Drawing Gantt Charts in \LaTeX\ with TikZ
The \texttt{pgfgantt} Package

Wolfgang Skala*

v4.0
2013/06/01

The \texttt{pgfgantt} package provides the \texttt{ganttchart} environment, which
draws a Gantt chart within a TikZ picture. The user may add various
elements to the chart, for example, titles, bars, groups, milestones and dif-
f erent links between these elements. The appearance of the chart elements
is highly customizable, and even new chart elements may be defined.

Contents

1 Introduction 2

2 User Guide 3
2.1 Overview .................................................. 3
2.2 Specifying Keys ........................................... 3
2.3 The Canvas ................................................ 4
2.4 Line Breaks between Chart Elements ..................... 10
2.5 Titles ....................................................... 12
2.6 Predefined Chart Elements ............................... 20
  2.6.1 Options: Chart Element Appearance ................. 23
  2.6.2 Options: Label Formatting ........................... 24
  2.6.3 Options: Chart Element Positioning ................. 26
  2.6.4 Options: Progress ................................... 29
  2.6.5 New Node Shapes .................................... 33
2.7 Defining Custom Chart Elements ......................... 34
2.8 Links ...................................................... 36
2.9 Style Examples ........................................... 45

*Division of Structural Biology, Department of Molecular Biology, University of Salzburg, Aus-
tria; Wolfgang.Skala@sbg.ac.at
1 Introduction

The pgfgantt package allows you to draw Gantt charts in \LaTeX. Thus, you can describe simple project schedules without having to include images produced by external programs. Similar to Martin Kumm’s gantt package\footnote{http://www.martin-kumm.de/tex_gantt_package.php} (which inspired pgfgantt’s fundamental aspects), pgfgantt bases upon PGF and its TikZ frontend\footnote{http://ctan.org/tex-archive/graphics/pgf/}. Besides, it provides a comprehensive (and portable) alternative to pst-gantt\footnote{http://ctan.org/tex-archive/graphics/pstricks/contrib/pst-gantt/}.

pgfgantt requires a current PGF installation. Note that the version number must at least be 2.10, dated October 25th, 2010. Furthermore, pgfgantt v4.0 and above is not fully downwards compatible.

Acknowledgements I would like to thank Petr Pošík (Czech Technical University in Prague), Raphaël Clifford (University of Bristol), Holger Karl (Universität Paderborn), Jakob Døllner Mønster (Technical University of Denmark), Sascha Yousefi (Universität Freiburg) and Callum Webb for their ideas concerning new features.
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 pgf\texttt{gantt}, 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
\gantt milestone{Milestone}{7} \ganttnewline
\ganttbar{Final Task}{8}{12} \\
\ganttlink{elem2}{elem3} \\
\ganttlink{elem3}{elem4} \\
\end{ganttchart}

\begin{center}
\begin{tabular}{|c|c|c|c|c|c|c|c|c|c|}
\hline
\multicolumn{12}{|c|}{2011} \\
\hline
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\
\hline
\end{tabular}
\end{center}

Group 1

\begin{center}
\begin{tabular}{|c|}
\hline
Task 1 \\
\hline
Task 2 \\
\hline
Milestone \\
\hline
Final Task \\
\hline
\end{tabular}
\end{center}

2.2 Specifying Keys

Keys (sometimes called options) modify the output from pgf\texttt{gantt}’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\{\texttt{key=value list}\} macro, which sets its keys within the current \TeX group:
\ganttset\{\texttt{bar height=.6}\}

Since \texttt{pgfgantt} uses the \texttt{pgfkeys} package for key management, all its keys reside in the /\texttt{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 \textit{chart} consists of several \textit{lines}, which may contain one or more \textit{title elements} (at the top) or \textit{chart elements} (such as bars, groups and milestones). From left to right, the chart is divided into an integer number of \textit{time slots} that represent the basic \textit{x}-unit.

The \texttt{ganttchart} environment draws a single Gantt chart:
\begin{ganttchart}
....
\end{ganttchart}

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

\texttt{/\texttt{pgfgantt/time slot format }=\texttt{(format)}
\texttt{simple}}

Sets the \texttt{(format)} of time slot specifiers. A \textit{time slot specifier} (abbreviated “\texttt{tss}”) denotes a certain time slot along the horizontal axis. \texttt{pgfgantt} defines a range of formats:

- \texttt{simple} – positive integers (the single format used by \texttt{pgfgantt} prior to v4.0). See also the \texttt{time slot format/start date} key below.
  \textit{Examples:} 1, 3, 24

- \texttt{isodate} – dates in ISO-standard format (\texttt{yyyy-mm-dd}). In this format and any other, you may omit the leading zero if month or day are less than 10.
  \textit{Examples:} 2013-03-14, 2013-5-1

- \texttt{isodate-yearmonth} – ISO-standard dates without days (\texttt{yyyy-mm}). Such dates are automatically converted to the first day of the respective month.
  \textit{Examples:} 2013-03, 2013-5

- \texttt{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 \texttt{time slot format/base century} (see below).

\textit{Examples:} 14-03-2013, 14/03/13, 14.3.2013

- \texttt{middle-endian} – middle-endian, i.e. month–day–year (the common US date format). For valid separators and automatic year completion, see \texttt{little-endian}.
\textit{Examples:} 03-14-2013, 03/14/13, 3.14.2013

- \texttt{big-endian} – Gregorian big-endian, i.e. year–month–day (the ISO-standard order). For valid separators and automatic year completion, see \texttt{little-endian}.
\textit{Examples:} 2013-03-14, 13/03/14, 2013.3.14

Two subkeys of \texttt{time slot format} let you configure \texttt{pgfgantt}'s behavior regarding automatic completion of abbreviated dates:

\begin{verbatim}
\texttt{pgfgantt/time slot format/base century =\{year\}} \hspace{1cm} 2000
\textit{Sets the century for auto-completion of two-digit years (used by the time slot formats \texttt{little-endian}, \texttt{middle-endian} and \texttt{big-endian}). Consequently, default settings convert a year like 13 to 2013.}

\texttt{pgfgantt/time slot format/start date =\{ISO-standard date\} \hspace{1cm} 2000-01-01}
\textit{Numbers denoting time slots in the \texttt{simple} format are internally converted to a date, where 1 is converted to \texttt{ISO-standard date}, 2 to \texttt{ISO-standard date} + 1 etc.}
\end{verbatim}

Advanced users may add their own time slot formats:

\begin{verbatim}
\newgantttimeslotformat{(name)}{(converter code)}
\end{verbatim}

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

For example, we would like to create a format called \texttt{stardate}, where dates are given as “\texttt{\{year\}.\{day of year\}”}. Thus, we will enter 24th February 2259 as “2259.55”. To this end, we write the following code:

\begin{verbatim}
1 \newgantttimeslotformat{stardate}%
2 \def\decomposestardate##1.##2\relax{%
3 \def\stardateyear{##1}\def\stardateday{##2}%
4 }%
5 \decomposestardate#1\relax%
6 \pgfcalendardatestojulian{\stardateyear-01-01}{#2}%
7 \advance\#2 by-1\relax%
8 \advance\#2 by\stardateday\relax%
9 }
\end{verbatim}
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\kZ 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\kZ keys suitable for the `(options)` of a Ti\kZ node (such as `shape=rectangle, fill` or `draw`; see chapter 16 of the Ti\kZ 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
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}

$hgrid$ 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 \texttt{hgrid} and \texttt{hgrid=true} activate the horizontal grid, which is then drawn in the default style \texttt{dotted}. Finally, \texttt{hgrid=(style list)} draws the horizontal grid in the given \texttt{(style list)} (see below).

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

The \texttt{vgrid} key governs the vertical grid; otherwise, use it exactly like \texttt{hgrid}.

Style lists allow you to draw the grid lines in different styles. Each style list consists of several \texttt{style list items} separated by a comma. A style list item has the general syntax \texttt{*{⟨n⟩}{⟨style⟩}} and orders the package to repeat the \texttt{⟨style⟩} \texttt{⟨n⟩}-times. (This syntax is reminiscent of column specifications in a \texttt{tabular} environment.) Thus, the list \texttt{*2{red}, *1{green}, *{10}{blue, dashed}} instructs \texttt{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 \texttt{*1}. Hence, the following style lists are equal:

\texttt{ {*1{red}, *1{blue, dashed}}}  \\
\texttt{ {{red}, {blue, dashed}}}  \\
\texttt{ {red, {blue, dashed}}}

However, if you wish to use a single style comprising two or more keys for all grid lines, e.g. \texttt{red, dotted}, you must retain the multiplier (i.e., \texttt{*1{red, dotted}}).
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 \textit{style list} to \texttt{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}

\begin{tikzpicture}
\draw[dotted] (0,0) -- (12,0);
\draw[red, dashed, line width=1pt] (0,1cm) -- (12,1cm);
\end{tikzpicture}

\begin{verbatim}
/pgf/gantt/today =⟨tss⟩ none
/pgf/gantt/today offset =⟨number⟩ 1
/pgf/gantt/today rule /.style=⟨style⟩ dashed, line width=1pt
/pgf/gantt/today label =⟨text⟩ TODAY
/pgf/gantt/today label font =⟨font commands⟩ \normalfont
/pgf/gantt/today label node /.style=⟨style⟩ anchor=north, font=\ganttvalueof{today label font}
\end{verbatim}
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

`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⟩` \hspace{1cm} `true`

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

```
\begin{ganttchart}{hgrid, vgrid}{1}{6}
\gantttitle{Title 1}{3}
```

```
\begin{ganttchart}[
  hgrid, vgrid
]{{1}|{6}}
\gantttitle{Title 1}{3}
```
However, enabling this shortcut prevents you from entering multi-line node text (see section 16.4.3 of the TiKZ manual). Thus, \texttt{pgfgantt} provides the macro \texttt{\ganttnewline} for breaking lines in the node text.

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 \texttt{[grid]} for \texttt{\ganttnewline} (or \texttt{\\}), which draws a grid rule in \texttt{hgrid style} between the current and the new line. Alternatively, directly give the desired style as optional argument.
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, \texttt{pgfgantt} implements several titling commands.

\begin{ganttchart}[\texttt{hgrid}, \texttt{vgrid}]{1}{12}
\gantttitle{2011}{12} \\
\ganttbar{}{1}{4} \\
\ganttbar{}{6}{11}
\end{ganttchart}

\begin{Verbatim}
\begin{ganttchart}[hgrid style/.style=red]{1}{12}
\gantttitle{Title}{12} \\
\gantttitle*[\texttt{thick}, \texttt{blue}]{\langle label \rangle}{\langle number of time slots \rangle}
\end{ganttchart}
\end{Verbatim}

\begin{ganttchart}[hgrid, vgrid]{1}{12}
\gantttitle{2011}{12} \\
\ganttbar{}{1}{4} \\
\ganttbar{}{6}{11}
\end{ganttchart}

\texttt{\gantttitle} draws a single title element:

\begin{ganttchart}[hgrid, vgrid]{1}{12}
\gantttitle{2011}{12} \\
\ganttbar{}{1}{4} \\
\ganttbar{}{6}{11}
\end{ganttchart}

The \texttt{\langle label \rangle} appears in the title element, which covers the \texttt{\langle number of time slots \rangle} 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 \texttt{\gantttitle} for titles that span several time slots.
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:

```latex
\gantttitlelist[\langle options\rangle] \langle \pgffor list \rangle \langle \text{length of each element} \rangle
```

This macro generates one title element for each element of the `\pgffor list`. The second mandatory argument specifies the `\langle length of each element \rangle`. 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 `\langle pgffor list \rangle` is `1,...,12`.

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

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

```latex
\gantttitle{1} \gantttitle{2} \ldots \gantttitle{12} \\end{ganttchart}
```

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:

`/pgf\gantt/title list options = \langle pgffor options \rangle`

Changes the `\langle pgffor options \rangle` 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 \texttt{\textbackslash x}.

```latex
\begin{ganttchart}[hgrid, vgrid, x unit=1cm]{1}{7}
\gantttitlelist[
  title list options=%
  \{var=\textbackslash y, evaluate=\textbackslash y as \textbackslash x\}]
\end{ganttchart}
```
using "$\texttt{\textbackslash pgfcalendar\textbackslash weekdayshortname\{y\}}$"
$\{0,\ldots,6\}{\{1\}}$ \\
$\texttt{\textbackslash ganttbar\{\{1\}\{4\}}}$
$\texttt{\textbackslash ganttbar\{\{6\}\{7\}}}$
$\texttt{\textbackslash end\{\texttt{ganttchart}\}}$

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

$\texttt{\textbackslash gantttitlecalendar\{\{options\}\{\{calendar lines\}\}}}$

Prints a title calendar that spans the whole chart and contains one or more $\langle\text{calendar lines}\rangle$. The starred form of the macro prints a calendar from $\langle\text{start tss}\rangle$ to $\langle\text{end tss}\rangle$:

$\texttt{\textbackslash gantttitlecalendar\star\{\{options\}\{\{start tss\}\{\{end tss\}\{\{calendar lines\}\}}}$

$\langle\text{calendar lines}\rangle$ is a comma-separated list of line types:

<table>
<thead>
<tr>
<th>Line type</th>
<th>$\langle\text{output format}\rangle$</th>
<th>Example output</th>
</tr>
</thead>
<tbody>
<tr>
<td>year</td>
<td>n/a</td>
<td>2012, 2013, ...</td>
</tr>
<tr>
<td>month</td>
<td>$\langle\text{output format}\rangle$</td>
<td>01, 02, ..., 12</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>January, February, ...</td>
</tr>
<tr>
<td></td>
<td>shortcode</td>
<td>Jan, Feb, ...</td>
</tr>
<tr>
<td>week</td>
<td>$\langle\text{number}\rangle$</td>
<td>n/a</td>
</tr>
<tr>
<td>weekday</td>
<td>$\langle\text{output format}\rangle$</td>
<td>(none)</td>
</tr>
<tr>
<td></td>
<td>name</td>
<td>Monday, Tuesday, ...</td>
</tr>
<tr>
<td></td>
<td>shortcode</td>
<td>Mon, Tue, ...</td>
</tr>
<tr>
<td>day</td>
<td>n/a</td>
<td>01, 02, ..., 31</td>
</tr>
</tbody>
</table>

The $\langle\text{number}\rangle$ for the $\text{week}$ line type is the number of the first week in the calendar.

$\begin{\texttt{\textbackslash ganttchart}\{\}$
| hgrid, vgrid, x unit=4mm, time slot format=isodate |
| $\{2012-12-25\}\{2013-02-01\}$ |
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 TikZ 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:

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

```latex
\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}
```
Changes the text displayed in a week title element. In \( \langle \text{format} \rangle \), four additional macros are available: \texttt{\currentweek} is the current week number; \texttt{\startyear}, \texttt{\startmonth} and \texttt{\startday} expand to the year, month and day of the current week's Monday.

```latex
\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}
```
By default, one calendar day is one time slot wide. With `compress calendar=true`, one month corresponds to one time slot. Consequently, in compressed calendars only year and month are sensible line types for \texttt{gantttitlecalendar}. The time slot format \texttt{isodate-yearmonth} is especially suited for compressed calendars.

\begin{ganttchart}[ hgrid, vgrid, time slot format=isodate-yearmonth, compress calendar ]{2012-01}{2013-05} \gantttitlecalendar{year, month} \ganttbar{}{2012-05}{2013-01} \end{ganttchart}

\begin{ganttchart}[ hgrid, vgrid, time slot format=isodate-yearmonth, compress calendar ]{2012-01}{2013-05} \gantttitlecalendar{year, month} \ganttbar{}{2012-05}{2013-01} \end{ganttchart}

\texttt{/pgf\_gantt/title /.style=(style)}
\begin{itemize}
\item \texttt{shape=rectangle, inner sep=0pt, draw, fill=white}
\end{itemize}

Sets the appearance of a title element.

\begin{itemize}
\item \texttt{\usetikzlibrary{shadows}}
\item \texttt{\usetikzlibrary{shadings}}
\item \texttt{\begin{ganttchart} vgrid,}
\end{itemize}
\begin{ganttchart}
\vgrid, \hgrid,  
\title label font=\LARGE\color{violet},  
\title label node/.append style={anchor=west}
\}{1}{6}  
\gantttitle{2011}{6}  
\ganttbar{}{1}{2}  
\ganttbar{}{4}{6}  
\end{ganttchart}
The first three keys shift the coordinates of a title element’s borders (or rather of its corners), while \textit{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.
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 \texttt{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]{}{7} \gantttitlelist{1,...,7}{1} \ganttbar{}{1}{3} \ganttbar{}{4}{7} \end{ganttchart}

\textbf{2.6 Predefined Chart Elements}

\texttt{pgf}gantt predefines three chart elements:

1. \textit{Bars} indicate the duration of a task or one of its parts. \texttt{\ganttbar\{\texttt{options}\}\{\texttt{label}\}\{\texttt{start tss}\}\{\texttt{end tss}\}}

2. \textit{Groups} combine several subtasks (represented by bars) into a single task. \texttt{\ganttgroup\{\texttt{options}\}\{\texttt{label}\}\{\texttt{start tss}\}\{\texttt{end tss}\}}

3. \textit{Milestones} signify that an important task has been completed or that a crucial goal has been reached. \texttt{\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}

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⟩}
\ganttl linkingmilestone[⟨options⟩]{⟨label⟩}{⟨tss⟩}

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

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

\textit{pgf\texttt{gantt/chart element start border =}left | right}

Determines which border of the start time slot a chart element touches. \texttt{left} is the behavior usually expected, while \texttt{right} strictly interprets the start time slot as an \textit{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}
2.6.1 Options: Chart Element Appearance

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

- \texttt{/pgfgantt/bar/.style=\{style\}}
  \begin{itemize}
  \item shape=ganttbar, inner sep=0pt, draw, fill=white
  \end{itemize}

- \texttt{/pgfgantt/group/.style=\{style\}}
  \begin{itemize}
  \item shape=ganttgroup, inner sep=0pt, fill=black
  \end{itemize}

- \texttt{/pgfgantt/milestone/.style=\{style\}}
  \begin{itemize}
  \item shape=gantt milestone, inner sep=0pt, draw, fill=black
  \end{itemize}

Determines the appearance of the chart element. The shapes \texttt{ganttbar}, \texttt{gantt group} and \texttt{gantt milestone} 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[
\text{bar/.append style=\{shape=ellipse, fill=yellow, dashed\}}
]{Task 2}{4}{10} \\
\gantt milestone{Milestone 1}{11}
\end{ganttchart}
2.6.2 Options: Label Formatting

```
\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}
```

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).
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 left border of the chart and apply the `<options>` given by `inline label node` of the respective chart element and apply the `<options>` given by `inline label node`.

```latex
\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}
```
2.6.3 Options: Chart Element Positioning

\texttt{/{pgf}gantt/bar left shift =\langle factor\rangle } = 0
\texttt{/{pgf}gantt/bar right shift =\langle factor\rangle } = 0
\texttt{/{pgf}gantt/bar top shift =\langle factor\rangle } = .3
\texttt{/{pgf}gantt/bar height =\langle factor\rangle } = .4
\texttt{/{pgf}gantt/group left shift =\langle factor\rangle } = -.1
\texttt{/{pgf}gantt/group right shift =\langle factor\rangle } = .1
\texttt{/{pgf}gantt/group top shift =\langle factor\rangle } = .4
\texttt{/{pgf}gantt/group height =\langle factor\rangle } = .2
\texttt{/{pgf}gantt/milestone left shift =\langle factor\rangle } = .6
\texttt{/{pgf}gantt/milestone right shift =\langle factor\rangle } = .4
\texttt{/{pgf}gantt/milestone top shift =\langle factor\rangle } = .3
\texttt{/{pgf}gantt/milestone height =\langle factor\rangle } = .4

Shift the coordinates of a chart element’s borders (\ldots shift) and change its height (\ldots 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.

![Gantt chart diagram]

Milestone with standard values
- milestone height (here: 0.7)
- milestone top shift (here: 0.15)
- milestone left shift (here: 0.2)
- milestone right shift (here: -0.4)

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,
]
2.6.4 Options: Progress

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

\texttt{/pgf\textbackslash gantt/progress =none | today | \langle \textit{number} \rangle}

\texttt{\textbackslash ganttchart[vgrid, hgrid]{1}{12}}

\texttt{\gantttitle{Title}{12} \\\n\ganttgroup[progress=45]{Group 1}{1}{12} \\\n\ganttbar[progress=100]{Subtask 1}{1}{3} \\\n\ganttbar[progress=37]{Subtask 2}{4}{8} \\\n\ganttbar[progress=none]{Subtask 3}{9}{12} \\\n\end{ganttchart}
The value \texttt{today} instructs \texttt{pgfgantt} to calculate progress according to the value of the \texttt{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

\[
P = \frac{T - S}{E - S} \times 100\%
\]
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.
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={$\frac{#1}{100}$} ]{Subtask 2}{5}{12} \end{ganttchart}
2.6.5 New Node Shapes

`pgfgantt` defines three new node shapes:

1. The `ganttbar` node shape derives from shape `rectangle` (section 48.2 of the TiKZ manual). It provides four additional anchors: `on top`, `on bottom`, `on left` and `on right`. The \( \text{fraction} \) 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.

![Diagram of ganttbar node shape](image)

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`.

```
```

![Diagram of ganttgroup node shape](image)
(3) The \texttt{ganttmilestone} node shape derives from shape \texttt{diamond} (section 48.3 of the TikZ 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).

\begin{center}
\includegraphics[width=0.8\textwidth]{ganttmilestone}
\end{center}

\section{2.7 Defining Custom Chart Elements}

You may define completely new chart elements with

\begin{verbatim}
\newganttchartelement\{\langle name\rangle\}\{\langle new default key values\rangle\}
\newganttchartelement*\{\langle name\rangle\}\{\langle new default key values\rangle\}
\end{verbatim}

\texttt{\newganttchartelement} (unstarred) defines a new chart element \texttt{\gantt\langle name\rangle} and the corresponding \texttt{\ganttlinked\langle name\rangle}. These chart element macros take one optional argument \texttt{\langle options\rangle} and three mandatory arguments \texttt{\langle label\rangle}, \texttt{\langle start tss\rangle} and \texttt{\langle end tss\rangle} (like \texttt{\ganttbar}).

Chart element macros defined by the starred form, \texttt{\newganttchartelement*}, take the same single optional argument, but \texttt{two} mandatory arguments \texttt{\langle label\rangle} and \texttt{\langle tss\rangle} (like \texttt{\ganttmilestone}).

For each new chart element, \texttt{\newganttchartelement} also introduces a set of nine value-storing keys and five style keys and assigns default values to them:
Consequently, a new chart element will look like the standard \ganttbar unless you introduce some \textit{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={
    /pgf/gantt/foobar, 
    draw=foobaryellow, 
    bottom color=foobaryellow!50
  },
  foobar label font=\slshape,
  foobar left shift=-.1,
  foobar right shift=.1
}
So far, we have drawn charts whose elements were quite independent of each other. However, relations or \textit{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.

\begin{verbatim}
\ganttlink[⟨options⟩]{⟨start element name⟩}{⟨end element name⟩}
\end{verbatim}

\texttt{/pgfgantt/name =⟨name⟩} \hspace*{1em} (empty)

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

\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}

/pgf\texttt{gantt/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}
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:

   - `$/pgfgantt/link mid = \langle factor \rangle$` \hspace{1cm} 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.

   - `$/pgfgantt/link bulge = \langle factor \rangle$` \hspace{1cm} 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.

   - `$/pgfgantt/link tolerance = \langle factor \rangle$` \hspace{1cm} 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 $\langle factor \rangle$ (this is the case for the second link in the example below), the package draws a five-part link.

```latex
\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}
```

![Gantt chart example](image)

38
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 TikZ: 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}

\begin{figure}[h]
  \centering
  \begin{tabular}{l|c|c}
    \hline
    & Task 1 & \hspace{1cm} \\
    \hline
    Title & & \hspace{1cm} \\
    \hline
    Task 1 & \begin{tikzpicture}
      \draw[red, ultra thick, -latex, fill=red] (-0.2,0) -- (0.2,0);
    \end{tikzpicture} & \begin{tikzpicture}
      \draw[red, ultra thick, -latex, fill=red] (0,0) -- (0.2,0);
    \end{tikzpicture} \\
    Task 2 & \begin{tikzpicture}
      \draw[red, ultra thick, -latex, fill=red] (0,0) -- (0.2,0);
    \end{tikzpicture} & \begin{tikzpicture}
      \draw[red, ultra thick, -latex, fill=red] (0,0) -- (0.2,0);
    \end{tikzpicture} \\
    Task 3 & \begin{tikzpicture}
      \draw[red, ultra thick, -latex, fill=red] (0,0) -- (0.2,0);
    \end{tikzpicture} & \begin{tikzpicture}
      \draw[red, ultra thick, -latex, fill=red] (0,0) -- (0.2,0);
    \end{tikzpicture} \\
    Task 4 & & \hspace{1cm} \\
    \hline
  \end{tabular}
  \caption{Example of straight links in a Gantt chart.}
\end{figure}

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 3.10 presents the TikZ code of these links). For instance, **pgfgantt** provides one additional link type, \texttt{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}
\end{ganttchart}
The following macro creates new link types:

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

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:

- 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 2.6.5 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.

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

```latex
\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.

```latex
\begin{ganttchart}[vgrid, hgrid]{1}{12}
  \gantttitle{Title}{12} \\
  \ganttbar{Task 1}{2}{3} \\
  \ganttbar{Task 2}{7}{12}
  \ganttlink[link type=drur]{elem0}{elem1}
\end{ganttchart}
```
\newganttlinktype{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} \\n\ganttbar[inline]{Task 1}{2}{4} \n\ganttbar[inline]{Task 2}{8}{11} \\n\ganttlink[link type=drur]{elem0}{elem1}
\end{ganttchart}
The `link label` key locally overrides any label specified by \setganttlinklabel. The \texttt{font commands} and \texttt{options} are applied to the link label node. By default, the label appears to the right of the straight link's center.

\begin{tikzpicture}
\begin{ganttchart}[vgrid, hgrid, link label font=\small\bfseries\color{purple}]
\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}
\end{tikzpicture}
2.9 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,
  compress calendar,
  title/.append style={draw=none, fill=RoyalBlue!50!black},
  title label font=\sfamily\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} \\
\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.)
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
\{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} \\%
\ganttgroup[progress=0]{WBS 2 Summary Element 2}{4}{10} \\%
\textbf{47}
\ganttchart{\textbf{WBS 1.1} Activity A}{4}{5}\
\ganttchart{\textbf{WBS 1.2} Activity B}{6}{8}\
\ganttchart{\textbf{WBS 1.3} Activity C}{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}

\begin{tabular}{l|c|c|c|c|c}
\hline
\textbf{WBS 1 Summary Element 1} & 1 & 2 & 3 & 4 & 5  \\
\hline
WBS 1.1 Activity A & 0 & 0 & 0 & 0 & 57% complete  \\
\hline
WBS 1.2 Activity B & 0 & 0 & 0 & 67% complete & 0% complete  \\
\hline
WBS 1.3 Activity C & 0 & 0 & 0 & 25% complete & 0% complete  \\
\hline
WBS 1.4 Activity D & 0 & 0 & 0 & 0% complete & 0% complete  \\
\hline
\end{tabular}

\begin{tabular}{l|c|c|c|c|c}
\hline
\textbf{WBS 2 Summary Element 2} & 1 & 2 & 3 & 4 & 5  \\
\hline
WBS 2.1 Activity E & 0% complete & 0% complete & 0% complete & 0% complete & 0% complete  \\
\hline
WBS 2.2 Activity F & 0% complete & 0% complete & 0% complete & 0% complete & 0% complete  \\
\hline
WBS 2.3 Activity G & 0% complete & 0% complete & 0% complete & 0% complete & 0% complete  \\
\hline
\end{tabular}
3 Implementation

3.1 Packages

\texttt{pgfgantt} is modest in terms of dependencies: It only requires the Ti\texttt{k}Z and \texttt{pgfcalendar} packages.

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

3.2 Macros for Key and Error Management

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

\begin{verbatim}
\def\ganttset#1{\pgfqkeys{/pgfgantt}{#1}}
\end{verbatim}

The following auxiliary macros save us some code when we devise keys later on. \texttt{\@gtt@keydef}\texttt{\{\langle key\rangle\}\{\langle initial value\rangle\}} declares the key /\texttt{pgfgantt/\langle key\rangle} and stores its \langle initial value\rangle. \texttt{\ganttvalueof}\texttt{\{\langle key\rangle\}} retrieves the value stored by a \langle key\rangle. Link type authors should be able to use this macro in their code; thus, it lacks any @s.

\begin{verbatim}
\def\@gtt@keydef#1#2{\pgfkeyssetvalue{/pgfgantt/#1}{#2}}
\end{verbatim}

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

\begin{verbatim}
\def\@gtt@keydef#1#2{\pgfkeyssetvalue{/pgfgantt/#1}{#2}}
\end{verbatim}

\texttt{\@gtt@stylekeydef}\texttt{\{\langle key\rangle\}\{\langle initial style\rangle\}} declares a style \langle key\rangle with an \langle initial style\rangle.

\begin{verbatim}
\def\@gtt@stylekeydef#1#2{\pgfkeys{/pgfgantt/#1/.style={#2}}}
\end{verbatim}

\texttt{\@gtt@PackageError}\texttt{\{\langle message\rangle\}} and \texttt{\@gtt@PackageWarning}\texttt{\{\langle message\rangle\}} issue a package error or warning \langle message\rangle, respectively.
3.3 The Horizontal and Vertical Grid

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

\begin{verbatim}
\newcount\g@currentline
\newcount\g@lasttitleline
\newcount\g@currgrid
\newcount\g@chartwidth

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

\begin{verbatim}
\@\g@stylekeydef{hgrid style}{dotted}
\newif\if\g@hgrid
\ganttset{%
  hgrid/.code={%
    \def\@tempa{#1}%
    \def\@tempb{false}%
    \ifx\@tempa\@tempb%
      \g@hgridfalse%
    \else%
      \g@hgridtrue%
      \def\@tempb{true}%
      \ifx\@tempa\@tempb%
        \def\g@hgridstyle{dotted}%
      \else%
        \def\g@hgridstyle{#1}%
      \fi%
    \fi%
  }
}
\end{verbatim}

\end{verbatim}

The \verb+\g@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 \verb+\expandafter+ construction, which enables tail recursion.

\begin{verbatim}
\def\g@hgrid@do#1,{%
  \ifx\relax#1\else
    \ifnum\g@currgrid<\g@currentline
      \g@hgrid@analyze#1\relax%
    \expandafter\expandafter\expandafter\g@hgrid@do%
  \fi%
  \fi%
}
\end{verbatim}

The \verb+\g@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 \verb+\expandafter+ construction, which enables tail recursion.
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{% 
\@ifstar{\gtt@hgrid@draw}{\gtt@hgrid@draw1}%
}

\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}% 
\draw [\#2] (0pt, \y@upper pt) -- (\gtt@chartwidth * \ganttvalueof{x unit}, \y@upper pt);\relax% 
\global\advance\gtt@currgrid by-1\relax\ifnum\gtt@currgrid<\gtt@currentline\breakforeach\fi%
}%
}

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,{%
3.4 Time Slot Formats

\gtt@smugglecount\{(count)\} smuggles the local value of a count register over the end of a T\TeX\ group.
\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 \(\text{count 1}\) to a time slot and stores the latter in \(\text{count 2}\). 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. If the chart is compressed, one time slot corresponds to one month; otherwise, it corresponds to one day.

\newcommand\gtt@juliantotimeslot[2]{% \begingroup% \@tempcnta=#1\relax% \ifgtt@compresscalendar% \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\@tempc\relax% \endgroup%}
\advance\@tempcnta by-\gtt@startmonth\relax%
\advance\@tempcnta by1\relax
\else%
\advance\@tempcnta by-\gtt@startjulian\relax%
\advance\@tempcnta by1\relax
\fi%
#2=\@tempcnta\relax
\gtt@smugglecount#2%
\endgroup%
}

\newgantttimeslotformat{(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\newgantttimeformat[2]{%
\expandafter\def\csname gtt@tsstojulian@#1\endcsname##1##2{%#2=#1\relax
\advance#2 by\gtt@tsf@startjulian\relax%
\advance#2 by-1\relax%
}}%

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

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

\newgantttimeformat{isodate}{%
\pgfcalendardatetojulian{#1}{#2}%
}%

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

\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elax{%
\def\local@day{##1}
\def\local@month{##2}
\def\local@year{##3}%
}%
\expandafter\local@decompose\local@firstarg..elax%
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}%
}
The key \texttt{time slot format=⟨name⟩} checks whether the format \texttt{⟨name⟩} exists and then defines the macro \texttt{gtt@tsstojulian} to be equivalent to \texttt{gtt@tsstojulian⟨name⟩}.

\begin{tikzpicture}
\end{tikzpicture}

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

\begin{tikzpicture}
\end{tikzpicture}

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

\begin{tikzpicture}
\end{tikzpicture}

\subsection{3.5 The Main Environment}

Keys that store the basis vectors of the chart.

\begin{tikzpicture}
\end{tikzpicture}
Keys related to the canvas and the today rule.

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

\@gtt@stylekeydef{canvas}{shape=rectangle, draw, fill=white}
\@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}
 anchor=north, font=\ganttvalueof{today label font}]
\end{verbatim}

Boolean key that determines if \texttt{\textbackslash \newline} is equivalent to \texttt{\textbackslash \ganttnewline}. \begin{verbatim}
\newif\ifgtt@newlineshortcut
\ganttset{\newline shortcut/.is if=gtt@newlineshortcut,\newline shortcut=true}
\end{verbatim}

The boolean \texttt{\ifgtt@tikzpicture} is true if a Gantt chart appears within a Ti\textit{kZ} picture. \texttt{\ifgtt@intitle} is true at the start of a \texttt{ganttchart} environment and set to false as soon as the first non-title element is encountered. \texttt{\gtt@lasttitleslot} corresponds to the \texttt{x}-coordinate of its right border. \texttt{\gtt@elementid} enumerates the automatic names of chart elements. \texttt{\gtt@today@slot} is the time slot of the today rule. \texttt{\gtt@startjulian} and \texttt{\gtt@endjulian} contain the Julian dates corresponding to the first and last time slot, respectively.

At the beginning of a \texttt{ganttchart} environment, the keys in its optional argument are executed. Initialize the macros and counts that contain start dates, end dates, the chart width, ...
... the time slot of the today rule, ...
\begin{Verbatim}
def\@tempb{\ganttvalueof{today}}
\ifx\@tempa\@tempb\else
\gtt@tsstojulian{\ganttvalueof{today}}{\gtt@today@slot}
\gtt@juliantotimeslot{\gtt@today@slot}{\gtt@today@slot}
\fi
\end{Verbatim}

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

\begin{Verbatim}
\global\gtt@elementid=0\relax
\global\gtt@currentline=0\relax
\global\gtt@lasttitleline=0\relax
\global\gtt@lasttitleslot=0\relax
\end{Verbatim}

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

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

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

\begin{Verbatim}
\begin{scope}[on background layer]
\ifgantt@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}}
\end{scope}
\end{Verbatim}
\[ + (\texttt{@currentline} - \texttt{@lasttitleline} - 1) \times \texttt{@chartwidth} \]
\[ \times \texttt{ganttvalueof(y unit chart)} \]
\[ \texttt{pgfmathsetmacro}\texttt{y@mid}{(\texttt{y@upper} + \texttt{y@lower}) / 2} \]
\[ \texttt{pgfmathsetmacro}\texttt{y@size}{abs(\texttt{y@lower} - \texttt{y@upper})} \]
\[ \texttt{pgfmathsetmacro}\texttt{x@size}{\texttt{chartwidth} * \texttt{ganttvalueof(x unit)}} \]
\[ \texttt{node \[/pgfgantt/canvas, minimum width=\texttt{x@size pt}, minimum height=\texttt{y@size pt}] at (\texttt{x@size pt / 2}, \texttt{y@mid pt}) \}; \]

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

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.

The last task of \texttt{ganttchart} is to apply the \texttt{today} key if its value differs from \texttt{none}. \[ \texttt{x@mid} \]
At the end of a \ganttchart, we also close the \tikzpicture if we started it implicitly.

3.6 Starting a New Line

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%
\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 Titles

Keys that influence title elements. Note that \texttt{\textbackslash ganttkeydef} cannot define title list options, since \texttt{\textbackslash gantttitlelistoptions} is expanded after a \texttt{\textbackslash foreach} statement, where \texttt{\textbackslash ganttvalueof} will not work.

\begin{verbatim}
\texttt{\textbackslash ganttstylekeydef}{title}{shape=rectangle, inner sep=0pt, draw, fill=white}
\texttt{\textbackslash ganttkeydef}{title label font}{\texttt{\small}}
\texttt{\textbackslash ganttstylekeydef}{title label node}{
  anchor=center, font=\texttt{\textbackslash ganttvalueof}{title label font}}
}
\texttt{\textbackslash ganttset}{
  title list options/.code={
    \def\texttt{\textbackslash gantttitlelistoptions}{[#1]}\
  },
  title list options={var=\texttt{x}, evaluate=\texttt{x}}
}
\texttt{\textbackslash ganttkeydef}{title left shift}{0}
\texttt{\textbackslash ganttkeydef}{title right shift}{0}
\texttt{\textbackslash ganttkeydef}{title top shift}{0}
\texttt{\textbackslash ganttkeydef}{title height}{0.6}
\texttt{\newif\texttt{\textbackslash ganttinclude}}
\texttt{\texttt{\textbackslash ganttset}{
  include title in canvas/.is if=\texttt{\textbackslash ganttinclude},
  include title in canvas=false
}}
\end{verbatim}

Keys for title calendars.

\begin{verbatim}
\texttt{\textbackslash ganttkeydef}{calendar week text}{Week~\texttt{\textbackslash currentweek}}
\texttt{\newif\texttt{\textbackslash ganttcompresscalendar}}
\texttt{\texttt{\textbackslash ganttset}{
  compress calendar/.is if=\texttt{\textbackslash ganttcompresscalendar},
  compress calendar=false
}}
\end{verbatim}

\texttt{\textbackslash 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 \texttt{\textbackslash ganttlasttitleslot}. This count is updated at the end of the macro.

\begin{verbatim}
\newcommand\texttt{\textbackslash gantttitle}[3][]{
\texttt{\textbackslash begin{group}}
\texttt{\textbackslash ganttset[#1]}\
\texttt{\pgfmathsetmacro}{x@left}{\texttt{\textbackslash ganttvalueof}\texttt{\textbackslash ganttlasttitleslot} + \texttt{\textbackslash ganttvalueof}{title left shift}}\
\texttt{\pgfmathsetmacro}{x unit}{\texttt{\textbackslash ganttvalueof}{x unit}}\
\texttt{\texttt{\textbackslash ganttvalueof}\texttt{\textbackslash ganttlasttitleslot} + 3 + \texttt{\textbackslash ganttvalueof}{title right shift}}\
\texttt{\texttt{\textbackslash end{group}}}
\texttt{\texttt{\textbackslash gantttitle}}
\end{verbatim}
\begin{verbatim}
\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 [/pgfantt/title, minimum width=x@size pt, minimum height=y@size pt] {}
  node [/pgfantt/title label node] {#2};
\global\advance\gtt@lasttitleslot by#3\relax
\endgroup
\end{verbatim}

\gantttitlelist generates title elements by repeatedly calling \gantttitle. Since \gantttitlelist does not have to calculate the respective x-coordinates explicitly.

\begin{verbatim}
\newcommand\gantttitlelist[3][{}]{\begin{group}
\ganttset{#1}
\expandafter\foreach\gtt@titlelistoptions in {#2} {\gantttitle{\x}{#3}}
\end{group}}
\end{verbatim}

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

\begin{verbatim}
\ifgantt@titlecalendarstar
\newcommand\gantttitlecalendar{\ifttt@titlecalendarstar}
\else
\newcommand\gantttitlecalendar{\ifttt@titlecalendarstar}
\@endif
\end{verbatim}

61
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%
\fi%
}

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}%
}

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%
\pgfkeys{/pgfgantt/calendar}{#3}%
}
}
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

For each ⟨line type⟩, we define a corresponding key /pgfgantt/calendar/⟨line type⟩. This key performs the necessary calculations and draws one or several \gantttitles.

Line type year draws years.

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

\begin{verbatim}
calendar/month/.code={% 
  \ifganttcalendar@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}% 
        }{% 
        \ifgantt pgfcalendarmonth#1\else 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}% 
          }{% 
          \ifgantt compresscalendar1\else the\gtt@calendar@slots\fi% 
          }% 
        \gtt@calendar@slots=1\relax% 
      }% 
      \fi% 
    }{% 
    }% 
  }% 
  \endgroup% 
  \gtt@calendar@eoltrue% 
},%,% 
\end{verbatim}

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`. 

\begin{verbatim}
calendar/week/.code={% 
  \ifganttcalendar@eol\ganttnewline\fi% 
  \begingroup% 
  \gtt@calendar@slots=1\relax% 
  \pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{% 
    \ifdate{equals=\pgfcalendarendiso}{% 
      \ifnum\gtt@calendar@slots=1\relax\else% 
        \advance\gtt@calendar@slots by-1\relax% 
        \gantttitle{\csname pgfcalendarmonth#1\endcsname{\pgfcalendarcurrentmonth}% 
          }{% 
          \ifgantt compresscalendar1\else the\gtt@calendar@slots\fi% 
          }% 
        \gtt@calendar@slots=1\relax% 
      }% 
      \fi% 
    }{% 
    }% 
  }% 
  \endgroup% 
  \gtt@calendar@eoltrue% 
},% 
\end{verbatim}

64
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.
Line type day=⟨format⟩ draws days of the month.

3.8 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.

```latex
\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%
},% 
} \ifgantt@keydef{name}{} \newif\ifgantt@ce@startatleftborder 
\ganttset{%
chart element start border/.is choice,% 
chart element start border/left/.code=\ifgantt@ce@startatleftbordertrue,%, 
chart element start border/right/.code=\ifgantt@ce@startatleftborderfalse,%, 
chart element start border=left% 
} 
```

66
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[⟨options⟩]{⟨label⟩}{⟨start tss⟩}{⟨end tss⟩}{⟨type⟩}` is the generic command for drawing chart elements of a certain `⟨type⟩`. First, it 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%
\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}\%%
```

67
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.
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.

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

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.
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 \langle label \rangle is not empty, a label is printed. Its anchor is either at the \langle type \rangle 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.

\def\newganttchartelement{%
\@ifstar\@newganttchartelement@one\@newganttchartelement@two%
\newganttchartelement*

70
Both \texttt{\newganttchartelement@one\{\textit{type}\}} and \texttt{\newganttchartelement@two\{\textit{type}\}} define two macros \texttt{\gantt\{\textit{type}\}} and \texttt{\ganttlinked\{\textit{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. \texttt{\gantt milestone} vs. \texttt{\gantt bar}). At the end, we execute \texttt{\newganttchartelement@definekeys} to process the second mandatory argument of \texttt{\newganttchartelement}.

\begin{verbatim}
\newcommand\newganttchartelement@one[1] {%
  \expandafter\newcommand\csname gantt#1\endcsname[3][{]{
    \gtt@chartelement[##1]{##2}{##3}{##3}{#1}%
  }%}
\end{verbatim}

\begin{verbatim}
\newcommand\newganttchartelement@two[1] {%
  \expandafter\newcommand\csname gantt#1\endcsname[4][{]{
    \gtt@chartelement[##1]{##2}{##3}{##4}{#1}%
  }%}
\end{verbatim}

\begin{verbatim}
\newcommand\newganttchartelement@definekeys[2] {[{]}
  \@gt@stylekeydef{#1}{shape=rectangle, inner sep=0pt, draw, fill=white}%
  \@gt@stylekeydef{#1 incomplete}{/pgfgantt/#1, fill=black!25}%
  \@gt@keydef{#1 label font}{\normalsize}%
  \@gt@stylekeydef{#1 label node}{
    anchor=east, font=\ganttvalueof{#1 label font}%
  }%
  \@gt@keydef{#1 inline label anchor}{(center)}%
\end{verbatim}
\ganttstylekeydef{#1 inline label node} {% 
  anchor=center, font=\ganttvalueof{#1 label font} 
}%
\ganttkeydef{#1 progress label anchor}{east} 
\ganttkeydef{#1 progress label font}{\scriptsize} 
\ganttstylekeydef{#1 progress label node} {% 
  anchor=west, font=\ganttvalueof{#1 progress label font} 
}%
\ganttkeydef{#1 left shift}{0} 
\ganttkeydef{#1 right shift}{0} 
\ganttkeydef{#1 top shift}{.3} 
\ganttkeydef{#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.
\newganttchartelement{bar} {% 
  bar/.style={shape=ganttbar, inner sep=0pt, draw, fill=white},%
  bar incomplete/.style={/pgfgantt/bar, fill=black!25},%
  bar label text=\strut\,%, 
  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`.

```latex
\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.

\@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%
    }%
  }
}%
```

73
Code for the predefined chart element type milestone.

\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 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%
}%
%
3.9 Node Shapes

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

\@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.

\pgfdeclarerectangle[on top fraction]{.5}
\pgfdeclarerectangle[on bottom fraction]{.5}
\pgfdeclarerectangle[on left fraction]{.5}
\pgfdeclarerectangle[on right fraction]{.5}
Code for node shape `ganttgroup`. Anchors derive from node shape `ganttbar`. The two additional anchors `left peak` and `right peak` are defined.

```latex
\pgfdeclare形状{ganttgroup}[]{\pgfsaved ancor形状s[\从=矩形]
\pgf继承了\保存的形状[\从=矩形]{中心}
\pgf继承了\保存的形状[\从=矩形]{mid}
\pgf继承了\保存的形状[\从=矩形]{基础}
\pgf继承了\保存的形状[\从=矩形]{北}
\pgf继承了\保存的形状[\从=矩形]{南}
\pgf继承了\保存的形状[\从=矩形]{西}
\pgf继承了\保存的形状[\从=矩形]{mid西}
\pgf继承了\保存的形状[\从=矩形]{base西}
\pgf继承了\保存的形状[\从=矩形]{北西}
\pgf继承了\保存的形状[\从=矩形]{南西}
\pgf继承了\保存的形状边[\从=矩形]
\pgf继承了\保存的形状[\从=矩形]{on top}
\pgf继承了\保存的形状[\从=矩形]{on bottom}
\pgf继承了\保存的形状[\从=矩形]{on left}
\pgf继承了\保存的形状[\从=矩形]{on right}
\pgf anchor{left peak}{
\pgf@process{
\pgf@pointadd{
\southwest
}{\pgf@process{%\pgfkeysvalueof{/pgf/outer x sep}}\%\pgfkeysvalueof{/pgf/outer y sep}}}
}
\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}
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.
\begin{verbatim}
1220 \pgf@x=0pt
1221 \pgfmathparse{
1222 \ganttvalueof{on right fraction} < 0.5
1223 ? \ganttvalueof{on right fraction}
1224 : 1 - \ganttvalueof{on right fraction}
1225 }
1226 \advance\pgf@x by\pgfmathresult\pgf@xa
1227 }
1228 \anchor{on left}{
1229 \pgf@process{\outernortheast}
1230 \pgf@ya=-2\pgf@y
1231 \advance\pgf@y by\ganttvalueof{on left fraction}\pgf@ya
1232 \pgf@xa=-2\pgf@x
1233 \pgf@x=0pt
1234 \pgfmathparse{
1235 \ganttvalueof{on left fraction} < 0.5
1236 ? \ganttvalueof{on left fraction}
1237 : 1 - \ganttvalueof{on left fraction}
1238 }
1239 \advance\pgf@x by\pgfmathresult\pgf@xa
1240 }
1241
3.10 Links

Keys for configuring links.

\begin{verbatim}
1243 \@gtt@stylekeydef{link}{-latex, rounded corners=1pt}
1244 \@gtt@keydef{link type}{auto}
1245 \@gtt@keydef{link label}{}
1246 \@gtt@keydef{link label font}{\scriptsize\itshape}
1247 \@gtt@stylekeydef{link label node}{
1248 anchor=west, font=\ganttvalueof{link label font}%
1249 }
1250 \newganttlinktype{⟨type⟩}{⟨code⟩} stores ⟨code⟩ in an internal macro
\@gtt@linktype@⟨type⟩, which is later called by \gtt@drawlink.
1251 \newcommand\newganttlinktype[2]{%
1252 \expandafter\def\csname @gtt@linktype@#1\endcsname{#2}%
1253 }
1254 \setganttlinklabel{⟨type⟩}{⟨label⟩} stores a given ⟨label⟩ in an internal macro \@gtt@linktype@⟨type⟩@label, which is later used by \gtt@drawlink.
1255 \newcommand\setganttlinklabel[2]{%
1256 \expandafter\def\csname @gtt@linktype@#1label\endcsname{#2}%
1257 }
\end{verbatim}
\end{verbatim}
\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).

\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:
\@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, \yLower)$) \% 
  ($($\xLeft, \yLower)!\ganttvalueof{link mid}!(\xRight, \yLower)$) \% 
  (\xRight, \yLower); \% 
}\}
\newganttlinktypealias{default}{r}

(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] \% 
}\}
Now we may define line 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, \texttt{pgfmathparse} yields 0.

\begin{verbatim}
\newganttlinktype{auto}{\%
  \pgfmathparse{abs(\yUpper - \yLower) <= 1}%
  \ifcase\pgfmathresult\%
  \else\%
  \fi%
  \pgfmathparse{\%
    (\xRight - \xLeft) >= \ganttvalueof{link tolerance} * \ganttvalueof{x unit}\%
  \ifcase\pgfmathresult\%
  \else\%
  \fi%}
  \ifcase\pgfmathresult\%
  \else\%
  \fi%
}
\end{verbatim}

The \texttt{dr} type is explained in section 2.8.
Here is the definition of the four straight link types and their labels.

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

83
\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%
}

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

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

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

\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\{\langle anchor\rangle\} is similar to the command above. However, it stores the anchor coordinates in the auxiliary macros \xRight and \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\{\langle start element\rangle\}\{\langle options\rangle\}\{\langle end element\rangle\} executes the \langle options\rangle and stores the names of the connected elements \langle E1\rangle and \langle E2\rangle in \@gtt@link@startelement and \@gtt@link@endelement.

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

\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.

\let\ganttsetstartanchor=\@gtt@setstartanchor\%
\let\ganttsetendanchor=\@gtt@setendanchor\%
\ganttsetstartanchor{east}
4 Index

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

Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>%</td>
<td>650</td>
</tr>
<tr>
<td>@@\gantttitlecalendar</td>
<td>\gantttitlecalendar</td>
<td>478, 487, 490</td>
</tr>
<tr>
<td>@@\gantttitlecalendar</td>
<td>\gantttitlecalendar</td>
<td>476, 484</td>
</tr>
<tr>
<td>\gantttitlecalendar</td>
<td>\gantttitlecalendar</td>
<td>468, 469, 472</td>
</tr>
<tr>
<td>\ganttPackageError</td>
<td>\ganttPackageError</td>
<td>18, 183, 225</td>
</tr>
<tr>
<td>\ganttPackageWarning</td>
<td>\ganttPackageWarning</td>
<td>18, 727, 1376</td>
</tr>
<tr>
<td>\gantt@currlabel</td>
<td>\gantt@currlabel</td>
<td>468, 469, 472</td>
</tr>
<tr>
<td>\gantt@currlinktype</td>
<td>\gantt@currlinktype</td>
<td>1374, 1384, 1391</td>
</tr>
<tr>
<td>\gantt@keydef</td>
<td>\gantt@keydef</td>
<td>9, 250-252, 254, 255, 257, 258, 393, 403-406, 412, 653, 849, 853, 857, 858, 862-865, 920-925, 972-975, 1244-1246, 1267-1269</td>
</tr>
<tr>
<td>\gantt@link@anchor</td>
<td>\gantt@link@anchor</td>
<td>1394, 1402, 1415, 1422</td>
</tr>
<tr>
<td>\gantt@link@endelement</td>
<td>\gantt@link@endelement</td>
<td>1422, 1427</td>
</tr>
<tr>
<td>\gantt@link@startelement</td>
<td>\gantt@link@startelement</td>
<td>1415, 1427</td>
</tr>
<tr>
<td>\gantt@linkanchordef</td>
<td>\gantt@linkanchordef</td>
<td>1399</td>
</tr>
<tr>
<td>\gantt@setendanchor</td>
<td>\gantt@setendanchor</td>
<td>1420, 1433</td>
</tr>
<tr>
<td>\gantt@setstartanchor</td>
<td>\gantt@setstartanchor</td>
<td>1413, 1432</td>
</tr>
<tr>
<td>\gantt@stylekeydef</td>
<td>\gantt@stylekeydef</td>
<td>15, 25, 253, 256, 259, 392, 394, 847, 848, 850, 854, 859, 1243, 1247</td>
</tr>
<tr>
<td>\newganttchart@definekeys</td>
<td>\newganttchart@definekeys</td>
<td>829, 843, 846</td>
</tr>
<tr>
<td>\newganttchart@one</td>
<td>\newganttchart@one</td>
<td>815, 818</td>
</tr>
<tr>
<td>\newganttchart@two</td>
<td>\newganttchart@two</td>
<td>815, 818</td>
</tr>
<tr>
<td>\anchor</td>
<td>\anchor</td>
<td>995, 1005, 1015, 1025, 1060, 1079, 1187, 1201, 1215, 1228</td>
</tr>
<tr>
<td>\backgroundpath</td>
<td>\backgroundpath</td>
<td>1110</td>
</tr>
</tbody>
</table>

A

environments:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>ganttchart</td>
<td>ganttchart</td>
<td>4, 275</td>
</tr>
</tbody>
</table>

B

\ganttalignnewline \ganttalignnewline \ganttalignnewline \ganttalignnewline \ganttalignnewline \ganttalignnewline

bar (option) \ganttalignnewline 23, 875
bar height (option) \ganttalignnewline 26, 875
bar incomplete (option) \ganttalignnewline 30, 875
bar inline label anchor (option) \ganttalignnewline
bar inline label node (option) \ganttalignnewline 25, 875
bar label font (option) \ganttalignnewline 24, 875
bar label node (option) \ganttalignnewline 24, 875
bar label text (option) \ganttalignnewline 24, 875
bar left shift (option) \ganttalignnewline 26, 875
bar progress label anchor (option) \ganttalignnewline
bar progress label font (option) \ganttalignnewline 31, 875
bar progress label node (option) \ganttalignnewline
bar right shift (option) \ganttalignnewline 26, 875
bar top shift (option) \ganttalignnewline 26, 875

calendar week text (option) \ganttalignnewline 16, 412
canvas (option) \ganttalignnewline 6, 253
chart element start border (option) \ganttalignnewline 22, 641
compress calendar (option) \ganttalignnewline 17, 412 \currentweek \currentweek \currentweek \currentweek \currentweek

eventsoptions:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ganttalignnewline</td>
<td>\ganttalignnewline</td>
<td>11, 297</td>
</tr>
</tbody>
</table>

G

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ganttalignnewline</td>
<td>\ganttalignnewline</td>
<td>11, 297</td>
</tr>
<tr>
<td>\ganttbar</td>
<td>\ganttbar</td>
<td>20, 875</td>
</tr>
</tbody>
</table>
link type (option) 37, 1243
\local@day 169, 174, 179, 203, 208, 209, 211, 219
\local@decompose 168, 171, 173, 176, 178, 181, 185, 186, 189, 190, 193, 194
\local@firstarg 167, 171, 176, 181
\local@month 169, 172, 174, 177, 179, 182, 203, 211, 219
\local@none 719
\local@timeslotmodifier 675, 737, 738, 756, 760
\local@year 169, 174, 179, 185, 186, 189, 190, 193, 194, 200, 201, 203, 211, 216, 217, 219

M
milestone (option) 23, 947
milestone height (option) 26, 947
milestone incomplete (option) 31, 947
milestone inline label anchor (option) 25, 947
milestone inline label node (option) 25, 947
milestone label font (option) 24, 947
milestone label node (option) 24, 947
milestone label text (option) 24, 947
milestone left shift (option) 26, 947
milestone progress label anchor (option) 32, 947
milestone progress label font (option) 32, 947
milestone progress label node (option) 32, 947
milestone right shift (option) 26, 947
milestone top shift (option) 26, 947

N
name (option) 36, 641
\newganttchartelement 34, 814, 875, 898, 947
\newganttchartelement* 34, 814
\newganttlinktype 40, 1250, 1270, 1278, 1288, 1308, 1324, 1334, 1344, 1354, 1364
\newganttlinktypealias 41, 1258, 1276
\newgantttimeslotformat 5, 146, 152, 158, 162, 198, 206, 214
newline shortcut (option) 10, 262
\northeast 998, 1006, 1016, 1028, 1082, 1113

O
on bottom fraction (option) 33, 972
on left fraction (option) 33, 972
on right fraction (option) 33, 972
on top fraction (option) 33, 972

options:
bar 23, 875
bar height 26, 875
bar incomplete 30, 875
bar inline label anchor 25, 875
bar inline label node 25, 875
bar label font 24, 875
bar label node 24, 875
bar label text 24, 875
bar left shift 26, 875
bar progress label anchor 31, 875
bar progress label font 31, 875
bar progress label node 31, 875
bar right shift 26, 875
bar top shift 26, 875
calendar week text 16, 412
canvas 6, 253
chart element start border 22, 641
compress calendar 17, 412
group 23, 898
group height 26, 898
group incomplete 31, 898
group inline label anchor 25, 898
group inline label node 25, 898
group label font 24, 898
group label node 24, 898
group label text 24, 898
group left peak height 28, 920
group left peak tip position 28, 920
group left peak width 28, 920
group left shift 26, 898
group peaks height 28, 920
group peaks tip position 28, 920
group peaks width 28, 920
group progress label anchor 31, 898
group progress label font 32, 898
group progress label node 32, 898
group right peak height 28, 920
group right peak tip position .................................. \texttt{28, 920}
group right peak width ........................................ \texttt{28, 920}
group right shift ................................................ \texttt{26, 898}
group top shift .................................................... \texttt{26, 898}
hgrid ............................................................... \texttt{7, 29}
hgrid style ........................................................ \texttt{7, 29}
include title in canvas ......................................... \texttt{20, 392}
inline ............................................................... \texttt{25, 641}
link ................................................................. \texttt{37, 1243}
link bulge ........................................................ \texttt{38, 1267}
link label ........................................................... \texttt{44, 1243}
link label font .................................................... \texttt{44, 1243}
link label node .................................................... \texttt{44, 1243}
link mid ............................................................. \texttt{38, 1267}
link tolerance ..................................................... \texttt{38, 1267}
link type ........................................................... \texttt{37, 1243}
milestone .......................................................... \texttt{23, 947}
milestone height ................................................... \texttt{26, 947}
milestone incomplete ............................................. \texttt{31, 947}
milestone inline label anchor ................................... \texttt{25, 947}
milestone inline label node ....................................... \texttt{25, 947}
milestone label font ............................................. \texttt{24, 947}
milestone label node ............................................. \texttt{24, 947}
milestone label text ............................................. \texttt{24, 947}
milestone left shift ............................................. \texttt{26, 947}
milestone progress label anchor ................................ \texttt{32, 947}
milestone progress label font ................................... \texttt{32, 947}
milestone progress label node ................................... \texttt{32, 947}
milestone right shift ............................................ \texttt{26, 947}
milestone top shift .............................................. \texttt{26, 947}
name ............................................................... \texttt{36, 641}
newline shortcut .................................................. \texttt{10, 262}
on bottom fraction ................................................ \texttt{33, 972}
on left fraction .................................................... \texttt{33, 972}
on right fraction .................................................. \texttt{33, 972}
on top fraction ..................................................... \texttt{33, 972}
progress .......................................................... \texttt{29, 641}
progress label text ............................................. \texttt{31, 641}
time slot format .................................................. \texttt{4, 222}
time slot format/base century (option) ....................... \texttt{5, 233}
time slot format/start date (option) ......................... \texttt{5, 241}
title .................................................................. \texttt{17, 392}
time slot format ................................................... \texttt{4, 222}
time slot format/base century ................................... \texttt{5, 233}
time slot format/start date ....................................... \texttt{5, 241}
title .................................................................. \texttt{17, 392}
progress (option) .................................................. \texttt{29, 641}
progress label text (option) .................................... \texttt{31, 641}
setganttlinklabel .................................................. \texttt{41, 1254, 1342, 1352, 1362, 1372}
southwest ........................................................... \texttt{996, 1008, 1018, 1026, 1063, 1094, 1126}
startday ............................................................... \texttt{16, 575}
startmonth ............................................................ \texttt{16, 575}
startyear .............................................................. \texttt{16, 575}
tikzalign@newline ................................................ \texttt{304}
time slot format (option) ........................................ \texttt{4, 222}
time slot format/base century (option) ....................... \texttt{5, 233}
time slot format/start date (option) ......................... \texttt{5, 241}
title (option) ....................................................... \texttt{17, 392}
title height .......................................................... \texttt{18, 392}
title label font ..................................................... \texttt{18, 392}
title label node ..................................................... \texttt{18, 392}
title left shift ...................................................... \texttt{18, 392}
title list options .................................................. \texttt{13, 392}
title top shift ...................................................... \texttt{18, 392}
title right shift .................................................... \texttt{18, 392}
today (option) ...................................................... \texttt{9, 253}
today label (option) .............................................. \texttt{9, 253}
today label font (option) ........................................ \texttt{9, 253}
today rule ........................................................... \texttt{9, 253}
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 TikZ 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.
\newgantttchartelement 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.