Drawing Gantt Charts in \LaTeX\ with TikZ

The pgfgantt Package

Wolfgang Skala∗

v5.0
2018/01/10

The pgfgantt package provides the 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 different links between these elements. The appearance of the chart elements is highly customizable, and even new chart elements may be defined.

∗Department of Biosciences, University of Salzburg, Austria; Wolfgang.Skala@sbg.ac.at
Contents

1 Introduction 3

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

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

4 Index 95

5 Change History 101
1 Introduction

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

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

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

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

\(^1\)http://www.martin-kumm.de/tex_gantt_package.php
\(^2\)http://ctan.org/tex-archive/graphics/pgf/
\(^3\)http://ctan.org/tex-archive/graphics/pstricks/contrib/pst-gantt/
2 User Guide

2.1 Overview

To load the package, simply put
\usepackage{pgfgantt}

into the document preamble.

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

\begin{ganttchart}{1}{12}
\gantttitle{2011}{12} \\
\gantttitlelist{1,...,12}{1} \\
\ganttgroup{Group 1}{1}{7} \\
\ganttbar{Task 1}{1}{2} \\
\ganttlinkedbar{Task 2}{3}{7} \ganttnewline
\ganttbar{Milestone}{7} \ganttnewline
\ganttbar{Final Task}{8}{12} \\
\ganttlink{elem2}{elem3} \\
\ganttlink{elem3}{elem4}
\end{ganttchart}

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

\begin{tikzpicture}
\draw (0,0) node {Group 1} -- (4,0) -- (5,0) -- (6,0) -- (7,0) -- (8,0) -- (12,0);
\draw (0,2) node {Task 1} -- (1,2) -- (2,2) -- (3,2) -- (4,2) -- (5,2) -- (6,2) -- (7,2) -- (8,2) -- (9,2) -- (10,2) -- (11,2) -- (12,2);
\draw (0,4) node {Task 2} -- (1,4) -- (2,4) -- (3,4) -- (4,4) -- (5,4) -- (6,4) -- (7,4) -- (8,4) -- (9,4) -- (10,4) -- (11,4) -- (12,4);
\draw (0,6) node {Milestone} -- (7,6) -- (8,6) -- (9,6) -- (10,6) -- (11,6) -- (12,6);
\draw (0,8) node {Final Task} -- (12,8);
\end{tikzpicture}

2.2 Specifying Keys

Keys (sometimes called \textit{options}) modify the output from \texttt{pgfgantt}'s commands. You may specify a key in two ways: (1) Pass it to the optional argument present in each command, e.g.
\ganttbar[bar height=.6]{Task 1}{1}{2}
Locally change a key for the element(s) drawn by that command. Alternatively, specify a key by the \ganttset{⟨key=value list⟩} macro, which sets its keys within the current \TeX{} group:
\ganttset{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 \texttt{chart} consists of several \texttt{lines}, which may contain one or more \texttt{title elements} (at the top) or \texttt{chart elements} (such as bars, groups and milestones). From left to right, the chart is divided into an integer number of \texttt{time slots} that represent the basic \texttt{x-unit}.

The \texttt{ganttchart} environment draws a single Gantt chart:
\begin{ganttchart}{⟨options⟩}{⟨start tss⟩}{⟨end tss⟩}
\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{pgfgantt}/\texttt{time slot format} =⟨\texttt{format}⟩

Sets the \texttt{⟨format⟩} of time slot specifiers. A \texttt{time slot specifier} (abbreviated “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{isodate-year} – year only (\texttt{yyyy}). Such dates are automatically converted to the first day of January.
  \textit{Examples:} 2013, 2014
• **little-endian** – Gregorian little-endian, i.e. day–month–year (the common German date format). Valid day/month and month/year separators are the hyphen (-), slash (/) and period (.). If you enter a two-digit year (for example, 13 instead of 2013), it will be completed according to the value of `time slot format/base century` (see below).

  *Examples: 14-03-2013, 14/03/13, 14.3.2013*

• **middle-endian** – middle-endian, i.e. month–day–year (the common US date format). For valid separators and automatic year completion, see **little-endian**.

  *Examples: 03-14-2013, 03/14/13, 3.14.2013*

• **big-endian** – Gregorian big-endian, i.e. year–month–day (the ISO-standard order). For valid separators and automatic year completion, see **little-endian**.

  *Examples: 2013-03-14, 13/03/14, 2013.3.14*

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

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

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

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

2000-01-01

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

Advanced users may add their own time slot formats:

\begin{verbatim}
\newgantttimeslotformat{⟨name⟩}{⟨converter code⟩}
\end{verbatim}

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

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

```
1 \newgantttimeslotformat{stardate}{
2 \def\decomposestardate##1.##2\relax{
3 \def\stardateyear{##1}\def\stardateday{##2}
4 }\%
5 \decomposestardate#1\relax%
6 \pgfcalendardatetojulian{\stardateyear-01-01}{\stardateday}#2\%}
```
\begin{ganttchart}[hgrid, vgrid, time slot format=stardate]
\{2259.55\}{2259.67}
\gantttitlecalendar{year, month=name, day} \\
\end{ganttchart}

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

\texttt{/pgfgantt/canvas ./style=(style) \hspace{0.5cm} shape=rectangle, draw, fill=white}

The \texttt{canvas} key changes the appearance of the canvas. \texttt{(style)} is a list of TikZ keys suitable for the \texttt{(options)} of a TikZ node (such as \texttt{shape=rectangle}, \texttt{fill} or \texttt{draw}; see chapter 16 of the TikZ manual). By default, the canvas is a white rectangle with a black frame.
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.

```latex
\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}
```

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

```latex
\pgfsetgraphunit{\pgfautobecalcmultidimension[\pgfautobecalcmultidimension{1cm},\pgfautobecalcmultidimension{.6cm},\pgfautobecalcmultidimension{1.5cm}]}{\pgfautobecalcmultidimension{1cm}}{\pgfautobecalcmultidimension{.6cm}}{\pgfautobecalcmultidimension{1.5cm}}
```
Title

Bar 1

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

\begin{tabular}{|c|}
  \hline
  Title
  \hline
  Bar 1
  \hline
\end{tabular}

/pgf/gantt/hgrid [false | true | \{style list\}] \hspace{1cm} false
/pgf/gantt/hgrid style /style=\{style\} \hspace{1cm} dotted
/pgf/gantt/vgrid [false | true | \{style list\}] \hspace{1cm} false

hgrid draws a horizontal grid which starts immediately below the last title element. The key can be specified in four different ways: Firstly, \texttt{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).

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\} \{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{pgf/gantt} 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} \\\n  \ganttbar{}{1}{8} \\n\end{ganttchart}
In most situations, you can omit the multiplier \(*1\). Hence, the following style lists are equal:

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

However, if you wish to use a single style comprising two or more keys for all grid lines, e.g. \texttt{red, dotted}, you \textit{must} retain the multiplier (i.e., \{*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.
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 `\texttt{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 `\texttt{style}` denoted by `today rule`. The node that contains the `\texttt{text}` given by `today label` appears below the rule. It is formatted by `today label font` and `today label node`.
2.4 Line Breaks between Chart Elements

pgf\texttt{gantt} does not automatically begin a new line after finishing a chart element. Instead, you must insert an explicit line break with \texttt{\ganttnewline}.

\texttt{/pgf\texttt{gantt}/newline \texttt{shortcut} =\texttt{boolean}} \hspace{1cm} \texttt{true}

If true, \texttt{\textbackslash} is defined as a shortcut for \texttt{\ganttnewline} within a \texttt{ganttchart} environment, so that the syntax is reminiscent of \LaTeX{}’s \texttt{tabular} environment.
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{\ganttalignnewline} for breaking lines in the node text.

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

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

Even if you prefer a canvas without a horizontal grid, you may nevertheless want to
separate certain lines by a grid rule. For this purpose, specify the optional argument
\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.

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

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

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

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

\begin{ganttchart}[hgrid, vgrid]{1}{12}
  \gantttitlelist{(options)}{(pgf for list)}{(length of each element)}
\end{ganttchart}
This macro generates one title element for each element of the \textit{pgffor list}. The second mandatory argument specifies the \textit{length of each element}. Refer to section 56 of the TikZ manual for the detailed syntax for the \textit{pgffor list}.

A simple application is to draw twelve title elements that contain the numbers from 1 to 12. The \textit{pgffor list} is $1, \ldots, 12$.

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

![Gantt chart](image)

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

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

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

```
/pgf/gantt/title list options =\textit{pgffor options} \quad \textit{var} = \textit{x}, \textit{evaluate} = \textit{x}
```

Changes the \textit{pgffor options} of the \texttt{foreach} command called by \texttt{gantttitlelist} (see section 56 of the TikZ manual). The macro that yields the labels to be printed by \texttt{gantttitlelist} must be called \texttt{x}.

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

![Gantt chart](image)

```
\begin{ganttchart}[hgrid, vgrid, x unit=1cm]{1}{7}
\gantttitlelist[
  title list options=\%
  \{var=\y, evaluate=\y as \x\%
  using "\texttt{pgfcalendarweekdayshortname}{\y}"\}]
{0,...,6}{1} \\
\ganttbar{}{1}{4} \\
\ganttbar{}{6}{7}
\end{ganttchart}
```
While you actually may build any chart title with the two commands described previously, \texttt{\gantttitlecalendar} saves a lot of time when you wish to create elaborate calendars:

\begin{verbatim}
\gantttitlecalendar\(\langle\text{options}\rangle\}{\langle\text{calendar lines}\rangle}
\end{verbatim}

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

\begin{verbatim}
\gantttitlecalendar*\(\langle\text{options}\rangle\){\langle\text{start tss}\rangle}{\langle\text{end tss}\rangle}{\langle\text{calendar lines}\rangle}
\end{verbatim}

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

<table>
<thead>
<tr>
<th>Line type</th>
<th>\texttt{\langle\text{output format}\rangle}</th>
<th>Example output</th>
</tr>
</thead>
<tbody>
<tr>
<td>decade</td>
<td>n/a</td>
<td>2000s, 2010s, ...</td>
</tr>
<tr>
<td>year</td>
<td>n/a</td>
<td>2012, 2013, ...</td>
</tr>
<tr>
<td>month</td>
<td>\texttt{\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>shortname</td>
<td>Jan, Feb, ...</td>
</tr>
<tr>
<td>week</td>
<td>\texttt{\langle\text{number}\rangle}</td>
<td>n/a</td>
</tr>
<tr>
<td>weekday</td>
<td>\texttt{\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>shortname</td>
<td>Mon, Tue, ...</td>
</tr>
<tr>
<td>day</td>
<td>n/a</td>
<td>01, 02, ..., 31</td>
</tr>
</tbody>
</table>

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

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

\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 \(\text{format}\), four additional macros are available: \currentweek is the current week number; \startyear, \currentweek \startmonth and \currentweek \startday expand to the year, month and day of the current week’s Monday.

\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 `time slot unit=month`, one month corresponds to one time slot. Consequently, in such calendars only year and month are sensible line types for \texttt{gantttitlecalendar}, and the time slot format \texttt{isodate-yearmonth} is especially suited.

\begin{ganttchart}{2012-03}{2014-1}
\gantttitlecalendar{year, month} \\
\ganttbar{}{2012-05}{2013-01}
\end{ganttchart}

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

\begin{ganttchart}{2012-03}{2014-1}
\hgrid, \vgrid, \xunit=7.5mm, \\
\time slot format=\texttt{isodate-year}, \time slot unit=year
\end{ganttchart}
\{2007\}–\{2020\}
\texttt{\textbackslash gantttitlecalendar\{decade, year\} \textbackslash \texttt{\textbackslash ganttbar\{\{2008\}\{2018\}\textbackslash\texttt{\textbackslash end\{ganttchart\}}

\begin{tabular}{|c|c|c|}
\hline
2000s & 2010s & 2020s \\
\hline
2007 & 2008 & 2009 \\
2010 & 2011 & 2012 \\
2013 & 2014 & 2015 \\
2016 & 2017 & 2018 \\
2019 & 2020 & \\
\hline
\end{tabular}

\texttt{\textbackslash pgfgantt/title \textbackslash .style=\{style\}}
\texttt{\textbackslash \textbackslash shape=rectangle, inner sep=0pt, draw, fill=white}

Sets the appearance of a title element.

\texttt{\textbackslash \usetikzlibrary{shadows}}
\texttt{\textbackslash \usetikzlibrary{shadings}}

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

\textbf{First week}

\begin{tikzpicture}
\node[fill=blue!20, rounded corners=2mm, drop shadow] at (0,0) {First week};
\end{tikzpicture}

\begin{tikzpicture}
\node at (0,0) {1 2 3 4 5 6 7};
\end{tikzpicture}

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

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

\begin{itemize}
  \item Start time slot (1)
  \item End time slot (6)
\end{itemize}

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

The figure below shows a Gantt chart with two lines and one (large) time slot and indicates the distances modified by these keys.
For example, you might devise a layout where the title element does not touch the borders of the start and end time slot.

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

//pgf/gantt/include title in canvas = ⟨boolean⟩

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

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

\texttt{\textbackslash ganttvrule}\[⟨\text{options}⟩\{⟨\text{label}⟩\}\{⟨\text{tss}⟩\}]

This macro draws a \textit{⟨label⟩ed} vertical rule at the given \textit{⟨tss⟩}.

\begin{verbatim}
/pgf/gantt/vrule offset =⟨number⟩
/pgf/gantt/vrule/.style=⟨style⟩ \textit{dashed, line width=1pt}
/pgf/gantt/vrule label font =⟨font commands⟩ \textit{\normalfont}
/pgf/gantt/vrule label node/.style = ⟨style⟩
\end{verbatim}

\textit{vrule offset} determines the exact \textit{y}-coordinate in the time slot and should lie between 0.0 (left border) and 1.0 (right border). The vertical rule appears in the \textit{⟨style⟩} denoted by \texttt{vrule}. The label is formatted by \texttt{vrule label font} and \texttt{vrule label node}.
2.7 Predefined Chart Elements

\texttt{pgfgantt} predefines three chart elements:

1. \textit{Bars} indicate the duration of a task or one of its parts.
   \begin{verbatim}
   \ganttbar[(options)\{\langle label \rangle\}\{\langle start tss \rangle\}\{\langle end tss \rangle\}]
   \end{verbatim}

2. \textit{Groups} combine several subtasks (represented by bars) into a single task.
   \begin{verbatim}
   \ganttgroup[(options)\{\langle label \rangle\}\{\langle start tss \rangle\}\{\langle end tss \rangle\}]
   \end{verbatim}

3. \textit{Milestones} signify that an important task has been completed or that a crucial goal has been reached.
   \begin{verbatim}
   \ganttmilestone[(options)\{\langle label \rangle\}\{\langle tss \rangle\}]
   \end{verbatim}

Each of these macros draws a \langle label \rangle ed chart element from the \langle start tss \rangle to the \langle end tss \rangle (or at the given \langle tss \rangle in case of \texttt{ganttmilestone}).

\begin{verbatim}
\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}
\end{verbatim}

24
For each predefined chart element, there is also a macro that additionally draws a link from the previous element. Otherwise, these macros work exactly like the standard versions:

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

In the following example, the code on the left is equivalent to the code on the right.
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 $x$-coordinate.

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

### 2.7.1 Options: Chart Element Appearance

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

```latex
\pgfkeys{/pgf/gantt/bar/.style={shape=ganttbar, inner sep=0pt, draw, fill=white}}
```
Determines the appearance of the chart element. The shapes ganttbar, ganttgroup and ganttmilestone are described below.

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

2.7.2 Options: Label Formatting

\begin{tabular}{ll}
/pgfgrantt/bar label text \text{=}(\text{text}) & \strut#1 \\
/pgfgrantt/group label text \text{=}(\text{text}) & \strut#1 \\
/pgfgrantt/milestone label text \text{=}(\text{text}) & \strut#1 \\
/pgfgrantt/bar label font \text{=}(\text{font commands}) & \normalsize \\
/pgfgrantt/group label font \text{=}(\text{font commands}) & \bfseries \\
/pgfgrantt/milestone label font \text{=}(\text{font commands}) & \itshape \\
\end{tabular}
The \textit{label text} keys configure the label \textit{⟨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 \texttt{\ganttbar} etc. The \texttt{\strut} in the standard value ensures equal vertical spacing of the labels. The \texttt{⟨font commands⟩} of \textit{label font} and the \texttt{⟨options⟩} of \textit{label node} are applied to the label node at the left border of the chart (see \texttt{inline} below).

\begin{verbatim}
\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 !!!}]
  \gantttitle{Title}{12} \\
  \ganttgroup{Group 1}{1}{10} \\
  \ganttbar{Task 1}{1}{3} \\
  \ganttbar{Task 2}{4}{10} \\
  \ganttmilestone{Milestone 1}{11}
\end{ganttchart}
\end{verbatim}
If two or more chart elements appear in a single line, their labels will overlap at the left border of the chart. Thus, you can place the label adjacent to a chart element by setting the boolean key `inline` to `true`. This key instructs the package to draw the label node at the ... inline label anchor of the respective chart element and apply the ⟨options⟩ given by ... inline label node.

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

2.7.3 Options: Chart Element Positioning

```latex
/pgfgantt/bar left shift =⟨factor⟩
```

0
Shift the coordinates of a chart element’s borders (... shift) and change its height (... height).

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

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

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

Change the appearance of the peaks at both ends of a group. By default, both the left and right peak are 0.4 units wide and 0.1 units high, their tips lie between the peak sides. The group peaks ... keys set the dimensions for both peaks simultaneously. The figure below exemplifies the keys that apply to the left peak.
For example, you might devise the following layout: Bars are small and rounded; they do not touch the borders of their start and end time slots. Groups stay within the start and end time slot, and the peaks are more acute.

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

2.7.4 Options: Progress

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

\texttt{/pgfgantt/progress =none | today | (number)}

\texttt{none}
Indicates that a chart element is \textit{(number)} percent complete. The value \texttt{none} turns progress calculations off.

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

<table>
<thead>
<tr>
<th>Group 1</th>
<th>45% complete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subtask 1</td>
<td>100% complete</td>
</tr>
<tr>
<td>Subtask 2</td>
<td>37% complete</td>
</tr>
<tr>
<td>Subtask 3</td>
<td></td>
</tr>
</tbody>
</table>

The value \texttt{today} instructs \texttt{pgf\texttt{gantt}} 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

\begin{equation}
P = \frac{T - S}{E - S} \times 100\%
\end{equation}

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

```latex
\begin{ganttchart}[
  vgrid, hgrid, time slot format=isodate, today=2013-04-06, progress=today, bar/.append style={fill=green}, bar incomplete/.append style={fill=red}, group incomplete/.append style={draw=black,fill=none} ]{2013-04-01}{2013-04-12}
\gantttitle{Title}{12} \\
\ganttgroup{Group 1}{2013-04-01}{2013-04-12} \\
\gantttbar{Subtask 1}{2013-04-01}{2013-04-03} \\
\gantttbar{Subtask 2}{2013-04-04}{2013-04-08} \\
\gantttbar{Subtask 3}{2013-04-09}{2013-04-12}
\end{ganttchart}
```
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.
\ganttchart[\textwidth]{Group 1}{}{10} \\ 
\ganttbar[ 
  bar progress label font={\color{green!25!black}\sffamily} 
]{Subtask 1}{}{3} \\ 
\ganttbar[ 
  progress label text={$\displaystyle\frac{#1}{100}$} 
]{Subtask 2}{}{12} 
\end{ganttchart}

\[-55pt\]

\begin{itemize}
  \item \textbf{2.7.5 New Node Shapes}
\end{itemize}

\texttt{pgfgantt} defines three new node shapes:

1. The \texttt{ganttbar} node shape derives from shape \texttt{rectangle} (section 48.2 of the \TeX{} manual). It provides four additional anchors: \texttt{on top}, \texttt{on bottom}, \texttt{on left} and \texttt{on right}. The \textit{\texttt{\langle fraction\rangle}} set by the following keys indicates a position between the left and right (for \texttt{on top} and \texttt{on bottom}) or upper and lower border (for \texttt{on left} and \texttt{on right}), similarly to the \texttt{/tikz/pos key}.

\begin{verbatim}
/pgfgantt/on top fraction = \langle fraction \rangle 0.5
/pgfgantt/on bottom fraction = \langle fraction \rangle 0.5
/pgfgantt/on left fraction = \langle fraction \rangle 0.5
/pgfgantt/on right fraction = \langle fraction \rangle 0.5
\end{verbatim}

In the following figures, \texttt{on top/bottom fraction} is 0.7, whereas \texttt{on left/right fraction} is 0.1.
(2) The \texttt{ganttgroup} node shape also derives from shape \texttt{rectangle}. It provides the additional anchors \texttt{on top}, \texttt{on bottom}, \texttt{on left}, \texttt{on right} (same as above), \texttt{left peak} and \texttt{right peak}.

(3) The \texttt{ganttmilestone} node shape derives from shape \texttt{diamond} (section 48.3 of the Ti\kern0pt Z manual), but does not consider any aspect ratio. It provides the additional anchors \texttt{on top}, \texttt{on bottom}, \texttt{on left} and \texttt{on right} (same as above).

\section*{2.8 Defining Custom Chart Elements}

You may define completely new chart elements with
newganttchartelement\(\langle\text{name}\rangle\}\{\langle\text{new default key values}\rangle\}
newganttchartelement*\(\langle\text{name}\rangle\}\{\langle\text{new default key values}\rangle\}

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

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

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

<table>
<thead>
<tr>
<th>Key</th>
<th>Default value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Style keys</strong></td>
<td></td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>shape=rectangle, inner sep=0pt, draw, fill=white</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>/pgfgantt/⟨name⟩, fill=black!25</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>anchor=east, font=\texttt{\textbackslash ganttvalueof{⟨name⟩ label font}}</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>anchor=center, font=\texttt{\textbackslash ganttvalueof{⟨name⟩ label font}}</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>anchor=west, font=\texttt{\textbackslash ganttvalueof{⟨name⟩ progress label font}}</td>
</tr>
<tr>
<td><strong>Value-storing keys</strong></td>
<td></td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>\texttt{\textnormalsize}</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>center</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>east</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>\texttt{\textscriptsize}</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>0</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>0</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>.3</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>.4</td>
</tr>
<tr>
<td>\langle name \rangle</td>
<td>\texttt{\texttt{\textbackslash strut#1}}</td>
</tr>
</tbody>
</table>

Consequently, a new chart element will look like the standard \(\text{\texttt{\textbackslash ganttbar}}\) unless you introduce some \(\langle\text{new default key values}\rangle\).

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,
2.9 Links

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

\begin{ganttchart}[
\vgrid,
progress=today,
progress label text=\relax,
today=6
]{1}{12}
\gantttitlecalendar{day} \\[grid\]
\ganttfoobar{Foobar 1}{1}{2} \\ 
\ganttfoobar{Foobar 2}{3}{7} \\ 
\ganttlinkedfoobar{Foobar 3}{9}{12}
\end{ganttchart}
The \texttt{\textbackslash ganttlink} macro connects two elements, which are specified by their \texttt{name}s. 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[\texttt{name=b1}] {Task 1}{1}{4} \ganttbar[\texttt{name=b2}] {Task 2}{5}{7} \ganttbar[\texttt{name=xyz}] {Task 3}{10}{12} \ganttlink{b1}{b2} \ganttlink{b2}{xyz} \end{ganttchart}

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

\begin{ganttchart}[
  vgrid, hgrid, \texttt{\textbackslash 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}

/\texttt{pgfgantt/name} = (name)

Sets the appearance of the link.

Set the \texttt{name} key before you specify \texttt{\textbackslash link} to give a name to each chart element. 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[\texttt{name=b1}] {Task 1}{1}{4} \ganttbar[\texttt{name=b2}] {Task 2}{5}{7} \ganttbar[\texttt{name=xyz}] {Task 3}{10}{12} \ganttlink{b1}{b2} \ganttlink{b2}{xyz} \end{ganttchart}

Set the appearance of the link.

/\texttt{pgfgantt/link} \texttt{/style=(style)} \texttt{-latex, rounded corners=1pt}

The \texttt{\textbackslash ganttlink} macro connects two elements, which are specified by their \texttt{name}s. 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.
Task 1

Task 2

\begin{ganttchart}[vgrid, hgrid, link mid=.25, link bulge=1.3]{1}{12}
\gantttitle{Title}{12} \\
ganttbar{Task 1}{1}{4} \\
ganttbar{Task 2}{5}{7} \\
ganttbar{Task 3}{10}{12} \\
ganttlink{elem0}{elem1} \\
ganttlink[link mid=.8]{elem1}{elem2}
\end{ganttchart}
2. **Straight links** are only meant for connecting two bars in order to establish start-to-finish relations (s-f), start-to-start relations (s-s) etc. Their \textit{(type)} identifiers are reminiscent of the syntax for specifying arrow tips in Ti\textit{k}Z: Each identifier is composed of two letters separated by a hyphen.

\begin{verbatim}
\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}
\end{verbatim}

3. **Custom links** allow you to define completely new link types. Strictly speaking, automatic and straight links are predefined custom links whose code supports the keys mentioned above (section 3.11 presents the Ti\textit{k}Z code of these links).
For instance, \texttt{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} \gantttitle{Title}{15} \ganttbar{A really long label}{1}{3} \ganttbar{Another really long label}{10}{12} \ganttbar{Task 3}{4}{6} \ganttbar{Task 4}{13}{15} \ganttlink[link/.append style=red]{elem0}{elem2} \ganttlink[link/.append style=green, link type=dr]{elem1}{elem3} \end{ganttchart}

\begin{tabular}{|c|c|}
\hline
Title & \hline
A really long label & Another really long label \\
\hline
Task 3 & Task 4 \\
\hline
\end{tabular}

The following macro creates new link types:

\texttt{\newganttlinktype\{\textit{type}\}\{\textit{TikZ code}\}}

It defines a new link \textit{type} which is drawn by the given \textit{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 \texttt{\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, \texttt{\ganttsetstartanchor\{\textit{anchor}\}} and \texttt{\ganttsetendanchor\{\textit{anchor}\}} select an \textit{anchor} of the start and end element, respectively. See the figures in section 2.7.5 for possible \textit{anchor}s of the default chart element shapes. You may specify a certain \textit{fraction} for anchors like \texttt{on top} by \texttt{\ganttsetstartanchor\{\textit{anchor}\}\{on top=\textit{fraction}\}}. \texttt{pgfgantt} sets the default anchors to \texttt{\ganttsetstartanchor\{east\}} and \texttt{\ganttsetendanchor\{west\}}, so you even may omit these two commands.

- The two macro pairs \texttt{\xLeft/\yUpper} and \texttt{\xRight/\yLower} provide the \textit{x}- and \textit{y}-coordinates of the link start and end points, respectively.
\begin{itemize}
\item \texttt{\textbackslash ganttlinklabel} contains the label that you may assign to each link type via \texttt{\textbackslash setganttlinklabel} or the \texttt{link label} key (see below).
\item You can access any values stored in the package’s \texttt{\langle key\rangle}s with the macro \texttt{\textbackslash ganttvalueof\{\langle key\rangle\}}.
\item Remember that you can use the style \texttt{/pgfgantt/link} to ensure a uniform appearance of all your link types.
\end{itemize}

\begin{Verbatim}
\newganttlinktypealias\{\textit{new type}\}\{\textit{existing type}\}
\end{Verbatim}

\texttt{\textbackslash newganttlinktypealias} lets a \textit{\langle new type\rangle} equal an \textit{\langle existing type\rangle}, also copying any label that has been set for the \textit{\langle existing type\rangle}.

\begin{Verbatim}
\setganttlinklabel\{\langle type\rangle\}\{\langle label\rangle\}
\end{Verbatim}

\texttt{\textbackslash setganttlinklabel} sets a \textit{\langle label\rangle} for the given \textit{\langle type\rangle}. In the following example, note how \texttt{sta-to-sta} and \texttt{s-s} share a common label, while we change the label of \texttt{fin-to-fin}.

\begin{Verbatim}
\newganttlinktypealias\{\textit{sta-to-sta}\}\{\textit{s-s}\}
\newganttlinktypealias\{\textit{fin-to-fin}\}\{\textit{f-f}\}
\setganttlinklabel\{\textit{fin-to-fin}\}\{\textit{f2f}\}

\begin{ganttchart}[vgrid, hgrid]{1}{12}
\gantttitle\{Title\}{12}\\
\ganttbar\{Task 1\}\{2\}\{3\}\\
\ganttbar\{Task 2\}\{2\}\{8\}\\
\ganttbar\{Task 3\}\{6\}\{8\}\\
\ganttlink\{link type=sta-to-sta\}\{elem0\}\{elem1\}\\
\ganttlink\{link type=fin-to-fin\}\{elem1\}\{elem2\}
\end{ganttchart}
\end{Verbatim}
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}{
\ganttsetstartanchor{on bottom=0.75}
\ganttsetendanchor{on left}
\draw [/pgfgantt/link]
% first segment (down)
(\xLeft, \yUpper) --
% second segment (right)
(\xLeft, \yUpper - \ganttvalueof{link bulge} * \ganttvalueof{y unit chart}) --
% link label
node [pos=.5, /pgfgantt/link label anchor] {\ganttlinklabel}
% third segment (up)
($(\xLeft,
\yUpper - \ganttvalueof{link bulge} * \ganttvalueof{y unit chart})!
\ganttvalueof{link mid}!%
(\xRight,
\yUpper - \ganttvalueof{link bulge} * \ganttvalueof{y unit chart}))$) --
% last segment (right again)
($(\xLeft, \yLower)!
\ganttvalueof{link mid}!%
(\xRight, \yLower)$) --
(\xRight, \yLower);
\setganttlinklabel{drur}{a fancy link}
\begin{ganttchart}{1}{12}
\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}
\ganttbar[inline]{Task 2}{8}{11} \n\ganttlink[link type=drur]{elem0}{elem1}
\end{ganttchart}

<table>
<thead>
<tr>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Diagram" /></td>
</tr>
</tbody>
</table>

46
The link label key locally overrides any label specified by \setganttlinklabel. The \textit{font commands} and \textit{options} are applied to the link label node. By default, the label appears to the right of the straight link’s center.

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

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

\begin{ganttchart}[
y unit title=0.4cm, 
y unit chart=0.5cm, 
vgrid, 
time slot format=isodate-yearmonth, 
time slot unit=month, 
title/.append style={draw=none, fill=RoyalBlue!50!black}, 
title label font=\sffamily\bfseries\color{white}, 
title label node/.append style={below=-1.6ex}, 
title left shift=.05, 
title right shift=-.05, 
title height=1, 
bar/.append style={draw=none, fill=OliveGreen!75}, 
bar height=.6, 
bar label font=\normalsize\color{black!50}, 
group right shift=0, 
group top shift=.6, 
group height=.3, 
group peaks height=.2, 
bar incomplete/.append style={fill=Maroon} 
]{2010-09}{2011-12} 
\gantttitlecalendar{year} \\
\ganttbar[
progress=100, 
bar progress label font=\small\color{OliveGreen!75}, 
bar progress label node/.append style={right=4pt}, 
bar label font=\normalsize\color{OliveGreen}, 
name=pp 
]{Preliminary Project}{2010-09}{2010-12} \\
\end{ganttchart}
The second example demonstrates that \texttt{pgf\texttt{\_}gantt} 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 \url{http://en.wikipedia.org/wiki/Gantt_chart}.)

\begin{ganttchart}
\definecolor{barblue}{RGB}{153,204,254}
\definecolor{groupblue}{RGB}{51,102,254}
\definecolor{linkred}{RGB}{165,0,33}
\renewcommand\sfdefault{phv}
\renewcommand\mddefault{mc}
\renewcommand\bfdefault{bc}
\setganttlinklabel{s-s}{START-TO-START}
\setganttlinklabel{f-s}{FINISH-TO-START}
\setganttlinklabel{f-f}{FINISH-TO-FINISH}
\sffamily
3 Implementation

3.1 Packages

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

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

\RequirePackage{pgfcalendar}

3.2 Macros for Key and Error Management

\@gtt@ifstar reimplements the \LaTeX\ kernel’s \ifstar macro. This makes it robust to amsgen’s reimplementation of \ifstar.

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

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

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

The following auxiliary macros save us some code when we devise keys later on. \@gtt@keydef

\@gtt@keydef{\langle key \rangle}{\langle initial value \rangle} declares the key /pgfgantt/\langle key \rangle and stores its \langle initial value \rangle.

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

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

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

\@gtt@PackageError\@gtt@PackageError{\langle message \rangle} and \@gtt@PackageWarning{\langle message \rangle} issue a \@gtt@PackageError package error or warning \langle message \rangle, respectively.

\def\@gtt@PackageError\@gtt@PackageError{\PackageError{}}

\def\@gtt@PackageWarning\@gtt@PackageWarning{\PackageWarning{}}

3.3 The Horizontal and Vertical Grid

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

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

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

\hgridstyle
\ifgtt@hgrid
\gtt@hgridstyle

\gtt@hgrid@do macro decomposes the style list for the horizontal grid into its comma-separated items. Each item is analyzed (see below) only if some grid lines are still left to draw. Note the “elegant” quadruple \expandafter construction, which enables tail recursion.
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{\ifnum\gtt@currgrid<\gtt@currentline\relax\expandafter\expandafter\expandafter\gtt@hgrid@draw\fi}

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

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

\texttt{gantt@smugglecount}(\texttt{\{count\}}) smuggling the local value of a count register over the end of a \LaTeX\ group.

\texttt{gantt@smugglecount}\{\texttt{\{count 1\}\{count 2\}}\} converts the Julian date stored in \texttt{\{count 1\}} to a time slot and stores the latter in \texttt{\{count 2\}}. This macro is called after the start of Gantt chart. Thus, \texttt{gantt@startyear}, \texttt{gantt@startmonth} and \texttt{gantt@startjulian} (see section 3.5) have already been initialized. Depending on the value of \texttt{time slot unit}, one time slot corresponds to one \texttt{day}, one \texttt{month} or one \texttt{year}. 
\newcommand\gtt@juliantotimeslot[2]{$\begingroup$
@tempcnta=#1\relax
@ifgtt@timeslotunit@day
  \advance@tempcnta by-\gtt@startjulian\relax
  \advance@tempcnta by1\relax
\fi
@ifgtt@timeslotunit@month
  \pgfcalendarjuliantodate{@tempcnta}{@tempa}{@tempb}{@tempc}%
  \@tempcnta=@tempa\relax
  \advance@tempcnta by-\gtt@startyear\relax
  \multiply@tempcnta by12\relax
  \advance@tempcnta by@tempb\relax
  \advance@tempcnta by-\gtt@startmonth\relax
  \advance@tempcnta by1\relax
\fi
#2=@tempcnta\relax
@gtt@smugglecount#2$
\endgroup}

\newgantttimeslotformat{langle}{rangle}{langle}{rangle} defines the macro \gtt@tsstojulian{langle}{rangle}{langle}{rangle}. This macro executes \langle code \rangle (within a group), which should convert \langle tss \rangle to a Julian date and store the date in \langle count \rangle.

\newcommand\newgantttimeslotformat[2]{$\expandafter\def\csname gtt@tsstojulian@#1\endcsname##1##2{
  \begingroup##1\gtt@smugglecount##2\endgroup}$

The predefined time slot formats \texttt{simple}, \texttt{isodate}, \texttt{isodate-yearmonth} and \texttt{isodate-year} are straight forward.

\newgantttimeslotformat{simple}{$#2=#1\relax$
  \advance#2 by@gtt@tsf@startjulian\relax
  \advance#2 by1\relax$
}

\newgantttimeslotformat{isodate}{$\pgfcalendarjuliantodate{#1}{#2}$
\newgantt{isodate-yearmonth}{\pgfcalendar\#1-01}{\#2}
\newgantt{isodate-year}{\pgfcalendar\#1-01-01}{\#2}

\gtt@tsf@getdmy\langle\date\rangle decomposes a \langle\date\rangle day[sep]month[sep]year (with \langlesep\rangle representing a period, hyphen or slash) into day, month and year and stores these numbers in \local@day, \local@month and \local@year, respectively.

\newcommand\gtt@tsf@getdmy[1]{
  \edef\local@firstarg{#1}
  \def\local@decompose##1.##2.##3\relax{\def\local@day{##1}\def\local@month{##2}\def\local@year{##3}}
  \expandafter\local@decompose\local@firstarg..\relax
  \ifx\local@month\@empty
    \def\local@decompose##1/##2/##3\relax{\def\local@day{##1}\def\local@month{##2}\def\local@year{##3}}
    \expandafter\local@decompose\local@firstarg//\relax
    \ifx\local@month\@empty
      \def\local@decompose##1--\relax{\def\local@year{##1}}
      \expandafter\local@decompose\local@year
    \else
      \def\local@decompose##1-##2-##3\relax{\def\local@year{##1}}
      \expandafter\local@decompose\local@year
    \fi
  \else
    \def\local@decompose##1..\relax{\def\local@year{##1}}
    \expandafter\local@decompose\local@year
  \fi
}

Time slot formats little-endian, big-endian and middle-endian only differ in their call of \pgfcalendar\#1-01-01. 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 \texttt{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%
\pgfcalendar.dateto julian{\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%
\pgfcalendar.dateto julian{\local@day-\local@month-\local@year}{#2}%
\}
\newgantttimeslotformat{middle-endian}{\gtt@tsf@getdmy{#1}}%
\ifnum\local@year<100\relax%
  \edef\local@year{\gtt@tsf@basecentury\local@year}%
\fi%
\pgfcalendar.dateto julian{\local@year-\local@day-\local@month}{#2}%

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⟩}.

\ganttset{\texttt{time slot format/.code=}%,
\@ifundefined{\gtt@tsstojulian@#1}{\@gtt@PackageError{\texttt{Time slot format ‘#1’ undefined.}}}{},%\expandafter\let\expandafter\gtt@tsstojulian\csname\gtt@tsstojulian@#1\endcsname%,
\texttt{time slot format=simple}},%
\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{verbatim}
\newgantttimeslotformat{little-endian}{\gtt@tsf@getdmy{#1}}%
\ifnum\local@year<100\relax%
  \edef\local@year{\gtt@tsf@basecentury\local@year}%
\fi%
\pgfcalendar.dateto julian{\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%
\pgfcalendar.dateto julian{\local@day-\local@month-\local@year}{#2}%
\}
\newgantttimeslotformat{middle-endian}{\gtt@tsf@getdmy{#1}}%
\ifnum\local@year<100\relax%
  \edef\local@year{\gtt@tsf@basecentury\local@year}%
\fi%
\pgfcalendar.dateto julian{\local@year-\local@day-\local@month}{#2}%

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

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

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

3.5 The Main Environment

Keys that store the basis vectors of the chart.

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

Keys related to the canvas and the today rule.

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

Boolean key that determines if \texttt{\textbackslash \textbackslash} is equivalent to \texttt{\textbackslash ganttnewline}.

\ifgtt@newlineshortcut
\texttt{\textbackslash ganttnewline}%
\newif\ifgtt@newlineshortcut
\ifgtt@newlineshortcut%
\newline shortcut = \texttt{\textbackslash ganttnewline}
\newline shortcut/.is if=gtt@newlineshortcut,%
\newline shortcut=true%
\}

The boolean \texttt{\ifgtt@tikzpicture} is true if a Gantt chart appears within a Ti\textit{k}Z 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 \textit{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

\textit{canvas} \textit{expand chart} \textit{today} \textit{today offset} \textit{today rule} \textit{today label} \textit{today label font} \textit{today label node} \textit{newline shortcut}
rule. \texttt{\gantt@startjulian} and \texttt{\gantt@endjulian} contain the Julian dates corresponding to the first and last time slot, respectively. \texttt{\gantt@chartid} assigns a consecutive number to each chart.

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

\begin{verbatim}
\def\ganttchartextrasize#1#2{\gdef\@\ganttchart#1@extrasize{#2}}
\end{verbatim}

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, ...

\begin{verbatim}
\newenvironment{ganttchart}[3][]{\ganttset{#1}\
gantt@tsstojulian{#2}{\gantt@startjulian}\
\global\gantt@startjulian=\gantt@startjulian\relax\
\gantt@tsstojulian{#3}{\gantt@endjulian}\
\global\gantt@endjulian=\gantt@endjulian\relax\
\pgfcalendarjuliantodate{\gantt@startjulian}{\gantt@startyear}{\gantt@startmonth}{\@tempa}\
\xdef\gantt@startyear{\gantt@startyear}\
\xdef\gantt@startmonth{\gantt@startmonth}{\@tempa}\
\xdef\gantt@chartwidth{\gantt@chartwidth}{\gantt@chartwidth\relax\

... the time slot of the today rule, ....
\end{verbatim}

\begin{verbatim}
\edef\@tempa{\ganttvalueof{today}}\
\edef\@tempb{\ganttvalueof{today}}\
\ifx\@tempa\@tempb\else\
\gantt@tsstojulian{\ganttvalueof{today}}{\gantt@today@slot}\
\gantt@juliantotimeslot{\gantt@today@slot}{\gantt@today@slot}\
\fi\
\global\gantt@elementid=0\relax
\end{verbatim}

60
If `expand chart` contains a value different from `none`, scale the chart so that its $x$-extent equals this value. To this end, use the information stored in the auxiliary file. `$\text{gtt@expanded@xunit}$` will contain the new value for `x unit`.

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

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

```latex
\ifundef\useasboundingbox\{%\\begin{tikzpicture}\%\gantt@intitle\%\ifgantt@newlineshortcut\%\let\\\ganttnewline\%\\let\ganttalignnewline\tikz@align@newline\%\\begin{scope}[on background layer]\%\\ifgantt@includefont\%\y@upper\%\\y@lower\%\\y@mid\%\y@size\%\begin{scope}[on background layer]\%\\ifgantt@includestyle\%\\edef\y@upper{0}\%\\else\%\\pgfmathsetmacro\y@upper{\@gtt@lasttitleline * \ganttvalueof{y unit title}}\%\\\%\\fi\%\\end{scope}\%\\end{scope}\%\\end{tikzpicture}\%\}\%
```

After the contents of the environment have been drawn, we add the canvas to the background layer. `pgfgantt` saves $x$- and $y$-coordinates in local internal macros called `$\text{x@left}$`, `$\text{x@right}$`, `$\text{x@size}$`, `$\text{y@upper}$`, `$\text{y@lower}$`, `$\text{y@mid}$` and `$\text{y@size}$`.
\pgfmathsetmacro\y@lower{\gtt@lasttitleline * \ganttvalueof{y unit title} + (\gtt@currentline - \gtt@lasttitleline - 1) * \ganttvalueof{y unit chart}}
\pgfmathsetmacro\y@mid{(\y@upper + \y@lower) / 2}
\pgfmathsetmacro\y@size{abs(\y@lower - \y@upper)}
\pgfmathsetmacro\x@size{\gtt@chartwidth * \ganttvalueof{x unit}}
\node [/pgfgantt/canvas, minimum width=\x@size pt, minimum height=\y@size pt] at (\x@size pt / 2, \y@mid pt) {};

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

\pgfmathsetmacro\y@upper{\gtt@lasttitleline * \ganttvalueof{y unit title}}
\ifgtt@vgrid
\gtt@currgrid=1\relax\global\advance\gtt@chartwidth by-1\relax\foreach \x in {1,...,\gtt@chartwidth} {%
  \expandafter\gtt@vgrid@do\gtt@vgridstyle,\relax,\ifnum\gtt@currgrid>\gtt@chartwidth\breakforeach\fi%
}\global\advance\gtt@chartwidth by1\relax\fi%

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

\ifgtt@hgrid%
\gtt@currgrid=\gtt@lasttitleline\relax\global\advance\gtt@chartwidth by-1\relax\foreach \t in {\gtt@currgrid,...,\gtt@currentline} {%
  \expandafter\gtt@hgrid@do\gtt@hgridstyle,\relax,\ifnum\gtt@currgrid<\gtt@currentline\breakforeach\fi%
}\global\advance\gtt@chartwidth by1\relax\fi%

The last task of \texttt{ganttchart} is to apply the \texttt{today} key if its value differs from \texttt{none}.

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

\pgfmathsetmacro\y@upper{\gtt@lasttitleline * \ganttvalueof{y unit title}}
\ifgtt@vgrid
\gtt@currgrid=1\relax\global\advance\gtt@chartwidth by-1\relax\foreach \x in {1,...,\gtt@chartwidth} {%
  \expandafter\gtt@vgrid@do\gtt@vgridstyle,\relax,\ifnum\gtt@currgrid>\gtt@chartwidth\breakforeach\fi%
}\global\advance\gtt@chartwidth by1\relax\fi%

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

\ifgtt@hgrid%
\gtt@currgrid=\gtt@lasttitleline\relax\global\advance\gtt@chartwidth by-1\relax\foreach \t in {\gtt@currgrid,...,\gtt@currentline} {%
  \expandafter\gtt@hgrid@do\gtt@hgridstyle,\relax,\ifnum\gtt@currgrid<\gtt@currentline\breakforeach\fi%
}\global\advance\gtt@chartwidth by1\relax\fi%

The last task of \texttt{ganttchart} is to apply the \texttt{today} key if its value differs from \texttt{none}.

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

\pgfmathsetmacro\y@upper{\gtt@lasttitleline * \ganttvalueof{y unit title}}
\ifgtt@vgrid
\gtt@currgrid=1\relax\global\advance\gtt@chartwidth by-1\relax\foreach \x in {1,...,\gtt@chartwidth} {%
  \expandafter\gtt@vgrid@do\gtt@vgridstyle,\relax,\ifnum\gtt@currgrid>\gtt@chartwidth\breakforeach\fi%
}\global\advance\gtt@chartwidth by1\relax\fi%

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

\ifgtt@hgrid%
\gtt@currgrid=\gtt@lasttitleline\relax\global\advance\gtt@chartwidth by-1\relax\foreach \t in {\gtt@currgrid,...,\gtt@currentline} {%
  \expandafter\gtt@hgrid@do\gtt@hgridstyle,\relax,\ifnum\gtt@currgrid<\gtt@currentline\breakforeach\fi%
}\global\advance\gtt@chartwidth by1\relax\fi%

The last task of \texttt{ganttchart} is to apply the \texttt{today} key if its value differs from \texttt{none}.
Store the $x$-extent of the bounding box in \@tempdima. Calculate the size by which the bounding box exceeds the “raw” canvas size. Write this information to the auxiliary file.

\pgfextractx{\@tempdima}{%}
\pgfpointdiff{\pgfpointanchor{current bounding box}{south west}}{%}
\pgfmathparse{\@tempdima - \gtt@chartwidth}%
\protected@write\@auxout{}{\string\gtt@chartextrasize{\the\gtt@chartid}{\pgfmathresult pt}}%

Increase the chart counter.
\global\advance\gtt@chartid by1\relax%

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

\if\gtt@tikzpicture\else\end{tikzpicture}\fi%

\end{scope}%

## 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{%}
\def\@tempa{grid}%
\if\local@drawarg\@empty\else%
\if\local@drawarg\@tempa%
\def\local@drawarg{\pgfgantt/hgrid style}%
\draw [/pgfgantt/today rule]
\pgfpointdiff{\pgfpointanchor{current bounding box}{south west}}{%}
\node [/pgfgantt/today label node] {\ganttvalueof{today label}};%
\fi%
\fi%
\end{scope}%

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{%}
\def\@tempa{grid}%
\if\local@drawarg\@empty\else%
\if\local@drawarg\@tempa%
\def\local@drawarg{\pgfgantt/hgrid style}%
\draw [/pgfgantt/today rule]
\pgfpointdiff{\pgfpointanchor{current bounding box}{south west}}{%}
\node [/pgfgantt/today label node] {\ganttvalueof{today label}};%
\fi%
\fi%
\end{scope}%

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{%}
\def\@tempa{grid}%
\if\local@drawarg\@empty\else%
\if\local@drawarg\@tempa%
\def\local@drawarg{\pgfgantt/hgrid style}%
\draw [/pgfgantt/today rule]
\pgfpointdiff{\pgfpointanchor{current bounding box}{south west}}{%}
\node [/pgfgantt/today label node] {\ganttvalueof{today label}};%
\fi%
\fi%
\end{scope}%
% 
\pgfmathsetmacro\y@upper{\gtt@lasttitleline * \ganttvalueof{y unit title} + (\gtt@currentline - \gtt@lasttitleline - 1) * \ganttvalueof{y unit chart}}
\expandafter\draw\expandafter[\local@drawarg]
(0pt, \y@upper pt) -- (\gtt@chartwidth * \ganttvalueof{x unit}, \y@upper pt);
\fi%
\global\advance\gtt@currentline by-1\relax%
\ifgtt@intitle\global\advance\gtt@lasttitleline by-1\relax\fi%
\global\gtt@lasttitleslot=0\relax%
\endgroup%
}

3.7 Vertical rules

Keys related to the vertical rules.

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

A count for storing the vrule time slot.

\newcount\gtt@vrule@slot

Calculate the coordinates for the vertical rule and draw it.

\newcommand\ganttvrule[3][]{%
\begingroup
\ganttset{#1}
\gtt@vrule@slot\gtt@vrule@slot
\gtt@lasttitleline * \ganttvalueof{y unit title} + (\gtt@currentline - \gtt@lasttitleline - 1) * \ganttvalueof{y unit chart})\%
}
3.8 Titles

Keys that influence title elements. The parameter token #1 in the value of \texttt{\stringtitle} \texttt{\stringlabel text} is replaced by the argument of \texttt{\stringgtt@titlelabeltext}. Note that \texttt{\string@gtt@keydef} cannot define \texttt{\stringtitle list options}, since \texttt{\string@gtt@titlelistoptions} is expanded after a \texttt{\stringforeach} statement, where \texttt{\stringganttvalueof} will not work.

\begin{verbatim}
\@gtt@keydef{\stringtitle}{\texttt{\stringshape=rectangle, inner sep=0pt, draw, fill=white}}
\@gtt@keydef{\stringtitle label font}{\texttt{\string\small}}
\@gtt@stylekeydef{\stringtitle label node}{\texttt{\stringanchor=center, font=\string\pfgantt/vrule label node \string\small}}
\ganttset{\texttt{\stringtitle label text/.code={\string\def\string\gtt@titlelabeltext##1{#1}}\texttt{\string\def\string\gtt@titlelistoptions{\string\#1}}\texttt{\string\if\string\if\string\ganttvalueof{\stringx unit}\string@mid pt\string@mid pt, \string\y@upper pt\string@lower pt\string\\node \string[/\string\pgfgantt/vrule \string\label \string\small node] \string\small \string\text{\string#2}\string\\endgroup\string\}}}}
\@gtt@keydef{\stringtitle left shift}{0}
\@gtt@keydef{\stringtitle right shift}{0}
\@gtt@keydef{\stringtitle top shift}{0}
\@gtt@keydef{\stringtitle height}{0.6}
\newif\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\if\i
\gantttitle draws a title element (i.e., a rectangle with a single node at its center). For reasons that will become clear below, the element essentially starts at the x-coordinate stored in \gtt@lasttitleslot. This count is updated at the end of the macro.

\newcommand\gantttitle[3][]{%
\begingroup%
\ganttset{#1}%
\pgfmathsetmacro\x@left{%
  (\gtt@lasttitleslot + \ganttvalueof{title left shift})%*
  \ganttvalueof{x unit}%
}%
\pgfmathsetmacro\x@right{%
  (\gtt@lasttitleslot + #3 + \ganttvalueof{title right shift})%*
  \ganttvalueof{x unit}%
}%
\pgfmathsetmacro\x@mid{%
  (\x@left + \x@right) / 2%
}%
\pgfmathsetmacro\x@size{%
  \x@right - \x@left%
}%
\pgfmathsetmacro\y@upper{%
  (\gtt@currentline - \ganttvalueof{title top shift})%*
  \ganttvalueof{y unit title}%
}%
\pgfmathsetmacro\y@lower{%
  (\gtt@currentline - \ganttvalueof{title top shift})%*
  - \ganttvalueof{title height})%*
  \ganttvalueof{y unit title}%
}%
\pgfmathsetmacro\y@mid{%
  (\y@upper + \y@lower) / 2%
}
\gantttitlelist generates title elements by repeatedly calling \gantttitle. Since \gantttitlelist\gantttitlelist\gantttitlelist does not have to calculate the respective x-coordinates explicitly.

\newcommand\gantttitlelist[3][\{}% \begin{group}\ganttset{#1}%. \expandafter\foreach\gtt@titlelistoptions in \{#2\} {\gantttitle\x{#3}}% \end{group}%
\gantttitlecalendar checks whether it is invoked in the starred or nonstarred form, \ifgtt@titlecalendarstar accordingly and then starts a command relaying chain.
\newif\ifgtt@titlecalendarstar
\newcommand\gantttitlecalendar{% \@gtt@ifstar{% \gtt@titlecalendarstartrue\@gantttitlecalendar% \} \else{% \expandafter\@@gantttitlecalendar\expandafter% \expandafter\gtt@startjulian\expandafter% \expandafter\gtt@endjulian\expandafter% \} %
\begin{group}\ganttset{#1}%. \ifgtt@titlecalendarstar% \expandafter\@@gantttitlecalendar% \else% \expandafter\@@@gantttitlecalendar\expandafter% \expandafter\gtt@startjulian\expandafter% \expandafter\gtt@endjulian\expandafter% \} %
\newcommand\@gantttitlecalendar[1][\{}% \gtt@titlecalendarstartrue\@gantttitlecalendar% \gtt@titlecalendarfalse\@gantttitlecalendar% 
\newcommand\@gantttitlecalendar[1][\{}% \begin{group}\ganttset{#1}%. \expandafter\@@gantttitlecalendar\expandafter% \expandafter\gtt@startjulian\expandafter% \expandafter\gtt@endjulian\expandafter% \} %
The second command in the relaying chain, \@@gantttitlecalendar{(start tss)\{\langle end tss\rangle)}, 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})\{\langle end Julian\rangle\}{\langle calendar lines\rangle)}, 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 \langle calendar lines\rangle, which reside in path /pgfgantt/calendar.

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

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

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

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

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

For each \langle line type \rangle, we define a corresponding key /pgfgantt/calendar/(line type). This key performs the necessary calculations and draws one or several \gantttitles. Line type decade draws decades.

\ganttset{%
\calendardecade/.code={%
  \ifganttcalendar@eol\ganttnewline\fi
  \begingroup
  \ganttcalendar@slots=1\relax
  \ifgantt@timeslotunit@year
    \pgfcalendar{\gantt@calendar@startdate}{\gantt@calendar@enddate}{%
      \ifdate{end of decade=2009-12-31}{%
        \gantttitle{\expandafter\@gtt@getdecade\pgfcalendarcurrentyear
          0s}{\the\ganttcalendar@slots}%
        \ganttcalendar@slots=0\relax%
      }{%}
      \ifdate{equals=01-01}{%
        \ifnum\pgfcalendarcurrentjulian>\pgfcalendarbeginjulian\relax%
        \advance\ganttcalendar@slots by1\relax%
        \fi
      }{%}
    }%
  }%
  \gantt@calendar@slots=0\relax
  \fi
%
}%
\endgroup
}%
Line type year draws years.

\begin{tikzpicture}[x=\textwidth, y=\textwidth]
\begin{scope}
\tt@calendar{\the\tt@calendar@slots}
\end{scope}
\end{tikzpicture}
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.
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`. `\currentweek`
Line type \texttt{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 \texttt{\pgfcalendarweekday(format)} and possibly converted.

\begin{verbatim}
\gtt@calendar@eoltrue
\endgroup
\pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
\gantttitle{\csname pgfcalendarweekday#1\endcsname{\pgfcalendarcurrentweekday}}{1}%
}%
\endgroup
\gtt@calendar@eoltrue
\end{verbatim}

Line type \texttt{day=(format)} draws days of the month.

\begin{verbatim}
\gtt@calendar@eoltrue
\pgfcalendar{}{\gtt@calendar@startdate}{\gtt@calendar@enddate}{%
\gantttitle{\pgfcalendarcurrentday}{1}%
}%
\end{verbatim}

\subsection{3.9 Chart Elements}

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

\begin{verbatim}
\ganttset{%
\gtt@progress/.code={%
\def\gtt@progress{#1}%
},%
\gtt@progress=none,%
\gtt@progress label text/.code={%
\def\gtt@progresslabeltext#1(#1)%
},%
\gtt@progress label text name
\gtt@progress label text start border
\gtt@progress label text inline%
\ifgttinline
\end{verbatim}
progress label text={%
\pgfmathprintnumber[precision=0,verbatim]{#1}\% complete%
}%
\@gtt@keydef{name}{}
\newif\ifgtt@ce@startatleftborder
\ganttset{%
chart element start border/.is choice,%
chart element start border/left/.code=\@gtt@ce@startatleftbordertrue,%,
chart element start border/right/.code=\@gtt@ce@startatleftborderfalse,%
chart element start border=left%
}
\newif\ifgtt@inline
\ganttset{%
inline/.is if=\@gtt@inline,%
inline=false%
}
\gtt@lastelement
\gtt@currentelement
\if\@gtt@draw@complete
\if\@gtt@draw@incomplete
\if\@gtt@draw@clip
\gtt@left@slot
\gtt@right@slot
\The macros \@gtt@lastelement and \@gtt@currentelement save the name of the current and last chart element drawn. Thereby, the \ganttlinked... macros can add a link connecting them. \@gtt@draw@complete, \@gtt@draw@incomplete and \@gtt@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\if\@gtt@draw@complete
\newif\if\@gtt@draw@incomplete
\newif\if\@gtt@draw@clip
\newcount\@gtt@left@slot
\newcount\@gtt@right@slot
\@gtt@chartelement[⟨options⟩]{⟨label⟩}{⟨start tss⟩}{⟨end tss⟩}{⟨type⟩}
\ganttcharterelement
is the generic command for drawing chart elements of a certain ⟨type⟩. First, \@gtt@chartelement converts ⟨start tss⟩ to \@gtt@left@slot and ⟨end tss⟩ to \@gtt@right@slot. Then it calculates the usual coordinates. \@gtt@name stores the name of the chart element.
\newcommand\@gtt@chartelement[5][]{%
\begingroup%
\ganttset{#1}%
\@gtt@sstojulian{#3}{\@gtt@left@slot}%
\@gtt@juliantotimeslot{\@gtt@left@slot}{\@gtt@left@slot}%
\@gtt@sstojulian{#4}{\@gtt@right@slot}%
\@gtt@juliantotimeslot{\@gtt@right@slot}{\@gtt@right@slot}%
\def\local@timeslotmodifier{-1}%
\if\@gtt@ce@startatleftborder else
\endgroup%
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.

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

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

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

```latex
\def\@tempa{#2}\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.

```latex
\xdef\gtt@lastelement{\gtt@currentelement}
```
\newganttchartelement checks whether it was invoked in the starred or nonstarred form and executes \newganttchartelement@one or \newganttchartelement@two, respectively.

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

\newcommand\newganttchartelement@one[1]\{\%
\expandafter\newcommand\csname gantt#1\endcsname[3]\{\%
\gtt@chartelement[##1]{##2}{##3}{##3}{#1}\%
\expandafter\newcommand\csname ganttlinked#1\endcsname[3]\{\%
\begingroup\%
\ganttset{##1}\%
\gtt@chartelement{##2}{##3}{##3}{#1}\%
\ganttlink{\gtt@lastelement}{\gtt@currentelement}\%
\endgroup\%
\}\%
\newganttchartelement@definekeys[#1]\%
\}
\newcommand\newganttchartelement@two[1]\{\%
\expandafter\newcommand\csname gantt#1\endcsname[4]\{\%
\gtt@chartelement[##1]{##2}{##3}{##4}{#1}\%
\expandafter\newcommand\csname ganttlinked#1\endcsname[4]\{\%
\begingroup\%
\ganttset{##1}\%
\gtt@chartelement{##2}{##3}{##4}{#1}\%
\ganttlink{\gtt@lastelement}{\gtt@currentelement}\%
\endgroup\%
\}\%
\newganttchartelement@definekeys[#1]\%
\}
\@newganttchartelement@definekeys{(type)\{\{key-value list\}\} introduces 14 keys for the newly generated chart element (type).

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

Code for the predefined chart element type bar.

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

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

\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.

\ganttset{%
  group peaks tip position/.code={%
    \ganttset{
      group left peak tip position=#1,%
      group right peak tip position=#1%
    }%
  },%
  group peaks width/.code={%
    \ganttset{
      group left peak width=#1,%
      group right peak width=#1%
    }%
  },%
  group peaks height/.code={%
    \ganttset{
      group left peak height=#1,%
      group right peak height=#1%
    }%
  }%}
Code for the predefined chart element type milestone.

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

3.10 Node Shapes

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

\begin{verbatim}
\@gtt@keydef{on top fraction}{.5}
\@gtt@keydef{on bottom fraction}{.5}
\@gtt@keydef{on left fraction}{.5}
\@gtt@keydef{on right fraction}{.5}
\end{verbatim}

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.

\begin{verbatim}
\pgfdeclareshape{ganttbar}{
    \inheritsavedanchors[from=rectangle]
    \inheritanchor[from=rectangle]{center}
    \inheritanchor[from=rectangle]{mid}
    \inheritanchor[from=rectangle]{base}
}
\end{verbatim}
\pgf@yb=\ganttvalueof{on right fraction}\pgf@yb
\advance\pgf@ya by\pgf@yb
\pgf@y=\pgf@ya
}
\inheritbackgroundpath[from=rectangle]

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

Code for node shape gantt milestone. Anchors and background path derive from node shape diamond. The four additional anchors on top, on bottom, on left and on right are defined.
\pgfdeclareshape{gantt milestone}{
\inheritsavedanchors[from=diamond]
\inheritanchor[from=diamond]{text}
\inheritanchor[from=diamond]{center}
\inheritanchor[from=diamond]{mid}
\inheritanchor[from=diamond]{base}
\inheritanchor[from=diamond]{north}
\inheritanchor[from=diamond]{south}
\inheritanchor[from=diamond]{west}
\inheritanchor[from=diamond]{north west}
\inheritanchor[from=diamond]{south west}
\inheritanchor[from=diamond]{east}
\inheritanchor[from=diamond]{north east}
\inheritanchor[from=diamond]{south east}
\inheritanchorborder[from=diamond]
\inheritbackgroundpath[from=diamond]
\anchor{on top}{
\pgf@process{\outernortheast}
\pgf@xa=2\pgf@x
\pgf@x=-\pgf@x
\advance\pgf@x by\ganttvalueof{on top fraction}\pgf@xa
\pgf@ya=2\pgf@y
\pgf@y=0pt
\pgfmathparse{
\ganttvalueof{on top fraction} < 0.5
? \ganttvalueof{on top fraction}
: 1 - \ganttvalueof{on top fraction}
}
\advance\pgf@y by\pgfmathresult\pgf@ya}
\anchor{on bottom}{
\pgf@process{\outernortheast}
\pgf@xa=2\pgf@x
\pgf@x=-\pgf@x
\advance\pgf@x by\ganttvalueof{on bottom fraction}\pgf@xa
\pgf@ya=-2\pgf@y
\pgf@y=0pt
\pgfmathparse{
\ganttvalueof{on bottom fraction} < 0.5
? \ganttvalueof{on bottom fraction}
: 1 - \ganttvalueof{on bottom fraction}
}
\advance\pgf@y by\pgfmathresult\pgf@ya}
\anchor{on right}{
\pgf@process{\outernortheast}
\pgf@ya=-2\pgf@y
\advance\pgf@y by\ganttvalueof{on right fraction}\pgf@ya
\pgf@xa=2\pgf@x}
3.11 Links

Keys for configuring links.

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

\newganttlinktype{(type)}{(code)} stores \textit{code} in an internal macro \texttt{\@gtt@linktype@{type}@label}, which is later called by \texttt{\gtt@drawlink}.

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

\setganttlinklabel{(type)}{(label)} stores a given \textit{label} in an internal macro \texttt{\@gtt@linktype@{type}@label}, which is later used by \texttt{\gtt@drawlink}.

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

88
\begin{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:

\begin{verbatim}
\@gtt@keydef{link mid}{.5}  
\@gtt@keydef{link bulge}{.4}  
\@gtt@keydef{link tolerance}{.6}  
\end{verbatim}

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

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

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

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

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

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

\end{verbatim}
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%
 \gtt@drawlink{rdldr}%
 \else%
 \gtt@drawlink{rdr}%
 \fi%
}\
\end{verbatim}

Once again, two possibilities arise: Either the elements to be connected are at least separated by \texttt{link tolerance} time slots, in which case we draw a three-part arrow (i.e., link type \texttt{rdr}). Alternatively, the elements lie in adjacent time slots or even overlap, in which case we draw a five-part arrow (i.e., link type \texttt{rdldr}).

\begin{verbatim}
\pgfmathparse{
 \pgfmathtruncatemacro{\gtt@drawlink}{\ifcase(\xRight - \xLeft)^\pgfmathresult\fi}%
}\
\end{verbatim}

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

\begin{verbatim}
\gtt@drawlink{r}%
}\
\end{verbatim}

The \texttt{dr} type is explained in section 2.9.
Here is the definition of the four straight link types and their labels.

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

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

\begin{verbatim}
\setganttlinklabel{s-s}{start-to-start}
\end{verbatim}

\begin{verbatim}
\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}; 
\end{verbatim}

\begin{verbatim}
\setganttlinklabel{s-f}{start-to-finish}
\end{verbatim}

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

\begin{verbatim}
\setganttlinklabel{f-s}{finish-to-start}
\end{verbatim}

\begin{verbatim}
\newganttlinktype{f-f}{% 
\ganttsetstartanchor{south east}\
\ganttsetendanchor{north east}\
\draw [/pgfgantt/link] \\
\( (\xLeft, \yUpper) \) -- \\
\( (\xRight, \yLower) \) \\
node [pos=.5, /pgfgantt/link label node] {\ganttlinklabel}; 
\end{verbatim}
\ganttlink{(link type)} first checks if the \langle link type \rangle 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 \texttt{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 \texttt{/pgf/gantt/link anchor/(anchor)} is undefined, \texttt{pgfgantt} stores \langle anchor \rangle in \texttt{/gantt/link@anchor}.

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

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

\def\@gtt@linkanchordef#1{%
  \ganttset{%
    link anchor/#1/.code=%
  }%
\edef\@gtt@link@anchor{#1}  
\ganttset{#1 fraction=##1}%
},%
link anchor/#1/.default=.5%
}%
\@gtt@linkanchordef{on top}
\@gtt@linkanchordef{on bottom}
\@gtt@linkanchordef{on left}
\@gtt@linkanchordef{on right}
\@gtt@setstartanchor{⟨anchor⟩} recalls the coordinates of the anchor stored in 
\@gtt@link@anchor from chart element \@gtt@link@startelement. It stores these 
coordinates in the auxiliary macros \xLeft and \yUpper.
\newcommand\@gtt@setstartanchor[1][]{%
\pgfqkeys{/pgfgantt/link anchor}{#1}%
\pgfpointanchor{\@gtt@link@startelement}{\@gtt@link@anchor}%
\edef\xLeft{\the\pgf@x}%
\edef\yUpper{\the\pgf@y}%
}%
\@gtt@setendanchor{⟨anchor⟩} is similar to the command above. However, it stores 
the anchor coordinates in the auxiliary macros \xRight and \yLower.
\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{⟨anchor⟩}{⟨anchortype⟩}{⟨anchor⟩} executes the ⟨anchortype⟩ and stores the names 
of the connected elements ⟨anchortype1⟩ and ⟨anchortype2⟩ in \@gtt@link@startelement and 
\@gtt@link@endelement. 
\newcommand\ganttlink[3][]{%
\begingroup%
\ganttset{}%
\def\@gtt@link@startelement{#1}%
\def\@gtt@link@endelement{#2}%
\ganttsetstartanchor{east}%
ganttsetendanchor and ganttsetstartanchor are only valid in the second argu-
ment of \newganttlinktype. Since you may wish to omit one of those commands,
we set default anchors for the link.
We call \texttt{\texttt{drawlink}} with the value of \texttt{link type}.

\texttt{\texttt{drawlink}}\texttt{(\texttt{ganttvalueof(link type)})}\%
4 Index

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

Symbols
\% .......................... 811
@@@\texttt{gantttitlecalendar} 570, 579, 582
@@\texttt{gantttitlecalendar} 568, 576
@@\texttt{gantttitlecalendar} 560, 561, 564
\@\texttt{GanttPackageError} 19, 195, 237
\@\texttt{GanttPackageWarning} 19, 318, 888, 1537
\@\texttt{currlabel} 1542
\@\texttt{currlinktype} 1535, 1545, 1552
\@\texttt{getdecade} 595, 623, 637
\@\texttt{getfourthdigit} 595, 602, 603
\@\texttt{ifstar} 8, 61, 109, 559, 976
\@\texttt{curranchor} 1555, 1563, 1576, 1583
\@\texttt{linkanchordef} 1560
\@\texttt{setendanchor} 1581, 1594
\@\texttt{setstartanchor} 1574, 1593
\@\texttt{stylekeydef} 16, 30, 265, 269, 272, 432, 434, 463, 465, 1008, 1009, 1011, 1015, 1020, 1404, 1408
\@\texttt{ganttcharlement\texttt{definkeys}} 990, 1004, 1007
\@\texttt{ganttcharlement\texttt{definone}} 976, 979
\@\texttt{ganttcharlement\texttt{dettwo}} 976, 979
\\ ................................ 12, 328

B
bar (option) ............. 26, 1036
bar height (option) .......... 30, 1036
bar incomplete (option) .... 34, 1036
bar inline label anchor (option) .......... 29, 1036
bar progress label anchor (option) .......... 35, 1036
bar progress label font (option) .... ... 35, 1036
bar progress label node (option) .... ... 35, 1036
bar right shift (option) .......... 29, 1036
bar top shift (option) .......... 29, 1036

C
calendar week text (option) ...... 18, 487
canvas (option) ............ 7, 265
chart element start border (option) .......... 26, 802
\texttt{currentweek} ........... 18, 487, 736
\texttt{endofdecade} .......... 602, 604
environments:
ganttchart ............ 5, 292
expand chart (option) .......... 8, 265

G
\texttt{ganttalignnewline} .......... 13, 328
\texttt{ganttbar} .......... 24, 1036
\texttt{ganttchart} (environment) .......... 5, 292
\texttt{ganttgroup} .......... 24, 1059
\texttt{ganttlink} 40, 987, 1001, 1588
\texttt{ganttlinkedbar} .......... 25, 1036
\texttt{ganttlinkedgroup} .......... 25, 1059
\texttt{ganttlinkedmilestone} .......... 25, 1108
\texttt{ganttlinklabel} .......... 44, 1435, 1491, 1501, 1511, 1521, 1531, 1542
\texttt{ganttbar} ............ 24, 1036
\texttt{ganttnewline} .......... 12, 328
\texttt{ganttset} .... 4, 9, 32, 80, 234, 276, 293, 325, 442, 468, 483, 491, 513, 552, 566, 614, 802, 816, 823, 838, 985, 999, 1027, 1087, 1089, 1095, 1101, 1555, 1561, 1564, 1590
\texttt{ganttnewline} .......... 12
\texttt{ganttsetendanchor} .......... 43, 1487, 1497, 1507, 1517, 1527, 1593
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.
\newganttchartelelement defines completely new chart elements.
\newganttetimeslotformat 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 timeslotformat/base century provides the century for autocompletion of two-digit years.
The key timeslotformat/start date specifies the internal date representation of digit 1 in the simple time slot format.
The key timeslotformat changes the format of time slot specifiers.
The key time slot modifier was renamed to chart element start border.
The key title label anchor was renamed to title label node.
The key today accepts a time slot specifier.
The keys bar/group/milestone label anchor were renamed to bar/group/milestone label node.
The keys bar/group/milestone label inline anchor were renamed to bar/group/milestone inline label node.
The keys bar/group/milestone label shape anchor were renamed to bar/group/milestone inline label anchor.
The keys bar/group/milestone progress label anchor were added.
The keys calendar week text and compress calendar were added.
The keys group right/left peak and group peaks were replaced by group right/left peak tip position, group peaks tip position, group right/left peak width, group peaks width, group right/left peak height and group peaks height.
The keys today offset, today label font and today label node were added.

v5.0
General: \gantttitlecalendar now recognizes the decade key.
Key compress calendar has been replaced by timeslot unit to allow an additional level of compression (year).
Made pgfgantt robust to amsgen's redefinition of \Atlas.
The command \ganttvrule allows to draw general vertical rules (similar to the today rule). The keys vrule, vrule offset, vrule label font and vrule label text configure those rules.
The key expand chart was added, which specifies that a chart should expand horizontally to a given dimension.
The key *title label text* was added to allow fine-tuning of title label formatting.