chart elements is highly customizable, and even new chart elements may be defined.

---

[*] Department of Biosciences, University of Salzburg, Austria; `Wolfgang.Skala@sbg.ac.at`

Contents

1 Introduction **3**

2 User Guide **4**

- 2.1 Overview . . . 4
- 2.2 Specifying Keys . . . 4
- 2.3 The Canvas . . . 5
- 2.4 Line Breaks between Chart Elements . . . 12
- 2.5 Titles . . . 14
- 2.6 Vertical rules . . . 23
- 2.7 Predefined Chart Elements . . . 24
  - 2.7.1 Options: Chart Element Appearance . . . 26
  - 2.7.2 Options: Label Formatting . . . 27
  - 2.7.3 Options: Chart Element Positioning . . . 29
  - 2.7.4 Options: Progress . . . 32
  - 2.7.5 New Node Shapes . . . 36
- 2.8 Defining Custom Chart Elements . . . 37
- 2.9 Links . . . 39
- 2.10 Style Examples . . . 48

3 Implementation **52**

- 3.1 Packages . . . 52
- 3.2 Macros for Key and Error Management . . . 52
- 3.3 The Horizontal and Vertical Grid . . . 53
- 3.4 Time Slot Formats . . . 55
- 3.5 The Main Environment . . . 59
- 3.6 Starting a New Line . . . 63
- 3.7 Vertical rules . . . 64
- 3.8 Titles . . . 65
- 3.9 Chart Elements . . . 73
- 3.10 Node Shapes . . . 82
- 3.11 Links . . . 88

4 Index **95**

5 Change History **101**

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 PGF and its Ti*k*Z frontend[2]. Besides, it provides a comprehensive (and portable) alternative to pst-gantt[3].

Requirements pgfgantt requires a *current* PGF installation. Note that the version number must at least be 2.10, dated October 25th, 2010. Furthermore, pgfgantt v5.0 and above is not fully downwards compatible.

Suggestions Please report any suggestions and improvements at the project's GitHub page (`https://github.com/skafdasschaf/latex-pgfgantt`).

Acknowledgements I would like to thank the following people for their ideas concerning new features: Christian Buhtz, Raphaël Clifford (University of Bristol), Jakob Døllner Mønster (Technical University of Denmark), Holger Karl (Universität Paderborn), Tristan Miller (Technische Universität Darmstadt), Scott Pakin (Los Alamos National Laboratory), Petr Pošík (Czech Technical University in Prague), Una Smith, Sascha Yousefi (Universität Freiburg), and Callum Webb.

---

[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/`

2 User Guide

2.1 Overview

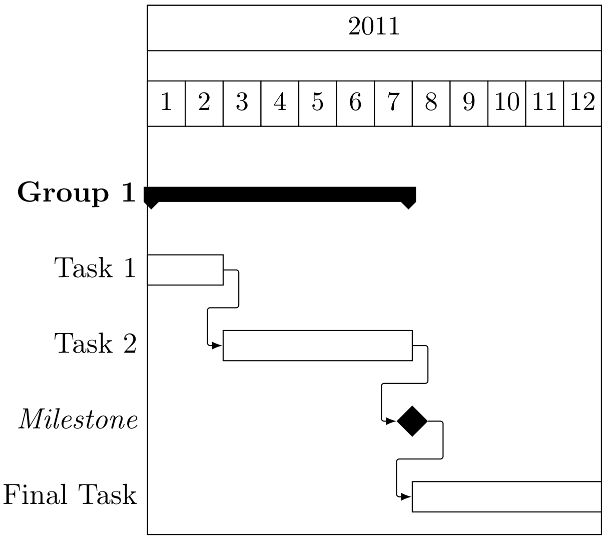
To load the package, simply put

```
\usepackage{pgfgantt}
```

into the document preamble.

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

```
\begin{ganttchart}{1}{12}
  \gantttitle{2011}{12} \\
  \gantttitlelist{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{elem2}{elem3}
  \ganttlink{elem3}{elem4}
\end{ganttchart}
```



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 height=.6]{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 within the current TeX group:

`\ganttset`

```
\ganttset{bar height=.6}
```

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 *lines*, which may contain one or more *title elements* (at the top) or *chart elements* (such as bars, groups and milestones). From left to right, the chart is divided into an integer number of *time slots* that represent the basic x-unit.

The `ganttchart` environment draws a single Gantt chart:

`ganttchart`

```
\begin{ganttchart}[⟨options⟩]{⟨start tss⟩}{⟨end tss⟩}
  ...
\end{ganttchart}
```

The environment has one optional argument, which specifies the ⟨*options*⟩ for the chart, and two mandatory arguments, which indicate the start and end time slot specifier. Although you will often put a `ganttchart` into a `tikzpicture` environment, you may actually use this environment on its own. pgfgantt checks whether a chart is surrounded by a `tikzpicture` and adds this environment if necessary.

`/pgfgantt/time slot format =`⟨*format*⟩ `simple`

Sets the ⟨*format*⟩ of time slot specifiers. A **time slot specifier** (abbreviated "tss") denotes a certain time slot along the horizontal axis. pgfgantt defines a range of formats:

- `simple` – positive integers (the single format used by pgfgantt prior to v4.0). See also the `time slot format/start date` key below.
  Examples: `1`, `3`, `24`
- `isodate` – dates in ISO-standard format (`yyyy-mm-dd`). In this format and any other, you may omit the leading zero if month or day are less than 10.
  Examples: `2013-03-14`, `2013-5-1`
- `isodate-yearmonth` – ISO-standard dates without days (`yyyy-mm`). Such dates are automatically converted to the first day of the respective month.
  Examples: `2013-03`, `2013-5`
- `isodate-year` – year only (`yyyy`). Such dates are automatically converted to the first day of January.
  Examples: `2013`, `2014`

- `little-endian` – Gregorian little-endian, i. e. day–month–year (the common German date format). Valid day/month and month/year separators are the hyphen (`-`), slash (`/`) and period (`.`). If you enter a two-digit year (for example, `13` instead of `2013`), it will be completed according to the value of `time slot format/base century` (see below).
  Examples: `14-03-2013`, `14/03/13`, `14.3.2013`
- `middle-endian` – middle-endian, i. e. month–day–year (the common US date format). For valid separators and automatic year completion, see *little-endian.*
  Examples: `03-14-2013`, `03/14/13`, `3.14.2013`
- `big-endian` – Gregorian big-endian, i. e. year–month–day (the ISO-standard order). For valid separators and automatic year completion, see *little-endian.*
  Examples: `2013-03-14`, `13/03/14`, `2013.3.14`

Two subkeys of `time slot format` let you configure `pgfgantt`'s behavior regarding automatic completion of abbreviated dates:

`/pgfgantt/time slot format/base century =`⟨*year*⟩ `2000`

Sets the century for auto-completion of two-digit years (used by the time slot formats `little-endian`, `middle-endian` and `big-endian`). Consequently, default settings convert a year like `13` to `2013`.

`/pgfgantt/time slot format/start date =`⟨*ISO-standard date*⟩ `2000-01-01`

Numbers denoting time slots in the `simple` format are internally converted to a date, where `1` is converted to ⟨*ISO-standard date*⟩, `2` to ⟨*ISO-standard date*⟩ + 1 etc.

Advanced users may add their own time slot formats:

`\newganttimeslotformat`

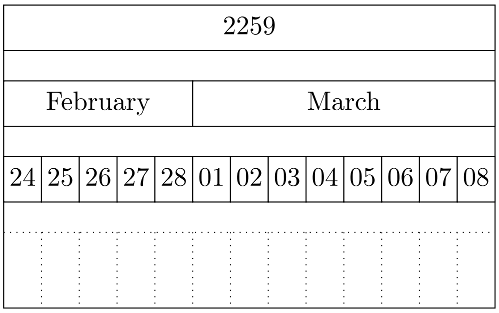
`\newganttimeslotformat{`⟨*name*⟩`}{`⟨*converter code*⟩`}`

Defines a new time slot format called ⟨*name*⟩. The ⟨*converter code*⟩ must convert the time slot specifier stored in `#1` to its corresponding Julian day number (see section 57 of the Ti*k*Z manual) and assign this number to the count register `#2`. The ⟨*converter code*⟩ is executed within a TEX group, so you may use temporary macros like `\@tempa`, counts like `\@tempcnta` etc.

For example, we would like to create a format called `stardate`, where dates are given as "⟨*year*⟩`.`⟨*day of year*⟩". Thus, we will enter 24th February 2259 as "`2259.55`". To this end, we write the following code:

```
\newganttimeslotformat{stardate}{%
  \def\decomposestardate##1.##2\relax{%
    \def\stardateyear{##1}\def\stardateday{##2}%
  }%
  \decomposestardate#1\relax%
  \pgfcalendardatetojulian{\stardateyear-01-01}{#2}%
```

```
 7    \advance#2 by-1\relax%
 8    \advance#2 by\stardateday\relax%
 9  }
10
11  \begin{ganttchart}[
12      hgrid,
13      vgrid,
14      time slot format=stardate
15    ]{2259.55}{2259.67}
16    \gantttitlecalendar{year, month=name, day} \\
17  \end{ganttchart}
```

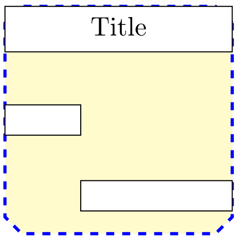


The macro `\decomposestardate` (lines 2–4) has two delimited arguments: The first one is delimited by a period and the second one by `\relax`. The call in line 5 decomposes the tss stored in `#1` and saves the day in `\stardateday` and the year in `\stardateyear`. `\pgfcalendardatetojulian` (section 57.1.1 of the Ti*k*Z manual) calculates the Julian date of the first day of `\stardateyear` and stores it in `#2` (line 6). We then subtract 1 from `#2` (line 7) and add the `\stardateday` (line 8).

`/pgfgantt/canvas ./style=`⟨*style*⟩ `shape=rectangle, draw, fill=white`

The `canvas` key changes the appearance of the canvas. ⟨*style*⟩ is a list of Ti*k*Z keys suitable for the ⟨*options*⟩ of a Ti*k*Z node (such as `shape=rectangle`, `fill` or `draw`; see chapter 16 of the Ti*k*Z manual). By default, the canvas is a white rectangle with a black frame.

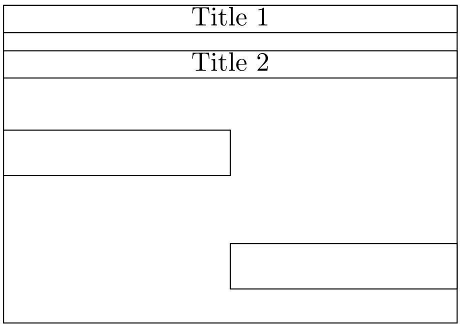
```
\begin{tikzpicture} % optional
  \begin{ganttchart}[
      canvas/.style=%
        {shape=chamfered rectangle, fill=yellow!25,
        draw=blue, dashed, very thick}
    ]{1}{6}
    \gantttitle{Title}{6} \\
    \ganttbar{}{1}{2} \\
    \ganttbar{}{3}{6}
  \end{ganttchart}
\end{tikzpicture} % optional
```



`/pgfgantt/x unit =`⟨*dimension*⟩ `.5cm`
`/pgfgantt/y unit title =`⟨*dimension*⟩ `1cm`
`/pgfgantt/y unit chart =`⟨*dimension*⟩ `1cm`

These keys specify the width of a time slot and the height of title or chart lines, respectively. Typically, the x/y-dimension ratio approximates $1 : 2$, and the line height is equal over the whole chart. Other dimensions are well possible, but you might have to change several spacing-related keys in order to obtain a pleasing chart.

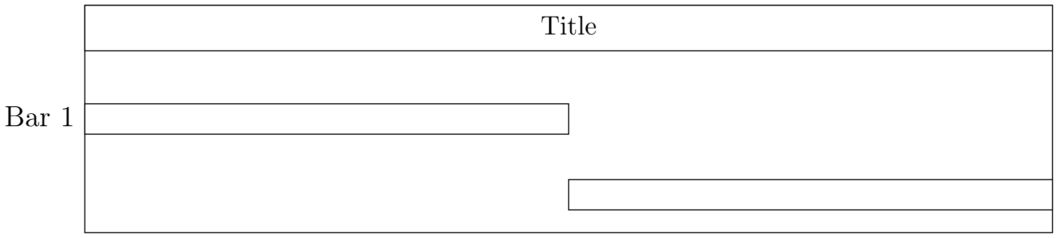
```
\begin{ganttchart}[
    x unit=1cm,
    y unit title=.6cm,
    y unit chart=1.5cm
  ]{1}{6}
  \gantttitle{Title 1}{6} \\
  \gantttitle{Title 2}{6} \\
  \ganttbar{}{1}{3} \\
  \ganttbar{}{4}{6}
\end{ganttchart}
```



`/pgfgantt/expand chart [=none |` ⟨*dimension*⟩`]` `none`

If the value of this key differs from `none`, the Gantt chart will expand horizontally to ⟨*dimension*⟩. Use this key to produce charts that automatically expand to the text width. Two LaTeX runs are required to calculate the correct size of the chart.

```
\begin{ganttchart}[
    expand chart=\textwidth
  ]{1}{6}
  \gantttitle{Title}{6} \\
  \ganttbar{Bar 1}{1}{3} \\
  \ganttbar{}{4}{6}
\end{ganttchart}
```



`/pgfgantt/hgrid` [=`false` | `true` | ⟨*style list*⟩] `false`
`/pgfgantt/hgrid style` `/.style=`⟨*style*⟩ `dotted`
`/pgfgantt/vgrid` [=`false` | `true` | ⟨*style list*⟩] `false`

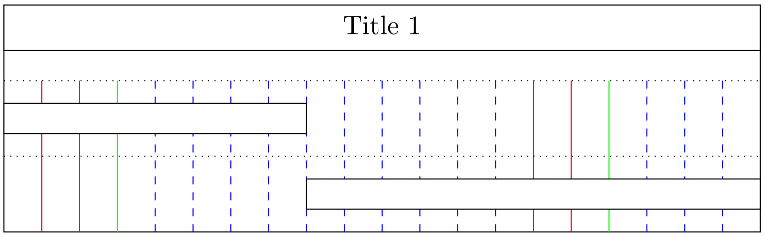
`hgrid` draws a horizontal grid which starts immediately below the last title element. The key can be specified in four different ways: Firstly, `hgrid=false` eliminates the horizontal grid. You may omit this declaration, since it is the default. Secondly, both `hgrid` and `hgrid=true` activate the horizontal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

`hgrid style` changes the style of single horizontal grid lines that are drawn with `\ganttnewline[grid]` (see section 2.4).

The `vgrid` key governs the vertical grid; otherwise, use it exactly like `hgrid`.

Style lists allow you to draw the grid lines in different styles. Each style list consists of several *style list items* separated by a comma. A style list item has the general syntax `*{`⟨*n*⟩`}{`⟨*style*⟩`}` and orders the package to repeat the ⟨*style*⟩ ⟨*n*⟩-times. (This syntax is reminiscent of column specifications in a `tabular` environment.) Thus, the list `*2{red}, *1{green}, *{10}{blue, dashed}` instructs `pgfgantt` to draw first two red vertical grid lines, then a green one and finally ten dashed blue lines. If any grid lines remain to be drawn at the end of the list, the package starts again at the beginning of the list.

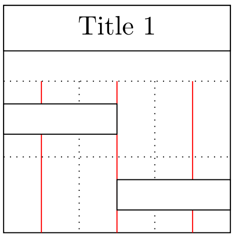
```
\begin{ganttchart}[
    hgrid=true,
    vgrid={*2{red}, *1{green}, *{10}{blue, dashed}}
  ]{1}{20}
  \gantttitle{Title 1}{20} \\
  \ganttbar{}{1}{8} \\
  \ganttbar{}{9}{20}
\end{ganttchart}
```

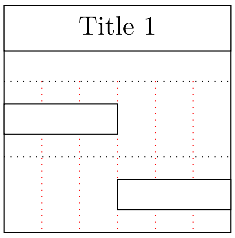


In most situations, you can omit the multiplier `*1`. Hence, the following style lists are equal:
`{*1{red}, *1{blue, dashed}}`
`{{red}, {blue, dashed}}`
`{red, {blue, dashed}}`
However, if you wish to use a single style comprising two or more keys for all grid lines, e. g. `red, dotted`, you *must* retain the multiplier (i. e., `{*1{red, dotted}}`).

```
% wrong code

\begin{ganttchart}[
    hgrid=true,
    vgrid={{red, dotted}}
  ]{1}{6}
  \gantttitle{Title 1}{6} \\
  \ganttbar{}{1}{3} \\
  \ganttbar{}{4}{6}
\end{ganttchart}
```

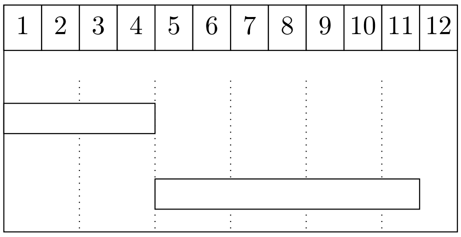
```
% correct code

\begin{ganttchart}[
    hgrid=true,
    vgrid={*1{red, dotted}}
  ]{1}{6}
  \gantttitle{Title 1}{6} \\
  \ganttbar{}{1}{3} \\
  \ganttbar{}{4}{6}
\end{ganttchart}
```





In a chart with many time slots, drawing vertical grid lines between all of them will lead to a confusing appearance. In such a case, you can pass an appropriate ⟨*style list*⟩ to `vgrid` in order to draw every second grid line, for example.

```
\begin{ganttchart}[vgrid={draw=none, dotted}]{1}{12}
  \gantttitlelist{1,...,12}{1} \\
  \ganttbar{}{1}{4} \\
  \ganttbar{}{5}{11}
\end{ganttchart}
```



`/pgfgantt/today` =⟨*tss*⟩ `none`

`/pgfgantt/today offset` =⟨*number*⟩ `1`

`/pgfgantt/today rule /.style=`⟨*style*⟩ `dashed, line width=1pt`

`/pgfgantt/today label` =⟨*text*⟩ `TODAY`

`/pgfgantt/today label font` =⟨*font commands*⟩ `\normalfont`

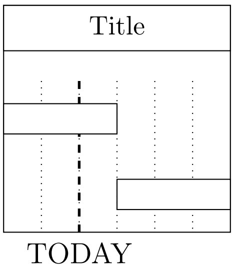
`/pgfgantt/today label node /.style=`⟨*style*⟩

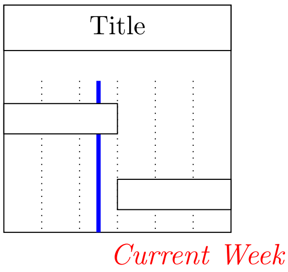
`anchor=north, font=\ganttvalueof{today label font}`

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 ⟨*tss*⟩. `today offset` determines the exact y-coordinate in the time slot and should lie between 0.0 (left border) and 1.0 (right border). The today rule appears in the ⟨*style*⟩ denoted by `today rule`. The node that contains the ⟨*text*⟩ given by `today label` appears below the rule. It is formatted by `today label font` and `today label node`.

```
\begin{ganttchart}[
    vgrid,
    today=2
  ]{1}{6}
  \gantttitle{Title}{6} \\
  \ganttbar{}{1}{3} \\
  \ganttbar{}{4}{6}
\end{ganttchart}
```

```
\begin{ganttchart}[
    vgrid,
    time slot format=isodate,
    today=2013-05-03,
    today offset=.5,
    today label=Current Week,
    today label node/.append style=%
      {anchor=north west},
    today label font=\itshape\color{red},
    today rule/.style=%
      {draw=blue, ultra thick}
  ]{2013-05-01}{2013-05-06}
  \gantttitle{Title}{6} \\
  \ganttbar{}{2013-05-01}{2013-05-03} \\
  \ganttbar{}{2013-05-04}{2013-05-06}
\end{ganttchart}
```





2.4 Line Breaks between Chart Elements

`\ganttnewline`

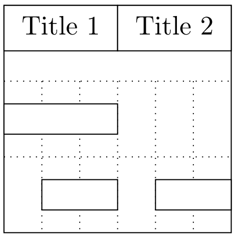
pgfgantt does not automatically begin a new line after finishing a chart element. Instead, you must insert an explicit line break with `\ganttnewline`.

`/pgfgantt/newline shortcut` =⟨*boolean*⟩ `true`

`\\`

If true, `\\` is defined as a shortcut for `\ganttnewline` within a `ganttchart` environment, so that the syntax is reminiscent of LaTeX's `tabular` enviroment.

```
\begin{ganttchart}[hgrid, vgrid]{1}{6}
  \gantttitle{Title 1}{3}
  \gantttitle{Title 2}{3} \\
  \ganttbar{}{1}{3} \ganttnewline
  \ganttbar{}{2}{3}
  \ganttbar{}{5}{6}
\end{ganttchart}
```

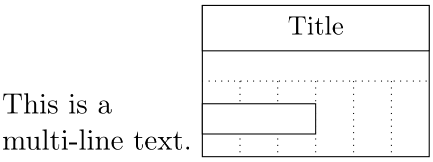


However, enabling this shortcut prevents you from entering multi-line node text (see section 16.4.3 of the Ti*k*Z manual). Thus, pgfgantt provides the macro `\ganttalignnewline` for breaking lines in the node text.

`\ganttalignnewline`

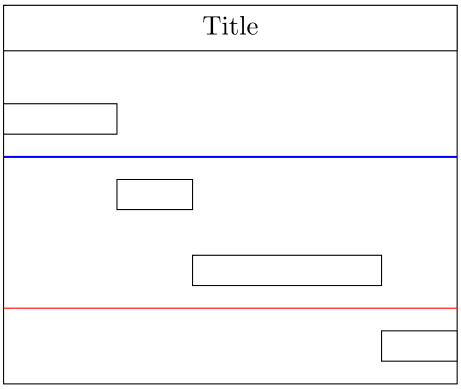
```
\begin{ganttchart}[
    hgrid,
    vgrid,
    newline shortcut=false,
    bar label node/.append style=%
      {align=left}
  ]{1}{6}
  \gantttitle{Title}{6} \ganttnewline
  \ganttbar{%
    This is a\\
    multi-line text.%
  }{1}{3}
\end{ganttchart}
```

```
\begin{ganttchart}[
    hgrid,
    vgrid,
    newline shortcut=true,
    bar label node/.append style=%
      {align=left}
  ]{1}{6}
  \gantttitle{Title}{6} \\
  \ganttbar{%
    This is a\ganttalignnewline
    multi-line text.%
  }{1}{3}
\end{ganttchart}
```



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 in `hgrid style` between the current and the new line. Alternatively, directly give the desired style as optional argument.

```
\begin{ganttchart}[hgrid style/.style=red]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{}{1}{3} \ganttnewline[thick, blue]
  \ganttbar{}{4}{5} \\
  \ganttbar{}{6}{10} \\[grid]
  \ganttbar{}{11}{12}
\end{ganttchart}
```



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 several titling commands.

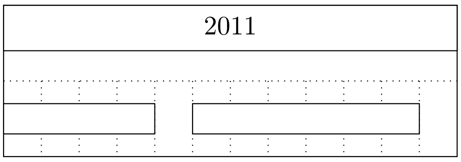
`\gantttitle` draws a single title element:

`\gantttitle`

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

The ⟨*label*⟩ appears in 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{ganttchart}[hgrid, vgrid]{1}{12}
  \gantttitle{2011}{12} \\
  \ganttbar{}{1}{4}
  \ganttbar{}{6}{11}
\end{ganttchart}
```



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

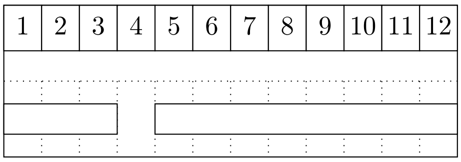
`\gantttitlelist`

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

This macro generates one title element for each element of the ⟨*pgffor list*⟩. The second mandatory argument specifies the ⟨*length of each element*⟩. Refer to section 56 of the Ti*k*Z manual for the detailed syntax for the ⟨*pgffor list*⟩.

A simple application is to draw twelve title elements that contain the numbers from 1 to 12. The ⟨*pgffor list*⟩ is `1,...,12`.

```
\begin{ganttchart}[hgrid, vgrid]{1}{12}
  \gantttitlelist{1,...,12}{1} \\
  \ganttbar{}{1}{3}
  \ganttbar{}{5}{12}
\end{ganttchart}
```



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

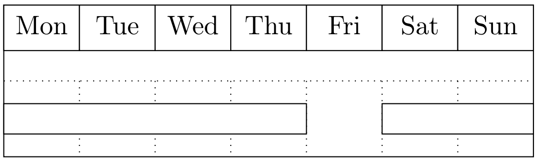
```
\gantttitle{1}{1} \gantttitle{2}{1} ... \gantttitle{12}{1} \\
```

As an advanced example, we will draw seven title elements containing the names of the weekdays (“Mon” to “Sun”). To this end, we introduce an additional key:

`/pgfgantt/title list options` =⟨*pgffor options*⟩ `var=\x, evaluate=\x`

Changes the ⟨*pgffor options*⟩ of the `\foreach` command called by `\gantttitlelist` (see section 56 of the Ti*k*Z manual). The macro that yields the labels to be printed by `\gantttitlelist` must be called `\x`.

```
\begin{ganttchart}[hgrid, vgrid, x unit=1cm]{1}{7}
  \gantttitlelist[
    title list options=%
      {var=\y, evaluate=\y as \x%
      using "\pgfcalendarweekdayshortname{\y}"}
  ]{0,...,6}{1} \\
  \ganttbar{}{1}{4}
  \ganttbar{}{6}{7}
\end{ganttchart}
```



While you actually may build any chart title with the two commands described previously, `\gantttitlecalendar` saves a lot of time when you wish to create elaborate calendars:

`\gantttitlecalendar`

```
\gantttitlecalendar[⟨options⟩]{⟨calendar lines⟩}
```

Prints a title calendar that spans the whole chart and contains one or more ⟨*calendar lines*⟩. The starred form of the macro prints a calendar from ⟨*start tss*⟩ to ⟨*end tss*⟩:

`\gantttitlecalendar*`

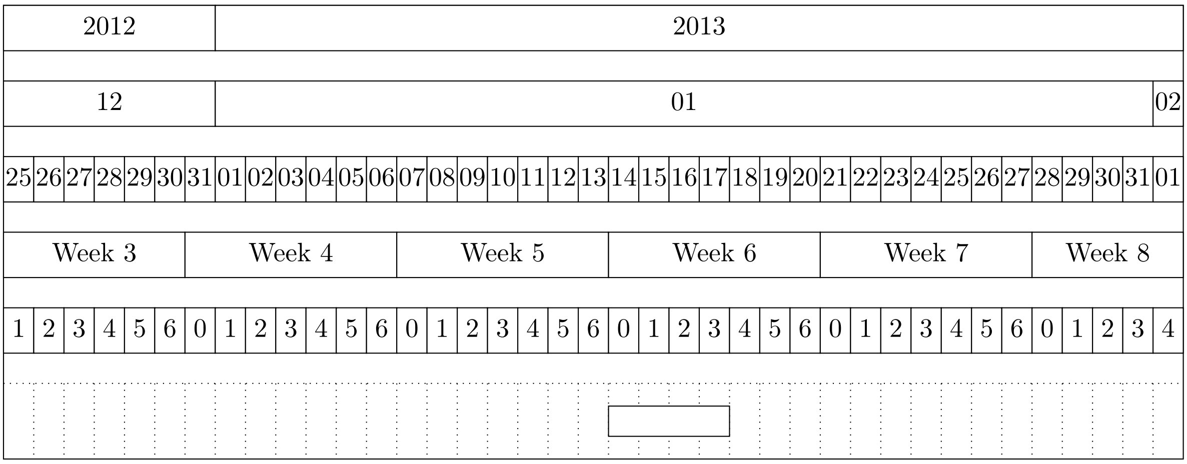
```
\gantttitlecalendar*[⟨options⟩]{⟨start tss⟩}{⟨end tss⟩}{⟨calendar lines⟩}
```

⟨*calendar lines*⟩ is a comma-separated list of line types:

Line type	⟨*output format*⟩	*Example output*
`decade`	n/a	2000s, 2010s, . . .
`year`	n/a	2012, 2013, . . .
`month [=`⟨*output format*⟩`]`	(none)	01, 02, . . . , 12
	`name`	January, February, . . .
	`shortname`	Jan, Feb, . . .
`week [=`⟨*number*⟩`]`	n/a	Week 1, Week 2, . . .
`weekday [=`⟨*output format*⟩`]`	(none)	0, 1, . . . , 6
	`name`	Monday, Tuesday, . . .
	`shortname`	Mon, Tue, . . .
`day`	n/a	01, 02, . . . , 31

The ⟨*number*⟩ for the `week` line type is the number of the first week in the calendar.

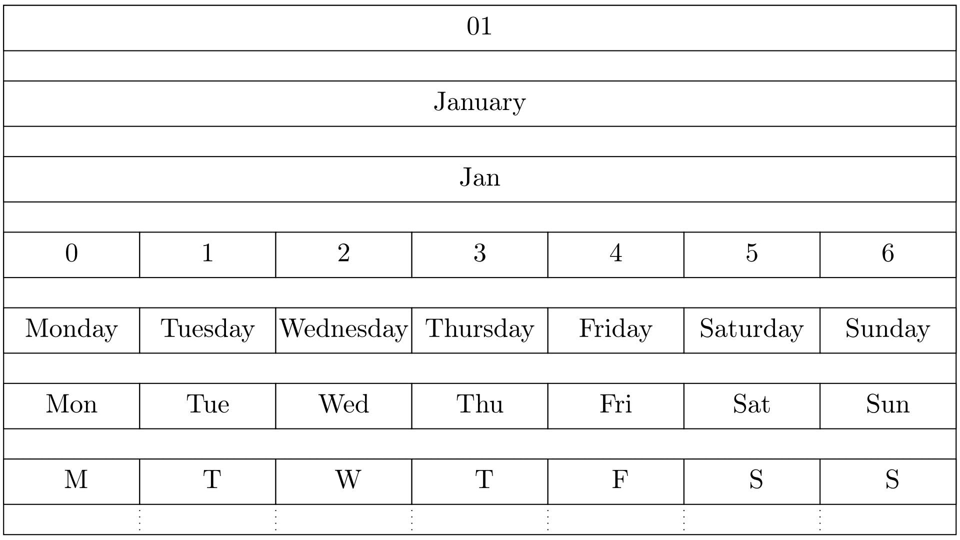
```
\begin{ganttchart}[
    hgrid,
    vgrid,
    x unit=4mm,
    time slot format=isodate
  ]{2012-12-25}{2013-02-01}
  \gantttitlecalendar{year, month, day, week=3, weekday} \\
  \ganttbar{}{2013-01-14}{2013-01-17}
\end{ganttchart}
```



You can easily add new output formats for `month` and `weekday`. The predefined ones use the macros described in section 57.1.3 of the Ti*k*Z manual. For example, `weekday=name` calls `\pgfcalendarweekdayname`. Thus, new macros called `\pgfcalendarmonth`⟨*output format*⟩ or `\pgfcalendarweekday`⟨*output format*⟩ will provide additional ⟨*output format*⟩s for `month` and `weekday`, respectively.

A weekday output format called `letter`, which displays a weekday as single letter, might be implemented as follows:

```
\def\pgfcalendarweekdayletter#1{%
  \ifcase#1M\or T\or W\or T\or F\or S\or S\fi%
}

\begin{ganttchart}[
    hgrid,
    vgrid,
    x unit=18mm,
    time slot format=little-endian
  ]{7.1.2013}{13.1.2013}
  \gantttitlecalendar*{7.1.2013}{13.1.2013}{
    month, month=name, month=shortname, weekday,
    weekday=name, weekday=shortname, weekday=letter
  }
\end{ganttchart}
```

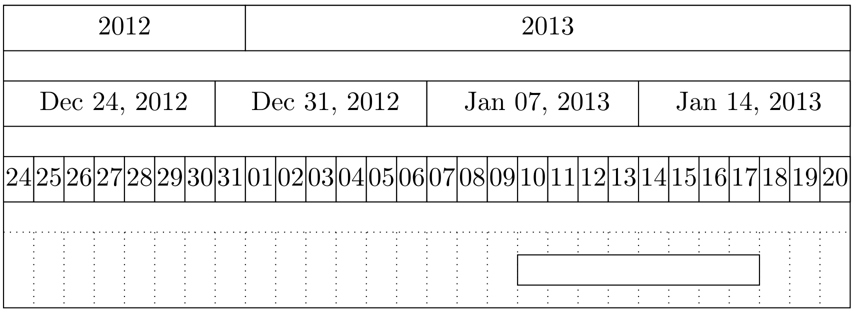


`/pgfgantt/calendar week text` =⟨*format*⟩ `Week~\currentweek`

Changes the text displayed in a week title element. In ⟨*format*⟩, four additional macros are available: `\currentweek` is the current week number; `\startyear`, `\startmonth` and `\startday` expand to the year, month and day of the current week's Monday.

`\currentweek`
`\startyear`
`\startmonth`
`\startday`

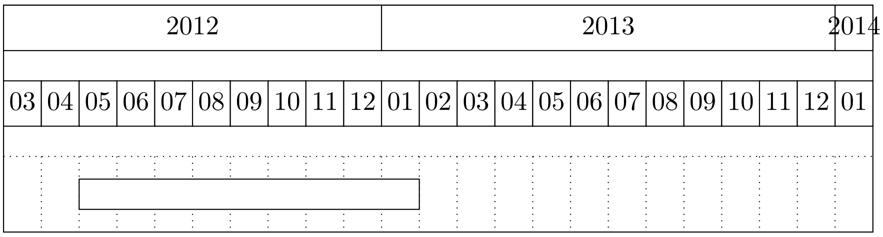
```
\ganttset{%
  calendar week text={%
    \pgfcalendarmonthshortname{\startmonth}~\startday, \startyear%
  }%
}
\begin{ganttchart}[
    hgrid,
    vgrid,
    x unit=4mm,
    time slot format=isodate
  ]{2012-12-24}{2013-01-20}
  \gantttitlecalendar{year, week, day} \\
  \ganttbar{}{2013-01-10}{2013-01-17}
\end{ganttchart}
```



/pgfgantt/**time slot unit** =day | month | year   day

By default, one *calendar day* is one time slot wide. With `time slot unit=month`, one *month* corresponds to one time slot. Consequently, in such calendars only `year` and `month` are sensible line types for `\gantttitlecalendar`, and the time slot format `isodate-yearmonth` is especially suited.

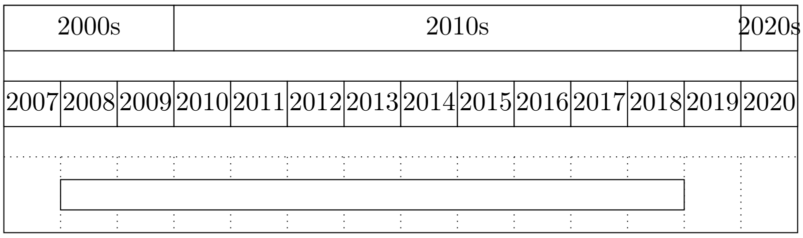
```
\begin{ganttchart}[
    hgrid,
    vgrid,
    time slot format=isodate-yearmonth,
    time slot unit=month
  ]{2012-03}{2014-1}
  \gantttitlecalendar{year, month} \\
  \ganttbar{}{2012-05}{2013-01}
\end{ganttchart}
```



With `time slot unit=year`, one *year* corresponds to one time slot. Consequently, in such calendars only `decade` and `year` are sensible line types for `\gantttitlecalendar`, and the time slot format `isodate-year` is especially suited.

```
\begin{ganttchart}[
    hgrid,
    vgrid,
    x unit=7.5mm,
    time slot format=isodate-year,
    time slot unit=year
```

```
  ]{2007}{2020}
  \gantttitlecalendar{decade, year} \\
  \ganttbar{}{2008}{2018}
\end{ganttchart}
```

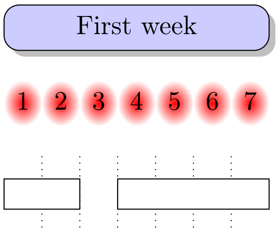


`/pgfgantt/title /.style=`⟨*style*⟩

`shape=rectangle, inner sep=0pt, draw, fill=white`

Sets the appearance of a title element.

```
\usetikzlibrary{shadows}
\usetikzlibrary{shadings}
  ...
\begin{ganttchart}[
    vgrid,
    canvas/.style={draw=none},
    title/.append style=%
      {fill=blue!20, rounded corners=2mm, drop shadow}
  ]{1}{7}
  \gantttitle{First week}{7} \\
  \gantttitlelist[
    title/.style={draw=none, inner color=red}
  ]{1,...,7}{1} \\
  \ganttbar{}{1}{2}
  \ganttbar{}{4}{7}
\end{ganttchart}
```



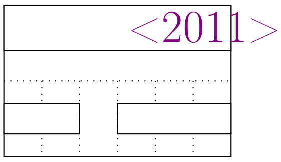
`/pgfgantt/title label font =`⟨*font commands*⟩ `\small`

`/pgfgantt/title label node /.style=`⟨*options*⟩
`anchor=center, font=\ganttvalueof{title label font}`

`/pgfgantt/title label text =`⟨*text*⟩ `\strut#1`

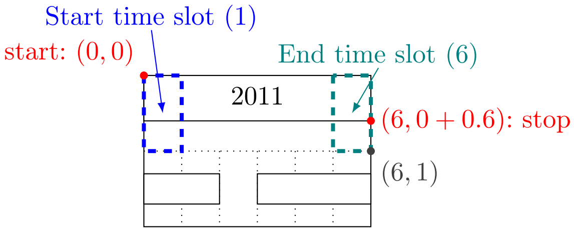
The ⟨*font commands*⟩ and ⟨*options*⟩ are applied to the title label node, which is positioned at the center of each title element. ⟨*text*⟩ should contain a single parameter token (`#1`), which is replaced by the first mandatory argument of `\gantttitle`. The `\strut` in the standard value ensures equal vertical spacing of the labels.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    title label font=\LARGE\color{violet},
    title label node/.append style={anchor=west},
    title label text=<#1>
  ]{1}{6}
  \gantttitle{2011}{6} \\
  \ganttbar{}{1}{2}
  \ganttbar{}{4}{6}
\end{ganttchart}
```

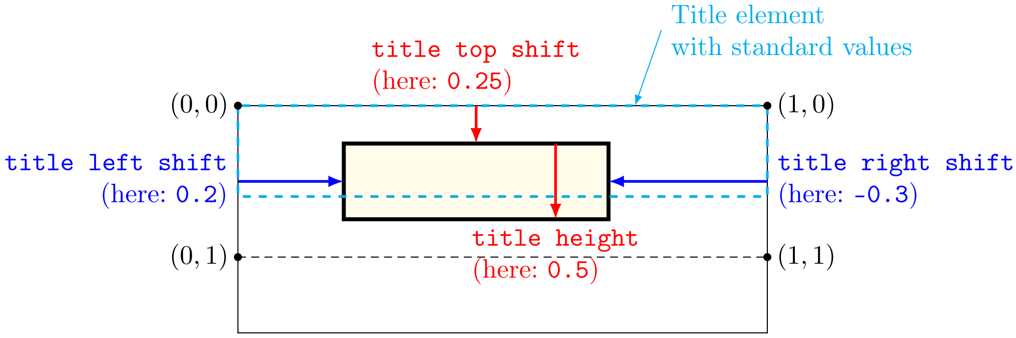


`/pgfgantt/title left shift =`⟨*factor*⟩ `0`
`/pgfgantt/title right shift =`⟨*factor*⟩ `0`
`/pgfgantt/title top shift =`⟨*factor*⟩ `0`
`/pgfgantt/title height =`⟨*factor*⟩ `0.6`

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:

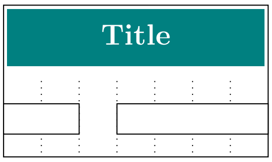


The figure below shows a Gantt chart with two lines and one (large) time slot and indicates the distances modified by these keys.



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

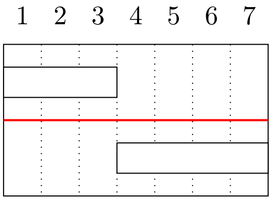
```
\begin{ganttchart}[
    vgrid,
    title/.style={fill=teal, draw=none},
    title label font=\color{white}\bfseries,
    title left shift=.1,
    title right shift=-.1,
    title top shift=.05,
    title height=.75
  ]{1}{7}
  \gantttitle{Title}{7} \\
  \ganttbar{}{1}{2}
  \ganttbar{}{4}{7}
\end{ganttchart}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



`/pgfgantt/include title in canvas` =⟨*boolean*⟩ `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{ganttchart}[
    hgrid={*1{draw=red, thick}},
    vgrid,
    y unit title=.5cm,
    title/.style={draw=none, fill=none},
    include title in canvas=false
  ]{1}{7}
  \gantttitlelist{1,...,7}{1} \\
  \ganttbar{}{1}{3} \\
  \ganttbar{}{4}{7}
\end{ganttchart}
```



2.6 Vertical rules

A *vertical rule* indicates an important date like a deadline. Such rules represent a generalization of the today rule and are drawn by the `\ganttvrule` macro: `\ganttvrule`

```
\ganttvrule[⟨options⟩]{⟨label⟩}{⟨tss⟩}
```

This macro draws a ⟨*label*⟩ed vertical rule at the given ⟨*tss*⟩.

`/pgfgantt/vrule offset =`⟨*number*⟩ `1`

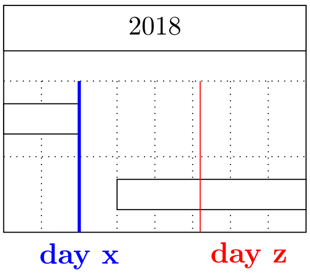
`/pgfgantt/vrule /.style=`⟨*style*⟩ `dashed, line width=1pt`

`/pgfgantt/vrule label font =`⟨*font commands*⟩ `\normalfont`

`/pgfgantt/vrule label node /.style=`⟨*style*⟩
`anchor=north, font=\ganttvalueof{vrule label font}`

`vrule offset` determines the exact *y*-coordinate in the time slot and should lie between 0.0 (left border) and 1.0 (right border). The vertical rule appears in the ⟨*style*⟩ denoted by `vrule`. The label is formatted by `vrule label font` and `vrule label node`.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    vrule/.style={very thick, blue},
    vrule label font=\bfseries
  ]{1}{8}
  \gantttitle{2018}{8} \\
  \ganttbar{}{1}{2} \\
  \ganttbar{}{4}{8}
  \ganttvrule{day x}{2}
  \ganttvrule[
    vrule/.append style={red, thin},
    vrule offset=.2,
    vrule label node/.append style={anchor=north west}
  ]{day z}{6}
\end{ganttchart}
```



2.7 Predefined Chart Elements

pgfgantt predefines three chart elements:

1. *Bars* indicate the duration of a task or one of its parts. `\ganttbar`

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

2. *Groups* combine several subtasks (represented by bars) into a single task. `\ganttgroup`

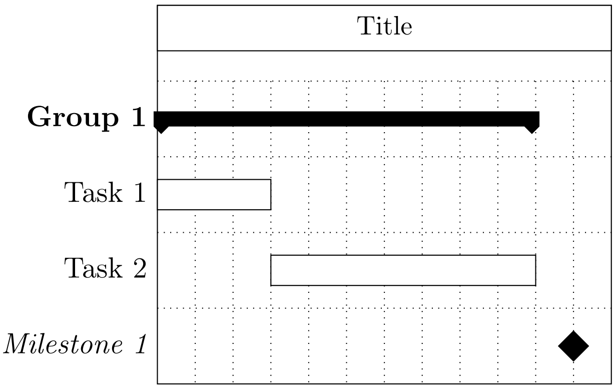
   `\ganttgroup[⟨options⟩]{⟨label⟩}{⟨start tss⟩}{⟨end tss⟩}`

3. *Milestones* signify that an important task has been completed or that a crucial goal has been reached. `\ganttmilestone`

   `\ganttmilestone[⟨options⟩]{⟨label⟩}{⟨tss⟩}`

Each of these macros draws a ⟨*label*⟩ed chart element from the ⟨*start tss*⟩ to the ⟨*end tss*⟩ (or at the given ⟨*tss*⟩ in case of `\ganttmilestone`).

```
\begin{ganttchart}[vgrid, hgrid]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1}{10} \\
  \ganttbar{Task 1}{1}{3} \\
  \ganttbar{Task 2}{4}{10} \\
  \ganttmilestone{Milestone 1}{11}
\end{ganttchart}
```



\ganttlinkedbar
\ganttlinkedgroup
\ganttlinkedmilestone

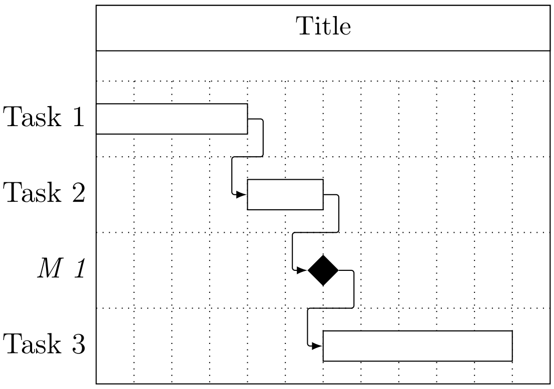
For each predefined chart element, there is also a macro that additionally draws a link from the previous element. Otherwise, these macros work exactly like the standard versions:

```
\ganttlinkedbar[⟨options⟩]{⟨label⟩}{⟨start tss⟩}{⟨end tss⟩}
\ganttlinkedgroup[⟨options⟩]{⟨label⟩}{⟨start tss⟩}{⟨end tss⟩}
\ganttlinkedmilestone[⟨options⟩]{⟨label⟩}{⟨tss⟩}
```

In the following example, the code on the left is equivalent to the code on the right.

```
% Short version

\begin{ganttchart}[
    vgrid,
    hgrid
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{1}{4} \\
  \ganttlinkedbar{Task 2}{5}{6} \\
  \ganttlinkedmilestone{M 1}{6} \\
  \ganttlinkedbar{Task 3}{7}{11}
\end{ganttchart}
```

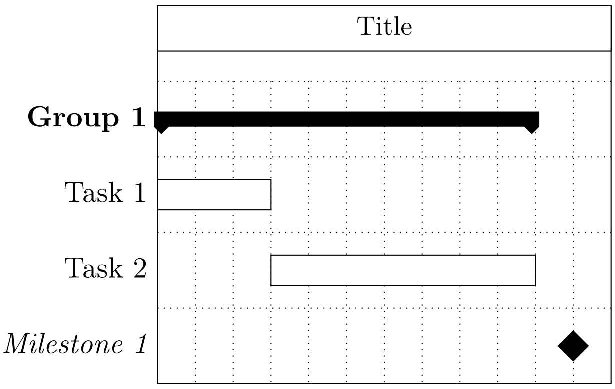
```
% Long version

\begin{ganttchart}[
    vgrid,
    hgrid
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{1}{4} \\
  \ganttbar{Task 2}{5}{6} \\
  \ganttmilestone{M 1}{6} \\
  \ganttbar{Task 3}{7}{11}
  \ganttlink{elem0}{elem1}
  \ganttlink{elem1}{elem2}
  \ganttlink{elem2}{elem3}
\end{ganttchart}
```



/pgfgantt/chart element start border =left | right left

Determines which border of the start time slot a chart element touches. `left` is the behavior usually expected, while `right` strictly interprets the start time slot as an x-coordinate.

```
\begin{ganttchart}[vgrid, hgrid, chart element start border=right]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{0}{10} \\
  \ganttbar{Task 1}{0}{3} \\
  \ganttbar{Task 2}{3}{10} \\
  \ganttmilestone{Milestone 1}{11}
\end{ganttchart}
```



2.7.1 Options: Chart Element Appearance

The following options are similar for all predefined (and user-defined) chart elements:

/pgfgantt/bar /.style=⟨*style*⟩
`shape=ganttbar, inner sep=0pt, draw, fill=white`

`/pgfgantt/group /.style=⟨style⟩`

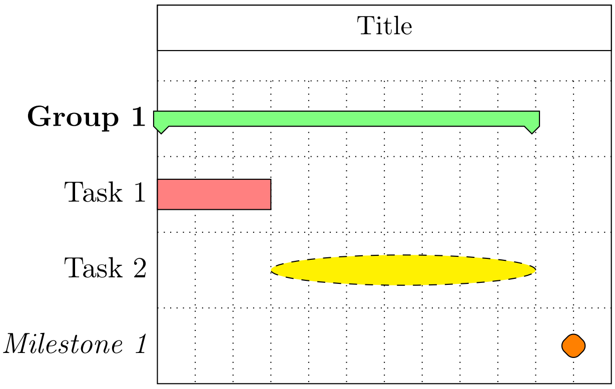
`shape=ganttgroup, inner sep=0pt, fill=black`

`/pgfgantt/milestone /.style=⟨style⟩`

`shape=ganttmilestone, inner sep=0pt, draw, fill=black`

Determines the appearance of the chart element. The shapes `ganttbar`, `ganttgroup` and `ganttmilestone` are described below.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    bar/.append style={fill=red!50},
    group/.append style={draw=black, fill=green!50},
    milestone/.append style={fill=orange, rounded corners=3pt}
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1}{10} \\
  \ganttbar{Task 1}{1}{3} \\
  \ganttbar[
    bar/.append style={shape=ellipse, fill=yellow, dashed}
  ]{Task 2}{4}{10} \\
  \ganttmilestone{Milestone 1}{11}
\end{ganttchart}
```



2.7.2 Options: Label Formatting

`/pgfgantt/bar label text =⟨text⟩`	`\strut#1`
`/pgfgantt/group label text =⟨text⟩`	`\strut#1`
`/pgfgantt/milestone label text =⟨text⟩`	`\strut#1`
`/pgfgantt/bar label font =⟨font commands⟩`	`\normalsize`
`/pgfgantt/group label font =⟨font commands⟩`	`\bfseries`
`/pgfgantt/milestone label font =⟨font commands⟩`	`\itshape`

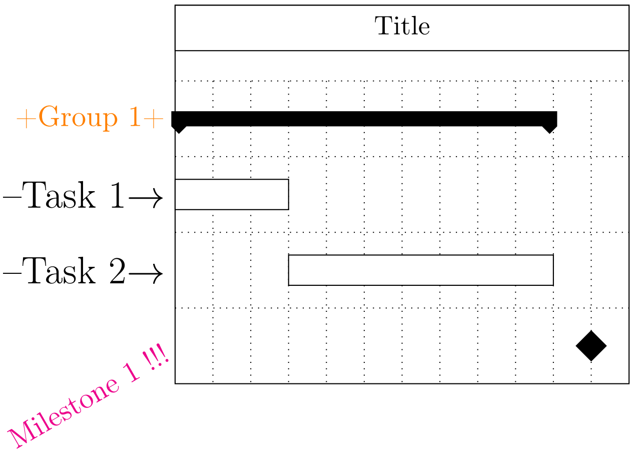
/pgfgantt/**bar label node** ./style=⟨*options*⟩
`anchor=east, font=\ganttvalueof{bar label font}`

/pgfgantt/**group label node** ./style=⟨*options*⟩
`anchor=east, font=\ganttvalueof{group label font}`

/pgfgantt/**milestone label node** ./style=⟨*options*⟩
`anchor=east, font=\ganttvalueof{milestone label font}`

The `... label text` keys configure the label ⟨*text*⟩ next to each chart element. Each of these keys should contain a single parameter token (`#1`), which is replaced by the first mandatory argument of `\ganttbar` etc. The `\strut` in the standard value ensures equal vertical spacing of the labels. The ⟨*font commands*⟩ of `... label font` and the ⟨*options*⟩ of `... label node` are applied to the label node at the left border of the chart (see `inline` below).

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    bar label font=\Large,
    bar label text={--#1$\rightarrow$},
    group label font=\color{orange},
    group label text={+#1+},
    milestone label font=\color{magenta},
    milestone label node/.append style={rotate=30},
    milestone label text={#1 !!!}
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1}{10} \\
  \ganttbar{Task 1}{1}{3} \\
  \ganttbar{Task 2}{4}{10} \\
  \ganttmilestone{Milestone 1}{11}
\end{ganttchart}
```



`/pgfgantt/inline =⟨boolean⟩` (default: `false`)

`/pgfgantt/bar inline label anchor =⟨anchor⟩` (default: `center`)

`/pgfgantt/group inline label anchor =⟨anchor⟩` (default: `center`)

`/pgfgantt/milestone inline label anchor =⟨anchor⟩` (default: `center`)

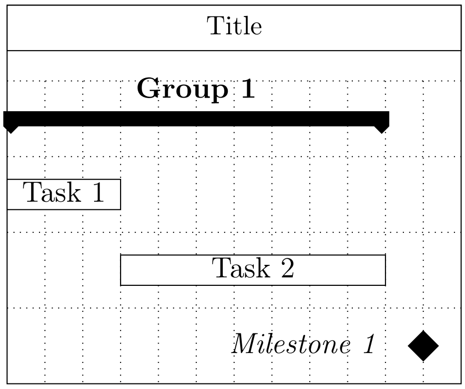
`/pgfgantt/bar inline label node /.style=⟨options⟩`
`anchor=center, font=\ganttvalueof{bar label font}`

`/pgfgantt/group inline label node /.style=⟨options⟩`
`anchor=south, font=\ganttvalueof{group label font}`

`/pgfgantt/milestone inline label node /.style=⟨options⟩`
`anchor=south, font=\ganttvalueof{milestone label font}`

If two or more chart elements appear in a single line, their labels will overlap at the left border of the chart. Thus, you can place the label adjacent to a chart element by setting the boolean key `inline` to `true`. This key instructs the package to draw the label node at the `... inline label anchor` of the respective chart element and apply the ⟨*options*⟩ given by `... inline label node`.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    inline,
    milestone inline label node/.append style={left=5mm}
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1}{10} \\
  \ganttbar{Task 1}{1}{3} \\
  \ganttbar{Task 2}{4}{10} \\
  \ganttmilestone{Milestone 1}{11}
\end{ganttchart}
```

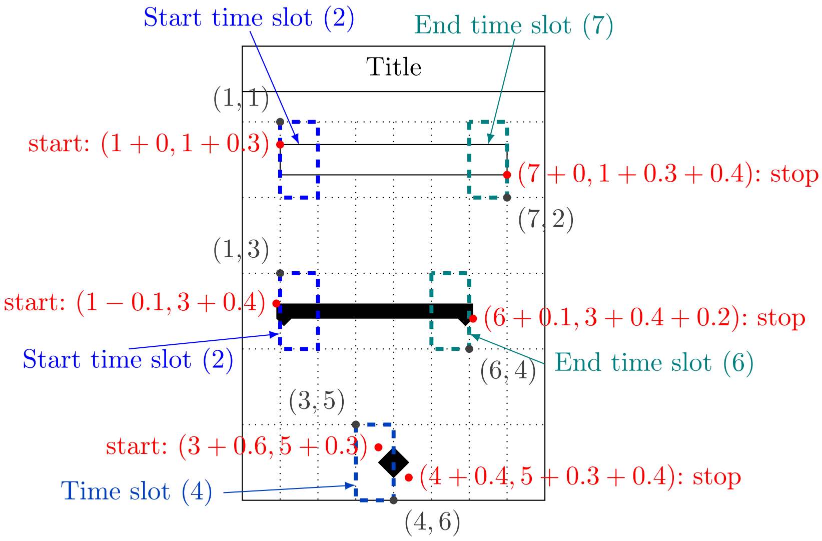


2.7.3 Options: Chart Element Positioning

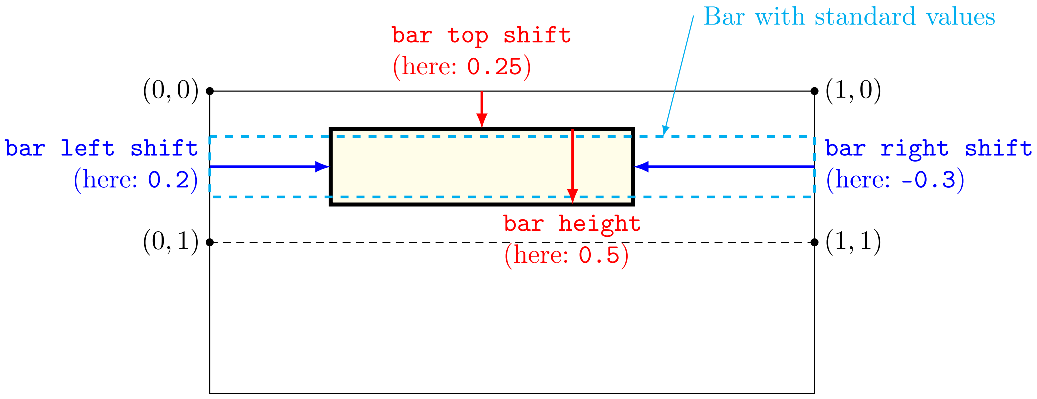
`/pgfgantt/bar left shift =⟨factor⟩` (default: `0`)

`/pgfgantt/bar right shift =`⟨*factor*⟩ `0`
`/pgfgantt/bar top shift =`⟨*factor*⟩ `.3`
`/pgfgantt/bar height =`⟨*factor*⟩ `.4`
`/pgfgantt/group left shift =`⟨*factor*⟩ `-.1`
`/pgfgantt/group right shift =`⟨*factor*⟩ `.1`
`/pgfgantt/group top shift =`⟨*factor*⟩ `.4`
`/pgfgantt/group height =`⟨*factor*⟩ `.2`
`/pgfgantt/milestone left shift =`⟨*factor*⟩ `.6`
`/pgfgantt/milestone right shift =`⟨*factor*⟩ `.4`
`/pgfgantt/milestone top shift =`⟨*factor*⟩ `.3`
`/pgfgantt/milestone height =`⟨*factor*⟩ `.4`

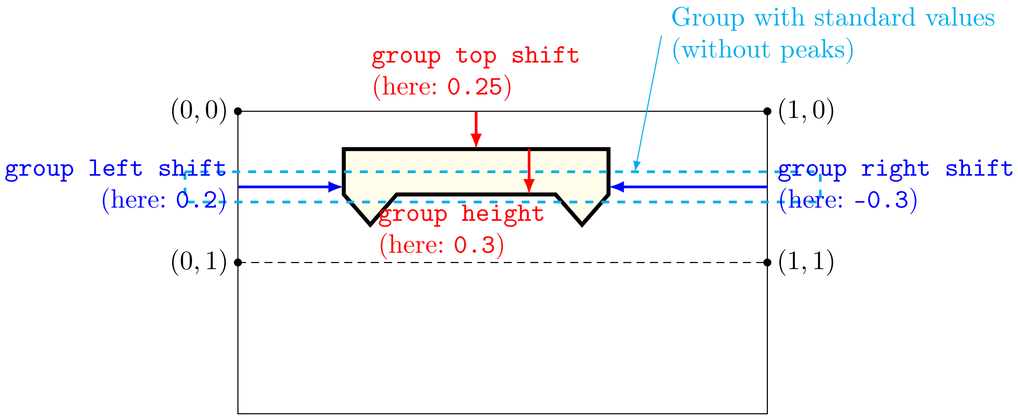
Shift the coordinates of a chart element's borders (`... shift`) and change its height (`... height`).



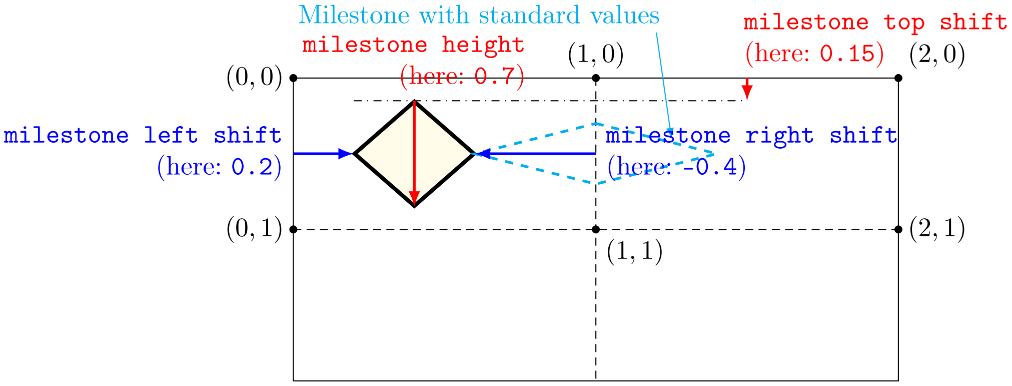
The three following figures illustrate the distances modified by these keys. The first figure shows a Gantt chart with a bar, two lines and one time slot.



The second one shows a Gantt chart with a group, two lines and one time slot.

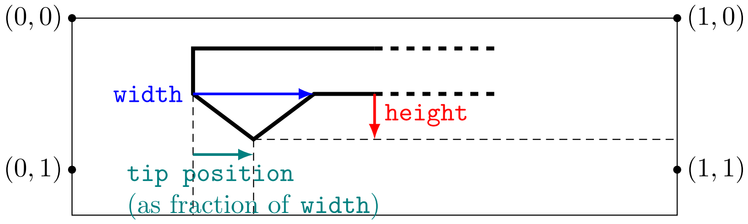


The third one shows a Gantt chart with a milestone, two lines and two time slots.



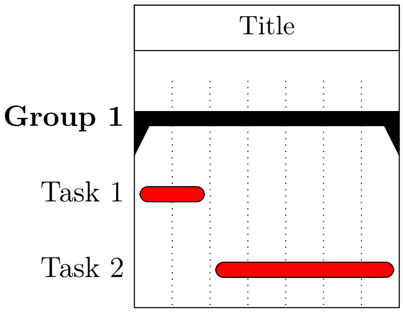
`/pgfgantt/group right peak tip position` =⟨*fraction*⟩ `0.5`
`/pgfgantt/group right peak width` =⟨*factor*⟩ `0.4`
`/pgfgantt/group right peak height` =⟨*factor*⟩ `0.1`
`/pgfgantt/group left peak tip position` =⟨*fraction*⟩ `0.5`
`/pgfgantt/group left peak width` =⟨*factor*⟩ `0.4`
`/pgfgantt/group left peak height` =⟨*factor*⟩ `0.1`
`/pgfgantt/group peaks tip position` =⟨*fraction*⟩ (none)
`/pgfgantt/group peaks width` =⟨*factor*⟩ (none)
`/pgfgantt/group peaks height` =⟨*factor*⟩ (none)

Change the appearance of the peaks at both ends of a group. By default, both the left and right peak are 0.4 units wide and 0.1 units high, their tips lie between the peak sides. The `group peaks ...` keys set the dimensions for both peaks simultaneously. The figure below exemplifies the keys that apply to the left peak.



For example, you might devise the following layout: Bars are small and rounded; they do not touch the borders of their start and end time slots. Groups stay within the start and end time slot, and the peaks are more acute.

```
\begin{ganttchart}[
    vgrid,
    bar/.append style={fill=red, rounded corners=3pt},
    bar left shift=.15,
    bar right shift=-.15,
    bar top shift=.4,
    bar height=.2,
    group left shift=0,
    group right shift=0,
    group peaks tip position=0,
    group peaks height=.4
  ]{1}{7}
  \gantttitle{Title}{7} \\
  \ganttgroup{Group 1}{1}{7} \\
  \ganttbar{Task 1}{1}{2} \\
  \ganttbar{Task 2}{3}{7}
\end{ganttchart}
```



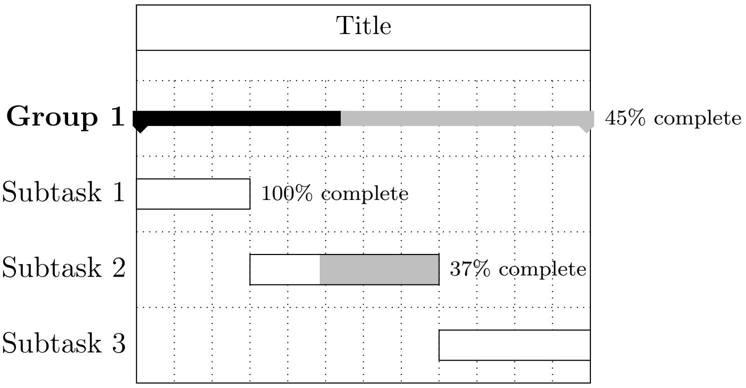
2.7.4 Options: Progress

The *progress* of a chart element illustrates the extent to which this element has been completed.

`/pgfgantt/progress` =`none` | `today` | ⟨*number*⟩ `none`

Indicates that a chart element is ⟨*number*⟩ percent complete. The value `none` turns progress calculations off.

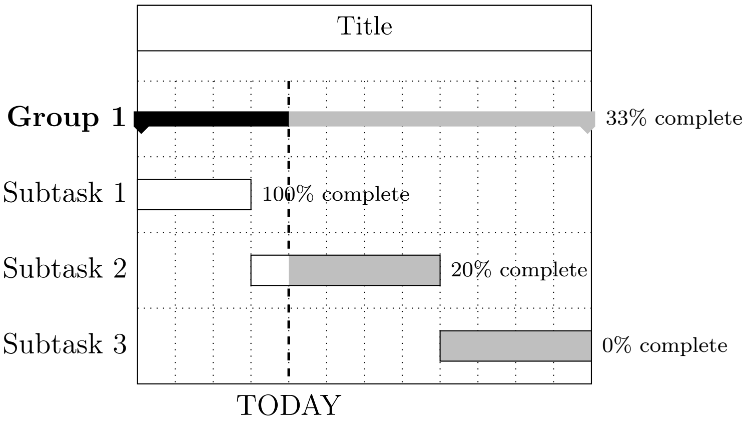
```
\begin{ganttchart}[vgrid, hgrid]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup[progress=45]{Group 1}{1}{12} \\
  \ganttbar[progress=100]{Subtask 1}{1}{3} \\
  \ganttbar[progress=37]{Subtask 2}{4}{8} \\
  \ganttbar[progress=none]{Subtask 3}{9}{12}
\end{ganttchart}
```



The value `today` instructs `pgfgantt` to calculate progress according to the value of the `today` key. Thus, if the current date T is earlier than the start date S of a chart element, its progress is 0 %; if the current date is later than the end date E of a chart element, its progress is 100 %; otherwise, its progress P is calculated according to

$$P = \frac{T - S}{E - S} \times 100\,\% \tag{1}$$

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    time slot format=little-endian,
    progress=today,
    today=4.5.13
  ]{1.5.13}{12.5.13}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1.5.13}{12.5.13} \\
  \ganttbar{Subtask 1}{1.5.13}{3.5.13} \\
  \ganttbar{Subtask 2}{4.5.13}{8.5.13} \\
  \ganttbar{Subtask 3}{9.5.13}{12.5.13}
\end{ganttchart}
```



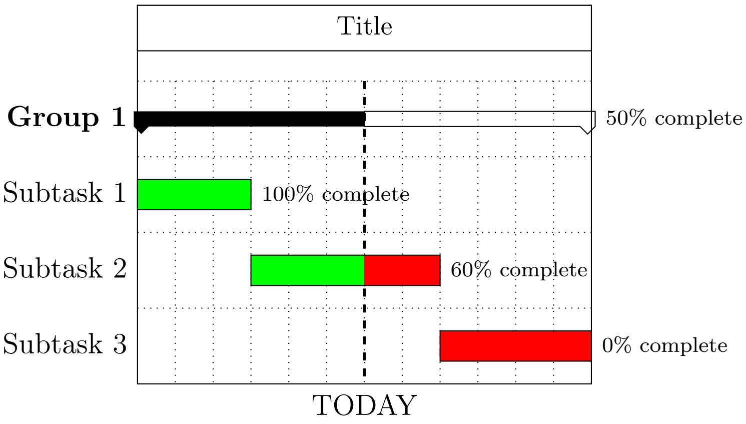
`/pgfgantt/bar incomplete /.style=⟨style⟩` `/pgfgantt/bar, fill=black!25`

`/pgfgantt/group incomplete /.style=⟨style⟩`
`/pgfgantt/group, fill=black!25`

`/pgfgantt/milestone incomplete /.style=⟨style⟩`
`/pgfgantt/milestone, fill=black!25`

If P is the progress of a chart element, $P\,\%$ of its area (starting from the left) appear in the basic style (i. e., `bar`, `group`, . . .), while the remainder is drawn in style `bar incomplete`, `group incomplete` etc.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    time slot format=isodate,
    today=2013-04-06,
    progress=today,
    bar/.append style={fill=green},
    bar incomplete/.append style={fill=red},
    group incomplete/.append style={draw=black,fill=none}
  ]{2013-04-01}{2013-04-12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{2013-04-01}{2013-04-12} \\
  \ganttbar{Subtask 1}{2013-04-01}{2013-04-03} \\
  \ganttbar{Subtask 2}{2013-04-04}{2013-04-08} \\
  \ganttbar{Subtask 3}{2013-04-09}{2013-04-12}
\end{ganttchart}
```



/pgfgantt/progress label text =⟨*text*⟩
\pgfmathprintnumber[precision=0, verbatim]{#1}\% complete

/pgfgantt/bar progress label anchor =⟨*anchor*⟩ east

/pgfgantt/bar progress label font =⟨*font commands*⟩ \scriptsize

/pgfgantt/bar progress label node /.style=⟨*options*⟩
anchor=west, font=\ganttvalueof{bar progress label font}

/pgfgantt/group progress label anchor =⟨*anchor*⟩ east

/pgfgantt/group progress label font =⟨*font commands*⟩ \scriptsize

/pgfgantt/group progress label node /.style=⟨*options*⟩
anchor=west, font=\ganttvalueof{group progress label font}

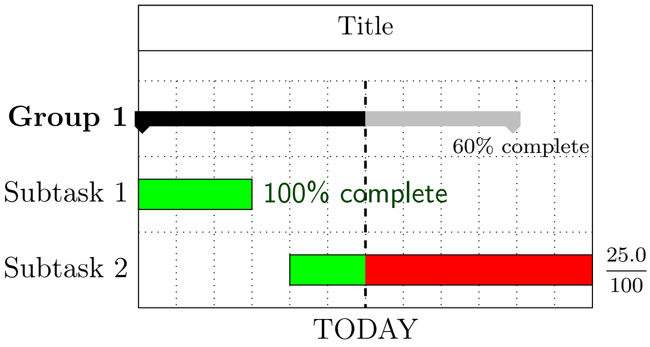
/pgfgantt/milestone progress label anchor =⟨*anchor*⟩ center

/pgfgantt/milestone progress label font =⟨*font commands*⟩ \scriptsize

/pgfgantt/milestone progress label node /.style=⟨*options*⟩
anchor=west, font=\ganttvalueof{milestone progress label font}

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 (possibly calculated) value of `progress`. The progress label node is drawn at the `... progress label anchor` of the respective chart element, with the ⟨*font commands*⟩ given by `... progress label font` and the ⟨*options*⟩ given by `... progress label node`.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    bar/.append style={fill=green},
    bar incomplete/.append style={fill=red},
    progress=today,
    today=6,
    group progress label node/.append style={below=3pt}
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1}{10} \\
  \ganttbar[
    bar progress label font=\color{green!25!black}\sffamily
  ]{Subtask 1}{1}{3} \\
  \ganttbar[
    progress label text={$\displaystyle\frac{#1}{100}$}
  ]{Subtask 2}{5}{12}
\end{ganttchart}
```



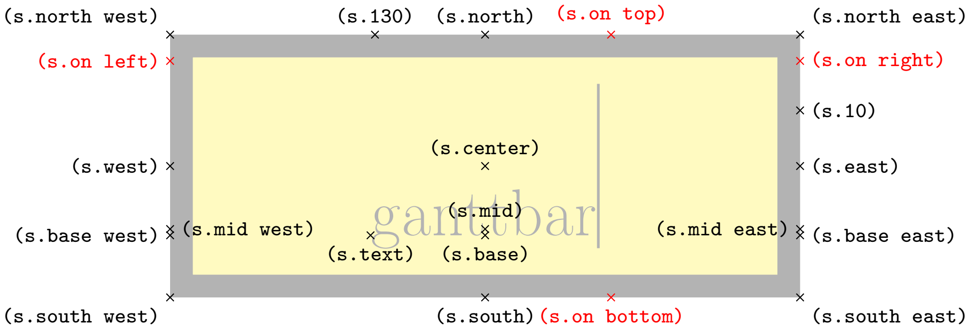
2.7.5 New Node Shapes

pgfgantt defines three new node shapes:

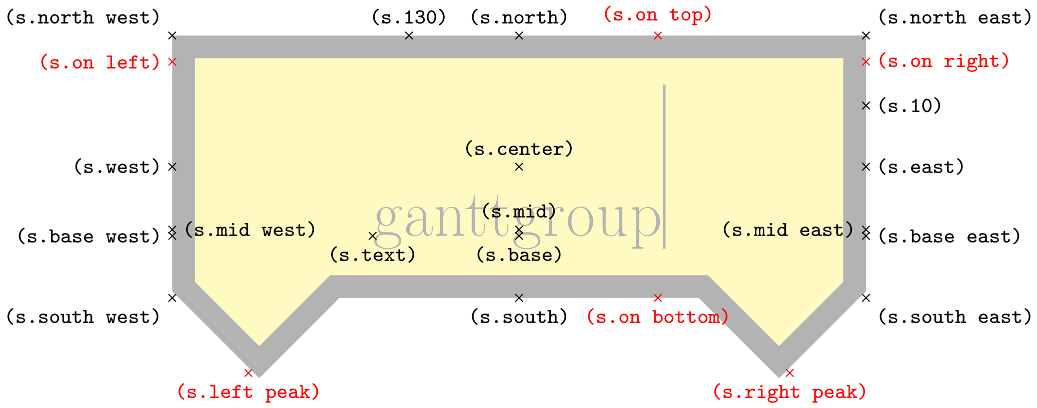
(1) The `ganttbar` node shape derives from shape `rectangle` (section 48.2 of the Ti*k*Z manual). It provides four additional anchors: `on top`, `on bottom`, `on left` and `on right`. The ⟨*fraction*⟩ set by the following keys indicates a position between the left and right (for `on top` and `on bottom`) or upper and lower border (for `on left` and `on right`), similarly to the `/tikz/pos` key.

`/pgfgantt/on top fraction` =⟨*fraction*⟩ `0.5`
`/pgfgantt/on bottom fraction` =⟨*fraction*⟩ `0.5`
`/pgfgantt/on left fraction` =⟨*fraction*⟩ `0.5`
`/pgfgantt/on right fraction` =⟨*fraction*⟩ `0.5`

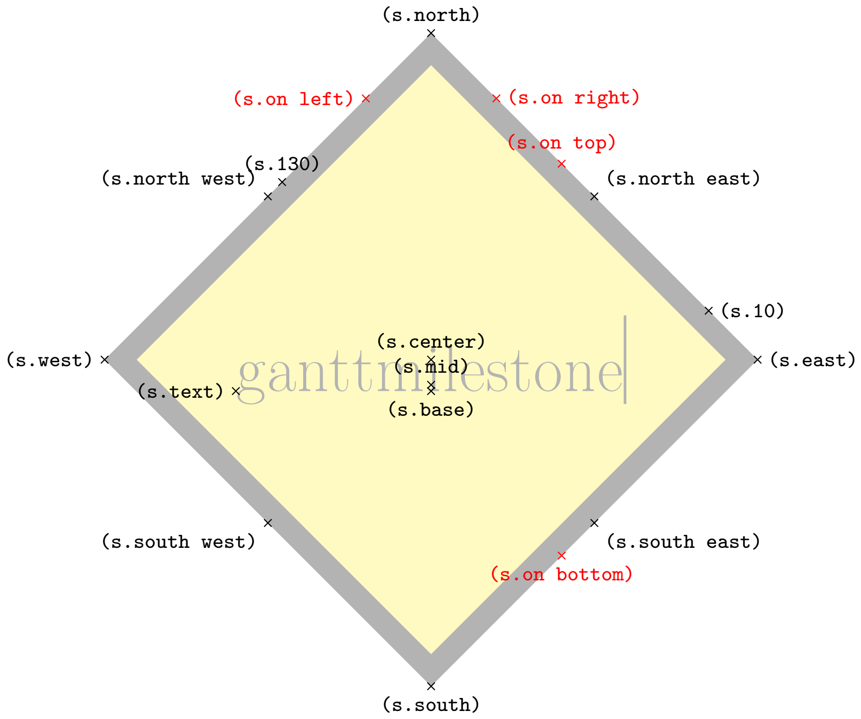
In the following figures, `on top/bottom fraction` is 0.7, whereas `on left/right fraction` is 0.1.



(2) The `ganttgroup` node shape also derives from shape `rectangle`. It provides the additional anchors `on top`, `on bottom`, `on left`, `on right` (same as above), `left peak` and `right peak`.



(3) The `ganttmilestone` node shape derives from shape `diamond` (section 48.3 of the Ti*k*Z manual), but does not consider any aspect ratio. It provides the additional anchors `on top`, `on bottom`, `on left` and `on right` (same as above).



2.8 Defining Custom Chart Elements

You may define completely new chart elements with

```
\newganttchartelement{⟨name⟩}{⟨new default key values⟩}
\newganttchartelement*{⟨name⟩}{⟨new default key values⟩}
```

`\newganttchartelement`

`\newganttchartelement` (unstarred) defines a new chart element `\gantt⟨name⟩` and the corresponding `\ganttlinked⟨name⟩`. These chart element macros take one optional argument ⟨*options*⟩ and *three* mandatory arguments ⟨*label*⟩, ⟨*start tss*⟩ and ⟨*end tss*⟩ (like `\ganttbar`).

`\newganttchartelement*`

Chart element macros defined by the starred form, `\newganttchartelement*`, take the same single optional argument, but *two* mandatory arguments ⟨*label*⟩ and ⟨*tss*⟩ (like `\ganttmilestone`).

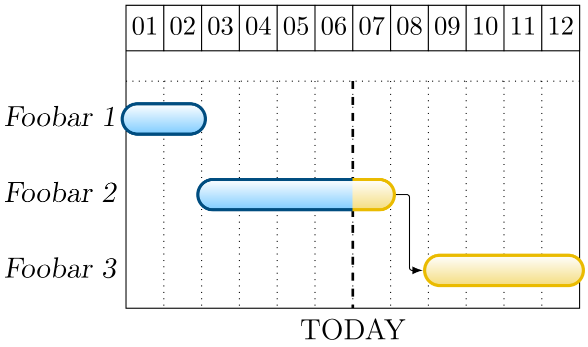
For each new chart element, `\newganttchartelement` also introduces a set of nine value-storing keys and five style keys and assigns default values to them:

Key	*Default value*
Style keys	
⟨*name*⟩	`shape=rectangle, inner sep=0pt, draw, fill=white`
⟨*name*⟩ `incomplete`	`/pgfgantt/`⟨*name*⟩`, fill=black!25`
⟨*name*⟩ `label node`	`anchor=east, font=\ganttvalueof{`⟨*name*⟩` label font}`
⟨*name*⟩ `inline label node`	`anchor=center, font=\ganttvalueof{`⟨*name*⟩` label font}`
⟨*name*⟩ `progress label node`	`anchor=west, font=\ganttvalueof{`⟨*name*⟩` progress label font}`
Value-storing keys	
⟨*name*⟩ `label font`	`\normalsize`
⟨*name*⟩ `inline label anchor`	`center`
⟨*name*⟩ `progress label anchor`	`east`
⟨*name*⟩ `progress label font`	`\scriptsize`
⟨*name*⟩ `left shift`	`0`
⟨*name*⟩ `right shift`	`0`
⟨*name*⟩ `top shift`	`.3`
⟨*name*⟩ `height`	`.4`
⟨*name*⟩ `label text`	`\strut#1`

Consequently, a new chart element will look like the standard `\ganttbar` unless you introduce some ⟨*new default key values*⟩.

Let us define a new chart element called “foobar”, which is basically a fancy-colored and -shaped bar:

```
\definecolor{foobarblue}{RGB}{0,153,255}
\definecolor{foobaryellow}{RGB}{234,187,0}

\newganttchartelement{foobar}{
  foobar/.style={
    shape=rounded rectangle,
    inner sep=0pt,
    draw=foobarblue!50!black,
    very thick,
    top color=white,
    bottom color=foobarblue!50
  },
  foobar incomplete/.style={
    /pgfgantt/foobar,
    draw=foobaryellow,
    bottom color=foobaryellow!50
  },
  foobar label font=\slshape,
  foobar left shift=-.1,
  foobar right shift=.1
}

\begin{ganttchart}[
    vgrid,
    progress=today,
    progress label text=\relax,
    today=6
  ]{1}{12}
  \gantttitlecalendar{day} \\[grid]
  \ganttfoobar{Foobar 1}{1}{2} \\
  \ganttfoobar{Foobar 2}{3}{7} \\
  \ganttlinkedfoobar{Foobar 3}{9}{12}
\end{ganttchart}
```



2.9 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 element name⟩}{⟨end element name⟩}
```

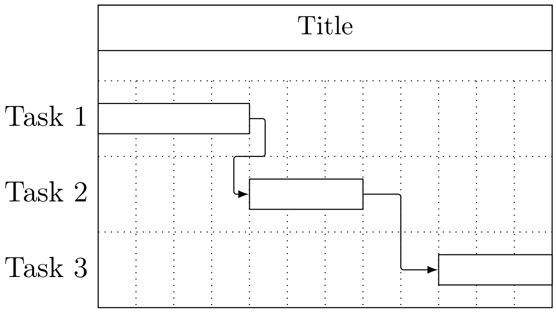
`/pgfgantt/name =⟨name⟩` (empty)

The `\ganttlink` macro connects two elements, which are specified by their ⟨*name*⟩s. By default, chart elements are named automatically: The first one receives the name `elem0`, the second one is called `elem1` and so on. However, the `name` key allows you to assign a name to each chart element.

`\ganttlink`

```
\begin{ganttchart}[
    vgrid,
    hgrid
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{1}{4} \\
  \ganttbar{Task 2}{5}{7} \\
  \ganttbar{Task 3}{10}{12}
  \ganttlink{elem0}{elem1}
  \ganttlink{elem1}{elem2}
\end{ganttchart}
```

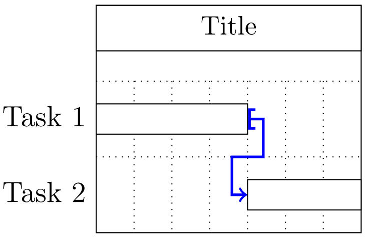
```
\begin{ganttchart}[
    vgrid,
    hgrid
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar[name=b1]%
    {Task 1}{1}{4} \\
  \ganttbar[name=b2]%
    {Task 2}{5}{7} \\
  \ganttbar[name=xyz]%
    {Task 3}{10}{12}
  \ganttlink{b1}{b2}
  \ganttlink{b2}{xyz}
\end{ganttchart}
```



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

Sets the appearance of the link.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    link/.style={[-to, line width=1pt, blue}
  ]{1}{7}
  \gantttitle{Title}{7} \\
  \ganttbar{Task 1}{1}{4} \\
  \ganttbar{Task 2}{5}{7}
  \ganttlink{elem0}{elem1}
\end{ganttchart}
```



`/pgfgantt/link type` =⟨*type*⟩ `auto`

Link types fall into several categories:

1. *Automatic links* are arrow-like. As you can see from the examples above, they 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 automatic links:

   `/pgfgantt/link mid` =⟨*factor*⟩ `0.5`

   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.

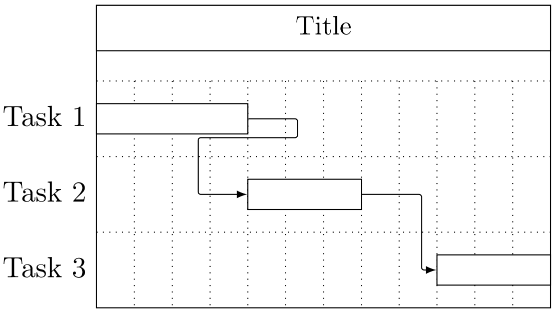
   `/pgfgantt/link bulge` =⟨*factor*⟩ `0.4`

   In five-part links, the upper and lower vertical segments are shifted along the x-axis by $+\langle factor \rangle$ and $-\langle factor \rangle$, respectively.

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

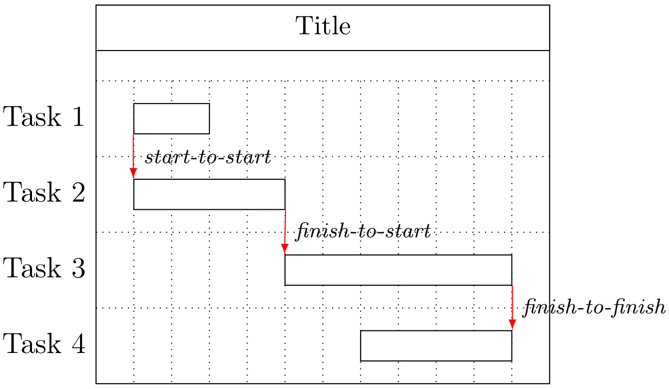
   Decides whether pgfgantt draws a five- or a three-part link. If the true x-coordinates of the link start and end differ by at least ⟨*factor*⟩ (this is the case for the second link in the example below), the package draws a five-part link.

```
\begin{ganttchart}[vgrid, hgrid, link mid=.25, link bulge=1.3]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{1}{4} \\
  \ganttbar{Task 2}{5}{7} \\
  \ganttbar{Task 3}{10}{12}
  \ganttlink{elem0}{elem1}
  \ganttlink[link mid=.8]{elem1}{elem2}
\end{ganttchart}
```



2. *Straight links* are only meant for connecting two bars in order to establish start-to-finish relations (`s-f`), start-to-start relations (`s-s`) etc. Their ⟨*type*⟩ identifiers are reminiscent of the syntax for specifying arrow tips in Ti*k*Z: Each identifier is composed of two letters separated by a hyphen.

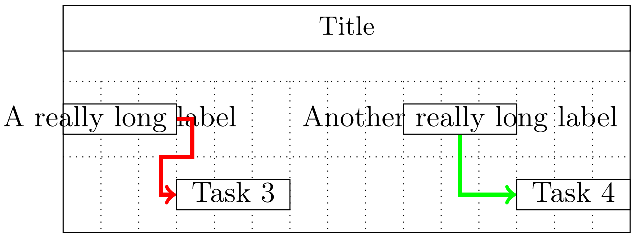
```
\begin{ganttchart}[
    vgrid,
    hgrid,
    link/.style={-latex, draw=red, fill=red}
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{2}{3} \\
  \ganttbar{Task 2}{2}{5} \\
  \ganttbar{Task 3}{6}{11} \\
  \ganttbar{Task 4}{8}{11}
  \ganttlink[link type=s-s]{elem0}{elem1}
  \ganttlink[link type=f-s]{elem1}{elem2}
  \ganttlink[link type=f-f]{elem2}{elem3}
\end{ganttchart}
```



3. *Custom links* allow you to define completely new link types. Strictly speaking, automatic and straight links are predefined custom links whose code supports the keys mentioned above (section 3.11 presents the Ti*k*Z code of these links).

For instance, pgfgantt provides one additional link type, `dr` (short for "down-right"). This type is convenient for connecting inline-labeled bars if the label of the start bar protrudes from its right border.

```
\begin{ganttchart}[
    vgrid,
    hgrid,
    inline,
    link/.style={->, ultra thick}
  ]{1}{15}
  \gantttitle{Title}{15} \\
  \ganttbar{A really long label}{1}{3}
  \ganttbar{Another really long label}{10}{12} \\
  \ganttbar{Task 3}{4}{6}
  \ganttbar{Task 4}{13}{15}
  \ganttlink[link/.append style=red]{elem0}{elem2}
  \ganttlink[link/.append style=green, link type=dr]{elem1}{elem3}
\end{ganttchart}
```



The following macro creates new link types:

```
\newganttlinktype{⟨type⟩}{⟨TikZ code⟩}
```

`\newganttlinktype`

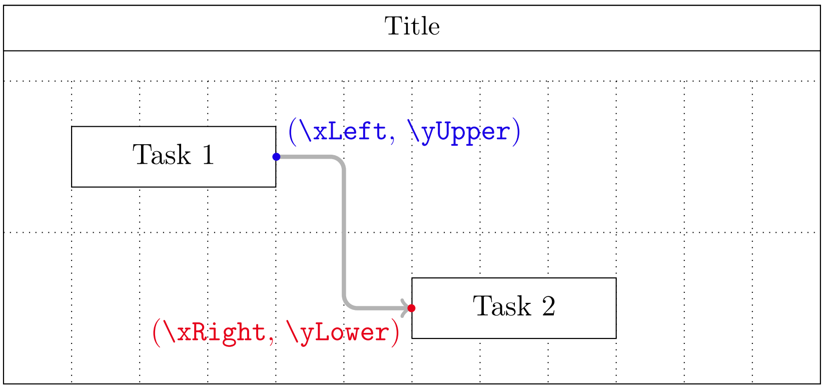
It defines a new link ⟨*type*⟩ which is drawn by the given ⟨*TikZ code*⟩. When you write this code, you do not have to know the final absolute coordinates of each link type instance. On the contrary, several commands that are only available in the second argument of `\newganttlinktype` help you to design generic link types:

`\ganttsetstartanchor`
`\ganttsetendanchor`

- First, you have to choose the border points of the chart elements the link will connect. For this purpose, `\ganttsetstartanchor{⟨anchor⟩}` and `\ganttsetendanchor{⟨anchor⟩}` select an ⟨*anchor*⟩ of the start and end element, respectively. See the figures in section 2.7.5 for possible ⟨*anchor*⟩s of the default chart element shapes. You may specify a certain ⟨*fraction*⟩ for anchors like `on top` by `\ganttsetstartanchor{on top=⟨fraction⟩}`. pgfgantt sets the default anchors to `\ganttsetstartanchor{east}` and `\ganttsetendanchor{west}`, so you even may omit these two commands.

`\xLeft`
`\yUpper`
`\xRight`
`\yLower`

- The two macro pairs `\xLeft`/`\yUpper` and `\xRight`/`\yLower` provide the x- and y-coordinates of the link start and end points, respectively.



- `\ganttlinklabel` contains the label that you may assign to each link type via `\setganttlinklabel` or the `link label` key (see below). `\ganttlinklabel`
- You can access any values stored in the package's ⟨*key*⟩s with the macro `\ganttvalueof{`⟨*key*⟩`}`. `\ganttvalueof`
- Remember that you can use the style `/pgfgantt/link` to ensure a uniform appearance of all your link types.

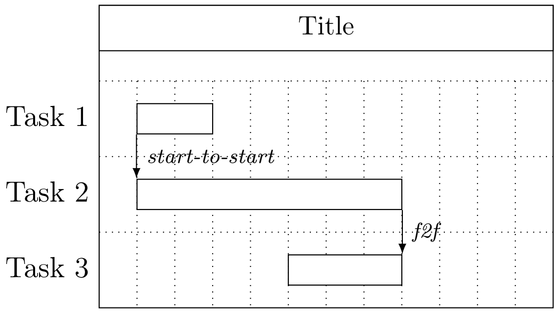
`\newganttlinktypealias{`⟨*new type*⟩`}{`⟨*existing type*⟩`}`

`\newganttlinktypealias` lets a ⟨*new type*⟩ equal an ⟨*existing type*⟩, also copying any label that has been set for the ⟨*existing type*⟩. `\newganttlinktypealias`

`\setganttlinklabel{`⟨*type*⟩`}{`⟨*label*⟩`}`

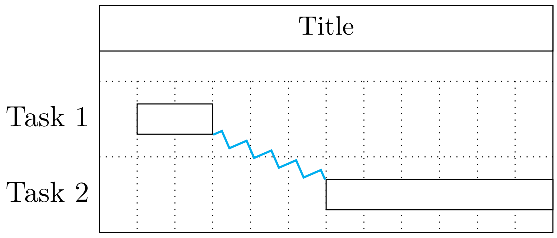
`\setganttlinklabel` sets a ⟨*label*⟩ for the given link ⟨*type*⟩. In the following example, note how `sta-to-sta` and `s-s` share a common label, while we change the label of `fin-to-fin`. `\setganttlinklabel`

```
\newganttlinktypealias{sta-to-sta}{s-s}
\newganttlinktypealias{fin-to-fin}{f-f}
\setganttlinklabel{fin-to-fin}{f2f}

\begin{ganttchart}[vgrid, hgrid]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{2}{3} \\
  \ganttbar{Task 2}{2}{8} \\
  \ganttbar{Task 3}{6}{8}
  \ganttlink[link type=sta-to-sta]{elem0}{elem1}
  \ganttlink[link type=fin-to-fin]{elem1}{elem2}
\end{ganttchart}
```



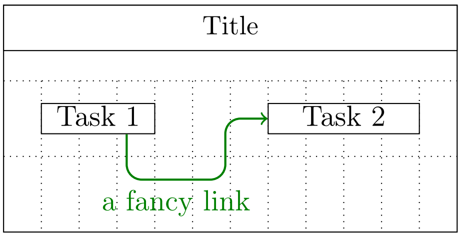
Let’s put it all together and devise two new link types. Firstly, `zigzag` connects the lower right corner of the start element and the upper left corner of the end element with a thick, cyan line decorated by a zigzag pattern.

```
\usetikzlibrary{decorations.pathmorphing}

\newganttlinktype{zigzag}{
  \ganttsetstartanchor{on right=1}
  \ganttsetendanchor{on left=0}
  \draw [decoration=zigzag, decorate, thick, cyan]
    (\xLeft, \yUpper) --
    (\xRight, \yLower);
}

\begin{ganttchart}[vgrid, hgrid]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{2}{3} \\
  \ganttbar{Task 2}{7}{12}
  \ganttlink[link type=zigzag]{elem0}{elem1}
\end{ganttchart}
```



Secondly, **drur** (short for down-right-up-right) draws a labelled arrow in the default style `link`. The link starts at the bottom of the first element and connects to the left border of the second one. In addition, the known keys `link mid` and `link bulge` decide where the line going up is positioned and how far the first line going right is below the start coordinate, respectively.

```
\newganttlinktype{drur}{
  \ganttsetstartanchor{on bottom=0.75}
  \ganttsetendanchor{on left}
  \draw [/pgfgantt/link]
    % first segment (down)
    (\xLeft, \yUpper) --
    % second segment (right)
    (\xLeft, \yUpper -
      \ganttvalueof{link bulge} * \ganttvalueof{y unit chart}) --
    % link label
    node [pos=.5, /pgfgantt/link label anchor] {\ganttlinklabel}
    % third segment (up)
    ($(\xLeft,
      \yUpper -
        \ganttvalueof{link bulge} * \ganttvalueof{y unit chart})!%
      \ganttvalueof{link mid}!%
      (\xRight,
      \yUpper -
        \ganttvalueof{link bulge} * \ganttvalueof{y unit chart})$) --
    % last segment (right again)
    ($(\xLeft, \yLower)!%
      \ganttvalueof{link mid}!%
      (\xRight, \yLower)$) --
    (\xRight, \yLower);
}
\setganttlinklabel{drur}{a fancy link}

\begin{ganttchart}[
    vgrid,
    hgrid,
    link/.style={thick, ->, green!50!black, rounded corners=2mm},
    link label anchor/.style=below,
    link mid=.7, link bulge=.6
  ]{1}12}
  \gantttitle{Title}{12} \\
  \ganttbar[inline]{Task 1}{2}{4}
  \ganttbar[inline]{Task 2}{8}{11} \\
  \ganttlink[link type=drur]{elem0}{elem1}
\end{ganttchart}
```

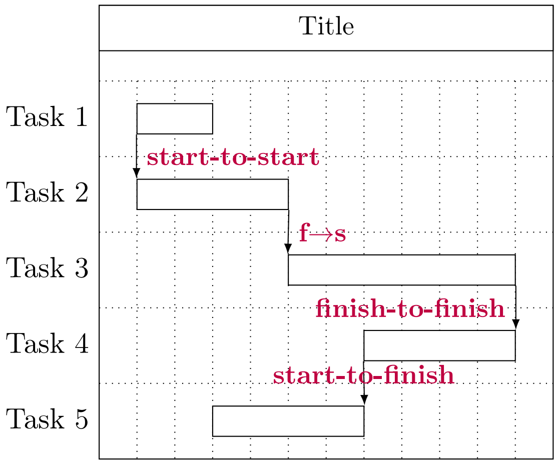


(Please remove the comments if you copy the code above – they will confuse Ti*k*Z and generate tons of errors.)

`/pgfgantt/link label =`⟨*label*⟩ (empty)
`/pgfgantt/link label font =`⟨*font commands*⟩ `\scriptsize\itshape`
`/pgfgantt/link label node /.style=`⟨*options*⟩
`anchor=west, font=\ganttvalueof{link label font}`

The `link label` key locally overrides any label specified by `\setganttlinklabel`. The ⟨*font commands*⟩ and ⟨*options*⟩ are applied to the link label node. By default, the label appears to the right of the straight link's center.

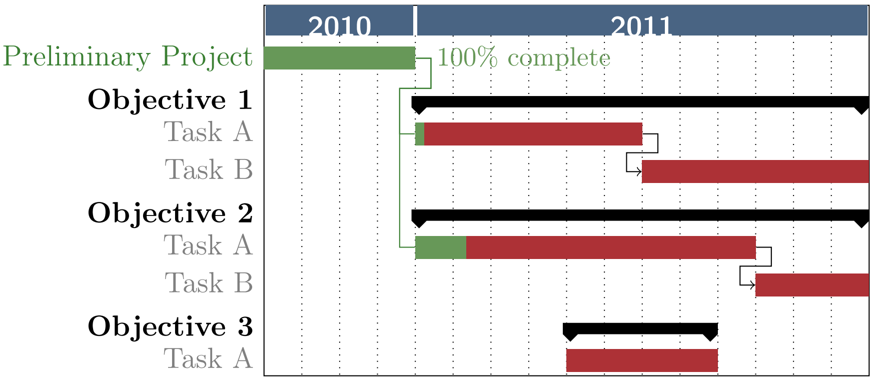
```
\begin{ganttchart}[
    vgrid,
    hgrid,
    link label font=\small\bfseries\color{purple}
  ]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{2}{3} \\
  \ganttbar{Task 2}{2}{5} \\
  \ganttbar{Task 3}{6}{11} \\
  \ganttbar{Task 4}{8}{11} \\
  \ganttbar{Task 5}{4}{7}
  \ganttlink[link type=s-s]{elem0}{elem1}
  \ganttlink[link type=f-s, link label={f$\to$s}]{elem1}{elem2}
  \ganttlink[
    link type=f-f,
    link label node/.append style={anchor=east}
  ]{elem2}{elem3}
  \ganttlink[
    link type=s-f,
    link label node/.append style={anchor=base}
  ]{elem3}{elem4}
\end{ganttchart}
```



2.10 Style Examples

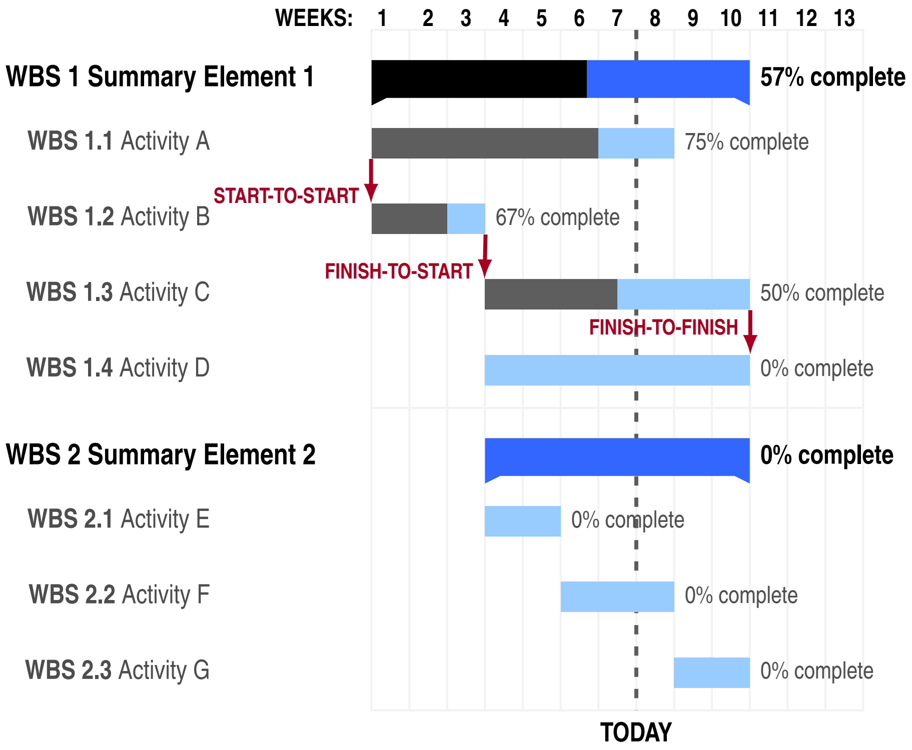
The first example plays around with colors and notably uses equal x- and y-vectors.

```
\begin{ganttchart}[
    y unit title=0.4cm,
    y unit chart=0.5cm,
    vgrid,
    time slot format=isodate-yearmonth,
    time slot unit=month,
    title/.append style={draw=none, fill=RoyalBlue!50!black},
    title label font=\sffamily\bfseries\color{white},
    title label node/.append style={below=-1.6ex},
    title left shift=.05,
    title right shift=-.05,
    title height=1,
    bar/.append style={draw=none, fill=OliveGreen!75},
    bar height=.6,
    bar label font=\normalsize\color{black!50},
    group right shift=0,
    group top shift=.6,
    group height=.3,
    group peaks height=.2,
    bar incomplete/.append style={fill=Maroon}
  ]{2010-09}{2011-12}
  \gantttitlecalendar{year} \\
  \ganttbar[
    progress=100,
    bar progress label font=\small\color{OliveGreen!75},
    bar progress label node/.append style={right=4pt},
    bar label font=\normalsize\color{OliveGreen},
    name=pp
  ]{Preliminary Project}{2010-09}{2010-12} \\
  \ganttset{progress label text={}, link/.style={black, -to}}
  \ganttgroup{Objective 1}{2011-01}{2011-12} \\
  \ganttbar[progress=4, name=T1A]{Task A}{2011-01}{2011-06} \\
  \ganttlinkedbar[progress=0]{Task B}{2011-07}{2011-12} \\
  \ganttgroup{Objective 2}{2011-01}{2011-12} \\
  \ganttbar[progress=15, name=T2A]{Task A}{2011-01}{2011-09} \\
  \ganttlinkedbar[progress=0]{Task B}{2011-10}{2011-12} \\
  \ganttgroup{Objective 3}{2011-05}{2011-08} \\
  \ganttbar[progress=0]{Task A}{2011-05}{2011-08}
  \ganttset{link/.style={OliveGreen}}
  \ganttlink[link mid=.4]{pp}{T1A}
  \ganttlink[link mid=.159]{pp}{T2A}
\end{ganttchart}
```



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}
\renewcommand\sfdefault{phv}
\renewcommand\mddefault{mc}
\renewcommand\bfdefault{bc}
\setganttlinklabel{s-s}{START-TO-START}
\setganttlinklabel{f-s}{FINISH-TO-START}
\setganttlinklabel{f-f}{FINISH-TO-FINISH}
\sffamily
\begin{ganttchart}[
    canvas/.append style={fill=none, draw=black!5, line width=.75pt},
    hgrid style/.style={draw=black!5, line width=.75pt},
    vgrid={*1{draw=black!5, line width=.75pt}},
    today=7,
    today rule/.style={
      draw=black!64,
      dash pattern=on 3.5pt off 4.5pt,
      line width=1.5pt
    },
    today label font=\small\bfseries,
    title/.style={draw=none, fill=none},
    title label font=\bfseries\footnotesize,
    title label node/.append style={below=7pt},
    include title in canvas=false,
    bar label font=\mdseries\small\color{black!70},
    bar label node/.append style={left=2cm},
    bar/.append style={draw=none, fill=black!63},
    bar incomplete/.append style={fill=barblue},
    bar progress label font=\mdseries\footnotesize\color{black!70},
    group incomplete/.append style={fill=groupblue},
    group left shift=0,
    group right shift=0,
    group height=.5,
    group peaks tip position=0,
    group label node/.append style={left=.6cm},
    group progress label font=\bfseries\small,
    link/.style={-latex, line width=1.5pt, linkred},
    link label font=\scriptsize\bfseries,
    link label node/.append style={below left=-2pt and 0pt}
  ]{1}{13}
  \gantttitle[
    title label node/.append style={below left=7pt and -3pt}
  ]{WEEKS:\quad1}{1}
  \gantttitlelist{2,...,13}{1} \\
  \ganttgroup[progress=57]{WBS 1 Summary Element 1}{1}{10} \\
  \ganttbar[
    progress=75,
    name=WBS1A
  ]{\textbf{WBS 1.1} Activity A}{1}{8} \\
  \ganttbar[
    progress=67,
    name=WBS1B
  ]{\textbf{WBS 1.2} Activity B}{1}{3} \\
  \ganttbar[
    progress=50,
    name=WBS1C
  ]{\textbf{WBS 1.3} Activity C}{4}{10} \\
  \ganttbar[
    progress=0,
    name=WBS1D
  ]{\textbf{WBS 1.4} Activity D}{4}{10} \\[grid]
  \ganttgroup[progress=0]{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[link type=s-s]{WBS1A}{WBS1B}
  \ganttlink[link type=f-s]{WBS1B}{WBS1C}
  \ganttlink[
    link type=f-f,
    link label node/.append style=left
  ]{WBS1C}{WBS1D}
\end{ganttchart}
```



3 Implementation

3.1 Packages

pgfgantt is modest in terms of dependencies: It only requires the TikZ and pgfcalendar packages.

```
\RequirePackage{tikz}
\usetikzlibrary{%
  arrows, backgrounds, calc,%
  patterns, positioning, shapes.geometric%
}
\RequirePackage{pgfcalendar}

```

3.2 Macros for Key and Error Management

`\@gtt@ifstar` reimplements the LaTeX 2ε kernel's `\@ifstar` macro. This makes it robust to amsgen's reimplementation of `\@ifstar`.

\@gtt@ifstar

```
\def\@gtt@ifstar#1{\kernel@ifnextchar*{\@firstoftwo{#1}}}
```

`\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 auxiliary macros save us some code when we devise keys later on. \@gtt@keydef{⟨*key*⟩}{⟨*initial value*⟩} declares the key /pgfgantt/⟨*key*⟩ and stores its ⟨*initial value*⟩.

\@gtt@keydef

```
\def\@gtt@keydef#1#2{%
  \pgfkeyssetvalue{/pgfgantt/#1}{#2}%
}
```

\ganttvalueof{⟨*key*⟩} retrieves the value stored by a ⟨*key*⟩. Link type authors should be able to use this macro in their code; thus, it lacks any @s.

\ganttvalueof

```
\def\ganttvalueof#1{%
  \pgfkeysvalueof{/pgfgantt/#1}%
}
```

\@gtt@stylekeydef{⟨*key*⟩}{⟨*initial style*⟩} declares a style ⟨*key*⟩ with an ⟨*initial style*⟩.

\@gtt@stylekeydef

```
\def\@gtt@stylekeydef#1#2{%
  \pgfkeys{/pgfgantt/#1/.style={#2}}%
}
```

\@gtt@PackageError{⟨*message*⟩} and \@gtt@PackageWarning{⟨*message*⟩} issue a package error or warning ⟨*message*⟩, respectively.

\@gtt@PackageError
\@gtt@PackageWarning

```
\def\@gtt@PackageError#1{%
```

```
  \PackageError{pgfgantt}{#1}{}%
}
\def\@gtt@PackageWarning#1{%
  \PackageWarning{pgfgantt}{#1}%
}

```

3.3 The Horizontal and Vertical Grid

The count register `\gtt@currentline` holds the current line; it starts from 0 and decreases. `\gtt@lasttitleline` equals the line of the title element drawn last. `\gtt@currgrid` is the index of the current grid line drawn. `\gtt@chartwidth` equals the number of time slots.

`\gtt@currentline`
`\gtt@lasttitleline`
`\gtt@currgrid`
`\gtt@chartwidth`

```
\newcount\gtt@currentline
\newcount\gtt@lasttitleline
\newcount\gtt@currgrid
\newcount\gtt@chartwidth
```

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

`hgrid`
`hgrid style`
`\ifgtt@hgrid`
`\gtt@hgridstyle`

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

```

The `\gtt@hgrid@do` macro decomposes the style list for the horizontal grid into its comma-separated items. Each item is analyzed (see below) only if some grid lines are still left to draw. Note the "elegant" quadruple `\expandafter` construction, which enables tail recursion.

`\gtt@hgrid@do`

```
\def\gtt@hgrid@do#1,{%
```

```
  \ifx\relax#1\else
    \ifnum\gtt@currgrid<\gtt@currentline\else%
      \gtt@hgrid@analyze#1\relax%
      \expandafter\expandafter\expandafter\gtt@hgrid@do%
    \expandafter\fi%
  \fi%
}

```

`\gtt@hgrid@analyze`

In the absence of a star as the first token in a style list item, `\gtt@hgrid@analyze` adds the multiplier `1` to the input stream.

```
\def\gtt@hgrid@analyze{%
  \@gtt@ifstar{\gtt@hgrid@draw}{\gtt@hgrid@draw1}%
}

```

`\gtt@hgrid@draw`

`\gtt@hgrid@draw` draws as many grid lines as required by the multiplier. It increases `\gtt@currgrid` after each line drawn and breaks the loop as soon as all grid rules have been drawn.

```
\def\gtt@hgrid@draw#1#2\relax{%
  \foreach \i in {1,...,#1} {%
    \pgfmathsetmacro\y@upper{%
      \gtt@lasttitleline * \ganttvalueof{y unit title} +%
      (\gtt@currgrid - \gtt@lasttitleline)%
      * \ganttvalueof{y unit chart}%
    }%
    \draw [#2]
      (0pt, \y@upper pt) --
      (\gtt@chartwidth * \ganttvalueof{x unit}, \y@upper pt);%
    \global\advance\gtt@currgrid by-1\relax%
    \ifnum\gtt@currgrid<\gtt@currentline\breakforeach\fi%
  }%
}

```

`vgrid`
`\ifgtt@vgrid`
`\gtt@vgridstyle`
`\gtt@vgrid@do`
`\gtt@vgrid@analyze`
`\gtt@vgrid@draw`

Analogously, we declare options and macros for printing the vertical grid.

```
\newif\ifgtt@vgrid
\ganttset{%
  vgrid/.code={%
    \def\@tempa{#1}%
    \def\@tempb{false}%
    \ifx\@tempa\@tempb%
      \gtt@vgridfalse%
    \else%
      \gtt@vgridtrue%
      \def\@tempb{true}%
      \ifx\@tempa\@tempb%
        \def\gtt@vgridstyle{dotted}%
      \else%
        \def\gtt@vgridstyle{#1}%
      \fi%
    \fi%
  },%
  vgrid/.default=dotted
}

\def\gtt@vgrid@do#1,{%
  \ifx\relax#1\else%
    \ifnum\gtt@currgrid>\gtt@chartwidth\else%
      \gtt@vgrid@analyze#1\relax%
      \expandafter\expandafter\expandafter\gtt@vgrid@do%
    \expandafter\fi%
  \fi%
}

\def\gtt@vgrid@analyze{%
  \@gtt@ifstar{\gtt@vgrid@draw}{\gtt@vgrid@draw1}%
}

\def\gtt@vgrid@draw#1#2\relax{%
  \foreach \i in {1,...,#1} {%
    \draw [#2]
      (\gtt@currgrid * \ganttvalueof{x unit}, \y@upper pt) --%
      (\gtt@currgrid * \ganttvalueof{x unit}, \y@lower pt);%
    \global\advance\gtt@currgrid by1\relax%
    \ifnum\gtt@currgrid>\gtt@chartwidth\breakforeach\fi%
  }%
}

```

3.4 Time Slot Formats

`\gtt@smugglecount{`⟨*count*⟩`}` smuggles the local value of a count register over the end of a TEX group. `\gtt@smugglecount`

```
\def\gtt@smugglecount#1\endgroup{%
  \edef\@tempa{\the#1}%
  \expandafter\endgroup\expandafter#1\expandafter=\@tempa%
}

```

`\gtt@juliantotimeslot{`⟨*count 1*⟩`}{`⟨*count 2*⟩`}` converts the Julian date stored in ⟨*count 1*⟩ to a time slot and stores the latter in ⟨*count 2*⟩. This macro is called after the start of Gantt chart. Thus, `\gtt@startyear`, `\gtt@startmonth` and `\gtt@startjulian` (see section 3.5) have already been initialized. Depending on the value of `time slot unit`, one time slot corresponds to one `day`, one `month` or one `year`. `\gtt@juliantotimeslot`

```
\newcommand\gtt@juliantotimeslot[2]{%
  \begingroup%
  \@tempcnta=#1\relax%
  \ifgtt@timeslotunit@day%
    \advance\@tempcnta by-\gtt@startjulian\relax%
    \advance\@tempcnta by1\relax%
  \fi%
  \ifgtt@timeslotunit@month%
    \pgfcalendarjuliantodate{\@tempcnta}{\@tempa}{\@tempb}{\@tempc}%
    \@tempcnta=\@tempa\relax%
    \advance\@tempcnta by-\gtt@startyear\relax%
    \multiply\@tempcnta by12\relax%
    \advance\@tempcnta by\@tempb\relax%
    \advance\@tempcnta by-\gtt@startmonth\relax%
    \advance\@tempcnta by1\relax%
  \fi%
  \ifgtt@timeslotunit@year%
    \pgfcalendarjuliantodate{\@tempcnta}{\@tempa}{\@tempb}{\@tempc}%
    \@tempcnta=\@tempa\relax%
    \advance\@tempcnta by-\gtt@startyear\relax%
    \advance\@tempcnta by1\relax%
  \fi%
  #2=\@tempcnta\relax%
  \gtt@smugglecount#2%
  \endgroup%
}

```

`\newganttimeslotformat`

`\newganttimeslotformat{`⟨*name*⟩`}{`⟨*code*⟩`}` defines the macro `\gtt@tsstojulian@`⟨*name*⟩`{`⟨*tss*⟩`}{`⟨*count*⟩`}`. This macro executes ⟨*code*⟩ (within a group), which should convert ⟨*tss*⟩ to a Julian date and store the date in ⟨*count*⟩.

```
\newcommand\newganttimeslotformat[2]{%
  \expandafter\def\csname gtt@tsstojulian@#1\endcsname##1##2{%
    \begingroup#2\gtt@smugglecount##2\endgroup%
  }%
}

```

The predefined time slot formats `simple`, `isodate`, `isodate-yearmonth` and `isodate-year` are straight forward.

```
\newganttimeslotformat{simple}{%
  #2=#1\relax%
  \advance#2 by\gtt@tsf@startjulian\relax%
  \advance#2 by-1\relax%
}

\newganttimeslotformat{isodate}{%
  \pgfcalendardatetojulian{#1}{#2}%
}
```

```

\newganttimeslotformat{isodate-yearmonth}{%
  \pgfcalendardatetojulian{#1-01}{#2}%
}

\newganttimeslotformat{isodate-year}{%
  \pgfcalendardatetojulian{#1-01-01}{#2}%
}

```

`\gtt@tsf@getdmy`

`\gtt@tsf@getdmy{`⟨*date*⟩`}` decomposes a ⟨*date*⟩ `day`[sep]`month`[sep]`year` (with [sep] representing a period, hyphen or slash) into `day`, `month` and `year` and stores these numbers in `\local@day`, `\local@month` and `\local@year`, respectively.

```
\newcommand\gtt@tsf@getdmy[1]{%
  \edef\local@firstarg{#1}%
  \def\local@decompose##1.##2.##3\relax{%
    \def\local@day{##1}\def\local@month{##2}\def\local@year{##3}%
  }%
  \expandafter\local@decompose\local@firstarg..\relax%
  \ifx\local@month\@empty%
    \def\local@decompose##1/##2/##3\relax{%
      \def\local@day{##1}\def\local@month{##2}\def\local@year{##3}%
    }%
    \expandafter\local@decompose\local@firstarg//\relax%
    \ifx\local@month\@empty%
      \def\local@decompose##1-##2-##3\relax{%
        \def\local@day{##1}\def\local@month{##2}\def\local@year{##3}%
      }%
      \expandafter\local@decompose\local@firstarg--\relax%
      \ifx\local@month\@empty%
        \@gtt@PackageError{Illegal time slot specifier ‘#1’.}%
      \else%
        \def\local@decompose##1--{\def\local@year{##1}}%
        \expandafter\local@decompose\local@year%
      \fi%
    \else%
      \def\local@decompose##1//{\def\local@year{##1}}%
      \expandafter\local@decompose\local@year%
    \fi%
  \else%
    \def\local@decompose##1..{\def\local@year{##1}}%
    \expandafter\local@decompose\local@year%
  \fi%
}

```

Time slot formats `little-endian`, `big-endian` and `middle-endian` only differ in their call of `\pgfcalendardatetojulian`. If the year (stored in `\local@year` or `\local@day`) lacks a century (e. g., 13 instead of 2013), it is completed according to the value of `time slot format/base century`.

```
\newgantttimeslotformat{little-endian}{%
  \gtt@tsf@getdmy{#1}%
  \ifnum\local@year<100\relax%
    \edef\local@year{\gtt@tsf@basecentury\local@year}%
  \fi%
  \pgfcalendardatetojulian{\local@year-\local@month-\local@day}{#2}%
}

\newgantttimeslotformat{big-endian}{%
  \gtt@tsf@getdmy{#1}%
  \ifnum\local@day<100\relax%
    \edef\local@day{\gtt@tsf@basecentury\local@day}%
  \fi%
  \pgfcalendardatetojulian{\local@day-\local@month-\local@year}{#2}%
}

\newgantttimeslotformat{middle-endian}{%
  \gtt@tsf@getdmy{#1}%
  \ifnum\local@year<100\relax%
    \edef\local@year{\gtt@tsf@basecentury\local@year}%
  \fi%
  \pgfcalendardatetojulian{\local@year-\local@day-\local@month}{#2}%
}

```

The key `time slot format`=⟨*name*⟩ checks whether the format ⟨*name*⟩ exists and then defines the macro `\gtt@tsstojulian` to be equivalent to `\gtt@tsstojulian@`⟨*name*⟩.

`time slot format`
`\gtt@tsstojulian`

```
\ganttset{%
  time slot format/.code={%
    \@ifundefined{gtt@tsstojulian@#1}{%
      \@gtt@PackageError{%
        Time slot format ‘#1’ undefined.%
      }%
    }{}%
    \expandafter\let\expandafter\gtt@tsstojulian%
      \csname gtt@tsstojulian@#1\endcsname%
  },%
  time slot format=simple,%
```

`time slot format/base century`=⟨*year*⟩ extracts the century from the four-digit ⟨*year*⟩ (e. g., 20 from 2000) and stores it in `\gtt@tsf@basecentury`.

`time slot format/base cen`
`\gtt@tsf@basecentury`

```
  time slot format/base century/.code={%
    \begingroup%
    \@tempcnta=#1\relax%
    \divide\@tempcnta by100\relax%
    \xdef\gtt@tsf@basecentury{\the\@tempcnta}%
    \endgroup%
  },%
  time slot format/base century=2000,%
```

time slot format/start date=⟨*isodate*⟩ stores the Julian date corresponding to ⟨*isodate*⟩ in `\gtt@tsf@startjulian`.

time slot format/start da
`\gtt@tsf@startjulian`

```
  time slot format/start date/.code={%
    \begingroup%
    \pgfcalendardatetojulian{#1}{\@tempcnta}%
    \xdef\gtt@tsf@startjulian{\the\@tempcnta}%
    \endgroup%
  },%
  time slot format/start date=2000-01-01%
}

```

3.5 The Main Environment

Keys that store the basis vectors of the chart.

x unit
y unit title
y unit chart

```
\@gtt@keydef{x unit}{.5cm}
\@gtt@keydef{y unit title}{1cm}
\@gtt@keydef{y unit chart}{1cm}
```

Keys related to the canvas and the today rule.

canvas
expand chart
today
today offset
today rule
today label
today label font
today label node

```
\@gtt@stylekeydef{canvas}{shape=rectangle, draw, fill=white}
\@gtt@keydef{expand chart}{none}
\@gtt@keydef{today}{none}
\@gtt@keydef{today offset}{1}
\@gtt@stylekeydef{today rule}{dashed, line width=1pt}
\@gtt@keydef{today label}{TODAY}
\@gtt@keydef{today label font}{\normalfont}
\@gtt@stylekeydef{today label node}{%
  anchor=north, font=\ganttvalueof{today label font}%
}
```

Boolean key that determines if `\\` is equivalent to `\ganttnewline`.

`\ifgtt@newlineshortcut`
newline shortcut

```
\newif\ifgtt@newlineshortcut
\ganttset{%
  newline shortcut/.is if=gtt@newlineshortcut,%
  newline shortcut=true%
}

```

The boolean `\ifgtt@tikzpicture` is true if a Gantt chart appears within a Ti*k*Z picture. `\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. `\gtt@lasttitleslot` corresponds to the x-coordinate of its right border. `\gtt@elementid` enumerates the automatic names of chart elements. `\gtt@today@slot` is the time slot of the today

`\ifgtt@tikzpicture`
`\ifgtt@intitle`
`\gtt@lasttitleslot`
`\gtt@elementid`
`\gtt@today@slot`
`\gtt@startjulian`
`\gtt@endjulian`
`\gtt@chartid`

rule. `\gtt@startjulian` and `\gtt@endjulian` contain the Julian dates corresponding to the first and last time slot, respectively. `\gtt@chartid` assigns a consecutive number to each chart.

```
\newif\ifgtt@tikzpicture
\newif\ifgtt@intitle
\newcount\gtt@lasttitleslot
\newcount\gtt@elementid
\newcount\gtt@today@slot
\newcount\gtt@startjulian
\newcount\gtt@endjulian
\newcount\gtt@chartid
```

`\gtt@chartextrasize`

Each `ganttchart` environment writes a `\gtt@chartextrasize{`⟨*chart id*⟩`}{`⟨*extra size*⟩`}` macro to the auxililary file. This macro stores its second argument in a macro of the form `\@gtt@chart@`⟨*chart id*⟩`@extrasize`. The ⟨*extra size*⟩ is the size of the chart's bounding box less the size of the canvas, calculated as `x unit` times the number of time slots.

```
\def\gtt@chartextrasize#1#2{%
  \global\@namedef{@gtt@chart@#1@extrasize}{#2}%
}
```

`ganttchart`

At the beginning of a `ganttchart` environment, the keys in its optional argument are executed. Initialize the macros and counts that contain start dates, end dates, the chart width, . . .

```
\newenvironment{ganttchart}[3][]{%
  \ganttset{#1}%
  \gtt@tsstojulian{#2}{\gtt@startjulian}%
  \global\gtt@startjulian=\gtt@startjulian\relax%
  \gtt@tsstojulian{#3}{\gtt@endjulian}%
  \global\gtt@endjulian=\gtt@endjulian\relax%
  \pgfcalendarjuliantodate{\gtt@startjulian}%
    {\gtt@startyear}{\gtt@startmonth}{\@tempa}%
  \xdef\gtt@startyear{\gtt@startyear}%
  \xdef\gtt@startmonth{\gtt@startmonth}%
  \gtt@juliantotimeslot{\gtt@endjulian}{\gtt@chartwidth}
  \global\gtt@chartwidth=\gtt@chartwidth\relax%
```

. . . the time slot of the today rule, ...

```
  \def\@tempa{none}%
  \edef\@tempb{\ganttvalueof{today}}%
  \ifx\@tempa\@tempb\else%
    \gtt@tsstojulian{\ganttvalueof{today}}{\gtt@today@slot}
    \gtt@juliantotimeslot{\gtt@today@slot}{\gtt@today@slot}%
  \fi%
```

. . . the current element number, and information for drawing actions.

```
  \global\gtt@elementid=0\relax%
  \global\gtt@currentline=0\relax%
  \global\gtt@lasttitleline=0\relax%
  \global\gtt@lasttitleslot=0\relax%
```

If `expand chart` contains a value different from `none`, scale the chart so that its x-extent equals this value. To this end, use the information stored in the auxiliary file. `\gtt@expanded@xunit` will contain the new value for `x unit`.

`\gtt@expanded@xunit`

```
  \def\@tempa{none}%
  \edef\@tempb{\ganttvalueof{expand chart}}%
  \ifx\@tempa\@tempb\else%
    \@ifundefined{@gtt@chart@\the\gtt@chartid @extrasize}{%
      \@gtt@PackageWarning{Gantt chart expansion may have changed.
                           Rerun to get expansion right}%
    }{%
      \pgfmathparse{(\ganttvalueof{expand chart}%
                     - \@nameuse{@gtt@chart@\the\gtt@chartid @extrasize})%
                    / \gtt@chartwidth}%
      \edef\gtt@expanded@xunit{\pgfmathresult pt}%
      \ganttset{x unit=\gtt@expanded@xunit}%
    }%
  \fi%
```

If a `ganttchart` appears outside of a `tikzpicture`, we implicitly start this environment. “Within a `tikzpicture`” means that `\useasboundingbox` is defined. Since we expect a chart to start with at least one title element, `\ifgtt@intitle` is true. If `newline shortcut` is true, make the control symbol `\\` equivalent to `\ganttnewline`. In any case, `\ganttalignnewline` is defined.

`\\`
`\ganttalignnewline`

```
  \@ifundefined{useasboundingbox}%
    {\gtt@tikzpicturefalse\begin{tikzpicture}}%
    {\gtt@tikzpicturetrue}%
  \gtt@intitletrue%
  \ifgtt@newlineshortcut%
    \let\\\ganttnewline%
  \fi%
  \let\ganttalignnewline\tikz@align@newline%
}{
```

After the contents of the environment have been drawn, we add the canvas to the background layer. pgfgantt saves x- and y-coordinates in local internal macros called `\x@left`, `\x@right`, `\x@mid`, `\x@size`, `\y@upper`, `\y@lower`, `\y@mid` and `\y@size`.

`\y@upper`
`\y@lower`
`\y@mid`
`\y@size`
`\x@size`

```
  \begin{scope}[on background layer]%
    \ifgtt@includetitle%
      \def\y@upper{0}%
    \else%
      \pgfmathsetmacro\y@upper{%
        \gtt@lasttitleline * \ganttvalueof{y unit title}%
      }%
    \fi%
    \pgfmathsetmacro\y@lower{%
      \gtt@lasttitleline * \ganttvalueof{y unit title}%
      + (\gtt@currentline - \gtt@lasttitleline - 1)%
      * \ganttvalueof{y unit chart}%
    }%
    \pgfmathsetmacro\y@mid{%
      (\y@upper + \y@lower) / 2%
    }%
    \pgfmathsetmacro\y@size{%
      abs(\y@lower - \y@upper)%
    }%
    \pgfmathsetmacro\x@size{%
      \gtt@chartwidth * \ganttvalueof{x unit}%
    }%
    \node [/pgfgantt/canvas, minimum width=\x@size pt,
           minimum height=\y@size pt]
      at (\x@size pt / 2, \y@mid pt) {};%
```

The contents of the vertical grid style list are evaluated at most `\gtt@chartwidth`-times, but the loop breaks as soon as all grid lines have been drawn.

```
  \pgfmathsetmacro\y@upper{%
    \gtt@lasttitleline * \ganttvalueof{y unit title}%
  }%
  \ifgtt@vgrid
    \gtt@currgrid=1\relax%
    \global\advance\gtt@chartwidth by-1\relax%
    \foreach \x in {1,...,\gtt@chartwidth} {%
      \expandafter\gtt@vgrid@do\gtt@vgridstyle,\relax,%
      \ifnum\gtt@currgrid>\gtt@chartwidth\relax\breakforeach\fi%
    }%
    \global\advance\gtt@chartwidth by1\relax%
  \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%
    \gtt@currgrid=\gtt@lasttitleline\relax%
    \ifgtt@includetitle\else%
      \advance\gtt@currgrid by-1\relax
    \fi%
    \edef\@tempa{\the\gtt@currgrid}%
    \foreach \t in {\@tempa,...,\gtt@currentline} {%
      \expandafter\gtt@hgrid@do\gtt@hgridstyle,\relax,%
      \ifnum\gtt@currgrid<\gtt@currentline\relax\breakforeach\fi%
    }%
  \fi%
```

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

```
    \def\@tempa{none}%
    \edef\@tempb{\ganttvalueof{today}}%
    \ifx\@tempa\@tempb\else%
      \pgfmathsetmacro\x@mid{%
        (\gtt@today@slot - 1 + \ganttvalueof{today offset})%
        * \ganttvalueof{x unit}%
      }%
      \draw [/pgfgantt/today rule]
        (\x@mid pt, \y@upper pt) -- (\x@mid pt, \y@lower pt)
        node [/pgfgantt/today label node] {\ganttvalueof{today label}};%
    \fi%
  \end{scope}%
```

Store the x-extent of the bounding box in `\@tempdima`. Calculate the size by which the bounding box exceeds the “raw” canvas size. Write this information to the auxiliary file.

```
  \pgfextractx{\@tempdima}{%
    \pgfpointdiff{\pgfpointanchor{current bounding box}{south west}}%
                 {\pgfpointanchor{current bounding box}{north east}}%
  }%
  \pgfmathparse{\@tempdima -\ganttvalueof{x unit} * \gtt@chartwidth}%
  \protected@write\@auxout{}{%
    \string\gtt@chartextrasize{\the\gtt@chartid}{\pgfmathresult pt}}%
```

Increase the chart counter.

```
  \global\advance\gtt@chartid by1\relax%
```

At the end of a `ganttchart`, we also close the `tikzpicture` if we started it implicitly.

```
  \ifgtt@tikzpicture\else\end{tikzpicture}\fi%
}

```

3.6 Starting a New Line

`\ganttnewline`
`\local@drawarg`

Unless the optional argument of `\ganttnewline` is empty, this macro adds a horizontal grid rule between the current and the new line. The style of this line, which is stored in `\local@drawarg`, is either `hgrid style` or the style specified in the optional argument. Anyway, `\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][]{%
  \begingroup%
  \def\local@drawarg{#1}%
  \def\@tempa{grid}%
  \ifx\local@drawarg\@empty\else%
    \ifx\local@drawarg\@tempa%
      \def\local@drawarg{/pgfgantt/hgrid style}%
    \fi%
    \pgfmathsetmacro\y@upper{%
      \gtt@lasttitleline * \ganttvalueof{y unit title}%
      + (\gtt@currentline - \gtt@lasttitleline - 1)%
      * \ganttvalueof{y unit chart}%
    }
    \expandafter\draw\expandafter[\local@drawarg]
      (0pt, \y@upper pt) --
      (\gtt@chartwidth * \ganttvalueof{x unit}, \y@upper pt);%
  \fi%
  \global\advance\gtt@currentline by-1\relax%
  \ifgtt@intitle\global\advance\gtt@lasttitleline by-1\relax\fi%
  \global\gtt@lasttitleslot=0\relax%
  \endgroup%
}

```

3.7 Vertical rules

Keys related to the vertical rules.

vrule offset
vrule
vrule label font
vrule label node

```
\@gtt@keydef{vrule offset}{1}
\@gtt@stylekeydef{vrule}{dashed, line width=1pt}
\@gtt@keydef{vrule label font}{\normalfont}
\@gtt@stylekeydef{vrule label node}{%
  anchor=north, font=\ganttvalueof{vrule label font}%
}

```

A count for storing the vrule time slot.

\gtt@vrule@slot

```
\newcount\gtt@vrule@slot

```

Calculate the coordinates for the vertical rule and draw it.

\ganttvrule

```
\newcommand\ganttvrule[3][]{%
  \begingroup
  \ganttset{#1}
  \gtt@tsstojulian{#3}{\gtt@vrule@slot}%
  \gtt@juliantotimeslot{\gtt@vrule@slot}{\gtt@vrule@slot}%
  \pgfmathsetmacro\y@upper{%
    \gtt@lasttitleline * \ganttvalueof{y unit title}%
  }%
  \pgfmathsetmacro\y@lower{%
    \gtt@lasttitleline * \ganttvalueof{y unit title}%
    + (\gtt@currentline - \gtt@lasttitleline - 1)%
    * \ganttvalueof{y unit chart}%
  }%
  \pgfmathsetmacro\x@mid{%
    (\gtt@vrule@slot - 1 + \ganttvalueof{vrule offset})%
    * \ganttvalueof{x unit}%
  }%
  \draw [/pgfgantt/vrule]
    (\x@mid pt, \y@upper pt) -- (\x@mid pt, \y@lower pt)
    node [/pgfgantt/vrule label node] {#2};%
  \endgroup
}

```

3.8 Titles

Keys that influence title elements. The parameter token `#1` in the value of `title label text` is replaced by the argument of `\gtt@titlelabeltext`. Note that `\@gtt@keydef` cannot define `title list options`, since `\@gtt@titlelistoptions` is expanded after a `\foreach` statement, where `\ganttvalueof` will not work.

`title`
`title label font`
`title label node`
`title label text`
`\gtt@titlelabeltext`
`title list options`
`\gtt@titlelistoptions`
`title left shift`
`title right shift`
`title top shift`
`title height`
`include title in canvas`
`\ifgtt@includetitle`

```
\@gtt@stylekeydef{title}{shape=rectangle, inner sep=0pt, draw, fill=white}
\@gtt@keydef{title label font}{\small}
\@gtt@stylekeydef{title label node}{%
  anchor=center, font=\ganttvalueof{title label font}%
}
\ganttset{%
  title label text/.code={%
    \def\gtt@titlelabeltext##1{#1}%
  },%
  title label text=\strut#1,%
  title list options/.code={%
    \def\gtt@titlelistoptions{[#1]}%
  },%
  title list options={var=\x, evaluate=\x}%
}
\@gtt@keydef{title left shift}{0}
\@gtt@keydef{title right shift}{0}
\@gtt@keydef{title top shift}{0}
\@gtt@keydef{title height}{.6}
\newif\ifgtt@includetitle
\ganttset{%
  include title in canvas/.is if=gtt@includetitle,%
  include title in canvas
}
```

Keys for title calendars.

`calendar week text`
`time slot unit`
`\ifgtt@timeslotunit@day`
`\ifgtt@timeslotunit@month`
`\ifgtt@timeslotunit@year`

```
\@gtt@keydef{calendar week text}{Week~\currentweek}
\newif\ifgtt@timeslotunit@day
\newif\ifgtt@timeslotunit@month
\newif\ifgtt@timeslotunit@year
\ganttset{%
  time slot unit/.is choice,
  time slot unit/day/.code={%
```

```
    \gtt@timeslotunit@daytrue%
    \gtt@timeslotunit@monthfalse%
    \gtt@timeslotunit@yearfalse%
  },%
  time slot unit/month/.code={%
    \gtt@timeslotunit@dayfalse%
    \gtt@timeslotunit@monthtrue%
    \gtt@timeslotunit@yearfalse%
  },%
  time slot unit/year/.code={%
    \gtt@timeslotunit@dayfalse%
    \gtt@timeslotunit@monthfalse%
    \gtt@timeslotunit@yeartrue%
  },%
  time slot unit=day
}

```

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

`\gantttitle`
`\x@left`
`\x@right`

```
\newcommand\gantttitle[3][]{%
  \begingroup%
  \ganttset{#1}%
  \pgfmathsetmacro\x@left{%
    (\gtt@lasttitleslot + \ganttvalueof{title left shift})%
    * \ganttvalueof{x unit}%
  }%
  \pgfmathsetmacro\x@right{%
    (\gtt@lasttitleslot + #3 + \ganttvalueof{title right shift})%
    * \ganttvalueof{x unit}%
  }%
  \pgfmathsetmacro\x@mid{%
    (\x@left + \x@right) / 2%
  }%
  \pgfmathsetmacro\x@size{%
    \x@right - \x@left%
  }%
  \pgfmathsetmacro\y@upper{%
    (\gtt@currentline - \ganttvalueof{title top shift})%
    * \ganttvalueof{y unit title}%
  }%
  \pgfmathsetmacro\y@lower{%
    (\gtt@currentline - \ganttvalueof{title top shift}%
    - \ganttvalueof{title height}) * \ganttvalueof{y unit title}%
  }%
  \pgfmathsetmacro\y@mid{%
    (\y@upper + \y@lower) / 2%
  }%
  \pgfmathsetmacro\y@size{%
    \y@upper - \y@lower%
  }%
  \path (\x@mid pt, \y@mid pt)
    node [/pgfgantt/title, minimum width=\x@size pt,
          minimum height=\y@size pt] {}
    node [/pgfgantt/title label node] {\gtt@titlelabeltext{#2}};%
  \global\advance\gtt@lasttitleslot by#3\relax%
  \endgroup%
}

```

\ganttitlelist

`\gantttitlelist` generates title elements by repeatedly calling `\gantttitle`. Since the latter always starts after the last time slot occupied by the previous element, `\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%
}

```

\ifgtt@titlecalendarstar
\gantttitlecalendar
\gantttitlecalendar*

`\gantttitlecalendar` checks whether it is invoked in the starred or nonstarred form, sets `\ifgtt@titlecalendarstar` accordingly and then starts a command relaying chain.

```
\newif\ifgtt@titlecalendarstar
\newcommand\gantttitlecalendar{%
  \@gtt@ifstar%
    {\gtt@titlecalendarstartrue\@gantttitlecalendar}%
    {\gtt@titlecalendarstarfalse\@gantttitlecalendar}%
}

```

\@gantttitlecalendar

The first command in the relaying chain, `\@gantttitlecalendar[`⟨*options*⟩`]`, processes the ⟨*options*⟩. If it was executed by the starred form of `\gantttitlecalendar`, it calls the second command in the chain. Otherwise, it directly calls the third command in the chain.

```
\newcommand\@gantttitlecalendar[1][]{
  \begingroup%
  \ganttset{#1}%
  \ifgtt@titlecalendarstar%
    \expandafter\@@gantttitlecalendar%
  \else%
    \expandafter\@@@gantttitlecalendar\expandafter%
      {\expandafter\gtt@startjulian\expandafter}\expandafter%
      {\expandafter\gtt@endjulian\expandafter}%
  \fi%
}

```

`\@@gantttitlecalendar`

The second command in the relaying chain,
`\@@gantttitlecalendar{`⟨*start tss*⟩`}{`⟨*end tss*⟩`}`,
reads two mandatory arguments from the input stream and converts them to Julian dates. Finally, it calls the third command in the chain.

```
\newcommand\@@gantttitlecalendar[2]{
  \gtt@tsstojulian{#1}{\@tempcnta}%
  \gtt@tsstojulian{#2}{\@tempcntb}%
  \@@@gantttitlecalendar{\@tempcnta}{\@tempcntb}%
}

```

`\@@@gantttitlecalendar`
`\gtt@calendar@startdate`
`\gtt@calendar@enddate`

The third and last command in the relaying chain, `\@@@gantttitlecalendar{`⟨*start Julian*⟩`}{`⟨*end Julian*⟩`}{`⟨*calendar lines*⟩`}`, stores the start and end ISO-standard dates of the calendar in `\gtt@calendar@startdate` and `\gtt@calendar@enddate`, respectively. Then, it executes the keys in ⟨*calendar lines*⟩, which reside in path `/pgfgantt/calendar`.

```
\newcommand\@@@gantttitlecalendar[3]{%
  \pgfcalendarjuliantodate{#1}{\@tempa}{\@tempb}{\@tempc}%
  \edef\gtt@calendar@startdate{\@tempa-\@tempb-\@tempc}%
  \pgfcalendarjuliantodate{#2}{\@tempa}{\@tempb}{\@tempc}%
  \edef\gtt@calendar@enddate{\@tempa-\@tempb-\@tempc}%
  \gtt@calendar@eolfalse%
  \pgfqkeys{/pgfgantt/calendar}{#3}%
  \endgroup%
}
```

`\ifgtt@calendar@eol`
`\gtt@calendar@slots`
`\gtt@calendar@weeknumber`
`\gtt@calendar@startofweek`

Booleans and counts for drawing title calendars: `\ifgtt@calendar@eol` is true if `\ganttcalendar` should start a new calendar line. `\gtt@calendar@slots` is the number of time slots a calendar element will cover. `\gtt@calendar@weeknumber` is the current week number in a calendar line of type `week`. `\gtt@calendar@startofweek` is the Julian date of the Monday in the current week.

```
\newif\ifgtt@calendar@eol
\newcount\gtt@calendar@slots
\newcount\gtt@calendar@weeknumber
\newcount\gtt@calendar@startofweek
```

`\@gtt@getfourthdigit`
`\@gtt@getdecade`

`\@gtt@getfourthdigit` returns the fourth digit of a year, while `\@gtt@getdecade` returns the first three digits of a year.

```
\def\@gtt@getfourthdigit#1#2#3#4{#4}
\def\@gtt@getdecade#1#2#3#4{#1#2#3}
```

We define a new check for `\pgfcalendarifdate` as described in the `pgfcalendar` manual: `end of decade=`⟨*date*⟩ returns true if a date marks the end of a decade as defined by ⟨*date*⟩. For instance, if ⟨*date*⟩ is `2009-12-31`, then the conditional will be true for the dates `1999-12-31`, `2009-12-31`, `2019-12-31` and so on.

```
\pgfkeys{%
  /pgf/calendar/end of decade/.code={
    \begingroup
    \pgfcalendardatetojulian{#1}{\@tempcnta}
    \pgfcalendarjuliantodate{\@tempcnta}{\@tempa}{\@tempb}{\@tempc}
    \edef\endofdecade{\expandafter\@gtt@getfourthdigit\@tempa}
    \edef\querydecade{\expandafter\@gtt@getfourthdigit\pgfcalendarifdateyear}
    \ifnum\endofdecade=\querydecade\relax%
      \ifnum\pgfcalendarifdatemonth=\@tempb\relax%
        \ifnum\pgfcalendarifdateday=\@tempc\relax%
          \global\pgfcalendarmatchestrue%
        \fi%
      \fi%
    \fi%
    \endgroup
  }%
}
```

For each ⟨*line type*⟩, we define a corresponding key `/pgfgantt/calendar/`⟨*line type*⟩. This key performs the necessary calculations and draws one or several `\gantttitle`s. Line type `decade` draws decades.

```
\ganttset{%
  calendar/decade/.code={%
    \ifgtt@calendar@eol\ganttnewline\fi%
    \begingroup%
    \gtt@calendar@slots=1\relax%
    \ifgtt@timeslotunit@year%
      \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
        \ifdate{end of decade=2009-12-31}{%
          \gantttitle{%
            \expandafter\@gtt@getdecade\pgfcalendarcurrentyear%
            0s%
          }{\the\gtt@calendar@slots}%
          \gtt@calendar@slots=0\relax%
        }{
          \ifdate{equals=01-01}{%
            \ifnum\pgfcalendarcurrentjulian>\pgfcalendarbeginjulian\relax%
              \advance\gtt@calendar@slots by1\relax%
            \fi
          }{}%
        }%
        \ifdate{equals=\pgfcalendarendiso}{%
          \ifnum\gtt@calendar@slots=0\relax\else%
            \gantttitle{%
              \expandafter\@gtt@getdecade\pgfcalendarcurrentyear%
              0s%
            }{\the\gtt@calendar@slots}%
          \fi%
        }{}%
      }%
    \fi%
    \endgroup%
    \gtt@calendar@eoltrue%
  },%
```

Line type `year` draws years.

```
calendar/year/.code={%
  \ifgtt@calendar@eol\ganttnewline\fi%
  \begingroup%
  \gtt@calendar@slots=1\relax%
  \ifgtt@timeslotunit@year%
    \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
      \ifdate{equals=12-31}{%
        \gantttitle{\pgfcalendarcurrentyear}{1}%
        \gtt@calendar@slots=1\relax%
      }{
        \advance\gtt@calendar@slots by1\relax%
      }%
      \ifdate{equals=\pgfcalendarendiso}{%
        \ifnum\gtt@calendar@slots=1\relax\else%
          \gantttitle{\pgfcalendarcurrentyear}{1}%
        \fi%
      }{}%
    }%
  \fi%
  \ifgtt@timeslotunit@month%
    \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
      \ifdate{equals=12-31}{%
        \gantttitle{\pgfcalendarcurrentyear}{\the\gtt@calendar@slots}%
        \gtt@calendar@slots=1\relax%
      }{%
        \ifdate{end of month=1}{%
          \advance\gtt@calendar@slots by1\relax%
        }{}%
      }%
      \ifdate{equals=\pgfcalendarendiso}{%
        \ifdate{end of month=1}{%
          \advance\gtt@calendar@slots by-1\relax%
        }{}%
        \ifdate{equals=12-31}{}{%
          \gantttitle{\pgfcalendarcurrentyear}{\the\gtt@calendar@slots}%
        }%
      }{}%
    }%
  \fi%
```

```
    \ifgtt@timeslotunit@day%
      \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
        \ifdate{equals=12-31}{%
          \gantttitle{\pgfcalendarcurrentyear}{\the\gtt@calendar@slots}%
          \gtt@calendar@slots=1\relax%
        }{%
          \advance\gtt@calendar@slots by1\relax%
        }%
        \ifdate{equals=\pgfcalendarendiso}{%
          \ifnum\gtt@calendar@slots=1\relax\else%
            \advance\gtt@calendar@slots by-1\relax%
            \gantttitle{\pgfcalendarcurrentyear}{\the\gtt@calendar@slots}%
          \fi%
        }{}%
      }%
    \fi%
    \endgroup%
    \gtt@calendar@eoltrue%
  },%
```

Line type `month`=⟨*format*⟩ draws months. Internally, a month is represented by a number between 1 (January) and 12 (December). However, when the title element is drawn, this number is fed to the macro `\pgfcalendarmonth`⟨*format*⟩ and possibly converted.

```
  calendar/month/.code={%
    \ifgtt@calendar@eol\ganttnewline\fi%
    \begingroup%
    \gtt@calendar@slots=1\relax%
    \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
      \ifdate{end of month=1}{%
        \gantttitle{%
          \csname pgfcalendarmonth#1\endcsname{\pgfcalendarcurrentmonth}%
        }{%
          \ifgtt@timeslotunit@month1\fi%
          \ifgtt@timeslotunit@day\the\gtt@calendar@slots\fi%
        }%
        \gtt@calendar@slots=1\relax%
      }{%
        \advance\gtt@calendar@slots by1\relax%
      }%
      \ifdate{equals=\pgfcalendarendiso}{%
        \ifnum\gtt@calendar@slots=1\relax\else%
          \advance\gtt@calendar@slots by-1\relax%
          \gantttitle{%
            \csname pgfcalendarmonth#1\endcsname{\pgfcalendarcurrentmonth}%
          }{%
            \ifgtt@timeslotunit@month1\fi%
            \ifgtt@timeslotunit@day\the\gtt@calendar@slots\fi%
          }%
        \fi%
      }{}%
    }%
    \endgroup%
    \gtt@calendar@eoltrue%
  },%
```

Line type `week`=⟨*number*⟩ draws weeks. The first week receives ⟨*number*⟩, which is also saved in `\currentweek`. This key also defines the macros `\startyear`, `\startmonth` and `\startday`, which store the year, month and day of the current week's Monday. These four macros can be used in the value of `calendar week text`.

`\startyear`
`\startmonth`
`\startday`
`\currentweek`

```
calendar/week/.code={%
  \ifgtt@calendar@eol\ganttnewline\fi%
  \begingroup%
  \gtt@calendar@slots=1\relax%
  \gtt@calendar@weeknumber=#1\relax%
  \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
    \ifdate{Sunday}{%
      \gtt@calendar@startofweek=\pgfcalendarcurrentjulian\relax%
      \advance\gtt@calendar@startofweek by1\relax%
      \advance\gtt@calendar@startofweek by-\gtt@calendar@slots\relax%
      \pgfcalendarjuliantodate{\gtt@calendar@startofweek}%
        {\startyear}{\startmonth}{\startday}%
      \def\currentweek{\the\gtt@calendar@weeknumber}%
      \gantttitle{%
        \ganttvalueof{calendar week text}%
      }{%
        \the\gtt@calendar@slots%
      }%
      \gtt@calendar@slots=1\relax%
      \advance\gtt@calendar@weeknumber by1\relax%
    }{%
      \advance\gtt@calendar@slots by1%
    }%
    \ifdate{equals=\pgfcalendarendiso}{%
      \ifnum\gtt@calendar@slots=1\relax\else%
        \advance\gtt@calendar@slots by-1\relax%
        \gtt@calendar@startofweek=\pgfcalendarcurrentjulian\relax%
        \advance\gtt@calendar@startofweek by1\relax%
        \advance\gtt@calendar@startofweek by-\gtt@calendar@slots\relax%
        \pgfcalendarjuliantodate{\gtt@calendar@startofweek}%
          {\startyear}{\startmonth}{\startday}%
        \def\currentweek{\the\gtt@calendar@weeknumber}%
        \gantttitle{%
          \ganttvalueof{calendar week text}%
        }{%
          \the\gtt@calendar@slots%
        }%
      \fi%
    }{}%
  }%
  \endgroup%
  \gtt@calendar@eoltrue%
},%
calendar/week/.default=1,%
```

Line type `weekday`=⟨*format*⟩ draws weekdays. Internally, a weekday is represented by a number between 0 (Monday) and 6 (Sunday). However, when the title element is drawn, this number is fed to the macro `\pgfcalendarweekday`⟨*format*⟩ and possibly converted.

```
  calendar/weekday/.code={%
    \ifgtt@calendar@eol\ganttnewline\fi%
    \begingroup%
    \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
      \gantttitle{%
        \csname pgfcalendarweekday#1\endcsname{\pgfcalendarcurrentweekday}%
      }{1}%
    }%
    \endgroup%
    \gtt@calendar@eoltrue%
  },%
```

Line type `day`=⟨*format*⟩ draws days of the month.

```
  calendar/day/.code={%
    \ifgtt@calendar@eol\ganttnewline\fi%
    \begingroup%
    \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
      \gantttitle{\pgfcalendarcurrentday}{1}%
    }%
    \endgroup%
    \gtt@calendar@eoltrue%
  }%
}

```

3.9 Chart Elements

Keys that apply to all chart elements. The parameter token `#1` in the value of `progress label text` is replaced by the argument of `\gtt@progresslabeltext`.

`progress`
`\gtt@progress`
`progress label text`
`\gtt@progresslabeltext`
`name`
`chart element start borde`
`\ifgtt@ce@startatleftbord`
`inline`
`\ifgtt@inline`

```
\ganttset{%
  progress/.code={%
    \def\gtt@progress{#1}%
  },%
  progress=none,%
  progress label text/.code={%
    \def\gtt@progresslabeltext##1{#1}%
  },%
  progress label text={%
    \pgfmathprintnumber[precision=0, verbatim]{#1}\% complete%
  }%
}
\@gtt@keydef{name}{}
\newif\ifgtt@ce@startatleftborder
\ganttset{%
  chart element start border/.is choice,%
  chart element start border/left/.code=\gtt@ce@startatleftbordertrue,%,
  chart element start border/right/.code=\gtt@ce@startatleftborderfalse,%
  chart element start border=left%
}
\newif\ifgtt@inline
\ganttset{%
  inline/.is if=gtt@inline,%
  inline=false%
}

```

`\gtt@lastelement`
`\gtt@currentelement`
`\ifgtt@draw@complete`
`\ifgtt@draw@incomplete`
`\ifgtt@draw@clip`
`\gtt@left@slot`
`\gtt@right@slot`

The macros `\gtt@lastelement` and `\gtt@currentelement` save the name of the current and last chart element drawn. Thereby, the `\ganttlinked...` macros can add a link connecting them. `\ifgtt@draw@complete`, `\ifgtt@draw@incomplete` and `\ifgtt@draw@clip` decide whether to draw the complete and incomplete part of a chart element and if these parts are clipped. `\gtt@left@slot` and `\gtt@right@slot` store a chart element's start and end time slot, respectively.

```
\def\gtt@lastelement{}
\def\gtt@currentelement{}
\newif\ifgtt@draw@complete
\newif\ifgtt@draw@incomplete
\newif\ifgtt@draw@clip
\newcount\gtt@left@slot
\newcount\gtt@right@slot

```

`\gtt@chartelement`
`\local@timeslotmodifier`
`\gtt@name`

`\gtt@chartelement[`⟨*options*⟩`]{`⟨*label*⟩`}{`⟨*start tss*⟩`}{`⟨*end tss*⟩`}{`⟨*type*⟩`}` is the generic command for drawing chart elements of a certain ⟨*type*⟩. First, `\gtt@chartelement` converts ⟨*start tss*⟩ to `\gtt@left@slot` and ⟨*end tss*⟩ to `\gtt@right@slot`. Then it calculates the usual coordinates. `\gtt@name` stores the name of the chart element.

```
\newcommand\gtt@chartelement[5][]{%
  \begingroup%
  \ganttset{#1}%
  \gtt@tsstojulian{#3}{\gtt@left@slot}%
  \gtt@juliantotimeslot{\gtt@left@slot}{\gtt@left@slot}%
  \gtt@tsstojulian{#4}{\gtt@right@slot}%
  \gtt@juliantotimeslot{\gtt@right@slot}{\gtt@right@slot}%
  \def\local@timeslotmodifier{-1}%
  \ifgtt@ce@startatleftborder\else%
    \ifnum\gtt@left@slot=\gtt@right@slot\relax\else%
      \def\local@timeslotmodifier{0}%
    \fi%
  \fi%
  \pgfmathsetmacro\x@left{%
    (\gtt@left@slot + \local@timeslotmodifier%
      + \ganttvalueof{#5 left shift})%
    * \ganttvalueof{x unit}%
  }%
  \pgfmathsetmacro\x@right{%
    (\gtt@right@slot + \ganttvalueof{#5 right shift})%
    * \ganttvalueof{x unit}%
  }%
  \pgfmathsetmacro\x@mid{%
    (\x@left + \x@right) / 2%
  }%
  \pgfmathsetmacro\x@size{%
    \x@right - \x@left%
  }%
  \pgfmathsetmacro\y@upper{%
    \gtt@lasttitleline * \ganttvalueof{y unit title}
    + (\gtt@currentline - \gtt@lasttitleline
    - \ganttvalueof{#5 top shift}) * \ganttvalueof{y unit chart}%
  }%
  \pgfmathsetmacro\y@lower{%
    \y@upper - \ganttvalueof{#5 height} * \ganttvalueof{y unit chart}%
  }%
  \pgfmathsetmacro\y@mid{%
    (\y@upper + \y@lower) / 2%
  }%
  \pgfmathsetmacro\y@size{%
    \y@upper - \y@lower%
  }%
  \edef\gtt@name{\ganttvalueof{name}}%
  \ifx\gtt@name\@empty\edef\gtt@name{elem\the\gtt@elementid}\fi%
```

`\local@none`

Depending on the values of `progress` and `today`, we determine the correct value for `\gtt@progress`. A value between 0 and 100 corresponds to a percentage of completeness. A value of 999 indicates that the chart element has no associated progress.

```
\def\local@none{none}%
\ifx\gtt@progress\local@none%
  \def\gtt@progress{999}%
\else%
  \def\@tempa{today}%
  \ifx\gtt@progress\@tempa%
    \edef\@tempa{\ganttvalueof{today}}%
    \ifx\@tempa\local@none%
      \@gtt@PackageWarning{%
        Value of today is ‘none’. Ignoring ‘progress=today’%
      }%
      \def\gtt@progress{999}%
    \else\ifnum\gtt@today@slot>\gtt@right@slot\relax%
      \def\gtt@progress{100}%
    \else\ifnum\gtt@today@slot<\gtt@left@slot\relax%
      \def\gtt@progress{0}%
    \else%
      \pgfmathsetmacro\gtt@progress{%
        (\gtt@today@slot - \gtt@left@slot - \local@timeslotmodifier)%
        / (\gtt@right@slot - \gtt@left@slot - \local@timeslotmodifier)%
        * 100%
      }%
    \fi\fi\fi%
  \fi%
\fi%
```

`\x@clip@size`

Now we determine whether only the complete part of the chart element, only its incomplete one or both are drawn. In the former two cases, we refrain from clipping the (in)complete part.

```
\gtt@draw@completetrue%
\gtt@draw@incompletetrue%
\gtt@draw@cliptrue%
\ifdim\gtt@progress pt<0.001pt\relax%
  \gtt@draw@completefalse%
  \gtt@draw@clipfalse%
\else\ifdim\gtt@progress pt>99.999pt\relax%
  \gtt@draw@incompletefalse%
  \gtt@draw@clipfalse%
\fi\fi%
\ifgtt@draw@clip%
  \pgfmathsetmacro\x@clip@size{%
    (\gtt@right@slot - \gtt@left@slot - \local@timeslotmodifier)%
    * \gtt@progress / 100%
  }
  \pgfmathsetmacro\x@clip{%
    (\gtt@left@slot + \local@timeslotmodifier + \x@clip@size%
    + \ganttvalueof{today offset} - 1) * \ganttvalueof{x unit}%
  }%
\fi%
```

We draw the chart element within a `pgfinterruptboundingbox`, since we clip a large area of the canvas in order to avoid removing parts of the chart element border.

```
\begin{pgfinterruptboundingbox}%
  \begin{scope}%
    \ifgtt@draw@clip%
      \clip (\x@left pt - 10cm, \y@upper pt + 10cm) rectangle
        (\x@clip pt, \y@lower pt - 10cm);%
    \fi%
    \ifgtt@draw@complete%
```

```
      \node [/pgfgantt/#5, minimum width=\x@size pt,
             minimum height=\y@size pt]
        (\gtt@name) at (\x@mid pt, \y@mid pt) {};%
    \fi%
  \end{scope}%
  \begin{scope}%
    \ifgtt@draw@clip%
      \clip (\x@clip pt, \y@upper pt + 10cm) rectangle
        (\x@right pt + 10cm, \y@lower pt - 10cm);%
    \fi%
    \ifgtt@draw@incomplete%
      \node [/pgfgantt/#5 incomplete, minimum width=\x@size pt,
             minimum height=\y@size pt]
        (\gtt@name) at (\x@mid pt, \y@mid pt) {};%
    \fi%
  \end{scope}%
\end{pgfinterruptboundingbox}%
```

If `progress` differs from `none` and `progress label text` differs from `\relax`, the progress label is drawn.

```
\ifdim\gtt@progress pt=999pt\relax\else%
  \expandafter\ifx\gtt@progresslabeltext\relax\relax\else%
    \node at (\gtt@name.\ganttvalueof{#5 progress label anchor})
      [/pgfgantt/#5 progress label node]
      {\gtt@progresslabeltext{\gtt@progress}};%
  \fi%
\fi%
```

If ⟨*label*⟩ is not empty, a label is printed. Its anchor is either at the ⟨*type*⟩ `inline label anchor` of the chart element (`inline=true`) or at the left canvas border halfway between the upper and lower y-coordinate of the chart element (`inline=false`).

```
\def\@tempa{#2}%
\ifx\@tempa\@empty\else%
  \ifgtt@inline%
    \node at (\gtt@name.\ganttvalueof{#5 inline label anchor})
      [/pgfgantt/#5 inline label node]
      {\csname gtt@#5labeltext\endcsname{#2}};%
  \else%
    \node at (0, \y@mid pt)
      [/pgfgantt/#5 label node]
      {\csname gtt@#5labeltext\endcsname{#2}};%
  \fi%
\fi%
```

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

```
\xdef\gtt@lastelement{\gtt@currentelement}%
```

```
  \xdef\gtt@currentelement{\gtt@name}%
  \global\advance\gtt@elementid by1\relax%
  \global\gtt@intitlefalse%
  \endgroup%
}

```

`\newganttchartelement` checks whether it was invoked in the starred or nonstarred form and executes `\@newganttchartelement@one` or `\@newganttchartelement@two`, respectively.

`\newganttchartelement`
`\newganttchartelement*`

```
\def\newganttchartelement{%
  \@gtt@ifstar\@newganttchartelement@one\@newganttchartelement@two%
}

```

Both `\@newganttchartelement@one{`⟨*type*⟩`}` and `...two{`⟨*type*⟩`}` define two macros `\gantt`⟨*type*⟩ and `\ganttlinked`⟨*type*⟩, which draw a singular chart element or one that is linked to its predecessor. However, the newly defined macros will take three or four mandatory arguments (cf. `\ganttmilestone` vs. `\ganttbar`). At the end, we execute `\@newganttchartelement@definekeys` to process the second mandatory argument of `\newganttchartelement`.

`\@newganttchartelement@on`
`\@newganttchartelement@tw`

```
\newcommand\@newganttchartelement@one[1]{%
  \expandafter\newcommand\csname gantt#1\endcsname[3][]{%
    \gtt@chartelement[##1]{##2}{##3}{##3}{#1}%
  }%
  \expandafter\newcommand\csname ganttlinked#1\endcsname[3][]{%
    \begingroup%
    \ganttset{##1}%
    \gtt@chartelement{##2}{##3}{##3}{#1}%
    \ganttlink{\gtt@lastelement}{\gtt@currentelement}%
    \endgroup%
  }%
  \@newganttchartelement@definekeys{#1}%
}

\newcommand\@newganttchartelement@two[1]{%
  \expandafter\newcommand\csname gantt#1\endcsname[4][]{%
    \gtt@chartelement[##1]{##2}{##3}{##4}{#1}%
  }%
  \expandafter\newcommand\csname ganttlinked#1\endcsname[4][]{%
    \begingroup%
    \ganttset{##1}%
    \gtt@chartelement{##2}{##3}{##4}{#1}%
    \ganttlink{\gtt@lastelement}{\gtt@currentelement}%
    \endgroup%
  }%
  \@newganttchartelement@definekeys{#1}%
}
```

`\@newganttchartelement@definekeys{`⟨*type*⟩`}{`⟨*key-value list*⟩`}` introduces 14 keys for the newly generated chart element ⟨*type*⟩.

\@newganttchartelement@de

```
\newcommand\@newganttchartelement@definekeys[2]{%
  \@gtt@stylekeydef{#1}{shape=rectangle, inner sep=0pt, draw, fill=white}%
  \@gtt@stylekeydef{#1 incomplete}{/pgfgantt/#1, fill=black!25}%
  \@gtt@keydef{#1 label font}{\normalsize}%
  \@gtt@stylekeydef{#1 label node}{%
    anchor=east, font=\ganttvalueof{#1 label font}%
  }%
  \@gtt@keydef{#1 inline label anchor}{center}%
  \@gtt@stylekeydef{#1 inline label node}{%
    anchor=center, font=\ganttvalueof{#1 label font}%
  }%
  \@gtt@keydef{#1 progress label anchor}{east}%
  \@gtt@keydef{#1 progress label font}{\scriptsize}%
  \@gtt@stylekeydef{#1 progress label node}{%
    anchor=west, font=\ganttvalueof{#1 progress label font}%
  }%
  \@gtt@keydef{#1 left shift}{0}%
  \@gtt@keydef{#1 right shift}{0}%
  \@gtt@keydef{#1 top shift}{.3}%
  \@gtt@keydef{#1 height}{.4}%
  \ganttset{%
    #1 label text/.code={%
      \expandafter\def\csname gtt@#1labeltext\endcsname####1{##1}%
    },%
    #1 label text=\strut##1,%
    #2%
  }%
}

```

Code for the predefined chart element type `bar`.

\ganttbar
\ganttlinkedbar
bar
bar incomplete
bar label text
bar label font
bar label node
bar inline label anchor
bar inline label node
bar progress label anchor
bar progress label font
bar progress label node
bar left shift
bar right shift
bar top shift
bar height

```
\newganttchartelement{bar}{%
  bar/.style={shape=ganttbar, inner sep=0pt, draw, fill=white},%
  bar incomplete/.style={/pgfgantt/bar, fill=black!25},%
  bar label text=\strut#1,%
  bar label font=\normalsize,%
  bar label node/.style={%
    anchor=east, font=\ganttvalueof{bar label font}%
  },%
  bar inline label anchor=center,%
  bar inline label node/.style={%
    anchor=center, font=\ganttvalueof{bar label font}%
  },%
  bar progress label anchor=east,%
  bar progress label font=\scriptsize,%
  bar progress label node/.style={%
    anchor=west, font=\ganttvalueof{bar progress label font}%
  },%
  bar left shift=0,%
  bar right shift=0,%
  bar top shift=.3,%
  bar height=.4%
}

```

Code for the predefined chart element type `group`.

\ganttgroup
\ganttlinkedgroup
group
group incomplete
group label text
group label font
group label node
group inline label anchor
group inline label node
group progress label anch
group progress label font
group progress label node
group left shift
group right shift
group top shift
group height

```
\newganttchartelement{group}{%
  group/.style={shape=ganttgroup, inner sep=0pt, fill=black},%
  group incomplete/.style={/pgfgantt/group, fill=black!25},%
  group label text=\strut#1,%
  group label font=\bfseries,%
  group label node/.style={%
    anchor=east, font=\ganttvalueof{group label font}%
  },%
  group inline label anchor=center,%
  group inline label node/.style={%
    anchor=south, font=\ganttvalueof{group label font}%
  },%
  group progress label anchor=east,%
  group progress label font=\scriptsize,%
  group progress label node/.style={%
    anchor=west, font=\ganttvalueof{group progress label font}%
  },%
  group left shift=-.1,%
  group right shift=.1,%
  group top shift=.4,%
  group height=.2%
}
```

More keys for the appearance of groups.

group right peak tip posi
group right peak width
group right peak height
group left peak tip posit
group left peak width
group left peak height
group peaks tip position
group peaks width
group peaks height

```
\@gtt@keydef{group right peak tip position}{.5}
\@gtt@keydef{group right peak width}{.4}
\@gtt@keydef{group right peak height}{.1}
\@gtt@keydef{group left peak tip position}{.5}
\@gtt@keydef{group left peak width}{.4}
\@gtt@keydef{group left peak height}{.1}
\ganttset{%
  group peaks tip position/.code={%
    \ganttset{%
      group left peak tip position=#1,%
      group right peak tip position=#1%
    }%
  },%
  group peaks width/.code={%
    \ganttset{%
      group left peak width=#1,%
      group right peak width=#1%
    }%
  },%
  group peaks height/.code={%
    \ganttset{%
      group left peak height=#1,%
      group right peak height=#1%
    }%
  }%
}

```

Code for the predefined chart element type `milestone`.

\ganttmilestone
\ganttlinkedmilestone
milestone
milestone incomplete
milestone label text
milestone label font
milestone label node
milestone inline label an
milestone inline label no
milestone progress label
milestone progress label
milestone progress label
milestone left shift
milestone right shift
milestone top shift
milestone height

```
\newganttchartelement*{milestone}{%
  milestone/.style={%
    shape=ganttmilestone, inner sep=0pt, draw, fill=black%
  },%
  milestone incomplete/.style={/pgfgantt/milestone, fill=black!25},%
  milestone label text=\strut#1,%
  milestone label font=\itshape,%
  milestone label node/.style={%
    anchor=east, font=\ganttvalueof{milestone label font}%
  },%
  milestone inline label anchor=center,%
  milestone inline label node/.style={%
    anchor=south, font=\ganttvalueof{milestone label font}%
  },%
  milestone progress label anchor=center,%
  milestone progress label font=\scriptsize,%
  milestone progress label node/.style={%
    anchor=west, font=\ganttvalueof{milestone progress label font}%
  },%
  milestone left shift=.6,%
  milestone right shift=.4,%
  milestone top shift=.3,%
  milestone height=.4%
}

```

3.10 Node Shapes

Keys for configuring the additional anchors of the new node shapes.

on top fraction
on bottom fraction
on left fraction
on right fraction

```
\@gtt@keydef{on top fraction}{.5}
\@gtt@keydef{on bottom fraction}{.5}
\@gtt@keydef{on left fraction}{.5}
\@gtt@keydef{on right fraction}{.5}

```

Code for node shape `ganttbar`. Anchors and background path derive from node shape `rectangle`. The four additional anchors `on top`, `on bottom`, `on left` and `on right` are defined.

```
\pgfdeclareshape{ganttbar}{
  \inheritsavedanchors[from=rectangle]
  \inheritanchor[from=rectangle]{center}
  \inheritanchor[from=rectangle]{mid}
  \inheritanchor[from=rectangle]{base}
```

```
\inheritanchor[from=rectangle]{north}
\inheritanchor[from=rectangle]{south}
\inheritanchor[from=rectangle]{west}
\inheritanchor[from=rectangle]{mid west}
\inheritanchor[from=rectangle]{base west}
\inheritanchor[from=rectangle]{north west}
\inheritanchor[from=rectangle]{south west}
\inheritanchor[from=rectangle]{east}
\inheritanchor[from=rectangle]{mid east}
\inheritanchor[from=rectangle]{base east}
\inheritanchor[from=rectangle]{north east}
\inheritanchor[from=rectangle]{south east}
\inheritanchorborder[from=rectangle]
\anchor{on top}{
  \southwest
  \pgf@xa=\pgf@x
  \northeast
  \pgf@xb=\pgf@x
  \advance\pgf@xb by-\pgf@xa
  \pgf@xb=\ganttvalueof{on top fraction}\pgf@xb
  \advance\pgf@xa by\pgf@xb
  \pgf@x=\pgf@xa
}
\anchor{on bottom}{
  \northeast
  \pgf@xb=\pgf@x
  \southwest
  \pgf@xa=\pgf@x
  \advance\pgf@xb by-\pgf@xa
  \pgf@xb=\ganttvalueof{on bottom fraction}\pgf@xb
  \advance\pgf@xa by\pgf@xb
  \pgf@x=\pgf@xa
}
\anchor{on left}{
  \northeast
  \pgf@ya=\pgf@y
  \southwest
  \pgf@yb=\pgf@y
  \advance\pgf@yb by-\pgf@ya
  \pgf@yb=\ganttvalueof{on left fraction}\pgf@yb
  \advance\pgf@ya by\pgf@yb
  \pgf@y=\pgf@ya
}
\anchor{on right}{
  \southwest
  \pgf@yb=\pgf@y
  \northeast
  \pgf@ya=\pgf@y
  \advance\pgf@yb by-\pgf@ya
```

```
    \pgf@yb=\ganttvalueof{on right fraction}\pgf@yb
    \advance\pgf@ya by\pgf@yb
    \pgf@y=\pgf@ya
  }
  \inheritbackgroundpath[from=rectangle]
}

```

Code for node shape `ganttgroup`. Anchors derive from node shape `ganttbar`. The two additional anchors `left peak` and `right peak` are defined.

```
\pgfdeclareshape{ganttgroup}{
  \inheritsavedanchors[from=rectangle]
  \inheritanchor[from=rectangle]{center}
  \inheritanchor[from=rectangle]{mid}
  \inheritanchor[from=rectangle]{base}
  \inheritanchor[from=rectangle]{north}
  \inheritanchor[from=rectangle]{south}
  \inheritanchor[from=rectangle]{west}
  \inheritanchor[from=rectangle]{mid west}
  \inheritanchor[from=rectangle]{base west}
  \inheritanchor[from=rectangle]{north west}
  \inheritanchor[from=rectangle]{south west}
  \inheritanchor[from=rectangle]{east}
  \inheritanchor[from=rectangle]{mid east}
  \inheritanchor[from=rectangle]{base east}
  \inheritanchor[from=rectangle]{north east}
  \inheritanchor[from=rectangle]{south east}
  \inheritanchorborder[from=rectangle]
  \inheritanchor[from=ganttbar]{on top}
  \inheritanchor[from=ganttbar]{on bottom}
  \inheritanchor[from=ganttbar]{on left}
  \inheritanchor[from=ganttbar]{on right}
  \anchor{left peak}{
    \pgf@process{
      \pgfpointadd{
        \southwest
      }{
        \pgfpoint%
          {\pgfkeysvalueof{/pgf/outer xsep}}%
          {\pgfkeysvalueof{/pgf/outer ysep}}
      }
    }
    \pgfmathsetlength\pgf@x{
      \pgf@x + \ganttvalueof{group left peak tip position}
      * \ganttvalueof{group left peak width} * \ganttvalueof{x unit}
    }
    \pgfmathsetlength\pgf@y{
      \pgf@y - \ganttvalueof{group left peak height}
               * \ganttvalueof{y unit chart}
    }
  }
  \anchor{right peak}{
    \pgf@process{
      \pgfpointadd{
        \northeast
      }{
        \pgfpointscale{-1}{
          \pgfpoint%
            {\pgfkeysvalueof{/pgf/outer xsep}}%
            {\pgfkeysvalueof{/pgf/outer ysep}}
        }
      }
    }
    \pgf@xa=\pgf@x
    \pgf@process{
      \pgfpointadd{
        \southwest
      }{
        \pgfpoint%
          {\pgfkeysvalueof{/pgf/outer xsep}}%
          {\pgfkeysvalueof{/pgf/outer ysep}}
      }
    }
    \pgfmathsetlength\pgf@x{
      \pgf@xa - \ganttvalueof{group right peak tip position}
      * \ganttvalueof{group right peak width} * \ganttvalueof{x unit}
    }
    \pgfmathsetlength\pgf@y{
      \pgf@y - \ganttvalueof{group right peak height}
               * \ganttvalueof{y unit chart}
    }
  }
  \backgroundpath{
    \pgf@process{
      \pgfpointadd{
        \northeast
      }{
        \pgfpointscale{-1}{
          \pgfpoint%
            {\pgfkeysvalueof{/pgf/outer xsep}}%
            {\pgfkeysvalueof{/pgf/outer ysep}}
        }
      }
    }
    \pgf@xb=\pgf@x
    \pgf@ya=\pgf@y
    \pgf@process{
      \pgfpointadd{
        \southwest
      }{
        \pgfpoint%
          {\pgfkeysvalueof{/pgf/outer xsep}}%
          {\pgfkeysvalueof{/pgf/outer ysep}}
      }
    }
    \pgf@xa=\pgf@x
    \pgf@yb=\pgf@y
    \pgfpathmoveto{\pgfpoint{\pgf@xa}{\pgf@ya}}
    \pgfpathlineto{\pgfpoint{\pgf@xb}{\pgf@ya}}
    \pgfpathlineto{\pgfpoint{\pgf@xb}{\pgf@yb}}
    \pgfmathsetlength\pgf@xc{
      \pgf@xb - \ganttvalueof{group right peak tip position}
      * \ganttvalueof{group right peak width} * \ganttvalueof{x unit}
    }
    \pgfmathsetlength\pgf@yc{
      \pgf@yb - \ganttvalueof{group right peak height}
                * \ganttvalueof{y unit chart}
    }
    \pgfpathlineto{\pgfpoint{\pgf@xc}{\pgf@yc}}
    \pgfmathsetlength\pgf@xc{
      \pgf@xb - \ganttvalueof{group right peak width}
                * \ganttvalueof{x unit}
    }
    \pgfpathlineto{\pgfpoint{\pgf@xc}{\pgf@yb}}
    \pgfmathsetlength\pgf@xc{
      \pgf@xa + \ganttvalueof{group left peak width}
                * \ganttvalueof{x unit}
    }
    \pgfpathlineto{\pgfpoint{\pgf@xc}{\pgf@yb}}
    \pgfmathsetlength\pgf@xc{
      \pgf@xa + \ganttvalueof{group left peak tip position}
      * \ganttvalueof{group left peak width} * \ganttvalueof{x unit}
    }
    \pgfmathsetlength\pgf@yc{
      \pgf@yb - \ganttvalueof{group left peak height}
                * \ganttvalueof{y unit chart}
    }
    \pgfpathlineto{\pgfpoint{\pgf@xc}{\pgf@yc}}
    \pgfpathlineto{\pgfpoint{\pgf@xa}{\pgf@yb}}
    \pgfpathclose
  }
}

```

Code for node shape `ganttmilestone`. Anchors and background path derive from node shape `diamond`. The four additional anchors `on top`, `on bottom`, `on left` and `on right` are defined.

```
\pgfdeclareshape{ganttmilestone}{
  \inheritsavedanchors[from=diamond]
  \inheritanchor[from=diamond]{text}
  \inheritanchor[from=diamond]{center}
  \inheritanchor[from=diamond]{mid}
  \inheritanchor[from=diamond]{base}
  \inheritanchor[from=diamond]{north}
  \inheritanchor[from=diamond]{south}
  \inheritanchor[from=diamond]{west}
  \inheritanchor[from=diamond]{north west}
  \inheritanchor[from=diamond]{south west}
  \inheritanchor[from=diamond]{east}
  \inheritanchor[from=diamond]{north east}
  \inheritanchor[from=diamond]{south east}
  \inheritanchorborder[from=diamond]
  \inheritbackgroundpath[from=diamond]
  \anchor{on top}{
    \pgf@process{\outernortheast}
    \pgf@xa=2\pgf@x
    \pgf@x=-\pgf@x
    \advance\pgf@x by\ganttvalueof{on top fraction}\pgf@xa
    \pgf@ya=2\pgf@y
    \pgf@y=0pt
    \pgfmathparse{
      \ganttvalueof{on top fraction} < 0.5
      ? \ganttvalueof{on top fraction}
      : 1 - \ganttvalueof{on top fraction}
    }
    \advance\pgf@y by\pgfmathresult\pgf@ya
  }
  \anchor{on bottom}{
    \pgf@process{\outernortheast}
    \pgf@xa=2\pgf@x
    \pgf@x=-\pgf@x
    \advance\pgf@x by\ganttvalueof{on bottom fraction}\pgf@xa
    \pgf@ya=-2\pgf@y
    \pgf@y=0pt
    \pgfmathparse{
      \ganttvalueof{on bottom fraction} < 0.5
      ? \ganttvalueof{on bottom fraction}
      : 1 - \ganttvalueof{on bottom fraction}
    }
    \advance\pgf@y by\pgfmathresult\pgf@ya
  }
  \anchor{on right}{
    \pgf@process{\outernortheast}
    \pgf@ya=-2\pgf@y
    \advance\pgf@y by\ganttvalueof{on right fraction}\pgf@ya
    \pgf@xa=2\pgf@x
    \pgf@x=0pt
    \pgfmathparse{
      \ganttvalueof{on right fraction} < 0.5
      ? \ganttvalueof{on right fraction}
      : 1 - \ganttvalueof{on right fraction}
    }
    \advance\pgf@x by\pgfmathresult\pgf@xa
  }
  \anchor{on left}{
    \pgf@process{\outernortheast}
    \pgf@ya=-2\pgf@y
    \advance\pgf@y by\ganttvalueof{on left fraction}\pgf@ya
    \pgf@xa=-2\pgf@x
    \pgf@x=0pt
    \pgfmathparse{
      \ganttvalueof{on left fraction} < 0.5
      ? \ganttvalueof{on left fraction}
      : 1 - \ganttvalueof{on left fraction}
    }
    \advance\pgf@x by\pgfmathresult\pgf@xa
  }
}
```

3.11 Links

Keys for configuring links.

`link`
`link type`
`link label`
`link label font`
`link label node`

```
\@gtt@stylekeydef{link}{-latex, rounded corners=1pt}
\@gtt@keydef{link type}{auto}
\@gtt@keydef{link label}{}
\@gtt@keydef{link label font}{\scriptsize\itshape}
\@gtt@stylekeydef{link label node}{%
  anchor=west, font=\ganttvalueof{link label font}%
}
```

`\newganttlinktype`

`\newganttlinktype{`⟨*type*⟩`}{`⟨*code*⟩`}` stores ⟨*code*⟩ in an internal macro `\@gtt@linktype@`⟨*type*⟩, which is later called by `\gtt@drawlink`.

```
\newcommand\newganttlinktype[2]{%
  \expandafter\def\csname @gtt@linktype@#1\endcsname{#2}%
}
```

`\setganttlinklabel`

`\setganttlinklabel{`⟨*type*⟩`}{`⟨*label*⟩`}` stores a given ⟨*label*⟩ in an internal macro `\@gtt@linktype@`⟨*type*⟩`@label`, which is later used by `\gtt@drawlink`.

```
\newcommand\setganttlinklabel[2]{%
  \expandafter\def\csname @gtt@linktype@#1@label\endcsname{#2}%
}
```

`\newganttlinktypealias{`⟨*new link type*⟩`}{`⟨*existing link type*⟩`}` copies both the link code and label of an ⟨*existing link type*⟩ into the internal macros associated with a ⟨*new link type*⟩.

`\newganttlinktypealias`

```
\newcommand\newganttlinktypealias[2]{%
  \expandafter\def\csname @gtt@linktype@#1\endcsname{%
    \csname @gtt@linktype@#2\endcsname%
  }%
  \expandafter\def\csname @gtt@linktype@#1@label\endcsname{%
    \csname @gtt@linktype@#2@label\endcsname%
  }%
}

```

We will define three link subtypes for the type `auto`, which require the following keys:

`link mid`
`link bulge`
`link tolerance`

```
\@gtt@keydef{link mid}{.5}
\@gtt@keydef{link bulge}{.4}
\@gtt@keydef{link tolerance}{.6}
```

(1) `r` (short for “right”) draws a straight arrow. Note that `r` and `default` are alias types.

```
\newganttlinktype{r}{%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    (\xRight, \yLower)
    node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel};
}
\newganttlinktypealias{default}{r}

```

(2) `rdr` (“right-down-right”) is an unlabeled three-part arrow. The value of `link mid` sets the position of the middle segment.

```
\newganttlinktype{rdr}{%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    ($(\xLeft, \yUpper)!\ganttvalueof{link mid}!
      (\xRight, \yUpper)$) --
    ($(\xLeft, \yLower)!\ganttvalueof{link mid}!
      (\xRight, \yLower)$) --
    (\xRight, \yLower);%
}

```

(3) `rdldr` (“right-down-left-down-right”) is an unlabeled five-part arrow, which considers the values of `link bulge` and `link mid`.

```
\newganttlinktype{rdldr}{%
  \draw [/pgfgantt/link]
```

```
    (\xLeft, \yUpper) --
    (\xLeft + \ganttvalueof{link bulge} * \ganttvalueof{x unit},
      \yUpper) --
    ($(\xLeft + \ganttvalueof{link bulge} * \ganttvalueof{x unit},
      \yUpper)!%
      \ganttvalueof{link mid}!%
      (\xLeft + \ganttvalueof{link bulge} * \ganttvalueof{x unit},
      \yLower)$) --
    ($(\xRight - \ganttvalueof{link bulge} * \ganttvalueof{x unit},
      \yUpper)!%
      \ganttvalueof{link mid}!%
      (\xRight - \ganttvalueof{link bulge} * \ganttvalueof{x unit},
      \yLower)$) --
    (\xRight - \ganttvalueof{link bulge} * \ganttvalueof{x unit},
      \yLower) --
    (\xRight, \yLower);%
}

```

Now we may define linke type `auto`: 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. We check if the connected elements lie in the same row or not (i. e., their y-coordinates differ at most 1 pt). In the latter case, `\pgfmathparse` yields 0.

```
\newganttlinktype{auto}{%
  \pgfmathparse{abs(\yUpper - \yLower) <= 1}%
  \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 (i. e., link type `rdr`). Alternatively, the elements lie in adjacent time slots or even overlap, in which case we draw a five-part arrow (i. e., link type `rdldr`).

```
    \pgfmathparse{
      (\xRight - \xLeft)
      >= \ganttvalueof{link tolerance} * \ganttvalueof{x unit}
    }%
    \ifcase\pgfmathresult%
      \gtt@drawlink{rdldr}%
    \else%
      \gtt@drawlink{rdr}%
    \fi%
```

For elements that lie in the same row, we draw a simple arrow (i. e., link type `r`).

```
  \else%
    \gtt@drawlink{r}%
  \fi%
}
```

The `dr` type is explained in section 2.9.

```
\newganttlinktype{dr}{%
  \ganttsetstartanchor{south}%
  \ganttsetendanchor{west}%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    (\xLeft, \yLower)
    node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel} --
    (\xRight, \yLower);%
}
```

Here is the definition of the four straight link types and their labels.

```
\newganttlinktype{s-s}{%
  \ganttsetstartanchor{south west}%
  \ganttsetendanchor{north west}%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    (\xRight, \yLower)
    node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel};
}
\setganttlinklabel{s-s}{start-to-start}

\newganttlinktype{s-f}{%
  \ganttsetstartanchor{on bottom=0}%
  \ganttsetendanchor{on top=1}%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    (\xRight, \yLower)
    node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel};
}
\setganttlinklabel{s-f}{start-to-finish}

\newganttlinktype{f-s}{%
  \ganttsetstartanchor{south east}%
  \ganttsetendanchor{north west}%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    (\xRight, \yLower)
    node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel};
}
\setganttlinklabel{f-s}{finish-to-start}

\newganttlinktype{f-f}{%
  \ganttsetstartanchor{south east}%
  \ganttsetendanchor{north east}%
  \draw [/pgfgantt/link]
    (\xLeft, \yUpper) --
    (\xRight, \yLower)
    node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel};
}
\setganttlinklabel{f-f}{finish-to-finish}

```

`\gtt@drawlink`
`\@gtt@currlinktype`

`\gtt@drawlink{`⟨*link type*⟩`}` first checks if the ⟨*link type*⟩ is defined, falling back to type `default` if it is unknown. `\@gtt@currlinktype` stores the link type for future reference.

```
\newcommand\gtt@drawlink[1]{%
  \@ifundefined{@gtt@linktype@#1}{%
    \@gtt@PackageWarning{Link type ‘#1’ unknown, using ‘default’.}%
    \def\@gtt@currlinktype{default}%
  }{%
    \def\@gtt@currlinktype{#1}%
  }%
```

`\@gtt@currlabel`
`\ganttlinklabel`

If the `link label` key contains any value, it locally overrides the label set by `\setganttlinklabel`. `\ganttlinklabel` is defined accordingly.

```
  \edef\@gtt@currlabel{\ganttvalueof{link label}}%
  \ifx\@gtt@currlabel\@empty%
    \def\ganttlinklabel{%
      \csname @gtt@linktype@\@gtt@currlinktype @label\endcsname%
    }%
  \else%
    \edef\ganttlinklabel{%
      \ganttvalueof{link label}%
    }%
  \fi%
```

Finally, we call the internal macro that stores the code for the desired link type.

```
  \csname @gtt@linktype@\@gtt@currlinktype\endcsname%
}

```

`\@gtt@link@anchor`

We need the following keys for setting the start and end anchor of a link: Whenever a key `/pgfgantt/link anchor/`⟨*anchor*⟩ is undefined, `pgfgantt` stores ⟨*anchor*⟩ in `\@gtt@link@anchor`.

```
\ganttset{%
  link anchor/.unknown/.code={%
    \edef\@gtt@link@anchor{\pgfkeyscurrentname}%
  },%
}
```

`\@gtt@linkanchordef`

`\@gtt@linkanchordef{`⟨*anchor*⟩`}` deals with the anchors `on top` etc.: It creates a code key `/pgfgantt/link anchor/`⟨*anchor*⟩, which stores its own name in `\@gtt@link@anchor` and sets the appropriate `...fraction` key.

```
\def\@gtt@linkanchordef#1{%
  \ganttset{%
    link anchor/#1/.code={%
```

```
      \edef\@gtt@link@anchor{#1}%
      \ganttset{#1 fraction=##1}%
    },%
    link anchor/#1/.default=.5%
  }%
}
\@gtt@linkanchordef{on top}
\@gtt@linkanchordef{on bottom}
\@gtt@linkanchordef{on left}
\@gtt@linkanchordef{on right}

```

`\@gtt@setstartanchor{`⟨*anchor*⟩`}` recalls the coordinates of the anchor stored in `\@gtt@link@anchor` from chart element `\@gtt@link@startelement`. It stores these coordinates in the auxiliary macros `\xLeft` and `\yUpper`.

`\@gtt@setstartanchor`
`\xLeft`
`\yUpper`

```
\newcommand\@gtt@setstartanchor[1]{%
  \pgfqkeys{/pgfgantt/link anchor}{#1}%
  \pgfpointanchor{\@gtt@link@startelement}{\@gtt@link@anchor}%
  \edef\xLeft{\the\pgf@x}%
  \edef\yUpper{\the\pgf@y}%
}

```

`\@gtt@setendanchor{`⟨*anchor*⟩`}` is similar to the command above. However, it stores the anchor coordinates in the auxiliary macros `\xRight` and `\yLower`.

`\@gtt@setendanchor`
`\xRight`
`\yLower`

```
\newcommand\@gtt@setendanchor[1]{%
  \pgfqkeys{/pgfgantt/link anchor}{#1}%
  \pgfpointanchor{\@gtt@link@endelement}{\@gtt@link@anchor}%
  \edef\xRight{\the\pgf@x}%
  \edef\yLower{\the\pgf@y}%
}

```

`\ganttlink[`⟨*options*⟩`]{`⟨*E1*⟩`}{`⟨*E2*⟩`}` executes the ⟨*options*⟩ and stores the names of the connected elements ⟨*E1*⟩ and ⟨*E2*⟩ in `\@gtt@link@startelement` and `\@gtt@link@endelement`.

`\ganttlink`
`\@gtt@link@startelement`
`\@gtt@link@endelement`

```
\newcommand\ganttlink[3][]{%
  \begingroup%
  \ganttset{#1}%
  \def\@gtt@link@startelement{#2}%
  \def\@gtt@link@endelement{#3}%
```

`\ganttsetstartanchor` and `\ganttsetendanchor` are only valid in the second argument of `\newganttlinktype`. Since you may wish to omit one of those commands, we set default anchors for the link.

`\ganttsetstartanchor`
`\ganttsetendanchor`

```
  \let\ganttsetstartanchor\@gtt@setstartanchor%
  \let\ganttsetendanchor\@gtt@setendanchor%
  \ganttsetstartanchor{east}%
  \ganttsetendanchor{west}%
```

We call `\gtt@drawlink` with the value of `link type`.

```
  \gtt@drawlink{\ganttvalueof{link type}}%
  \endgroup%
}

```

4 Index

Bold numbers refer to the page where the corresponding entry is described; italic numbers refer to the code line of the definition; upright numbers refer to the code lines where the entry is used.

Symbols

`\%` 811
`\@@@gantttitlecalendar` 570, 579, *582*
`\@@gantttitlecalendar` 568, *576*
`\@gantttitlecalendar` 560, 561, *564*
`\@gtt@PackageError` *19*, 195, 237
`\@gtt@PackageWarning` *19*, 318, 888, 1537
`\@gtt@currlabel` *1542*
`\@gtt@currlinktype` *1535*, 1545, 1552
`\@gtt@getdecade` *595*, 623, 637
`\@gtt@getfourthdigit` *595*, 602, 603
`\@gtt@ifstar` *8*, 61, 109, 559, 976
`\@gtt@keydef` *10*, 262–264, 266–268, 270, 271, 431, 433, 464, 478–481, 487, 814, 1010, 1014, 1018, 1019, 1023–1026, 1081–1086, 1133–1136, 1405–1407, 1428–1430
`\@gtt@link@anchor` *1555*, 1563, 1576, 1583
`\@gtt@link@endelement` 1583, *1588*
`\@gtt@link@startelement` 1576, *1588*
`\@gtt@linkanchordef` *1560*
`\@gtt@setendanchor` *1581*, 1594
`\@gtt@setstartanchor` *1574*, 1593
`\@gtt@stylekeydef` *16*, 30, 265, 269, 272, 432, 434, 463, 465, 1008, 1009, 1011, 1015, 1020, 1404, 1408
`\@newganttchartelement@definekeys` 990, 1004, *1007*
`\@newganttchartelement@one` 976, *979*
`\@newganttchartelement@two` 976, *979*
`\\` **12**, *328*

B

`bar` (option) **26**, *1036*
`bar height` (option) **30**, *1036*
`bar incomplete` (option) **34**, *1036*
`bar inline label anchor` (option) **29**, *1036*
`bar inline label node` (option) **29**, *1036*
`bar label font` (option) **27**, *1036*
`bar label node` (option) **27**, *1036*
`bar label text` (option) **27**, *1036*
`bar left shift` (option) **29**, *1036*
`bar progress label anchor` (option) **35**, *1036*
`bar progress label font` (option) **35**, *1036*
`bar progress label node` (option) **35**, *1036*
`bar right shift` (option) **29**, *1036*
`bar top shift` (option) **29**, *1036*

C

`calendar week text` (option) **18**, *487*
`canvas` (option) **7**, *265*
`chart element start border` (option) **26**, *802*
`\currentweek` **18**, 487, *736*

E

`\endofdecade` 602, 604
environments:
  `ganttchart` **5**, *292*
`expand chart` (option) **8**, *265*

G

`\ganttalignnewline` **13**, *328*
`\ganttbar` **24**, *1036*
`ganttchart` (environment) **5**, *292*
`\ganttgroup` **24**, *1059*
`\ganttlink` **40**, 987, 1001, *1588*
`\ganttlinkedbar` **25**, *1036*
`\ganttlinkedgroup` **25**, *1059*
`\ganttlinkedmilestone` **25**, *1108*
`\ganttlinklabel` **44**, 1435, 1491, 1501, 1511, 1521, 1531, *1542*
`\ganttmilestone` **24**, *1108*
`\ganttnewline` **12**, 333, *408*, 616, 648, 706, 737, 781, 792
`\ganttset` **4**, *9*, 32, 80, 234, 276, 293, 325, 442, 468, 483, 491, 513, 552, 566, 614, 802, 816, 823, 838, 985, 999, 1027, 1087, 1089, 1095, 1101, 1555, 1561, 1564, 1590
`\ganttsetendanchor` **43**, 1487, 1497, 1507, 1517, 1527, *1593*

\ganttsetstartanchor **43**, 1486, 1496, 1506, 1516, 1526, *1593*
\gantttitle **14**, *511*, 553, 622, 636, 654, 661, 669, 681, 689, 697, 711, 724, 749, 768, 784, 795
\gantttitlecalendar **15**, *557*
\gantttitlecalendar* **16**, *557*
\gantttitlelist **14**, *550*
\ganttvalueof *13*, **44**, 67, 69, 73, 115, 116, 273, 305, 307, 315, 321, 342, 346, 348, 357, 363, 386, 389, 390, 394, 401, 417, 419, 423, 435, 446, 449, 451, 454, 455, 466, 515, 516, 519, 520, 529, 530, 533, 534, 750, 769, 851, 852, 855, 856, 865, 867, 870, 878, 886, 922, 951, 959, 1012, 1016, 1021, 1042, 1046, 1051, 1065, 1069, 1074, 1116, 1120, 1125, 1162, 1172, 1182, 1192, 1232, 1233, 1236, 1237, 1263, 1264, 1267, 1268, 1300, 1301, 1304, 1305, 1309, 1310, 1314, 1315, 1319, 1320, 1323, 1324, 1352, 1356–1358, 1366, 1370–1372, 1379, 1383–1385, 1392, 1396–1398, 1409, 1442, 1444, 1452, 1454, 1456, 1457, 1459, 1461, 1462, 1464, 1474, 1542, 1549, 1597
\ganttvrule **23**, *440*
group (option) **26**, *1059*
group height (option) **30**, *1059*
group incomplete (option) **34**, *1059*
group inline label anchor (option) **29**, *1059*
group inline label node (option) **29**, *1059*
group label font (option) **27**, *1059*
group label node (option) **27**, *1059*
group label text (option) **27**, *1059*
group left peak height (option) **31**, *1081*
group left peak tip position (option) **31**, *1081*
group left peak width (option) **31**, *1081*
group left shift (option) **30**, *1059*
group peaks height (option) **31**, *1081*
group peaks tip position (option) **31**, *1081*
group peaks width (option) **31**, *1081*
group progress label anchor (option) **35**, *1059*
group progress label font (option) **35**, *1059*
group progress label node (option) **35**, *1059*
group right peak height (option) **31**, *1081*
group right peak tip position (option) **31**, *1081*
group right peak width (option) **31**, *1081*
group right shift (option) **30**, *1059*
group top shift (option) **30**, *1059*
\gtt@calendar@enddate *582*, 620, 652, 667, 687, 709, 741, 783, 794
\gtt@calendar@eolfalse 587
\gtt@calendar@eoltrue 645, 703, 734, 777, 789, 798
\gtt@calendar@slots *591*, 618, 625, 626, 630, 635, 639, 650, 655, 657, 660, 669, 670, 673, 678, 681, 689, 690, 692, 695–697, 708, 715, 717, 719, 722, 723, 728, 739, 745, 752, 754, 757, 760, 761, 764, 771
\gtt@calendar@startdate *582*, 620, 652, 667, 687, 709, 741, 783, 794
\gtt@calendar@startofweek *591*, 743–746, 762–765
\gtt@calendar@weeknumber *591*, 740, 748, 755, 767
\gtt@ce@startatleftborderfalse 819
\gtt@ce@startatleftbordertrue 818
\gtt@chartelement *836*, 981, 986, 995, 1000
\gtt@chartextrasize *289*, 403
\gtt@chartid *281*, 317, 322, 403, 404
\gtt@chartwidth *26*, 73, 101, 118, 302, 303, 323, 357, 367, 368, 370, 372, 401, 423
\gtt@currentelement *828*, 968, 969, 987, 1001
\gtt@currentline *26*, 53, 75, 311, 347, 380, 382, 418, 425, 450, 529, 533, 866
\gtt@currgrid *26*, 53, 68, 74, 75, 101, 115–118, 366, 370, 375, 377, 379, 382
\gtt@draw@clipfalse 910, 913
\gtt@draw@cliptrue 907
\gtt@draw@completefalse 909
\gtt@draw@completetrue 905

\gtt@draw@incompletefalse 912
\gtt@draw@incompletetrue 906
\gtt@drawlink 1477, 1479, 1482, *1535*, 1597
\gtt@elementid *281*, 310, 879, 970
\gtt@endjulian *281*, 296, 297, 302, 572
\gtt@expanded@xunit *314*
\gtt@hgrid@analyze 54, *60*
\gtt@hgrid@do *51*, 381
\gtt@hgrid@draw 61, *64*
\gtt@hgridfalse 37
\gtt@hgridstyle *30*, 381
\gtt@hgridtrue 39
\gtt@intitlefalse 971
\gtt@intitletrue 331
\gtt@juliantotimeslot *127*, 302, 308, 444, 840, 842
\gtt@lastelement *828*, 968, 987, 1001
\gtt@lasttitleline *26*, 67, 68, 312, 342, 346, 347, 363, 375, 417, 418, 426, 446, 449, 450, 865, 866
\gtt@lasttitleslot *281*, 313, 427, 515, 519, 546
\gtt@left@slot *828*, 839, 840, 845, 850, 894, 898, 899, 917, 921
\gtt@name *836*, 934, 945, 951, 959, 969
\gtt@progress *802*, 881, 882, 885, 891, 893, 895, 897, 908, 911, 918, 949, 953
\gtt@progresslabeltext *802*, 950, 953
\gtt@right@slot *828*, 841, 842, 845, 855, 892, 899, 917
\gtt@smugglecount *122*, 150, 156
\gtt@startjulian 131, *281*, 294, 295, 298, 571
\gtt@startmonth 140, 299, 301
\gtt@startyear 137, 146, 299, 300
\gtt@tikzpicturefalse 329
\gtt@tikzpicturetrue 330
\gtt@timeslotunit@dayfalse 499, 504
\gtt@timeslotunit@daytrue 494
\gtt@timeslotunit@monthfalse 495, 505
\gtt@timeslotunit@monthtrue 500
\gtt@timeslotunit@yearfalse 496, 501
\gtt@timeslotunit@yeartrue 506
\gtt@titlecalendarstarfalse 561
\gtt@titlecalendarstartrue 560
\gtt@titlelabeltext *463*, 545
\gtt@titlelistoptions *463*, 553
\gtt@today@slot *281*, 307, 308, 389, 892, 894, 898
\gtt@tsf@basecentury 213, 221, 229, *245*
\gtt@tsf@getdmy *178*, 211, 219, 227
\gtt@tsf@startjulian 162, *253*
\gtt@tsstojulian *234*, 294, 296, 307, 443, 577, 578, 839, 841
\gtt@vgrid@analyze *79*
\gtt@vgrid@do *79*, 369
\gtt@vgrid@draw *79*
\gtt@vgridfalse 85
\gtt@vgridstyle *79*, 369
\gtt@vgridtrue 87
\gtt@vrule@slot *438*, 443, 444, 454

H

hgrid (option) **9**, *30*
hgrid style (option) **9**, *30*

I

\ifgtt@calendar@eol *591*, 616, 648, 706, 737, 781, 792
\ifgtt@ce@startatleftborder *802*, 844
\ifgtt@draw@clip *828*, 915, 927, 938
\ifgtt@draw@complete *828*, 931
\ifgtt@draw@incomplete *828*, 942
\ifgtt@hgrid *30*, 374
\ifgtt@includetitle 338, 376, *463*
\ifgtt@inline *802*, 958
\ifgtt@intitle *281*, 426
\ifgtt@newlineshortcut *275*, 332
\ifgtt@tikzpicture *281*, 405
\ifgtt@timeslotunit@day 130, *487*, 686, 715, 728
\ifgtt@timeslotunit@month 134, *487*, 666, 714, 727
\ifgtt@timeslotunit@year 143, *487*, 619, 651
\ifgtt@titlecalendarstar *557*, 567
\ifgtt@vgrid *79*, 365
include title in canvas (option) **22**, *463*
inline (option) **28**, *802*

L

link (option) **40**, *1404*
link bulge (option) **41**, *1428*
link label (option) **47**, *1404*
link label font (option) **47**, *1404*
link label node (option) **47**, *1404*
link mid (option) **41**, *1428*
link tolerance (option) **41**, *1428*

- `link type` (option) **41**, *1404*
- `\local@day` 181, 186, 191, 215, 220, 221, 223, 231
- `\local@decompose` 180, 183, 185, 188, 190, 193, 197, 198, 201, 202, 205, 206
- `\local@drawarg` *408*
- `\local@firstarg` 179, 183, 188, 193
- `\local@month` 181, 184, 186, 189, 191, 194, 215, 223, 231
- `\local@none` *880*
- `\local@timeslotmodifier` *836*, 898, 899, 917, 921
- `\local@year` 181, 186, 191, 197, 198, 201, 202, 205, 206, 212, 213, 215, 223, 228, 229, 231

M

- `milestone` (option) **26**, *1108*
- `milestone height` (option) **30**, *1108*
- `milestone incomplete` (option) **34**, *1108*
- `milestone inline label anchor` (option) **29**, *1108*
- `milestone inline label node` (option) **29**, *1108*
- `milestone label font` (option) **27**, *1108*
- `milestone label node` (option) **28**, *1108*
- `milestone label text` (option) **27**, *1108*
- `milestone left shift` (option) **30**, *1108*
- `milestone progress label anchor` (option) **35**, *1108*
- `milestone progress label font` (option) **35**, *1108*
- `milestone progress label node` (option) **35**, *1108*
- `milestone right shift` (option) **30**, *1108*
- `milestone top shift` (option) **30**, *1108*

N

- `name` (option) **39**, *802*
- `\newganttchartelement` **38**, *975*, 1036, 1059, 1108
- `\newganttchartelement*` **38**, *975*
- `\newganttlinktype` **43**, *1411*, 1431, 1439, 1449, 1469, 1485, 1495, 1505, 1515, 1525
- `\newganttlinktypealias` **44**, *1419*, 1437
- `\newgantttimeslotformat` **6**, *154*, 160, 166, 170, 174, 210, 218, 226
- `newline shortcut` (option) **12**, *275*

O

- `on bottom fraction` (option) **36**, *1133*
- `on left fraction` (option) **36**, *1133*
- `on right fraction` (option) **36**, *1133*
- `on top fraction` (option) **36**, *1133*
- options:
  - `bar` **26**, *1036*
  - `bar height` **30**, *1036*
  - `bar incomplete` **34**, *1036*
  - `bar inline label anchor` **29**, *1036*
  - `bar inline label node` **29**, *1036*
  - `bar label font` **27**, *1036*
  - `bar label node` **27**, *1036*
  - `bar label text` **27**, *1036*
  - `bar left shift` **29**, *1036*
  - `bar progress label anchor` **35**, *1036*
  - `bar progress label font` **35**, *1036*
  - `bar progress label node` **35**, *1036*
  - `bar right shift` **29**, *1036*
  - `bar top shift` **29**, *1036*
  - `calendar week text` **18**, *487*
  - `canvas` **7**, *265*
  - `chart element start border` **26**, *802*
  - `expand chart` **8**, *265*
  - `group` **26**, *1059*
  - `group height` **30**, *1059*
  - `group incomplete` **34**, *1059*
  - `group inline label anchor` **29**, *1059*
  - `group inline label node` **29**, *1059*
  - `group label font` **27**, *1059*
  - `group label node` **27**, *1059*
  - `group label text` **27**, *1059*
  - `group left peak height` **31**, *1081*
  - `group left peak tip position` **31**, *1081*
  - `group left peak width` **31**, *1081*
  - `group left shift` **30**, *1059*
  - `group peaks height` **31**, *1081*
  - `group peaks tip position` **31**, *1081*
  - `group peaks width` **31**, *1081*
  - `group progress label anchor` **35**, *1059*
  - `group progress label font` **35**, *1059*
  - `group progress label node` **35**, *1059*
  - `group right peak height` **31**, *1081*
  - `group right peak tip position` **31**, *1081*
  - `group right peak width` **31**, *1081*
  - `group right shift` **30**, *1059*

group top shift **30**, *1059*
hgrid **9**, *30*
hgrid style **9**, *30*
include title in canvas **22**, *463*
inline **28**, *802*
link **40**, *1404*
link bulge **41**, *1428*
link label **47**, *1404*
link label font **47**, *1404*
link label node **47**, *1404*
link mid **41**, *1428*
link tolerance **41**, *1428*
link type **41**, *1404*
milestone **26**, *1108*
milestone height **30**, *1108*
milestone incomplete **34**, *1108*
milestone inline label anchor **29**, *1108*
milestone inline label node **29**, *1108*
milestone label font **27**, *1108*
milestone label node **28**, *1108*
milestone label text **27**, *1108*
milestone left shift **30**, *1108*
milestone progress label anchor **35**, *1108*
milestone progress label font **35**, *1108*
milestone progress label node **35**, *1108*
milestone right shift **30**, *1108*
milestone top shift **30**, *1108*
name **39**, *802*
newline shortcut **12**, *275*
on bottom fraction **36**, *1133*
on left fraction **36**, *1133*
on right fraction **36**, *1133*
on top fraction **36**, *1133*
progress **32**, *802*
progress label text **35**, *802*
time slot format **5**, *234*
time slot format/base century **6**, *245*
time slot format/start date **6**, *253*
time slot unit **19**, *487*
title **20**, *463*
title height **21**, *463*
title label font **20**, *463*
title label node **20**, *463*
title label text **20**, *463*
title left shift **21**, *463*
title list options **15**, *463*
title right shift **21**, *463*
title top shift **21**, *463*
today **11**, *265*
today label **11**, *265*
today label font **11**, *265*
today label node **11**, *265*
today offset **11**, *265*
today rule **11**, *265*
vgrid **9**, *79*
vrule **23**, *431*
vrule label font **23**, *431*
vrule label node **23**, *431*
vrule offset **23**, *431*
x unit **8**, *262*
y unit chart **8**, *262*
y unit title **8**, *262*

P

progress (option) **32**, *802*
progress label text (option) **35**, *802*

Q

\querydecade 603, 604

S

\setganttlinklabel **44**, *1415*, 1503, 1513, 1523, 1533
\startday **18**, *736*
\startmonth **18**, *736*
\startyear **18**, *736*

T

time slot format (option) **5**, *234*
time slot format/base century (option) **6**, *245*
time slot format/start date (option) **6**, *253*
time slot unit (option) **19**, *487*
title (option) **20**, *463*
title height (option) **21**, *463*
title label font (option) **20**, *463*
title label node (option) **20**, *463*
title label text (option) **20**, *463*
title left shift (option) **21**, *463*
title list options (option) **15**, *463*
title right shift (option) **21**, *463*
title top shift (option) **21**, *463*

`today` (option) **11**, *265*
`today label` (option) **11**, *265*
`today label font` (option) **11**, *265*
`today label node` (option) **11**, *265*
`today offset` (option) **11**, *265*
`today rule` (option) **11**, *265*

V

`vgrid` (option) **9**, *79*
`vrule` (option) **23**, *431*
`vrule label font` (option) **23**, *431*
`vrule label node` (option) **23**, *431*
`vrule offset` (option) **23**, *431*

X

`x unit` (option) **8**, *262*
`\x@clip` 920, 929, 939
`\x@clip@size` *905*
`\x@left` *511*, 849, 859, 862, 928
`\x@mid` *385*, 453, 458, 522, 542, 858, 934, 945
`\x@right` *511*, 854, 859, 862, 940
`\x@size` *337*, 525, 543, 861, 932, 943
`\xLeft` **43**, 1433, 1441, 1442, 1444, 1451, 1452, 1454, 1457, 1473, 1489, 1490, 1499, 1509, 1519, 1529, *1574*
`\xRight` **43**, 1434, 1443, 1445, 1446, 1459, 1462, 1464, 1466, 1473, 1492, 1500, 1510, 1520, 1530, *1581*

Y

`y unit chart` (option) **8**, *262*
`y unit title` (option) **8**, *262*
`\y@lower` 116, *337*, 393, 448, 458, 532, 537, 540, 869, 873, 876, 929, 940
`\y@mid` *337*, 536, 542, 872, 934, 945, 963
`\y@size` *337*, 539, 544, 875, 933, 944
`\y@upper` 66, 72, 73, 115, *337*, 362, 393, 416, 422, 423, 445, 458, 528, 537, 540, 864, 870, 873, 876, 928, 939
`\yLower` **43**, 1434, 1444–1446, 1458, 1463, 1465, 1466, 1470, 1490, 1492, 1500, 1510, 1520, 1530, *1581*
`\yUpper` **43**, 1433, 1441–1443, 1451, 1453, 1455, 1460, 1470, 1489, 1499, 1509, 1519, 1529, *1574*

5 Change History

v1.0

General: Initial release

v1.1

General: `bar label text` configures the text of a bar label.

`group label text` configures the text of a group label.

`link tolerance` decides whether a five- or a three-part link is drawn.

`milestone label text` configures the text of a milestone label.

The `time slot modifier` key has been added. If set to zero, all x-coordinates are interpreted as given, without regarding them as time slots.

The `vgrid lines list` key determines the number of vertical grid lines drawn.

The introduction clarifies what I mean by "a current PGF installation".

v2.0

General: Added style lists for the horizontal and vertical grid.

Completely rewrote the calculation of coordinates.

Removed the `hgrid shift` and `last line height` keys.

Removed the `vgrid lines list` key, as its behaviour can be simulated by an appropriate ⟨*style list*⟩ for `vgrid`.

Removed the `vgrid style` key.

The `x unit`, `y unit title` and `y unit chart` keys specify the width of time slots and the height of title or chart lines, respectively. Thus, one can draw titles whose height differs from the rest of the chart. Furthermore, the x- and y-dimensions of the chart are independent of the dimensions of the surrounding `tikzpicture`.

The optional argument of `\ganttnewline` now also accepts a style.

The syntax of `\ganttlink` was completely changed. The command now takes one optional and *two* mandatory arguments. The latter specify the name of the chart elements to be linked. Consequently, the keys `b-b`, `b-m`, `m-b` and `m-m` were removed. The keys `s-s`, `s-f`, `f-s` and `f-f` are now values for the `link type` key.

v2.1

General: Added three keys (`bar/group/milestone label inline anchor`) for placing inline labels.

The `ganttchart` environment may be used outside a `tikzpicture`.

The `inline` key moves labels close to their respective chart elements.

v3.0

General: `\@gtt@get` has been renamed to `\ganttvalueof` to provide a convenient access for link type authors.

`\@gtt@keydef` and `\@gtt@stylekeydef` have been rewritten to support `pgfkey`'s abilities to store key values.

`\setganttlinklabel` specifies the label for all links of a certain type. The `link label` key locally overrides any label set by this command.

All style keys (`canvas`, `bar` etc.) only support the common Ti*k*Z style key syntax.

Completely rewrote the code for links (again). Definition of new link types is now possible (via `\newganttlinktype` and `\newganttlinktypealias`).

New auxiliary macros for `\newganttlinkstyle`: `\xLeft`, `\xRight`, `\yUpper`, `\yLower`, `\ganttsetstartanchor`, `\ganttsetendanchor` and `\ganttlinklabel`.

The `bar/group/milestone label shape anchor` keys allow for a fine-tuned placement of chart element labels.

- The `chart element` shape supports four additional anchors (`on left`, `on top`, `on right` and `on bottom`).

v4.0

- General: `\gantttitlecalendar` prints a title calendar.
  - `\newganttchartelement` defines completely new chart elements.
  - `\newgantttimeslotformat` allows the user to define custom time slot formats.
  - Chart elements are now nodes, so the corresponding styles must specify a node shape.
  - The `ganttchart` environment now requires two mandatory arguments.
  - The canvas is now a node with shape `rectangle` by default.
  - The key `incomplete` was removed.
  - The key `link label anchor` was renamed to `link label node`.
  - The key `newline shortcut` determines whether the shortcut for line breaks is defined in the chart. In this case, `\ganttalignnewline` allows line breaks in the node text.
  - The key `progress label anchor` was replaced by `bar`/`group`/`milestone progress label node`.
  - The key `progress label font` was replaced by the keys `bar`/`group`/`milestone progress label font`.
  - The key `time slot format/base century` provides the century for autocompletion of two-digit years.
  - The key `time slot format/start date` specifies the internal date representation of digit 1 in the `simple` time slot format.
  - The key `time slot format` changes the format of time slot specifiers.
  - The key `time slot modifier` was renamed to `chart element start border`.
  - The key `title label anchor` was renamed to `title label node`.
  - The key `today` accepts a time slot specifier.
  - The keys `bar`/`group`/`milestone label anchor` were renamed to `bar`/`group`/`milestone label node`.
  - The keys `bar`/`group`/`milestone label inline anchor` were renamed to `bar`/`group`/`milestone inline label node`.
  - The keys `bar`/`group`/`milestone label shape anchor` were renamed to `bar`/`group`/`milestone inline label anchor`.
  - The keys `bar`/`group`/`milestone progress label anchor` were added.
  - The keys `calendar week text` and `compress calendar` were added.
  - The keys `group right`/`left peak` and `group peaks` were replaced by `group right`/`left peak tip position`, `group peaks tip position`, `group right`/`left peak width`, `group peaks width`, `group right`/`left peak height` and `group peaks height`.
  - The keys `today offset`, `today label font` and `today label node` were added.

v5.0

- General: `\gantttitlecalendar` now recognizes the `decade` key.
  - Key `compress calendar` has been replaced by `time slot unit` to allow an additional level of compression (year).
  - Made pgfgantt robust to amsgen's redefinition of `\@ifstar`.
  - The command `\ganttvrule` allows to draw general vertical rules (similar to the today rule). The keys `vrule`, `vrule offset`, `vrule label font` and `vrule label text` configure those rules.
  - The key `expand chart` was added, which specifies that a chart should expand horizontally to a given dimension.

The key `title label text` was added to allow fine-tuning of title label formatting.