

# timechart — A package for drawing chronological charts\*

Alan J. Cain

Released 2025-10-17

## Abstract

This package provides for the easy creation of chronological charts which show visually the relative historical positions of people and events. Each event or period can be specified by a single line of  $\text{\LaTeX}$  code comprising (possibly uncertain) start and finish dates and a label, and the package takes care of indicating the uncertainties and whether intervals extend beyond the specified bounds of the chart.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Requirements</b>	<b>3</b>
<b>3</b>	<b>Installation</b>	<b>3</b>
<b>4</b>	<b>Getting started</b>	<b>3</b>
<b>5</b>	<b>Example</b>	<b>3</b>
<b>6</b>	<b>Configuration</b>	<b>6</b>
<b>7</b>	<b>Specifying dates and date ranges</b>	<b>7</b>
<b>8</b>	<b>timechart environment</b>	<b>8</b>
8.1	General configuration of the <code>timechart</code> environment . . . . .	8
8.2	Grid configuration . . . . .	9
8.3	Axis configuration . . . . .	9
<b>9</b>	<b>Within the timechart environment</b>	<b>9</b>
9.1	Intervals . . . . .	9
	Interval configuration . . . . .	10
9.2	Text . . . . .	11
	Text configuration . . . . .	11
9.3	Space . . . . .	12
9.4	Positioning . . . . .	12
9.5	Completion . . . . .	13
9.6	Shortcut keys . . . . .	13

---

\*This file describes v0.56.1, last revised 2025-10-17.



## 1 Introduction

The timechart package provides a system for the easy creation of chronological charts — of the type pioneered by Joseph Priestley (1733–1804) in his ‘Chart of Biography’ and more famously in his ‘New Chart of History’<sup>1</sup> — which can show visually the relative historical positions of people and events. An example of what timechart can be used to produce is shown in [Figure 1](#) on page 4.

Essentially (and this was a design requirement), *only one line* of L<sup>A</sup>T<sub>E</sub>X code is required for each interval (which, in the case of [Figure 1](#), are mostly lifetimes). The timechart package takes care of indicating ranges of possible dates by suitable fading from or to transparency. It also handles indicators to show that intervals continue outside the specified scope of the chart.

timechart was developed from, and replaced, a set of macros used to create the chronological charts in the author’s book *Form & Number: A History of Mathematical Beauty*, which is available on the Internet Archive under a Creative Commons licence.<sup>2</sup>

**Licence.** timechart is released under the L<sup>A</sup>T<sub>E</sub>X Project Public Licence v1.3c or later.<sup>3</sup>

**Acknowledgements.** The author thanks Tânia Paulista for reading and commenting on an earlier draft of the documentation.

## 2 Requirements

timechart requires PGF/TikZ and a L<sup>A</sup>T<sub>E</sub>X kernel with expl3 support (any kernel version since 2020-02-02 should suffice).

## 3 Installation

To install manually, run `tex timechart.ins` and copy the file `timechart.sty` to somewhere L<sup>A</sup>T<sub>E</sub>X can find it.

## 4 Getting started

The timechart package is loaded as usual via `\usepackage{timechart}`. There are no package options.

The small example in [Section 5](#) illustrates the basic principles of timechart. [Section 14](#) shows the full code used to produce the large example in [Figure 1](#).

## 5 Example

This section illustrates how to create the small chart shown in [Figure 3](#) on page 6.

The basic environment is `timechart`, which includes the specification of the start and finish years. The start year 50 BCE is specified as `-50`, the finish year 75 CE as `75`. Each interval in the chart is specified using the `\timechartinterval` command, which

---

<sup>1</sup>URL: [https://commons.wikimedia.org/wiki/File:A\\_New\\_Chart\\_of\\_History\\_color.jpg](https://commons.wikimedia.org/wiki/File:A_New_Chart_of_History_color.jpg)

<sup>2</sup>URL: [https://archive.org/details/cain\\_formandnumber\\_ebook\\_large](https://archive.org/details/cain_formandnumber_ebook_large)

<sup>3</sup>URL: <https://www.latex-project.org/lppl.txt>

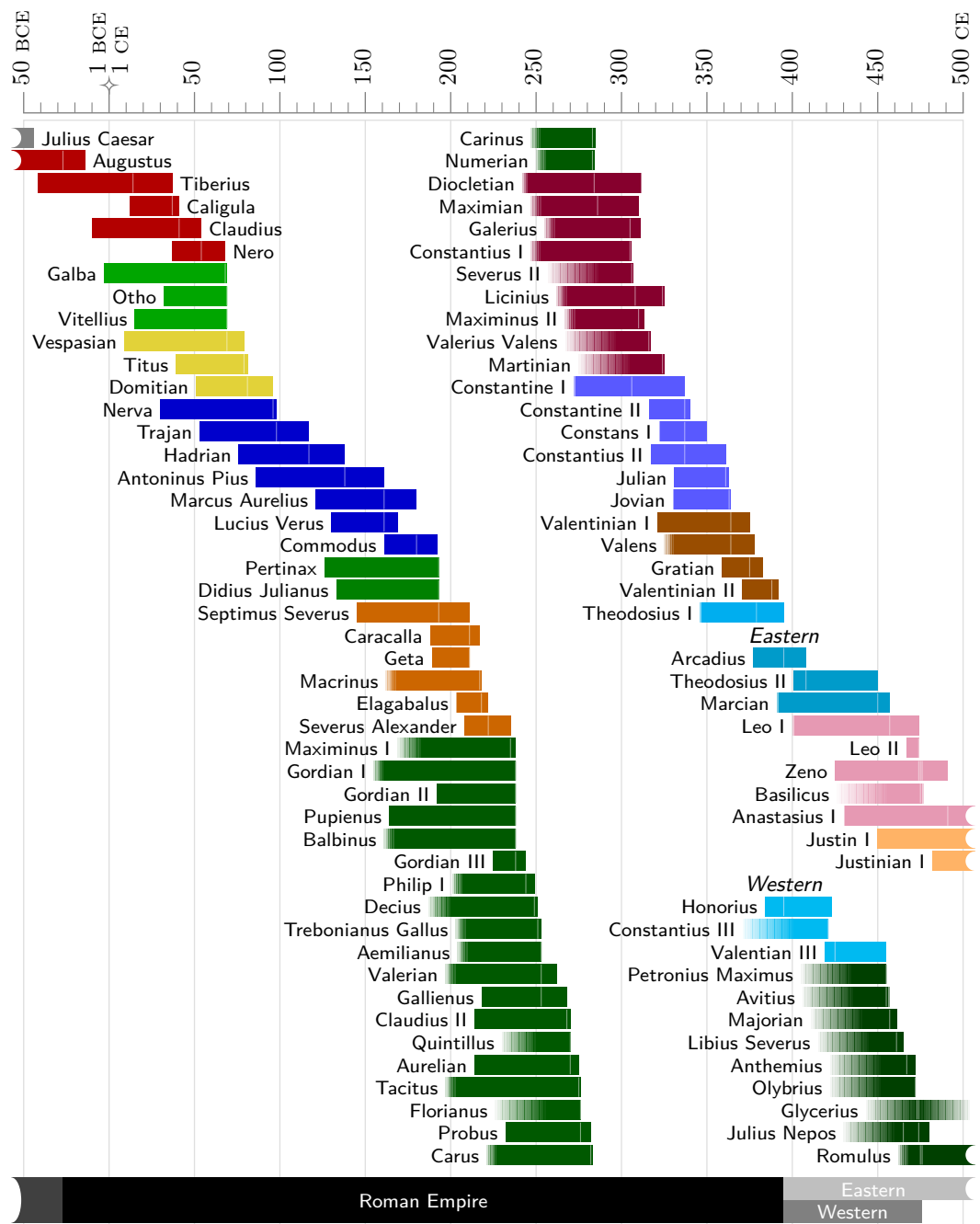


Figure 1: Timechart showing the lifetimes of Roman emperors from 50 BCE to 500 CE. Marks on each lifetime indicate the beginning (and, where relevant, the end) of that emperor's reign. Colours generally indicate dynasties, with shades of green indicating periods when the imperial power shifted between many short-reigning emperors.

takes three mandatory parameters: a start year, a finish year, and a label. The following code produces the flawed preliminary version shown in [Figure 2](#).

```

1 \begin{timechart}{-50}{75}
2   \timechartinterval{-63}{-14}{Augustus}
3   \timechartinterval{-42}{37}{Tiberius}
4   \timechartinterval{12}{41}{Caligula}
5   \timechartinterval{-10}{54}{Claudius}
6   \timechartinterval{37}{68}{Nero}
7 \end{timechart}

```

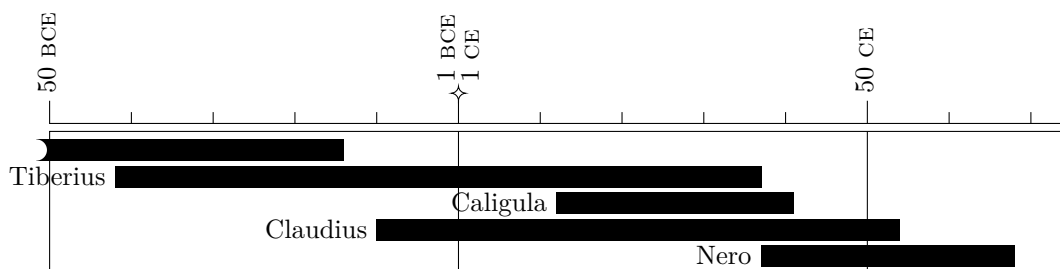


Figure 2: Flawed preliminary version of a chart showing the lifetimes of Roman emperors of the Julio-Claudian dynasty. (The final version is shown in [Figure 3](#).)

This first attempt result illustrates some of the principles of `timechart`. Each interval has been placed on its own line. More precisely, the  $y$  coordinate of the first interval is 0 and each use of `\timechartinterval` increments the ‘current  $y$  coordinate’ by a specified amount. (Various commands are available to set the  $y$  coordinate manually or to reset it automatically when it passes certain bounds; see [Subsection 9.4](#).) Horizontally, the chart starts and finishes at the specified years. The topmost interval, indicating Augustus’ life, has been truncated with an indicator that it begins before the specified start year of the chart. Vertically, the chart has been sized to fit around the specified intervals.

But this version is hardly satisfactory, for many reasons. The chart finishes between two minor ticks on the axis, because the intervals between major and minor ticks default respectively to 10 years and 50 years. The black intervals and text do not contrast well with the black axis and grid. The serif font is perhaps not best suited to label the intervals. And the label ‘Augustus’ has been lost, since labels are by default placed on the left of intervals. To rectify these problems, some changes are necessary, all of which can be made using key-value syntax in an optional parameter to the `timechart` environment or the `\timechartinterval` command:

1. Set the intervals between major and minor ticks to 5 years and 25 years respectively, by setting `minor tick interval=5` and `major tick interval=25`.
2. Change the colour of the grid to light grey by appending `draw=lightgray` to the `grid` style
3. Change the colour of the axis to grey by appending `draw=gray` to the `axis line`, `minor tick`, and `major tick` styles.
4. Change the font used for interval labels to small san-serif by appending `node font=\sffamily\small` to the `interval label` style

5. Change the colour of the intervals by defining a style `julioclaudian` that sets `interval bar color=red!80!black` and applying it to each interval via its optional argument. (While `interval bar color` could be set locally for each interval, it is better to define a style that corresponds to the semantic meaning of the colour: in this case, a single dynasty.)
6. Use the `right` key to place some labels on the right

The result is the following code, which produces [Figure 3](#).

```

1 \begin{timechart}[
2   minor tick interval=5,
3   major tick interval=25,
4   grid/.append style={ draw=lightgray },
5   axis line/.append style={ draw=gray },
6   minor tick/.append style={ draw=gray },
7   major tick/.append style={ draw=gray },
8   interval label/.append style={ node font=\sffamily\small },
9   julioclaudian/.style={ interval bar color=red!80!black },
10 ]{-50}{75}
11 \timechartinterval[right,julioclaudian]{-63}{-14}{Augustus}
12 \timechartinterval[right,julioclaudian]{-42}{37}{Tiberius}
13 \timechartinterval[julioclaudian]{12}{41}{Caligula}
14 \timechartinterval[julioclaudian]{-10}{54}{Claudius}
15 \timechartinterval[julioclaudian]{37}{68}{Nero}
16 \end{timechart}

```

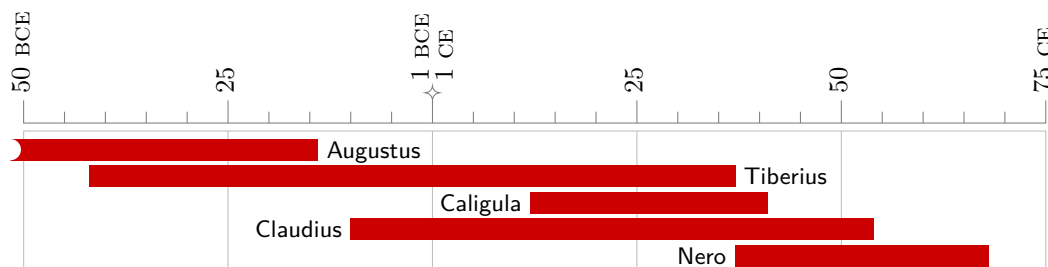


Figure 3: Chart showing the lifetimes of Roman emperors of the Julio-Claudian dynasty.

## 6 Configuration

All `timechart` configuration, global or local, is via PGF keys, so some familiarity with their use is beneficial; see the PGF/TikZ manual.

Configuration keys for `timechart` are contained in `/timechart/` in the PGF keys hierarchy. The `<options>` passed to the `timechart` environment or any of the commands `\timechartinterval`, `\timecharttext`, or `\timechartspace` are processed within `/timechart/` (since `/timechart/.cd` is executed before keys are processed).

The user may wish to define PGF styles for different kinds of interval within a chart. For example, one could define styles `science` and `art` that set a particular colour for the interval, and write `\timechartinterval[science]{<birth>}{<death>}{<name>}` or

`\timechartinterval[art]{\langle birth \rangle}{\langle death \rangle}{\langle name \rangle}` to distinguish visually the lifetimes of various scientists and artists.

## 7 Specifying dates and date ranges

Using `timechart` requires specification of dates and date ranges for the start and finish of each interval, both of which may be uncertain.

The basic specification of a date uses ISO 8601 format `YYYY-MM-DD`. This format specifies a date with day-level precision; use `YYYY-MM` and `YYYY` for month- and year-level precision. If the date is prefixed by `-`, it is treated as the corresponding date before the epoch. (This is a difference with ISO 8601, in which 0 represents 1 BCE, -1 represents 2 BCE, and so on.) So (assuming that one is using BCE/CE) one uses `-100` to indicate 100 BCE and `100` to indicate 100 CE. (The era indicators ‘BCE’ and ‘CE’ appear on the axis. Alternative era indicators — or a different epoch — can be specified; see [Section 10](#).)

A date can be prefixed with a `c` to indicate ‘circa’, such as `c-100` for ‘circa 100 BCE’ and `c100` for ‘circa 100 CE’. When an interval is drawn in a chart, ‘circa’ will be indicated by automatically creating (or extending) a range according to the value of the key `circa uncertainty` (see [Subsection 9.1](#)).

A date range comprises two dates (each with or without `c`) separated by a slash `/`, with the first date being earlier (or equal to) the second date. (The slash indicates a range of dates per ISO 8601.) A date range can be used to indicate a broader uncertainty than the default ‘circa’, or to indicate a definite range within which an interval starts or ends.

**Examples of correctly formatted dates and date ranges:** `-50`, `100`, `c-50`, `c100`, `-50/100`, `c-50 / +100`, `-50/c100`, `c-50/c100`, `-585-05-28`, `1947-12-01`, `1989-11`.

**Examples of incorrectly formatted date and date ranges:** `100?`, `100CE`, `100BCE`, `-50-100`, `100/-50`.

That is, the syntax for dates and date ranges is per the following (not-quite-formal) grammar:

<code>&lt;cdate-or-crangle&gt;</code>	<code>::=</code>	<code>&lt;cdate&gt;</code>
		<code>&lt;crangle&gt;</code>
<code>&lt;cdate&gt;</code>	<code>::=</code>	<code>&lt;date&gt;</code>
		<code>c&lt;date&gt;</code>
<code>&lt;date&gt;</code>	<code>::=</code>	<code>&lt;pdate&gt;</code>
		<code>-&lt;pdate&gt;</code>
<code>&lt;pdate&gt;</code>	<code>::=</code>	<code>&lt;year&gt;-&lt;month&gt;-&lt;day&gt;</code>
		<code>&lt;year&gt;-&lt;month&gt;</code>
		<code>&lt;year&gt;</code>
<code>&lt;crangle&gt;</code>	<code>::=</code>	<code>&lt;date<sub>1</sub>&gt;/&lt;date<sub>2</sub>&gt;</code>
		<code>c&lt;date<sub>1</sub>&gt;/&lt;date<sub>2</sub>&gt;</code>
		<code>&lt;date<sub>1</sub>&gt;/c&lt;date<sub>2</sub>&gt;</code>
		<code>c&lt;date<sub>1</sub>&gt;/c&lt;date<sub>2</sub>&gt;</code> (with <code>date<sub>1</sub> ≤ date<sub>2</sub></code> )

The bounds of the `timechart` environment (see [Section 8](#)) must satisfy `<cdate>` in this grammar (although only the `<year>` is used); the start and finish dates of an `\timechartinterval` command (see [Subsection 9.1](#)) must satisfy `<cdate-or-crangle>`; the parameter of an `\timecharttext` command (see [Subsection 9.2](#)) must satisfy `<date>`.

*Note.* For performance reasons, the date parser does only limited error-checking. Months outside the range from 01 to 12 or days outside the range of the specified month will be ignored. Otherwise malformed dates or date ranges may produce obscure error messages or unexpected results.

## 8 timechart environment

---

```
timechart \begin{timechart}[(options)]{(start)}{(finish)}
          <content>
          \end{timechart}
```

This is the main environment for creating a chronological chart. The mandatory arguments *start* and *finish* specify the first and last years of the chart. These can be dates with circa indicators (that is, they satisfy *cdate* in the grammar in [Section 7](#)), but the circa specifier has no effect and only the ‘year’ part of the date is used. The optional argument *options* supplies PGF keys that apply to the entire chart.

The *content* comprises commands like `\timechartinterval`, `\timecharttext`, `\timechartspace`, commands for positioning, as described in [Section 9](#), and the user’s own TikZ code.

### 8.1 General configuration of the timechart environment

- `/timechart/width=<dimension>` (default `\textwidth`)  
The width of the chart. This refers to the width of the grid and axis of the chart, not including intervals that pass beyond the specified limits of the chart, or axis labels that protrude beyond the width of the axis itself.
- `/timechart/tolerance=<dimension>` (default `5pt`)  
The length by which an interval is allowed to pass beyond the limits of the chart before it ‘counts’ as doing so and the appropriate indicator is drawn.
- `/timechart/beyond length=<dimension>` (default `5pt`)  
The length of the indicator that an interval passes beyond the limits of the chart.
- `/timechart/beyond x radius=<dimension>` (default `4pt`)  
The horizontal radius of the concave part of the indicator that an interval passes beyond the limits of the chart. (The vertical radius will be half the thickness of the bar.)
- `/timechart/ystep=<dimension>` (default `-10pt`)  
The default length (positive or negative) by which the current *y* coordinate is automatically adjusted after each interval, text, or space is placed.
- `/timechart/minor tick interval=<number>` (default `10`)  
The number of years (which must be positive) between each minor tick on the axis.
- `/timechart/major tick interval=<number>` (default `50`)  
The number of years (which must be positive) between each major tick on the axis and each vertical line in the grid.



## 8.2 Grid configuration

- `/timechart/no grid`  
Do not draw the grid.
- `/timechart/grid top ysep=<dimension>` (default 3pt)  
Distance between the top of the grid and the topmost interval or space.
- `/timechart/grid bottom ysep=<dimension>` (default 3pt)  
Distance between the bottom of the grid and the bottommost interval or space.
- `/timechart/grid` (style, initially empty)  
Style for drawing the grid.

## 8.3 Axis configuration

- `/timechart/axis=<position>` (default above)  
Where and whether to draw the axis. `<position>` may be above, below, or none.
- `/timechart/no axis`  
Do not draw the axis. Equivalent to `/timechart/axis=none`.
- `/timechart/axis line` (style, initially `line cap=rect`)  
Style for drawing the axis line.
- `/timechart/minor tick` (style, initially empty)  
Style for drawing minor ticks.
- `/timechart/minor tick length=<dimension>` (default 1.5mm)  
Length of minor ticks.
- `/timechart/major tick` (style, initially empty)  
Style for drawing major ticks.
- `/timechart/major tick length=<dimension>` (default 3mm)  
Length of major ticks.
- `/timechart/major tick label` (style, initially as described below)  
Style for labels on the major ticks on the axis. The initial style essentially sets `inner sep=0`, `outer sep=0`, `anchor=mid west`, `rotate=90`.
- `/timechart/major tick eras=<locations>` (default outer)  
Which major tick labels have era indicators. `<locations>` may be `none`, `all`, or `outer` (which means that only the first and last labels have era indicators). This option does not affect the epoch marker (see [Subsection 12.2](#)).

# 9 Within the timechart environment

## 9.1 Intervals

---

---

`\timechartinterval` `\timechartinterval[<options>]{<start>}{<finish>}{<label>}`

This command creates an interval in the chart at the current  $y$  coordinate between the specified `<start>` and `<finish>`, with the given `<label>`. These arguments are mandatory. Each of `<start>` and `<finish>` can be either a year or a range of years, possibly with circa markers. That is, each must satisfy `<year-or-crange>` in the grammar in [Section 7](#).

The optional argument `<options>` specifies PGF keys within `/timechart/` that are applied locally to the interval.

The current  $y$  coordinate will be adjusted according to `/timechart/ystep` unless `/timechart/no autostep` has been set.

## Interval configuration

`/timechart/no autostep`

Do not automatically alter the current  $y$  coordinate by the amount specified in `/timechart/ystep`.

`/timechart/ref=<label>` (default none)

Make the interval label a hyperlink to the position labelled by `<label>`.

`/timechart/mark=<comma-separated list of years>` (default empty)

Draw marks in the interval at the years contained in the list. Each entry in the list must be a definite year (that is, must satisfy `<year>` in the grammar in [Section 7](#)), must lie in the definite part of the interval (not in the start or finish ranges) and cannot lie beyond the bounds of the chart on a side where there interval exceeds tolerance. The colour of marks can be specified using `/timechart/interval mark color`. `/timechart/marks` is a synonym for this key.

`/timechart/circa uncertainty=<number>` (default 3)

Treat a circa indicator  $c$  as indicating an uncertainty of  $\pm<number>$ .

`/timechart/interval minimum width=<dimension>` (default 1pt)

Ensure that any interval has a width of at least `<dimension>`. This is useful to ensure that a single event is visible in the chart.

If an interval is specified with start and finish ranges, and with `start range=fade` and `finish range=fade`, then the ‘certain’ portion of the interval will also have width at least `<dimension>`. (This restriction prevents a common rendering error where start and finish fadings around a ‘certain’ interval of length 0 would not quite meet.)

`/timechart/interval bar color=<color>` (default black)

Fill the interval bar with `<color>`.

`/timechart/interval bar thickness=<dimension>` (default 8pt)

Set the vertical thickness of the interval bar to `<dimension>`.

`/timechart/interval bar node name=<string>` (default bar node)

Set the name of the node containing the interval bar to `<string>`.

`/timechart/interval mark color=<color>` (default gray)

Draw marks using `<color>`.

`/timechart/interval label` (style, initially empty)

Style to apply to an interval label.

`/timechart/interval label centered` (style, initially as below)

Style to apply to an interval label placed centrally. Initially, this style executes the style `/timechart/interval label` and sets `text=white`. The reason for a separate style for centred labels is that often a contrasting colour will be required. For instance, labels positioned to the left and right may be black, but if the bar is black, a centred label should be a light colour.



`/timechart/interval label centered background` (style, initially as below)



Style to apply to the ‘background’ interval label placed centrally. Initially, this style executes `/timechart/interval label`. The ‘background’ interval label is simply the usual label and is placed in the same location, but on a layer behind the bar and, unlike the label itself, is not clipped. The reason for this style is that if the bar is narrow, part of the label text (such as ascenders and/or descenders) may naturally extend beyond the bar itself and it may be useful that these should appear in a different colour.

`/timechart/interval label baseline=<dimension>` (default -3pt)  
 Position the baseline of the interval label  $\langle dimension \rangle$  below the current  $y$  coordinate (which is the midpoint of the interval bar).

`/timechart/interval label pos=<position>` (default left)  
 Specify where to place the label relative to the interval bar:  $\langle position \rangle$  may be `left`, `center`, or `right`. The position `center` places the label at the midpoint of *the visible segment of the solid part* of the interval bar (that is, not including fading at the start or finish of the bar, and not including part of the bar that would extend beyond the bounds of the chart). Further, a centred label is clipped to the size of the bar and a unclipped ‘background’ copy of it is drawn behind the bar, so that the portion appearing ‘on’ and ‘outside’ the bar can have different styles. (See the keys `/timechart/interval label centered` and `/timechart/interval label centered background`.)

`/timechart/interval label node name=<string>` (default label node)  
 Set the name of the node containing the interval label to  $\langle string \rangle$ .

`/timechart/start range=<range-type>` (default fade)  
 Type of indication of the range where an interval may start.  $\langle range-type \rangle$  can be `fade`, which produces an indicator like , or `slant`, which produces .

`/timechart/finish range=<range-type>` (default fade)  
 Type of indication of the range where an interval may finish.  $\langle range-type \rangle$  can be `fade`, which produces an indicator like , or `slant`, which produces .

## 9.2 Text

---

`\timecharttext` [ $\langle options \rangle$ ]{ $\langle year \rangle$ }{ $\langle text \rangle$ }

Place  $\langle text \rangle$  in the time chart at the current  $y$  coordinate and at the horizontal position of  $\langle year \rangle$ , which must be a definite year (that is, must satisfy  $\langle year \rangle$  in the grammar in [Section 7](#)).

The optional argument  $\langle options \rangle$  specifies PGF keys within `/timechart/` that are applied locally.

The current  $y$  coordinate will be adjusted according to `/timechart/ystep` unless `/timechart/no autostep` has been set.

### Text configuration

`/timechart/text node name=<string>` (default text node)  
 Set the name of the node containing the text to  $\langle string \rangle$ .

`/timechart/text` (style, initially empty)  
 Style to apply to the text.

`/timechart/text baseline=<dimension>` (default -3pt)  
 Position the baseline of the text  $\langle dimension \rangle$  below the current  $y$  coordinate.

`/timechart/text pos=<position>` (default left)  
 Specify where to place the label relative to the given  $\langle year \rangle$ :  $\langle position \rangle$  may be `left`, `center`, or `right`.

### 9.3 Space

---

`\timechartspace` `\timechartspace[<options>]`

Create a space in the time chart at the current  $y$  coordinate, with the same effect on vertical spacing as an interval. More precisely, the current  $y$  coordinate will be adjusted according to `/timechart/ystep` unless `/timechart/no autostep` has been set.

The optional argument `<options>` specifies PGF keys within `/timechart/` that are applied locally.

### 9.4 Positioning

The commands `\timechartinterval`, `\timecharttext`, and `\timechartspace` all act at the current  $y$  coordinate and change its value according to `/timechart/ystep` (unless `/timechart/no autostep` is used). There are several functions to set the current  $y$  coordinate and to have it reset automatically.

---

`\timechartsety` `\timechartsety{<dimension>}`

Set the current  $y$  coordinate to `<dimension>`.

---

`\timechartsavey` `\timechartsavey`

Save the current  $y$  coordinate. If `\timechartresety` is used, the  $y$  coordinate resets to the last saved  $y$  coordinate. If the current  $y$  coordinate exceeds the minimum or maximum set by `\timechartsetyminimumautoreset` and `\timechartsetymaximumautoreset`, it will be reset to the last saved  $y$  coordinate.

---

`\timechartresety` `\timechartresety`

Reset the  $y$  coordinate to the last coordinate saved using `\timechartsavey`, or to 0 if there has been no use of `\timechartsavey` within the current `timechart` environment.

---

`\timechartsetyminimumautoreset` `\timechartsetyminimumautoreset{<min-coordinate>}`

`\timechartsetymaximumautoreset` `\timechartsetymaximumautoreset{<max-coordinate>}`

Set  $y$  coordinates that automatically trigger `\timechartresety` if the current  $y$  coordinate goes below `<min-coordinate>` or above `<max-coordinate>`.

---

`\timechartstepy` `\timechartstepy[<count>]`

Manually step the current  $y$  coordinate by `<count>` times the value of `/timechart/ystep`. The default value of `<count>` is 1. (The `/timechart/no autostep` does not affect `\timechartstepy`.)

## 9.5 Completion

---

`\timechartfinish` Signal that the chart is complete and that the grid and axis should be drawn (unless the keys `/timechart/no grid` and/or `/timechart/no axis` have been used). It is not necessary to use this command: if it is not given, the grid and axis will be drawn at the end of the `timechart` environment. But after this command, the TikZ nodes `grid` and `axis` are available, containing (respectively) the grid and the axis. These can be used in for further TikZ drawing.

Note that after `\timechartfinish`, none of the various `\timechart...` commands are available inside that `timechart` environment.

## 9.6 Shortcut keys

`/timechart/left`

Equivalent to setting `/timechart/interval label pos` and `/timechart/text pos` to `left`.

`/timechart/center`

Equivalent to setting `/timechart/interval label pos` and `/timechart/text pos` to `center`.

`/timechart/right`

Equivalent to setting `/timechart/interval label pos` and `/timechart/text posright` to `right`.

## 10 Era indicators

---

`\timechartmakebeforeyear` `\timechartmakebeforeyear{<number>}`  
`\timechartmakeafteryear` `\timechartmakeafteryear{<number>}`

Typeset `<number>` (which should be a positive whole number) as a year before or after the epoch. By default, `\timechartmakebeforeyear{<number>}` produces `<number>` BCE and `\timechartmakeafteryear{<number>}` produces `<number>` CE.

These commands are used for axis labels and can be re-defined by the user. For example, if BC/AD is preferred to BCE/CE, the user can define

```
1 \renewcommand*\timechartmakebeforeyear}[1]{#1-\textsc{bc}}
2 \renewcommand*\timechartmakeafteryear}[1]{\textsc{ad}~#1}
```

Similarly, if AH/BH is preferred, the user can define

```
1 \renewcommand*\timechartmakebeforeyear}[1]{#1-\textsc{bh}}
2 \renewcommand*\timechartmakeafteryear}[1]{\textsc{ah}~#1}
```

## 11 Legend

`timechart` supplies a number of auxiliary macros for creating a legend to explain, for example the significance of different colours of intervals. For example, [Figure 4](#) shows a suitable legend for [Figure 1](#).

	Julio-Claudian dynasty
	Year of four emperors
	Flavian dynasty
	Nerva-Antonine dynasty
	Year of five emperors
	Severan dynasty
	Crisis of the third century
	Tetrarchy
	Constantinian dynasty
	Valentinianic dynasty
	Theodosian dynasty
	— in the east
	— in the west
	Leonid dynasty
	Justinian dynasty
	Last emperors in the west

Figure 4: Example legend for the timechart shown in Figure 1.

The `\timechartlength...` macros are *not* meant to be used inside a `timechart` environment, but in locations such as running text or a `tabular` environment.

---

`\timechartlegenditem` `\timechartlegenditem[options]`

Draw a bar suitable for use in a legend. *options* specifies PGF keys that are applied within `/timechart`. The same PGF keys that affect interval bars affect the drawn bar, as do the keys listed below.

---

`\timechartlegendstartrange` `\timechartlegendstartrange[options]`  
`\timechartlegendfinishrange` `\timechartlegendfinishrange[options]`

Draw a bar suitable for use in a legend, with a start or finish range. *options* specifies PGF keys that are applied within `/timechart/`. The same PGF keys that affect interval bars affect the drawn bar, as do the keys listed below.

`/timechart/legend item width=dimension` (default 9mm)

When using macros `\timechartlegenditem`, `\timechartlegendstartrange`, or `\timechartlegendfinishrange`, draw a bar of total width *dimension*.

`/timechart/legend item range width=dimension` (default 3mm)

When using `\timechartlegendstartrange` or `\timechartlegendfinishrange`, draw a bar with a range of width *dimension*.

## 12 Usage notes

### 12.1 Additional TikZ code

The `timechart` environment is a `tikzpicture`. The user can add any TikZ code before, between, or after content created using the `\timechart...` commands. Each use of `\timechartinterval` defines two nodes. One, by default named `bar node`, contains the interval bar; the other, by default named `label node`, contains the interval label. Similarly, text added using `\timecharttext` is contained in a node, by default

named `text node`. (The default names are re-used, but can be changed using the keys `/timechart/interval bar node name`, `/timechart/interval label node name`, and `/timechart/text node name`.) The user can use these nodes to position extra content.

If the `\timechartfinish` command is used (after which the `\timechart...` commands are no longer available within the `timechart` environment) the nodes `grid` and `axis`, which contain the grid and the axis, are also available.

## 12.2 ‘year zero’

Although calendars typically do not admit a ‘year zero’ (for instance, 1 BCE is immediately followed by 1 CE, with no intervening ‘year zero’), `timechart` does allow 0 for the `<start>` or `<finish>` of the `timechart` environment or as the `<start>` or `<finish>` of `\timechartinterval` or the `<year>` of `\timecharttext`. But ‘year zero’ is indicated on the axis by a special epoch marker showing the last year before and first year after the epoch.

## 13 Feature requests and bug reports

The development code and issue tracker are hosted at Codeberg: <https://codeberg.org/ajcain/timechart>

## 14 Appendix: full example source

This section contains the necessary source code to produce the example timechart and legend shown in [Figures 1 and 4](#) on pages [4](#) and [14](#).

### 14.1 Setup source

```
1 \pgfkeys{
2   /timechart/.cd,
3   julioclaudian/.style={
4     interval bar color=red!70!black,
5     interval mark color=red!70!black!50!white
6   },
7   fourempereors/.style={
8     interval bar color=green!65!black,
9     interval mark color=green!65!black!50!white
10  },
11  flavian/.style={
12    interval bar color=yellow!85!black,
13    interval mark color=yellow!85!black!50!white
14  },
15  nervaantonine/.style={
16    interval bar color=blue!80!black,
17    interval mark color=blue!80!black!50!white
18  },
19  fiveemperors/.style={
20    interval bar color=green!50!black,
21    interval mark color=green!50!black!50!white
22  },
```

```

23 severan/.style={
24     interval bar color=orange!80!black,
25     interval mark color=orange!80!black!50!white
26 },
27 thirdcentury/.style={
28     interval bar color=green!35!black,
29     interval mark color=green!35!black!50!white
30 },
31 tetrarchy/.style={
32     interval bar color=purple!70!black,
33     interval mark color=purple!70!black!50!white
34 },
35 constantinian/.style={
36     interval bar color=blue!65!white,
37     interval mark color=blue!65!white!50!white
38 },
39 valentinianic/.style={
40     interval bar color=orange!60!black,
41     interval mark color=orange!60!black!50!white
42 },
43 theodosian/.style={
44     interval bar color=cyan,
45     interval mark color=cyan!50!white
46 },
47 theodosian-east/.style={
48     interval bar color=cyan!80!black,
49     interval mark color=cyan!80!black!50!white
50 },
51 theodosian-west/.style={
52     interval bar color=cyan!80!white,
53     interval mark color=cyan!80!white!50!white
54 },
55 leonid/.style={
56     interval bar color=purple!40!white,
57     interval mark color=purple!40!white!50!white
58 },
59 justinian/.style={
60     interval bar color=orange!60!white,
61     interval mark color=orange!60!white!50!white
62 },
63 lastwest/.style={
64     interval bar color=green!25!black,
65     interval mark color=green!25!black!50!white
66 },
67 }

```

## 14.2 Timechart source

```

1 \begin{timechart}[
2     axis line/.append style={ draw=gray, line width=.5pt },
3     grid/.append style={ draw=lightgray!50!white, line width=.5pt },
4     minor tick/.append style={ draw=gray },
5     major tick/.append style={ line width=.5pt, draw=gray },
6     major tick label/.append style={ node font=\small },

```



```

7     interval label/.style={ node font=\sffamily\footnotesize },
8     text/.style={ node font=\sffamily\small\itshape, },
9     ystep=-3.25mm,
10    ]{-50}{500}
11
12    \pgfmathsetmacro{\mainlinecount}{46}
13
14    \timechartsetyminimumautoreset{-3.25mm*\mainlinecount+1mm}
15
16    \timechartinterval[interval bar color=gray,right,mark=-45]{-100}{-44}{Julius
17    Caesar}
18    \timechartinterval[julioclaudian,right,mark=-27]{-63}{-14}{Augustus}
19    \timechartinterval[julioclaudian,right,mark=14]{-42}{37}{Tiberius}
20    \timechartinterval[julioclaudian,right,mark=37]{12}{41}{Caligula}
21    \timechartinterval[julioclaudian,right,mark=41]{-10}{54}{Claudius}
22    \timechartinterval[julioclaudian,right,mark=54]{37}{68}{Nero}
23
24    \timechartinterval[fourempereors,mark=68]{-3}{69}{Galba}
25    \timechartinterval[fourempereors,mark=69]{32}{69}{Otho}
26    \timechartinterval[fourempereors,mark=69]{15}{69}{Vitellius}
27
28    \timechartinterval[flavian,mark=69]{9}{79}{Vespasian}
29    \timechartinterval[flavian,mark=79]{39}{81}{Titus}
30    \timechartinterval[flavian,mark=81]{51}{96}{Domitian}
31
32    \timechartinterval[nervaantonine,mark=96]{30}{98}{Nerva}
33    \timechartinterval[nervaantonine,mark=98]{53}{117}{Trajan}
34    \timechartinterval[nervaantonine,mark=117]{76}{138}{Hadrian}
35    \timechartinterval[nervaantonine,mark=138]{86}{161}{Antoninus Pius}
36    \timechartinterval[nervaantonine,mark=161]{121}{180}{Marcus Aurelius}
37    \timechartinterval[nervaantonine,mark=161]{130}{169}{Lucius Verus}
38    \timechartinterval[nervaantonine,mark=180]{161}{192}{Commodus}
39
40    \timechartinterval[fiveemperors,mark=193]{126}{193}{Pertinax}
41    \timechartinterval[fiveemperors,mark=193]{133}{193}{Didius Julianus}
42
43    \timechartinterval[severan,mark=193]{145}{211}{Septimus Severus}
44    \timechartinterval[severan,mark=211]{188}{217}{Caracalla}
45    \timechartinterval[severan,mark=211]{189}{211}{Geta}
46    \timechartinterval[severan,mark=217]{c165}{218}{Macrinus}
47    \timechartinterval[severan,mark=218]{203/204}{222}{Elagabalus}
48    \timechartinterval[severan,mark=222]{208}{235}{Severus Alexander}
49
50    \timechartinterval[thirdcentury,mark=235]{c172/c180}{238}{Maximinus I}
51    \timechartinterval[thirdcentury,mark=238]{c158}{238}{Gordian I}
52    \timechartinterval[thirdcentury,mark=238]{192}{238}{Gordian II}
53    \timechartinterval[thirdcentury,mark=238]{164}{238}{Pupienus}
54    \timechartinterval[thirdcentury,mark=238]{c164}{238}{Balbinus}
55    \timechartinterval[thirdcentury,mark=238]{225}{244}{Gordian III}
56    \timechartinterval[thirdcentury,mark=244]{c204}{249}{Philip I}
57    \timechartinterval[thirdcentury,mark=249]{c190/200}{251}{Decius}
58    \timechartinterval[thirdcentury,mark=251]{c206}{253}{Trebonianus Gallus}
59    \timechartinterval[thirdcentury,mark=253]{c207}{253}{Aemilianus}
60    \timechartinterval[thirdcentury,mark=253]{c200}{262}{Valerian}

```

60 \timechartinterval [thirdcentury,mark=253] {218}{268}{Gallienus}  
61 \timechartinterval [thirdcentury,mark=268] {214}{270}{Claudius II}  
62 \timechartinterval [thirdcentury,mark=270] {230/250}{270}{Quintillus}  
63 \timechartinterval [thirdcentury,mark=270] {214}{275}{Aurelian}  
64 \timechartinterval [thirdcentury,mark=275] {c200}{276}{Tacitus}  
65 \timechartinterval [thirdcentury,mark=276] {226/256}{276}{Florianus}  
66 \timechartinterval [thirdcentury,mark=276] {232}{282}{Probus}  
67 \timechartinterval [thirdcentury,mark=282] {c224}{283}{Carus}  
68 \timechartinterval [thirdcentury,mark=283] {c250}{285}{Carinus}  
69 \timechartinterval [thirdcentury,mark=283] {c253}{284}{Numerian}  
70  
71 \timechartinterval [tetrarchy,mark=284] {242/245}{311/312}{Diocletian}  
72 \timechartinterval [tetrarchy,mark=286] {c250}{310}{Maximian}  
73 \timechartinterval [tetrarchy,mark=305] {c258}{311}{Galerius}  
74 \timechartinterval [tetrarchy,mark=305] {c250}{306}{Constantius I}  
75 \timechartinterval [tetrarchy,mark=306] {257/287}{307}{Severus II}  
76 \timechartinterval [tetrarchy,marks={308,324}] {c265}{325}{Licinius}  
77 \timechartinterval [tetrarchy,mark=310] {c270}{313}{Maximinus II}  
78 \timechartinterval [tetrarchy,mark=316] {267/297}{317}{Valerius Valens}  
79 \timechartinterval [tetrarchy,mark=324] {275/305}{325}{Martinian}  
80  
81 \timechartinterval [constantinian,mark=306] {272/273}{337}{Constantine I}  
82 \timechartinterval [constantinian,mark=337] {316}{340}{Constantine II}  
83 \timechartinterval [constantinian,mark=337] {322/323}{350}{Constans I}  
84 \timechartinterval [constantinian,mark=337] {317}{361}{Constantius II}  
85 \timechartinterval [constantinian,mark=361] {331}{363}{Julian}  
86 \timechartinterval [constantinian,mark=363] {330/331}{364}{Jovian}  
87  
88 \timechartinterval [valentinianic,mark=364] {321}{375}{Valentinian I}  
89 \timechartinterval [valentinianic,mark=364] {c328}{378}{Valens}  
90 \timechartinterval [valentinianic,mark=375] {359}{383}{Gratian}  
91 \timechartinterval [valentinianic,mark=388] {371}{392}{Valentinian II}  
92  
93 \timechartinterval [theodosian,mark=379] {346/347}{395}{Theodosius I}  
94 \timecharttext [center] {395}{Eastern}  
95 \timechartinterval [theodosian-east,mark=395] {377}{408}{Arcadius}  
96 \timechartinterval [theodosian-east,mark=408] {401}{450}{Theodosius II}  
97 \timechartinterval [theodosian-east,mark=450] {391/392}{457}{Marcian}  
98  
99 \timechartinterval [leonid,mark=457] {400/401}{474}{Leo I}  
100 \timechartinterval [leonid,mark=474] {467}{474}{Leo II}  
101 \timechartinterval [leonid,mark={474,475,476}] {425}{491}{Zeno}  
102 \timechartinterval [leonid,mark=475] {426/456}{476/477}{Basilicus}  
103 \timechartinterval [leonid,mark=491] {430/431}{518}{Anastasius I}  
104  
105 \timechartinterval [justinian,mark=518] {450}{527}{Justin I}  
106 \timechartinterval [justinian,mark=527] {482}{565}{Justinian I}  
107  
108 \timecharttext [center] {395}{Western}  
109 \timechartinterval [theodosian-west,mark=395] {384}{423}{Honorius}  
110 \timechartinterval [theodosian-west,mark=421] {371/401}{421}{Constantius III}  
111 \timechartinterval [theodosian-west,mark=425] {419}{455}{Valentian III}  
112  
113 \timechartinterval [lastwest,mark=455] {405/435}{455}{Petronius Maximus}

```

114 \timechartinterval [lastwest,mark=455] {406/436}{456/457}{Avitius}
115 \timechartinterval [lastwest,mark=457] {411/441}{461}{Majorian}
116 \timechartinterval [lastwest,mark=461] {415/445}{465}{Libius Severus}
117 \timechartinterval [lastwest,mark=467] {422/452}{472}{Anthemius}
118 \timechartinterval [lastwest,mark=472] {422/452}{472}{Olybrius}
119 \timechartinterval [lastwest,mark={473,474}] {443/473}{474/504}{Glycerius}
120 \timechartinterval [lastwest,mark={474,465}] {430/460}{480}{Julius Nepos}
121 \timechartinterval [lastwest,mark={475,476}] {c465}{507/527}{Romulus}
122
123 \timechartsetyminimumautoreset{-\maxdimen}
124 \timechartsety{(-\mainlinecount-1)*3.25mm}
125
126 \timechartinterval [
127     center,
128     interval bar color=darkgray,
129     interval text/.append style={ text=white },
130     interval bar thickness=6.5mm
131 ]{-509}{-27}{ }
132 \timechartstepy[-1]
133 \timechartinterval [
134     center,
135     interval bar color=black,
136     interval text/.append style={ node font=\sffamily\normalsize,text=white },
137     interval bar thickness=6.5mm
138 ]{-27}{395}{Roman Empire}
139 \timechartstepy[-1.5]
140 \timechartinterval [
141     center,
142     interval bar color=lightgray,
143     interval bar thickness=3.25mm
144 ]{395}{1453}{Eastern}
145 \timechartinterval [
146     center,
147     interval bar color=gray,
148     interval bar thickness=3.25mm
149 ]{395}{476}{Western}
150 \end{timechart}

```

### 14.3 Legend source

```

1 \begin{tabular}{rl}
2 \timechartlegenditem[julioclaudian] & & Julio-Claudian dynasty & \\
3 \timechartlegenditem[foureemperors] & & Year of four emperors & \\
4 \timechartlegenditem[flavian] & & Flavian dynasty & \\
5 \timechartlegenditem[nervaantonine] & & Nerva-Antonine dynasty & \\
6 \timechartlegenditem[fiveemperors] & & Year of five emperors & \\
7 \timechartlegenditem[severan] & & Severan dynasty & \\
8 \timechartlegenditem[thirdcentury] & & Crisis of the third century & \\
9 \timechartlegenditem[tetrarchy] & & Tetrarchy & \\
10 \timechartlegenditem[constantinian] & & Constantinian dynasty & \\
11 \timechartlegenditem[valentinianic] & & Valentinianic dynasty & \\
12 \timechartlegenditem[theodosian] & & Theodosian dynasty & \\
13 \timechartlegenditem[theodosian-east] & & --- in the east & \\
14 \timechartlegenditem[theodosian-west] & & --- in the west & \\

```

```

15 \timechartlegenditem[leonid] & Leonid dynasty \\
16 \timechartlegenditem[justinian] & Justinian dynasty \\
17 \timechartlegenditem[lastwest] & Last emperors in the west \\
18 \end{tabular}

```

## 15 Implementation

```

1 <*package>
2 <@=timechart>

```

### 15.1 Coding standard

This package makes extensive use of `pgfmath` computations. The usual `expl3` standard of ending variables with a type indicator (`_bool`, `_int`, etc.) is therefore adapted as follows:

`_year` Stores a year, which could in principle be fractional.

`_pgf` Stores a length calculated by `pgfmath`. (Unlike `_dim`, there is no underlying dimension register.)

`_x` Stores a raw  $x$  coordinate (not in `TikZ`'s  $XY$ -coordinate system).

`_y` Stores a raw  $y$  coordinate (not in `TikZ`'s  $XY$ -coordinate system).

`_text` Stores text (not an `expl3` string).

### 15.2 Initial set-up

Package identification/version information.

```

3 \NeedsTeXFormat{LaTeX2e}[2020-02-02]
4 \ProvidesExplPackage{timechart}{2025-10-17}{0.56.1}
5 {Typesetting chronological charts}

```

### 15.3 Debugging macro

Macro to output (many) debugging messages.

```

\__timechart_debug:n Macro to output (many) debugging messages.
\__timechart_debug_real:n
6 \cs_new:Npn \__timechart_debug_real:n #1
7 {
8   \typeout{timechart:~#1}
9 }
10 \cs_set_eq:NN\__timechart_debug:n\use_none:n

```

*(End of definition for `\__timechart_debug:n` and `\__timechart_debug_real:n`.)*

### 15.4 Load TikZ

```

11 \RequirePackage{tikz}

```

In the remainder of the package, only a limited subset of `TikZ` is used, and `PGF` code is preferred. For `PGF` keys, it is necessary to use `~` in place of a space.

### 15.5 Scratch variables

Scratch boolean variables.

```

\l__timechart_tmpa_bool
\l__timechart_tmpb_bool
\l__timechart_tmpc_bool
\l__timechart_tmpd_bool
12 \bool_new:N\l__timechart_tmpa_bool
13 \bool_new:N\l__timechart_tmpb_bool

```

```

14 \bool_new:N\l__timechart_tmpc_bool
15 \bool_new:N\l__timechart_tmpd_bool

```

*(End of definition for \l\_\_timechart\_tmpa\_bool and others.)*

\l\_\_timechart\_tmpa\_dim Scratch dimension variables, reusing \l\_tmpa\_dim and \l\_tmpb\_dim with uniform names.

```

\l__timechart_tmpb_dim 16 \cs_set_eq:NN\l__timechart_tmpa_dim\l_tmpa_dim
\l__timechart_tmpc_dim 17 \cs_set_eq:NN\l__timechart_tmpb_dim\l_tmpb_dim
\l__timechart_tmpd_dim 18 \dim_new:N\l__timechart_tmpc_dim
19 \dim_new:N\l__timechart_tmpd_dim

```

*(End of definition for \l\_\_timechart\_tmpa\_dim and others.)*

## 15.6 Generic auxiliary functions

\\_\_timechart\_make\_ref:NN Make hyperreference from text, if the supplied target is non-empty.

**#1** : Reference for hyperlink target, or empty.  
**#2** : Text.

```

20 \cs_new:Npn\__timechart_make_ref:NN #1#2
21 {
22   \str_if_empty:NTF #1
23     { #2 }
24     { \hyperref[#1]{#2} }
25 }

```

*(End of definition for \\_\_timechart\_make\_ref:NN.)*

## 15.7 PGF auxiliary functions

\_\_timechart\_pgfmathsetbool:nn Set an expl3 boolean variable to the outcome of a pgfmath comparison. This macro is simply a wrapper around \pgfmathsetmacro using ifthenelse and returning the boolean literal true or false.

```

26 \cs_new:Npn\__timechart_pgfmathsetbool:nn #1#2
27 {
28   \pgfmathsetmacro{#1}{ifthenelse(#2,"c_true_bool","c_false_bool")}
29 }

```

*(End of definition for \_\_timechart\_pgfmathsetbool:nn.)*

\\_\_timechart\_if\_equal:nnF Use pgfmath to check whether #1 and #2 are equal. If not, execute #3.

```

30 \cs_new:Npn\__timechart_if_equal:nnF #1#2#3
31 {
32   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{#1==#2}
33   \bool_if:NF\l__timechart_tmpa_bool{#3}
34 }

```

*(End of definition for \\_\_timechart\_if\_equal:nnF.)*

\\_\_timechart\_pgffextractxy:nnn Extract coordinates of #3 (a PGF point) to dimension variables #1 and #2. This macro simply combines the functionality of \pgffextractx and \pgffextracty.

```

35 \cs_new:Npn\__timechart_pgffextractxy:nnn #1#2#3
36 {
37   \pgf@process{#3}
38   #1=\pgf@x\relax
39   #2=\pgf@y\relax
40 }

```

(End of definition for `\__timechart_pgffextractxy:nnn`.)

`\__timechart_hsmash_pgfnode:nnnn` Do the same as `\pgfnode` but only update the bounding box ‘vertically’.

```

41 \cs_new:Npn\__timechart_hsmash_pgfnode:nnnn #1#2#3#4#5
42 {
43   \pgfinterruptboundingbox
44   \pgfnode{#1}{#2}{#3}{#4}{#5}
45   \pgfcoordinate
46     {__timechart_tmpa_coord}
47     {\pgfpointanchor{current~bounding~box}{south}}
48   \pgfcoordinate
49     {__timechart_tmpb_coord}
50     {\pgfpointanchor{current~bounding~box}{north}}
51   \endpgfinterruptboundingbox
52   \pgfextractx
53     {\l__timechart_tmpa_dim}
54     {\pgfpointanchor{__timechart_tmpa_coord}{center}}
55   \pgfextractx
56     {\l__timechart_tmpb_dim}
57     {\pgfpointanchor{__timechart_tmpb_coord}{center}}
58   \pgfpathmoveto{\pgfpoint{\l__timechart_tmpa_dim}{0}}
59   \pgfpathmoveto{\pgfpoint{\l__timechart_tmpb_dim}{0}}
60   \pgfusepath{discard}
61 }

```

(End of definition for `\__timechart_hsmash_pgfnode:nnnn`.)

`\__timechart_make_rectangle_node:nnnn` Make a node with south west corner #1, north east corner #2, and name #3. #1 and #2 should be given as PGF points. #4 is a boolean literal indicating whether the path should be stroked.

```

62 \cs_new:Npn\__timechart_make_rectangle_node:nnnn #1#2#3#4
63 {
64   \group_begin:
65   \__timechart_pgffextractxy:nnn
66     {\l__timechart_tmpa_dim}{\l__timechart_tmpb_dim}{#1}
67   \__timechart_pgffextractxy:nnn
68     {\l__timechart_tmpc_dim}{\l__timechart_tmpd_dim}{#2}
69   \pgftransformshift{#1}
70   \pgfset{
71     minimum~width=\l__timechart_tmpc_dim-\l__timechart_tmpa_dim,
72     minimum~height=\l__timechart_tmpd_dim-\l__timechart_tmpb_dim,
73     inner~sep=0,
74     outer~sep=0,
75   }
76   \bool_if:NTF #4
77     { \pgfnode{rectangle}{south~west}{#3}{\pgfusepath{draw}} }
78     { \pgfnode{rectangle}{south~west}{#3}{\pgfusepath{discard}} }
79   \group_end:
80 }

```

(End of definition for `\__timechart_make_rectangle_node:nnnn`.)

`\__timechart_set_style_line_width:nn` Set macro #1 to be the line width set by the PGF style #2. Note that `\begingroup` and `\endgroup` are used here because of the definition of `\pgfmathsmuggle`.

```

81 \cs_new:Npn\__timechart_set_style_line_width:nn #1#2
82 {
83   \beginpgfkeys
84   \tikzset{#2}
85   \pgfmathsetlengthmacro{#1}{\pgflinewidth}
86   \pgfmathsmuggle #1
87   \endpgfkeys
88 }

```

(End of definition for `\__timechart_set_style_line_width:nn`.)

## 15.8 PGF keys

All PGF keys for this package are under `/timechart/`.

```

89 \pgfkeys{
90   /timechart/.cd,

```

Keys applicable to debugging.

```

91   debug/.code = {
92     \cs_set_eq:NN\__timechart_debug:n\__timechart_debug_real:n
93   },
94   no-debug/.code = {
95     \cs_set_eq:NN\__timechart_debug:n\use_none:n
96   },

```

Keys applicable to whole chart.

```

97   width/.initial=\textwidth,
98   ystep/.initial=-10pt,
99   minor~tick~interval/.initial=10,
100  major~tick~interval/.initial=50,
101  tolerance~start/.initial=5pt,
102  tolerance~finish/.initial=5pt,
103  beyond~length~start/.initial=5pt,
104  beyond~length~finish/.initial=5pt,
105  beyond~x~radius~start/.initial=4pt,
106  beyond~x~radius~finish/.initial=4pt,

```

Shortcuts for tolerance and ‘beyond’ indicators.

```

107  tolerance/.code = {
108    \pgfkeyssetvalue{/timechart/tolerance~start}{#1}
109    \pgfkeyssetvalue{/timechart/tolerance~finish}{#1}
110  },
111  beyond~length/.code = {
112    \pgfkeyssetvalue{/timechart/beyond~length~start}{#1}
113    \pgfkeyssetvalue{/timechart/beyond~length~finish}{#1}
114  },
115  beyond~x~radius/.code = {
116    \pgfkeyssetvalue{/timechart/beyond~x~radius~start}{#1}
117    \pgfkeyssetvalue{/timechart/beyond~x~radius~finish}{#1}
118  },

```

Keys applicable to the grid.

```

119  no~grid/.code = { \bool_set_false:N\__timechart_grid_bool },
120  grid~top~ysep/.initial={3pt},
121  grid~bottom~ysep/.initial={3pt},
122  grid/.style={},

```

Keys applicable to the axis.

```
123 axis~line/.style={
124   line~cap=rect,
125 },
126 axis~ysep/.initial=3pt,
127 axis/.is~choice,
128 axis~none/.code
129   = { \int_set:Nn\l__timechart_axis_int{0} },
130 axis~above/.code
131   = { \int_set:Nn\l__timechart_axis_int{1} },
132 axis~below/.code
133   = { \int_set:Nn\l__timechart_axis_int{2} },
134 no~axis/.code = { \int_set:Nn\l__timechart_axis_int{0} },
135 minor~tick/.style={},
136 minor~tick~length/.initial=1.5mm,
137 major~tick/.style={},
138 major~tick~length/.initial=3mm,
139 major~tick~label/.style={
140   inner~sep=0,
141   outer~sep=0,
142   anchor=mid~west,
143   rotate=90,
144 },
145 major~tick~eras/.is~choice,
146 major~tick~eras~none/.code
147   = { \int_set:Nn\l__timechart_major_tick_eras_int{0} },
148 major~tick~eras/all/.code
149   = { \int_set:Nn\l__timechart_major_tick_eras_int{1} },
150 major~tick~eras/outer/.code
151   = { \int_set:Nn\l__timechart_major_tick_eras_int{2} },
```

Keys applicable to intervals, texts, spaces, and legends.

```
152 no~autostep/.code = { \bool_set_false:Nz\l__timechart_autostep_bool },
153 ref/.initial={},
154 mark/.initial={},
155 marks/.forward~to=/timechart/mark,
156 circa~uncertainty/.initial=3,
157 interval~minimum~width/.initial=1pt,
158 interval~bar~color/.initial=black,
159 interval~bar~thickness/.initial=8pt,
160 interval~bar~node~name/.initial = {bar~node},
161 interval~mark~color/.initial=gray,
162 interval~label/.style={},
163 interval~label~centered/.style={/timechart/interval~label,text=white},
164 interval~label~centered~background/.style={/timechart/interval~label},
165 interval~label~baseline/.initial=-3pt,
166 interval~label~pos/.is~choice,
167 interval~label~pos/left/.code
168   = { \int_set:Nn\l__timechart_label_pos_int{0} },
169 interval~label~pos/center/.code
170   = { \int_set:Nn\l__timechart_label_pos_int{1} },
171 interval~label~pos/right/.code
172   = { \int_set:Nn\l__timechart_label_pos_int{2} },
173 interval~label~node~name/.initial = {label~node},
```



```

174 start~range/.is~choice,
175 start~range/fade/.code
176   = { \int_set:Nn\l__timechart_start_range_type_int{0} },
177 start~range/slant/.code
178   = { \int_set:Nn\l__timechart_start_range_type_int{1} },
179 finish~range/.is~choice,
180 finish~range/fade/.code
181   = { \int_set:Nn\l__timechart_finish_range_type_int{0} },
182 finish~range/slant/.code
183   = { \int_set:Nn\l__timechart_finish_range_type_int{1} },

```

Keys applicable only to texts.

```

184 text~node~name/.initial = {text~node},
185 text/.style={},
186 text~baseline/.initial=-3pt,
187 text~pos/.is~choice,
188 text~pos/left/.code = { \int_set:Nn\l__timechart_text_pos_int{0} },
189 text~pos/center/.code = { \int_set:Nn\l__timechart_text_pos_int{1} },
190 text~pos/right/.code = { \int_set:Nn\l__timechart_text_pos_int{2} },

```

Keys applicable only to legends.

```

191 legend~item~width/.initial=9mm,
192 legend~item~range~width/.initial=3mm,

```

Shortcuts for positioning.

```

193 left/.code = {
194   \int_set:Nn\l__timechart_label_pos_int{0}
195   \int_set:Nn\l__timechart_text_pos_int{0}
196 },
197 center/.code = {
198   \int_set:Nn\l__timechart_label_pos_int{1}
199   \int_set:Nn\l__timechart_text_pos_int{1}
200 },
201 right/.code = {
202   \int_set:Nn\l__timechart_label_pos_int{2}
203   \int_set:Nn\l__timechart_text_pos_int{2}
204 },
205 }

```

`\l__timechart_grid_bool` Boolean indicating whether the grid will be drawn. This variable is by default true but can be set false via the `/timechart/no grid` PGF key.

```

206 \bool_new:N\l__timechart_grid_bool
207 \bool_set_true:N\l__timechart_grid_bool

```

*(End of definition for `\l__timechart_grid_bool`.)*

`\l__timechart_axis_int` An integer indicating whether and where the axis will be drawn. This is set via the `/timechart/axis` PGF key.

```

208 \int_new:N\l__timechart_axis_int
209 \int_set:Nn\l__timechart_axis_int{1}

```

*(End of definition for `\l__timechart_axis_int`.)*

`\l__timechart_major_tick_eras_int` An integer indicating which major ticks will have era indicators. This is set via the `/timechart/major tick era` PGF key.

```
210 \int_new:N\l__timechart_major_tick_eras_int
211 \int_set:Nn\l__timechart_major_tick_eras_int{2}
```

*(End of definition for `\l__timechart_major_tick_eras_int`.)*

`\l__timechart_autostep_bool` Boolean indicating whether to automatically step the  $y$  coordinate after an interval, text, or space. This variable is by default true but can be set false via the `/timechart/no autostep` PGF key.

```
212 \bool_new:N\l__timechart_autostep_bool
213 \bool_set_true:N\l__timechart_autostep_bool
```

*(End of definition for `\l__timechart_autostep_bool`.)*

`\l__timechart_label_pos_int` An integer to hold the interval label position. This is set via the `/timechart/interval label pos` PGF key.

```
214 \int_new:N \l__timechart_label_pos_int
```

*(End of definition for `\l__timechart_label_pos_int`.)*

`\l__timechart_text_pos_int` An integer to hold the text position. This is set via the `/timechart/text pos` PGF key.

```
215 \int_new:N \l__timechart_text_pos_int
```

*(End of definition for `\l__timechart_text_pos_int`.)*

`\l__timechart_start_range_type_int` and `\l__timechart_finish_range_type_int` Integers to hold the type of the start/end ranges. These are set via the `/timechart/start range` and `/timechart/finish range` PGF keys.

```
216 \int_new:N \l__timechart_start_range_type_int
217 \int_new:N \l__timechart_finish_range_type_int
```

*(End of definition for `\l__timechart_start_range_type_int` and `\l__timechart_finish_range_type_int`.)*

## 15.9 Main environment

`timechart` The main environment.

`#1` : PGF keys to apply.

`#2` : Start year.

`#3` : End year.

```
218 \NewDocumentEnvironment{timechart}{0}{m m }
219 { \l__timechart_main_begin:nnn{#1}{#2}{#3} }
220 { \l__timechart_main_end: }
```

*(End of definition for `timechart`. This function is documented on page 8.)*

`\l__timechart_main_begin:nnn` This command uses values specified by PGF keys to make some necessary calculations to begin the chart.

```
221 \cs_new:Npn \l__timechart_main_begin:nnn #1#2#3
222 {
```

Process the supplied PGF keys and retrieve values that affect the chart as a whole.

```

223   \pgfkeys{
224     /timechart/.cd,
225     #1,
226     width/.get=\l__timechart_width_pgf,
227     ystep/.get=\l__timechart_ystep_pgf,
228     grid~top~ysep/.get=\l__timechart_grid_top_ysep_pgf,
229     grid~bottom~ysep/.get=\l__timechart_grid_bottom_ysep_pgf,
230     minor~tick~interval/.get=\l__timechart_minor_tick_interval_year,
231     major~tick~interval/.get=\l__timechart_major_tick_interval_year,
232   }

```

Start the TikZ picture and set up the necessary layers.

```

233   \tikzpicture
234   \pgfdeclarelayer{grid}
235   \pgfdeclarelayer{labelbg}
236   \pgfsetlayers{grid,labelbg,main}

```

Store the line width of the grid and axis, treating them as 0 pt if they are disabled.

```

237   \bool_if:NTF\l__timechart_grid_bool
238     {
239     \__timechart_set_style_line_width:nn
240       {\l__timechart_grid_line_width}
241       {/timechart/grid}
242     }
243     { \pgfmathsetlengthmacro{\l__timechart_grid_line_width}{0} }
244   \int_if_zero:NTF{ \l__timechart_axis_int }
245     {
246     \pgfmathsetlengthmacro{\l__timechart_axis_line_width}{0}
247     \pgfmathsetlengthmacro{\l__timechart_major_tick_line_width}{0}
248     \pgfmathsetlengthmacro{\l__timechart_minor_tick_line_width}{0}
249     }
250     {
251     \__timechart_set_style_line_width:nn
252       {\l__timechart_axis_line_width}
253       {/timechart/axis~line}
254     \__timechart_set_style_line_width:nn
255       {\l__timechart_major_tick_line_width}
256       {/timechart/major~tick}
257     \__timechart_set_style_line_width:nn
258       {\l__timechart_minor_tick_line_width}
259       {/timechart/minor~tick}
260     }

```

Store the start and finish years (ignoring circa, month, day), and then set up the conversion from years to  $x$  coordinates. `\l__timechart_x` is the  $x$ -distance corresponding to one year, and `yeartox` is the `pgfmath` function that does the conversion.

```

261   \__timechart_parse_date:NNn\l_tmpa_bool\l__timechart_start_year{#2}
262   \__timechart_parse_date:NNn\l_tmpa_bool\l__timechart_finish_year{#3}
263   \pgfmathsetmacro{\l__timechart_start_year}
264     {floor(\l__timechart_start_year)}
265   \pgfmathsetmacro{\l__timechart_finish_year}
266     {floor(\l__timechart_finish_year)}
267   \pgfmathsetmacro{\l__timechart_x}
268     {

```

```

269     (
270     \l__timechart_width_pgf
271     - max(
272     \l__timechart_grid_line_width,
273     \l__timechart_axis_line_width,
274     \l__timechart_major_tick_line_width,
275     \l__timechart_minor_tick_line_width
276     )
277     )/(\l__timechart_finish_year-\l__timechart_start_year)
278   }
279   \pgfkeys{
280     /pgf/declare-function={
281       yeartox(\n)=\l__timechart_x*(\n-\l__timechart_start_year);
282     },
283   }

```

Calculate the start and finish  $x$  coordinates.

```

284   \pgfmathsetmacro{\l__timechart_start_x}
285     {yeartox(\l__timechart_start_year)}
286   \pgfmathsetmacro{\l__timechart_finish_x}
287     {yeartox(\l__timechart_finish_year)}

```

Set up tracking of current  $y$  coordinate.

```

288   \pgfmathsetmacro{\l__timechart_current_y}{0}
289   \pgfmathsetmacro{\l__timechart_saved_y}{0}
290   \pgfmathsetmacro{\l__timechart_auto_reset_minimum_y}{-16000pt}
291   \pgfmathsetmacro{\l__timechart_auto_reset_maximum_y}{16000pt}

```

Calculate some years used in loops.

```

292   \pgfmathsetmacro{\l__timechart_start_plus_year}{
293     \l__timechart_start_year+\l__timechart_minor_tick_interval_year
294   }
295   \pgfmathsetmacro{\l__timechart_start_plusplus_year}{
296     \l__timechart_start_year+(2*\l__timechart_minor_tick_interval_year)
297   }
298   \pgfmathsetmacro{\l__timechart_end_minus_year}{
299     \l__timechart_finish_year-\l__timechart_minor_tick_interval_year
300   }

```

Begin a group and make available the user commands `\timechart...`. (The group will be ended by `\__timechart_main_end_user:.`)

```

301   \group_begin:
302   \cs_set_eq:NN\timechartinterval\__timechart_interval_user:0mmm
303   \cs_set_eq:NN\timecharttext\__timechart_text_user:0mm
304   \cs_set_eq:NN\timechartspace\__timechart_space_user:0
305   \cs_set_eq:NN\timechartsety\__timechart_set_y_user:m
306   \cs_set_eq:NN\timechartsavey\__timechart_save_y_user:
307   \cs_set_eq:NN\timechartresety\__timechart_reset_y_user:
308   \cs_set_eq:NN\timechartsetyminimumautoreset
309     \__timechart_set_y_minimum_auto_reset_user:m
310   \cs_set_eq:NN\timechartsetymaximumautoreset
311     \__timechart_set_y_maximum_auto_reset_user:m
312   \cs_set_eq:NN\timechartstepy\__timechart_step_y_user:0
313   \cs_set_eq:NN\timechartfinish\__timechart_main_end_user:
314   }

```

(End of definition for `\__timechart_main_begin:nnn`.)

`\__timechart_main_end:` Make sure the chart is complete and end the TikZ picture. `\__timechart_main_end_user:` ends the group begun by `\__timechart_main_begin:nnn`, so whether the user has *not* called it (as `\timechartfinish`) is equivalent to it being equal to `\timechartfinish`.

```
315 \cs_new:Npn\__timechart_main_end:
316   {
317     \cs_if_eq:NNT\timechartfinish\__timechart_main_end_user:
318       { \__timechart_main_end_user: }
319     \endtikzpicture
320   }
```

(End of definition for `\__timechart_main_end:.`)

`\__timechart_main_end_user:` End the group begun by `\__timechart_main_begin:nnn`, draw the axis and grid, and set the bounding box. This macro is made available as `\timechartfinish` inside the `timechart` environment.

```
321 \cs_new:Npn\__timechart_main_end_user:
322   {
```

The aim here is to set the bounding box (1) to fit horizontally the axis *not* including labels and the grid and (2) to fit vertically the axis including labels and the grid. All the ‘horizontal’ data is already known, and the ‘vertical’ data is determined by the *current* bounding box. So extract the ‘vertical’ data and then reset the bounding box.

```
323   \pgfextracty{\l__timechart_tmpa_dim}
324     { \pgfpointanchor{current~bounding~box}{south} }
325   \pgfextracty{\l__timechart_tmpb_dim}
326     { \pgfpointanchor{current~bounding~box}{north} }
327   \pgfresetboundingbox
```

If the timechart is empty, then the extracted  $y$  coordinates of ‘north’ and ‘south’ anchors of the bounding box will be  $-16\,000$  pt and  $16\,000$  pt respectively. Test for this and treat them as both having  $y$  coordinate 0 pt in this case.

```
328   \dim_compare:nNnTF{\l__timechart_tmpa_dim}>{\l__timechart_tmpb_dim}
329     {
330       \pgfmathsetmacro{\l__timechart_content_bottom_y}{0pt}
331       \pgfmathsetmacro{\l__timechart_content_top_y}{0pt}
332     }
333     {
334       \pgfmathsetmacro{\l__timechart_content_bottom_y}
335         {\l__timechart_tmpa_dim}
336       \pgfmathsetmacro{\l__timechart_content_top_y}
337         {\l__timechart_tmpb_dim}
338     }
```

Now draw the grid and axis if necessary and set the bounding box if not.

```
339   \bool_if:NTF{\l__timechart_grid_bool}
340     { \__timechart_grid_draw: }
341     { \__timechart_nogrid_bounding_box_set: }
342   \__timechart_axis_draw:
```

Finally, end the group begun by `\__timechart_main_begin:nnn`.

```
343   \group_end:
344 }
```

(End of definition for `\__timechart_main_end_user:.`)

## 15.10 Grid drawing

`\__timechart_grid_draw:` Draw the grid of the chart, assuming that the  $y$  coordinates of the top and bottom of the content have been calculated and stored in `\l__timechart_content_top_y` and `\l__timechart_content_bottom_y`. These variable will be updated to the grid top and bottom.

```

345 \cs_new:Npn\__timechart_grid_draw:
346   {
347     \pgfmathsetmacro{\l__timechart_content_bottom_y}{
348       \l__timechart_content_bottom_y-\l__timechart_grid_bottom_ysep_pgf
349     }
350     \pgfmathsetmacro{\l__timechart_content_top_y}{
351       \l__timechart_content_top_y+\l__timechart_grid_top_ysep_pgf
352     }
353     \pgfonlayer{ grid }
354     \scope[/timechart/grid]
355     \foreach \year in {
356       \l__timechart_start_plus_year,
357       \l__timechart_start_plusplus_year,
358       ...,
359       \l__timechart_end_minus_year
360     } {
361       \group_begin:

```

Only draw gridlines at major ticks.

```

362       \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
363       { Mod(\year,\l__timechart_major_tick_interval_year)==0 }
364       \bool_if:NT\l__timechart_tmpa_bool
365       {
366         \pgftransformshift{ \pgfpoint{yeartox(\year)}{0} }
367         \pgfpathmoveto{ \pgfpoint{0}{\l__timechart_content_top_y} }
368         \pgfpathlineto{ \pgfpoint{0}{\l__timechart_content_bottom_y} }
369         \pgfusepath{ draw }
370       }
371       \group_end:
372     }

```

Define and draw the grid node.

```

373     \__timechart_make_rectangle_node:nnnn
374     { \pgfpoint{\l__timechart_start_x}{\l__timechart_content_bottom_y} }
375     { \pgfpoint{\l__timechart_finish_x}{\l__timechart_content_top_y} }
376     { grid }
377     { \c_true_bool }
378   \endscope
379   \endpgfonlayer
380 }

```

*(End of definition for `\__timechart_grid_draw:`.)*

## 15.11 Axis drawing

`\__timechart_axis_draw:` Draw the axis, with large/small ticks and labels on appropriate years.

```

381 \cs_new:Npn\__timechart_axis_draw:
382   {
383     \group_begin:

```

```

384 \int_case:nn{ \l__timechart_axis_int }
385 {
386   {0}
387   { \prg_do_nothing: }
388   {1}
389   {
390     \pgftransformshift{
391       \pgfpoint{0}{
392         \l__timechart_content_top_y
393         +\pgfkeysvalueof{/timechart/axis~ysep}
394       }
395     }
396     \pgfmathsetmacro{\l__timechart_tick_orientation_pgf}{1}
397     \cs_set:Npn\l__timechart_tick_label_anchor_text
398       { mid~west }
399     \cs_set:Npn\l__timechart_zero_tick_before_label_anchor_text
400       { base~west }
401     \cs_set:Npn\l__timechart_zero_tick_after_label_anchor_text
402       { north~west }
403     \__timechart_axis_draw_aux:
404   }
405   {2}
406   {
407     \pgftransformshift{
408       \pgfpoint{0}{
409         \l__timechart_content_bottom_y
410         -\pgfkeysvalueof{/timechart/axis~ysep}
411       }
412     }
413     \pgfmathsetmacro{\l__timechart_tick_orientation_pgf}{-1}
414     \cs_set:Npn\l__timechart_tick_label_anchor_text
415       { mid~east }
416     \cs_set:Npn\l__timechart_zero_tick_before_label_anchor_text
417       { base~east }
418     \cs_set:Npn\l__timechart_zero_tick_after_label_anchor_text
419       { north~east }
420     \__timechart_axis_draw_aux:
421   }
422 }
423 \group_end:
424 }

```

(End of definition for \\_\_timechart\_axis\_draw:.)

\\_\_timechart\_axis\_draw\_aux: Draw the axis, with large/small ticks and labels on appropriate years, assuming that the  $y$  coordinate has been calculated and stored in \l\_\_timechart\_axis\_y.

```

425 \cs_new:Npn\__timechart_axis_draw_aux:
426 {
427   \pgfkeys{
428     /timechart/minor~tick~length/.get=\__timechart_minor_tick_length_pgf,
429     /timechart/major~tick~length/.get=\__timechart_major_tick_length_pgf,
430   }

```

Work out the first and last years which will have a major tick (since these are marked with the era).

```

431 \pgfmathsetmacro{\l__timechart_start_major_tick_year}
432 {
433   \l__timechart_start_year
434   -Mod(
435     \l__timechart_start_year,
436     \l__timechart_major_tick_interval_year
437   )
438 }
439 \pgfmathsetmacro{\l__timechart_start_major_tick_year}
440 {
441   ifthenelse(
442     \l__timechart_start_major_tick_year<\l__timechart_start_year,
443     \l__timechart_start_major_tick_year
444     +\l__timechart_major_tick_interval_year,
445     \l__timechart_start_major_tick_year
446   )
447 }
448 \pgfmathsetmacro{\l__timechart_start_plus_major_tick_year}
449 {
450   \l__timechart_start_major_tick_year
451   +\l__timechart_major_tick_interval_year
452 }
453 \pgfmathsetmacro{\l__timechart_finish_major_tick_year}
454 {
455   \l__timechart_finish_year
456   -Mod(
457     \l__timechart_finish_year,
458     \l__timechart_major_tick_interval_year
459   )
460 }

```

Loop over years and draw the minor ticks.

```

461 \foreach \year in {
462   \l__timechart_start_year,
463   \l__timechart_start_plus_year,
464   ...,
465   \l__timechart_finish_year
466 } {
467   \pgfmathsetmacro{\x}{yeartox(\year)}
468   \_timechart_pgfmathsetbool:nn
469   {\l__timechart_tmpa_bool}
470   {
471     Mod(
472       \year-\l__timechart_start_major_tick_year,
473       \l__timechart_major_tick_interval_year
474     )==0
475   }
476   \bool_if:NF\l__timechart_tmpa_bool
477   { \_timechart_axis_draw_minor_tick:N\x }
478 }

```

Loop over years and draw the major ticks.

```

479 \foreach \year in {
480   \l__timechart_start_major_tick_year,
481   \l__timechart_start_plus_major_tick_year,

```



```

482     ...,
483     \l__timechart_finish_major_tick_year
484 } {
485   \pgfmathsetmacro{\x}{yeartox(\year)}
486   \l__timechart_axis_draw_labelled_major_tick:NN\x\year
487 }

```

Define the axis line and define the axis node.

```

488 \l__timechart_draw_axis_line
489 \int_case:nn{\l__timechart_axis_int}
490 {
491   {1}
492   {
493     \pgfextracty{\l__timechart_tmpa_dim}
494     { \pgfpointanchor{current~bounding~box}{north} }
495     \pgfmathsetmacro{\l__timechart_axis_top_y}{\l__timechart_tmpa_dim}
496     \pgfmathsetmacro{\l__timechart_axis_bottom_y}{0}
497   }
498   {2}
499   {
500     \pgfextracty{\l__timechart_tmpa_dim}
501     { \pgfpointanchor{current~bounding~box}{south} }
502     \pgfmathsetmacro{\l__timechart_axis_top_y}{0}
503     \pgfmathsetmacro{\l__timechart_axis_bottom_y}{\l__timechart_tmpa_dim}
504   }
505 }
506 \l__timechart_make_rectangle_node:nnnn
507 { \pgfpoint{\l__timechart_start_x}{\l__timechart_axis_bottom_y} }
508 { \pgfpoint{\l__timechart_finish_x}{\l__timechart_axis_top_y} }
509 { axis }
510 { \c_false_bool }
511 }

```

*(End of definition for \l\_\_timechart\_axis\_draw\_aux:.)*

All the remaining axis-related macros (which begin `\l__timechart_axis_draw_`) assume that a transformation has been applied so that the axis line is at  $y = 0$ .

`\l__timechart_axis_draw_minor_tick:N` Draw an unlabelled tick at  $x$  coordinate #1.

```

512 \cs_new:Npn\l__timechart_axis_draw_minor_tick:N #1
513 {
514   \scope[/timechart/minor-tick]
515   \pgfpathmoveto{ \pgfpoint{#1}{0} }
516   \pgfpathlineto{
517     \pgfpoint{#1}{
518       \l__timechart_tick_orientation_pgf
519       *\l__timechart_minor_tick_length_pgf
520     }
521   }
522   \pgfusepath{draw}
523   \endscope
524 }

```

*(End of definition for \l\_\_timechart\_axis\_draw\_minor\_tick:N.)*

`\timechart_axis_draw_labelled_major_tick:NN` Draw a labelled major tick at  $x$  coordinate #1, with label for year #2, using the special epoch marker if the year is 0, and showing the era if and only if the year is for the first or last major tick. This macro assumes that `\__timechart_start_major_tick_year` and `\__timechart_finish_major_tick_year` have been calculated.

```

525 \cs_new:Npn\__timechart_axis_draw_labelled_major_tick:NN #1#2
526 {
527   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{#2==0}
528   \bool_if:NTF \l__timechart_tmpa_bool
529     { \__timechart_axis_draw_zero_tick:N #1 }
530     {
531       \int_case:nn { \l__timechart_major_tick_eras_int }
532       {
533         {0}{ \bool_set_false:N\l__timechart_tmpd_bool }
534         {1}{ \bool_set_true:N\l__timechart_tmpd_bool }
535         {2}
536       }
537       \__timechart_pgfmathsetbool:nn{\l__timechart_tmpb_bool}
538       { #2==\l__timechart_start_major_tick_year }
539       \__timechart_pgfmathsetbool:nn{\l__timechart_tmpc_bool}
540       { #2==\l__timechart_finish_major_tick_year }
541       \bool_set:Nn\l__timechart_tmpd_bool
542       { \l__timechart_tmpb_bool || \l__timechart_tmpc_bool }
543     }
544   }
545   \__timechart_axis_draw_major_tick:N #1
546   \__timechart_axis_draw_year_label:nnnnn
547   { #1 }
548   { #2 }
549   { \l__timechart_tmpd_bool }
550   { \l__timechart_tick_label_anchor_text }
551   { 0 }
552   {
553     \l__timechart_tick_orientation_pgf
554     *(\__timechart_major_tick_length_pgf+1mm)
555   }
556 }
557 }

```

*(End of definition for `\__timechart_axis_draw_labelled_major_tick:NN`.)*

`\__timechart_axis_draw_major_tick:N` Draw a major tick at  $x$  coordinate #1.

```

558 \cs_new:Npn\__timechart_axis_draw_major_tick:N #1
559 {
560   \scope[/timechart/major~tick]
561   \pgfpathmoveto{ \pgfpoint{#1}{0} }
562   \pgfpathlineto{
563     \pgfpoint{#1}{
564       \l__timechart_tick_orientation_pgf
565       *\__timechart_major_tick_length_pgf
566     }
567   }
568   \pgfusepath{draw}
569   \endscope
570 }

```

(End of definition for `\__timechart_axis_draw_major_tick:N`.)

`\__timechart_axis_draw_zero_tick:N` Draw a special (labelled) tick for year zero at  $x = \#1$ .

```
571 \cs_new:Npn\__timechart_axis_draw_zero_tick:N #1
572 {
573   \group_begin:
574   \pgftransformshift{ \pgfpoint{#1}{0} }
```

The mark is a cross made up of four arcs and the miter joins between them. The parameter `\r` is the arc radius. The parameter `\a` is how many degrees should be trimmed from the start/end of a quarter-circle to form each arc. Thus the length of the miter is dependent on `\a`.

```
575   \pgfmathsetmacro{\a}{5}
576   \pgfmathsetlengthmacro{\r}{1mm}
577   \pgfmathsetlengthmacro{\t}{\r*(cos(\a)-cos(90-\a))/(1-cos(90-\a))}
```

The drawing process is: move to the start of the tick, draw the tick, draw the four arcs, then draw a small part of the tick again. The last step is not mathematically necessary for a smooth join, but ensures that the join *appears* smooth.

```
578   \scope[/timechart/major~tick,line~join=miter]
579   \group_begin:
580   \pgftransformyscale{\l__timechart_tick_orientation_pgf}
581   \pgfpathmoveto{\pgfpointorigin}
582   \pgfpathlineto{\pgfpoint{0}{\__timechart_major_tick_length_pgf}}
583   \pgfpatharc{0}{90-\a}{\t-and-\r}
584   \pgfpatharc{270+\a}{360-\a}{\r}
585   \pgfpatharc{180+\a}{270-\a}{\r}
586   \pgfpatharc{90+\a}{180-\a}{\t-and-\r}
587   \pgfpathlineto{\pgfpoint{0}{\__timechart_major_tick_length_pgf-1pt}}
588   \pgfusepath{draw}
589   \group_end:
590   \endscope
```

There is no year 0, so label the zero mark with the 1 before and 1 after the epoch.

```
591   \__timechart_axis_draw_year_label:nnnnnn
592   { 0 }
593   { -1 }
594   { \c_true_bool }
595   { \l__timechart_zero_tick_before_label_anchor_text }
596   { -.5mm }
597   { \l__timechart_tick_orientation_pgf*5.5mm }
598   \__timechart_axis_draw_year_label:nnnnnn
599   { 0 }
600   { 1 }
601   { \c_true_bool }
602   { \l__timechart_zero_tick_after_label_anchor_text }
603   { .5mm }
604   { \l__timechart_tick_orientation_pgf*5.5mm }
605   \group_end:
606 }
```

(End of definition for `\__timechart_axis_draw_zero_tick:N`.)

`\__timechart_axis_draw_line` Draw the axis line itself.

```
607 \cs_new:Npn\__timechart_draw_axis_line
```

```

608 {
609   \scope[/timechart/axis~line]
610   \pgfpathmoveto{ \pgfpoint{\l__timechart_start_x}{0} }
611   \pgfpathlineto{ \pgfpoint{\l__timechart_finish_x}{0} }
612   \pgfusepath{draw}
613   \endscope
614 }

```

(End of definition for `\__timechart_axis_draw_line`.)

`\__timechart_axis_draw_year_label:nnnnnn`

Draw a year label.

**#1** :  $x$  coordinate.

**#2** : Year for label.

**#3** : Boolean literal indicating whether the era should be shown.

**#4** : Anchor for node.

**#5** :  $x$  offset (dimension).

**#6** :  $y$  offset (dimension).

```

615 \cs_new:Npn\__timechart_axis_draw_year_label:nnnnnn #1#2#3#4#5#6
616 {
617   \group_begin:
618   \pgftransformshift{ \pgfpoint{#1+#5}{#6} }
619   \pgfmathtruncatemacro{\absyear}{ abs(#2) }
620   \scope[/timechart/major~tick~label]
621   \bool_if:NTF #3
622   {
623     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{#2<0}
624     \bool_if:NTF\l__timechart_tmpa_bool
625     { \cs_set_eq:NN\__timechart_make_year:n\timechartmakebeforeyear }
626     { \cs_set_eq:NN\__timechart_make_year:n\timechartmakeafteryear }
627   }
628   { \cs_set_eq:NN\__timechart_make_year:n\use:n }
629   \__timechart_hsmash_pgfnode:nnnnn
630   {rectangle}
631   {#4}
632   {\__timechart_make_year:n{\absyear}}
633   {}
634   {}
635   \endscope
636   \group_end:
637 }

```

(End of definition for `\__timechart_axis_draw_year_label:nnnnnn`.)

`\timechartmakebeforeyear`

User-redefineable macros to format a year as before or after the epoch.

`\timechartmakeafteryear`

```

638 \cs_new:Npn\timechartmakebeforeyear #1
639 {
640   #1\nobreakspace\textsc{bce}
641 }
642 \cs_new:Npn\timechartmakeafteryear #1
643 {
644   #1\nobreakspace\textsc{ce}
645 }

```

(End of definition for `\timechartmakebeforeyear` and `\timechartmakeafteryear`. These functions are documented on page 13.)

## 15.12 Bounding box

`\_timechart_nogrid_bounding_box_set:` Set the bounding box when no grid is being drawn.

```
646 \cs_new:Npn\_timechart_nogrid_bounding_box_set:
647 {
648   \pgfpathmoveto
649     { \pgfpoint{\l__timechart_start_x}{\l__timechart_content_bottom_y} }
650   \pgfpathmoveto
651     { \pgfpoint{\l__timechart_finish_x}{\l__timechart_content_top_y} }
652   \pgfusepath{discard}
653 }
```

*(End of definition for \\_timechart\_nogrid\_bounding\_box\_set:.)*

## 15.13 Positioning

`\_timechart_set_y_user:m` Set current  $y$  coordinate to #1. This macro will be made available as `\timechartsety` in the `timechart` environment.

```
654 \cs_new:Npn\_timechart_set_y_user:m #1
655 {
656   \pgfmathsetmacro{\l__timechart_current_y}{#1}
657 }
```

*(End of definition for \\_timechart\_set\_y\_user:m.)*

`\_timechart_save_y_user:` Save the current  $y$  coordinate. This macro will be made available as `\timechartsavey` in the `timechart` environment.

```
658 \cs_new:Npn\_timechart_save_y_user:
659 {
660   \pgfmathsetmacro{\l__timechart_saved_y}{\l__timechart_current_y}
661 }
```

*(End of definition for \\_timechart\_save\_y\_user:.)*

`\_timechart_reset_y_user:` Set the current  $y$  coordinate to the last saved coordinate. This macro will be made available as `\timechartsresety` in the `timechart` environment.

```
662 \cs_new:Npn\_timechart_reset_y_user:
663 {
664   \pgfmathsetmacro{\l__timechart_current_y}{\l__timechart_saved_y}
665 }
```

*(End of definition for \\_timechart\_reset\_y\_user:.)*

`\_timechart_set_y_minimum_auto_reset_user:m` Set a  $y$  coordinate below which `\_timechart_step_y_user:` will automatically reset the current  $y$  coordinate to the last saved  $y$  coordinate. This macro will be made available as `\timechartssetyminimumautoreset` in the `timechart` environment.

```
666 \cs_new:Npn\_timechart_set_y_minimum_auto_reset_user:m #1
667 {
668   \pgfmathsetmacro{\l__timechart_auto_reset_minimum_y}{#1}
669 }
```

*(End of definition for \\_timechart\_set\_y\_minimum\_auto\_reset\_user:m.)*

`\__timechart_set_y_maximum_auto_reset_user:m` Set a  $y$  coordinate above which `\__timechart_step_y_user:` will automatically reset the current  $y$  coordinate to the last saved  $y$  coordinate. This macro will be made available as `\timechartssetymaximumautoreset` in the `timechart` environment.

```

670 \cs_new:Npn\__timechart_set_y_maximum_auto_reset_user:m #1
671 {
672   \pgfmathsetmacro{\l__timechart_auto_reset_maximum_y}{#1}
673 }

```

*(End of definition for `\__timechart_set_y_maximum_auto_reset_user:m`.)*

`\__timechart_step_y_user:0` Increment the current  $y$  coordinate by `#1` times the length specified in `/timechart/ystep`. This macro will be made available as `\timechartstepy` in the `timechart` environment.

```

674 \NewDocumentCommand{\__timechart_step_y_user:0}{ 0{1} }
675 {
676   \pgfmathsetmacro{\l__timechart_current_y}
677     {\l__timechart_current_y+#1*\l__timechart_ystep_pgf}
678   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
679     {
680     or(
681       \l__timechart_current_y<\l__timechart_auto_reset_minimum_y,
682       \l__timechart_current_y>\l__timechart_auto_reset_maximum_y
683     )
684   }
685   \bool_if:nT{\l__timechart_tmpa_bool}
686     { \pgfmathsetmacro{\l__timechart_current_y}{\l__timechart_saved_y} }
687 }

```

*(End of definition for `\__timechart_step_y_user:0`.)*

## 15.14 Bounds checking

`\__timechart_if_x_in_bounds:nT` Check if  $x$  coordinate `#1` is (strictly) within the bounds of the chart; if so, execute `#2`.

```

688 \cs_new:Npn\__timechart_if_x_in_bounds:nT #1#2
689 {
690   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{
691     and(
692       #1>=\l__timechart_start_x,
693       #1<=\l__timechart_finish_x
694     )
695   }
696   \bool_if:NT\l__timechart_tmpa_bool
697     {#2}
698 }
699

```

*(End of definition for `\__timechart_if_x_in_bounds:nT`.)*

`timechart_if_x_range_intersect_bounds_x:nnT` Check if the range between  $x$  coordinates `#1` and `#2` intersects the range of the bounds of the chart; if so, execute `#3`.

```

700 \cs_new:Npn\__timechart_if_x_range_intersect_bounds:nnT #1#2#3
701 {
702   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{
703     or(
704

```

```

705         and(
706             #2>=\l__timechart_start_x,
707             #2<=\l__timechart_finish_x
708         ),
709         and(
710             #1>=\l__timechart_start_x,
711             #1<=\l__timechart_finish_x
712         )
713     ),
714     and(
715         #1<\l__timechart_start_x,
716         #2>\l__timechart_finish_x
717     )
718 )
719 }
720 \bool_if:NT\l__timechart_tmpa_bool
721 {#3}
722 }

```

(End of definition for `\__timechart_if_x_range_intersect_bounds_x:nnT`.)

## 15.15 Date and date range parsing

`\__timechart_parse_date_or_daterange:NNNNNn`

Parse the text in #6, which should represent a date or date range, into parameters #1–#5.

- #1 : range indicator boolean variable.
- #2 : minimum circa indicator boolean variable.
- #3 : minimum variable.
- #4 : maximum circa indicator boolean variable.
- #5 : maximum variable.
- #6 : text to parse.

```

723 \cs_new:Npn\__timechart_parse_date_or_daterange:NNNNNn #1#2#3#4#5#6
724 {
725     \bool_set:Nn #1 {\__timechart_is_nondaterange_p:w #6/\q_stop}
726     \bool_set_inverse:N #1
727     \bool_if:nTF #1
728     { \__timechart_parse_range:w #2#3#4#5\q_mark #6\q_stop }
729     {
730         \__timechart_parse_date:NNn #2#3{#6}
731         \bool_set_eq:NN #4#2
732         \pgfmathsetmacro{#5}{#3}
733     }
734 }

```

(End of definition for `\__timechart_parse_date_or_daterange:NNNNNn`.)

`\__timechart_is_nondaterange_p:w`

To be called in the form `\__timechart_is_nondaterange_p:w⟨text⟩/\q_stop`. Return boolean true if and only if `⟨text⟩` (known to be either a date or date range) contains a range marker.

```

735 \cs_new:Npn\__timechart_is_nondaterange_p:w #1/#2\q_stop
736 {
737     \tl_if_empty_p:n{#2}
738 }

```

(End of definition for `\__timechart_is_nondaterange_p:w`.)

`\__timechart_parse_range:w` To be called in the form `\__timechart_parse_range:w<min><min><max><max>\q-mark<text>\q-stop`. Parse `<text>` (known to represent a date range) into minimum circa indicator boolean variable `<min>`, minimum variable `<min>`, maximum circa indicator boolean variable `<max>`, maximum variable `<max>`.

```

739 \cs_new:Npn\__timechart_parse_range:w #1#2#3#4\q_mark #5/#6\q_stop
740 {
741   \__timechart_parse_date:NNn #1#2{#5}
742   \__timechart_parse_date:NNn #3#4{#6}
743 }

```

*(End of definition for \\_\_timechart\_parse\_range:w.)*

`\__timechart_parse_date:NNn` Parse text (known to represent a date) into the supplied variables. Parameters `#1` and `#2` are the variables for (respectively) circa indicator boolean and date, and `#3` is the text to be parsed:

`#1` : circa indicator boolean variable.  
`#2` : date variable.  
`#3` : text to parse.

```

744 \cs_new:Npn\__timechart_parse_date:NNn #1#2#3
745 {
746   \bool_set:Nn #1 { \__timechart_is_circa_p:w #3c\q_stop }
747   \bool_if:NTF #1
748     { \__timechart_parse_circa_date:w #2\q_mark #3\q_stop }
749     { \__timechart_parse_noncirca_date:Nn #2{#3} }
750 }

```

*(End of definition for \\_\_timechart\_parse\_date:NNn.)*

`\__timechart_is_circa_p:w` To be called in the form `\__timechart_is_circa_p:w<text>c\q-stop`. Return boolean true if and only if `<text>` (known to be either a date or a date with a circa indicator) has a circa indicator.

```

751 \cs_new:Npn\__timechart_is_circa_p:w #1c#2\q_stop
752 {
753   \tl_if_empty_p:n{#1}
754 }

```

*(End of definition for \\_\_timechart\_is\_circa\_p:w.)*

`\__timechart_parse_circa_date:w` To be called in the form `\__timechart_parse_circa_date:w<var>\q-mark<text>\q-stop`. Parse `<text>` (known to represent a circa date) into the supplied variable. `#1` is the variable for the date and `#2` is the text to be parsed.

```

755 \cs_new:Npn\__timechart_parse_circa_date:w #1\q_mark c#2\q_stop
756 {
757   \__timechart_parse_noncirca_date:Nn #1{#2}
758 }

```

*(End of definition for \\_\_timechart\_parse\_circa\_date:w.)*

`\__timechart_parse_noncirca_date:w` To be called in the form `\__timechart_parse_noncirca_date:w<var>\q-mark<text>\q-stop`. Parse `<text>` (known to represent a non-circa date) into the supplied variable. `#1` is the variable for the date and `#2` is the text to be parsed.

```

759 \cs_new:Npn\__timechart_parse_noncirca_date:Nn #1#2
760 {

```



```

761 \bool_if:nTF { \__timechart_is_before_p:w #2-\q_stop }
762   { \__timechart_parse_before_date:w #1\q_mark #2\q_stop }
763   { \__timechart_parse_signed_date:w #1\q_mark #2-0-0\q_stop }
764 }

```

(End of definition for \\_\_timechart\_parse\_noncirca\_date:w.)

\\_\_timechart\_is\_before\_p:w To be called in the form \\_\_timechart\_is\_before\_p:w⟨text⟩-\q\_stop. Return boolean true if and only if ⟨text⟩ (known to be a date without a circa indicator) begins with a -.

```

765 \cs_new:Npn\__timechart_is_before_p:w #1-#2\q_stop
766 {
767   \tl_if_empty_p:n{#1}
768 }

```

(End of definition for \\_\_timechart\_is\_before\_p:w.)

\\_\_timechart\_parse\_before\_date:w To be called in the form \\_\_timechart\_parse\_before\_date:w⟨var⟩\q\_mark⟨text⟩\q\_stop. Parse ⟨text⟩ (known to represent a date with a leading -) into the supplied variable. #1 is the variable for the date and #2 is the text to be parsed.

```

769 \cs_new:Npn\__timechart_parse_before_date:w #1\q_mark-#2\q_stop
770 {
771   \__timechart_parse_signed_date:w #1-\q_mark #2-0-0\q_stop
772 }

```

(End of definition for \\_\_timechart\_parse\_before\_date:w.)

Now comes that actual parsing of an ISO-format date YYYY-MM-DD. The following macros serve as lookup tables for the number of days in the  $n$ -th month and the number of days in the year up to the start of the  $n$ -th month.

```

773 \cs_new:cpn{c__timechart_year_days_pgf}{365}
774 \cs_new:cpn{c__timechart_month_days_1_pgf}{31}
775 \cs_new:cpn{c__timechart_month_days_2_pgf}{28}
776 \cs_new:cpn{c__timechart_month_days_3_pgf}{31}
777 \cs_new:cpn{c__timechart_month_days_4_pgf}{30}
778 \cs_new:cpn{c__timechart_month_days_5_pgf}{31}
779 \cs_new:cpn{c__timechart_month_days_6_pgf}{30}
780 \cs_new:cpn{c__timechart_month_days_7_pgf}{31}
781 \cs_new:cpn{c__timechart_month_days_8_pgf}{31}
782 \cs_new:cpn{c__timechart_month_days_9_pgf}{30}
783 \cs_new:cpn{c__timechart_month_days_10_pgf}{31}
784 \cs_new:cpn{c__timechart_month_days_11_pgf}{30}
785 \cs_new:cpn{c__timechart_month_days_12_pgf}{31}
786 \cs_new:cpn{c__timechart_cumulative_days_1_pgf}{0}
787 \cs_new:cpn{c__timechart_cumulative_days_2_pgf}{31}
788 \cs_new:cpn{c__timechart_cumulative_days_3_pgf}{59}
789 \cs_new:cpn{c__timechart_cumulative_days_4_pgf}{90}
790 \cs_new:cpn{c__timechart_cumulative_days_5_pgf}{120}
791 \cs_new:cpn{c__timechart_cumulative_days_6_pgf}{151}
792 \cs_new:cpn{c__timechart_cumulative_days_7_pgf}{181}
793 \cs_new:cpn{c__timechart_cumulative_days_8_pgf}{212}
794 \cs_new:cpn{c__timechart_cumulative_days_9_pgf}{243}
795 \cs_new:cpn{c__timechart_cumulative_days_10_pgf}{273}
796 \cs_new:cpn{c__timechart_cumulative_days_11_pgf}{304}
797 \cs_new:cpn{c__timechart_cumulative_days_12_pgf}{334}

```

`\_timechart_parse_signed_date:w` To be called in the form `\_timechart_parse_positive_date:w⟨var⟩⟨sign⟩\q_mark⟨text⟩-0-0\q_stop`. Parse `⟨text⟩` (known to represent a non-circa date) into the supplied variable. #1 is the variable for the date and #2 is possibly -.

There is a trick in the parsing:

1. If `⟨text⟩` has the form `⟨year⟩-⟨month⟩-⟨day⟩`, then parameters #3, #4, and #5 will be, respectively, `⟨year⟩`, `⟨month⟩`, and `⟨day⟩-0-0`. Thus #5 will be evaluated by `pgfmath` to `⟨day⟩`.
2. If `⟨text⟩` has the form `⟨year⟩-⟨month⟩`, then parameters #3, #4, and #5 will be, respectively, `⟨year⟩`, `⟨month⟩`, and `0-0`. Thus #5 will be evaluated by `pgfmath` to 0.
3. If `⟨text⟩` is simply `⟨year⟩`, then parameters #3, #4, and #5 will be, respectively, `⟨year⟩`, 0, and 0.

```

798 \cs_new:Npn\_timechart_parse_signed_date:w #1#2\q_mark #3-#4-#5\q_stop
799 {
800   \pgfmathtruncatemacro{\_timechart_parsed_year_pgf}{#2#3}
801   \pgfmathtruncatemacro{\_timechart_parsed_month_pgf}{#4}
802   \pgfmathtruncatemacro{\_timechart_parsed_day_pgf}{#5}
803   \_timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
804     or(
805       \_timechart_parsed_month_pgf < 1,
806       \_timechart_parsed_month_pgf > 12,
807     )
808   }
809   \bool_if:NTF\l_tmpa_bool
810   {

```

*Case: no valid month is given.* Use only the year.

```

811   \pgfmathsetmacro{#1}{#2#3}
812   }
813   {

```

*Case: a valid month is given.* Get the number of days in the year, in the month, and in the year up to the month. Then check if the year is a leap year and, if so, make the appropriate adjustments.

```

814   \cs_set_eq:NN\l_timechart_year_days_pgf\c_timechart_year_days_pgf
815   \cs_set_eq:Nc\l_timechart_month_days_pgf
816     { c_timechart_month_days\_timechart_parsed_month_pgf _pgf }
817   \cs_set_eq:Nc\l_timechart_cumulative_days_pgf
818     { c_timechart_cumulative_days\_timechart_parsed_month_pgf _pgf }
819   \_timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
820     or(
821       Mod(\_timechart_parsed_year_pgf,400) == 0,
822       and(
823         Mod(\_timechart_parsed_year_pgf,4) == 0,
824         Mod(\_timechart_parsed_year_pgf,100) != 0
825       )
826     )
827   }
828   \bool_if:NT\l_tmpa_bool
829   {
830     \pgfmathsetmacro{\l_timechart_year_days_pgf}
831     { \l_timechart_year_days_pgf+1 }

```

```

832     \_timechart_pgfmathsetbool:nn{\l_tmpb_bool}
833     { \_timechart_parsed_month == 1 }
834     \bool_if:NF\l_tmpb_bool
835     {
836         \_timechart_pgfmathsetbool:nn{\l_tmpb_bool}
837         { \_timechart_parsed_month == 2 }
838         \bool_if:NF\l_tmpb_bool
839         {
840             \pgfmathsetmacro{\l_timechart_month_days_pgf}
841             { \l_timechart_month_days_pgf + 1 }
842         }
843         {
844             \pgfmathsetmacro{\l_timechart_cumulative_days_pgf}
845             { \l_timechart_cumulative_days_pgf + 1 }
846         }
847     }
848 }
849 \_timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
850     or(
851         \_timechart_parsed_day_pgf < 1,
852         \_timechart_parsed_day_pgf > \l_timechart_month_days_pgf,
853     )
854 }
855 \bool_if:NTF\l_tmpa_bool
856 {

```

*Sub-case: no valid day is given. Use only the year and month.*

```

857     \pgfmathsetmacro{#1}
858     {
859         #2#3
860         + \l_timechart_cumulative_days_pgf/\l_timechart_year_days_pgf
861     }
862 }
863 {

```

*Sub-case: a valid day is given. Use the year, month, and day.*

```

864     \pgfmathsetmacro{#1}
865     {
866         #2#3
867         + (
868             \l_timechart_cumulative_days_pgf
869             + \_timechart_parsed_day_pgf
870             )/\l_timechart_year_days_pgf
871     }
872 }
873 }
874 }

```

*(End of definition for \\_timechart\_parse\_signed\_date:w.)*

## 15.16 Interval drawing

### 15.16.1 Preliminaries

These boolean variables will be used to hold parsed data for the start and finish of an interval: whether it is a range, whether the beginning of that range is qualified by ‘circa’,

and whether the end of that range is qualified by ‘circa’.

```
875 \bool_new:N\l__timechart_start_is_range_bool
876 \bool_new:N\l__timechart_interval_start_min_circa_bool
877 \bool_new:N\l__timechart_interval_start_max_circa_bool
878 \bool_new:N\l__timechart_finish_is_range_bool
879 \bool_new:N\l__timechart_interval_finish_min_circa_bool
880 \bool_new:N\l__timechart_interval_finish_max_circa_bool
```

(End of definition for `\l__timechart_start_is_range_bool` and others.)

### 15.16.2 Error message definition

```
881 \msg_new:nnn{timechart}{interval_dates_invalid}
882 { Invalid~interval~dates:~#1~to~#2 }
```

### 15.16.3 Main macros

`\_timechart_interval_user:Ommm` Draw an interval. This macro will be made available as `\timechartinterval` inside the `timechart` environment. It is simply a wrapper around the internal `\_timechart_interval:nnnn` macro.

**#1** : PGF keys under `/timechart/` to apply.  
**#2** : Start year.  
**#3** : Finish year.  
**#4** : Label.

```
883 \NewDocumentCommand{\_timechart_interval_user:Ommm}{ O{} m m m }
884 {
885   \_timechart_interval:nnnn{#1}{#2}{#3}{#4}
886 }
```

(End of definition for `\_timechart_interval_user:Ommm`.)

`\_timechart_interval:nnnn` Internal macro for drawing an interval.

```
887 \cs_new:Npn \_timechart_interval:nnnn #1#2#3#4
888 {
889   \_timechart_debug:n{
890     *****
891   }
892   \_timechart_debug:n{Interval~#2~to~#3~"~#4"}
```

Open an group and parse the start and finish dates or date ranges.

```
893 \group_begin:
894 \_timechart_parse_date_or_daterange:NNNNNn
895 \l__timechart_start_is_range_bool
896 \l__timechart_interval_start_min_circa_bool
897 \l__timechart_interval_start_min_year
898 \l__timechart_interval_start_max_circa_bool
899 \l__timechart_interval_start_max_year
900 {#2}
901 \_timechart_parse_date_or_daterange:NNNNNn
902 \l__timechart_finish_is_range_bool
903 \l__timechart_interval_finish_min_circa_bool
904 \l__timechart_interval_finish_min_year
905 \l__timechart_interval_finish_max_circa_bool
906 \l__timechart_interval_finish_max_year
907 {#3}
```

Check the results of parsing.

```

908   \_timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
909     and(
910       \l_timechart_interval_start_min_year
911       <= \l_timechart_interval_start_max_year,
912       and(
913         \l_timechart_interval_start_max_year
914         <= \l_timechart_interval_finish_min_year,
915         \l_timechart_interval_finish_min_year
916         <= \l_timechart_interval_finish_max_year
917       )
918     )
919   }

```

Proceed if and only if the results of parsing are valid, otherwise output an error message. In both cases, close the group and, in the former case, step the  $y$  coordinate if appropriate.

```

920   \bool_if:NTF\l_tmpa_bool
921   {
922     \_timechart_interval_checked:nn{#1}{#4}
923     \group_end:
924     \bool_if:NT\l_timechart_autostep_bool
925     { \_timechart_step_y_user:0 }
926   }
927   {
928     \msg_error:nnnn{timechart}{interval_dates_invalid}{#2}{#3}
929     \group_end:
930   }
931 }

```

(End of definition for `\_timechart_interval:nnnn`.)

In `\_timechart_interval_user_aux:nnnn`, the start and finish dates were parsed and it has been checked that they satisfy

$$\begin{aligned}
 \l\_timechart\_interval\_start\_min\_year & \\
 & \leq \l\_timechart\_interval\_start\_max\_year \\
 & \leq \l\_timechart\_interval\_finish\_min\_year \\
 & \leq \l\_timechart\_interval\_finish\_max\_year.
 \end{aligned}$$

In the code that follows, horizontal coordinates `\l_timechart_interval_..._x` will be computed from `\l_timechart_interval_..._year`. Subsequently, only the former variables will be used.

`\_timechart_interval_checked:nn` Draw an interval using the parsed and checked start and finish dates  
**#1** : PGF keys under `/timechart/` to apply.  
**#2** : Label.

```

932 \cs_new:Npn \_timechart_interval_checked:nn #1#2
933 {

```

Process keys supplied locally and retrieve the only value needed at this stage.

```

934   \pgfqkeys{/timechart}{
935     #1,
936     circa-uncertainty/.get=\l_timechart_circa_uncertainty_year
937   }

```

Do the minimum amount of calculation necessary to check whether any part of interval is visible. This means calculating `\l__timechart_interval_start_min_x` and `\l__timechart_interval_finish_max_x` (including taking into account any ‘circa’).

```

938   \bool_if:NTF\l__timechart_interval_start_min_circa_bool
939   {
940     \pgfmathsetmacro{\l__timechart_interval_start_min_x}
941     { yeartox(\l__timechart_interval_start_min_year
942       - \l__timechart_circa_uncertainty_year) }
943   }
944   {
945     \pgfmathsetmacro{\l__timechart_interval_start_min_x}
946     { yeartox(\l__timechart_interval_start_min_year) }
947   }
948   \bool_if:NTF\l__timechart_interval_finish_max_circa_bool
949   {
950     \pgfmathsetmacro{\l__timechart_interval_finish_max_x}
951     { yeartox(\l__timechart_interval_finish_max_year
952       + \l__timechart_circa_uncertainty_year) }
953   }
954   {
955     \pgfmathsetmacro{\l__timechart_interval_finish_max_x}
956     { yeartox(\l__timechart_interval_finish_max_year) }
957   }

```

Proceed if some part of the interval is visible.

```

958   \__timechart_if_x_range_intersect_bounds:nnT
959   {\l__timechart_interval_start_min_x}
960   {\l__timechart_interval_finish_max_x}
961   { \__timechart_draw_visible_interval:nn{#1}{#2} }
962   }

```

*(End of definition for `\__timechart_interval_checked:nn`.)*

`\__timechart_start_beyond_bool` `\__timechart_finish_beyond_bool` Boolean variables to indicate whether the interval extends beyond the chart at the start or the finish more than the specified tolerance.

```

963   \bool_new:N\l__timechart_interval_start_beyond_bool
964   \bool_new:N\l__timechart_interval_finish_beyond_bool

```

*(End of definition for `\__timechart_start_beyond_bool` and `\__timechart_finish_beyond_bool`.)*

`\__timechart_draw_visible_interval:nn` Draw an interval of which some part is known to be visible.

**#1** : PGF keys

**#2** : Label

```

965   \cs_new:Npn\__timechart_draw_visible_interval:nn #1#2
966   {

```

Retrieve PGF key values.

```

967   \pgfqkeys{/timechart}{
968     tolerance~start/.get=\l__timechart_start_tolerance_pgf,
969     tolerance~finish/.get=\l__timechart_finish_tolerance_pgf,
970     beyond~length~start/.get=\l__timechart_start_beyond_length_pgf,
971     beyond~length~finish/.get=\l__timechart_finish_beyond_length_pgf,
972     beyond~x~radius~start/.get=\l__timechart_start_beyond_x_radius_pgf,
973     beyond~x~radius~finish/.get=\l__timechart_finish_beyond_x_radius_pgf,

```

```

974     ref/.get=\l__timechart_ref_text,
975     mark/.get=\l__timechart_mark_text,
976     interval~minimum~width/.get=\l__timechart_minimum_width_pgf,
977     interval~bar~color/.get=\l__timechart_bar_color,
978     interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
979     interval~mark~color/.get=\l__timechart_mark_color,
980     interval~label~baseline/.get=\l__timechart_text_baseline_pgf,
981     interval~label~node~name/.get=\l__timechart_interval_label_node_name,
982   }

```

Calculate the start and finish tolerance and beyond indicator  $x$  coordinates.

```

983   \pgfmathsetmacro{\l__timechart_start_tolerance_x}{
984     \l__timechart_start_x-(\l__timechart_start_tolerance_pgf)
985   }
986   \pgfmathsetmacro{\l__timechart_finish_tolerance_x}{
987     \l__timechart_finish_x+(\l__timechart_finish_tolerance_pgf)
988   }
989   \pgfmathsetmacro{\l__timechart_start_beyond_x}{
990     \l__timechart_start_x-(\l__timechart_start_beyond_length_pgf)
991   }
992   \pgfmathsetmacro{\l__timechart_finish_beyond_x}{
993     \l__timechart_finish_x+(\l__timechart_finish_beyond_length_pgf)
994   }

```

Do the remaining calculations: calculate  $\l__timechart\_interval\_start\_max\_x$  and  $\l__timechart\_interval\_finish\_min\_x$  (including taking into account any ‘circa’).

```

995   \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
996     {.5*\l__timechart_bar_thickness_pgf}
997   \cs_set:Npn\l__timechart_label_text{#2}
998   \bool_if:NTF\l__timechart_interval_start_max_circa_bool
999     {
1000     \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1001       { yeartox(\l__timechart_interval_start_max_year
1002         + \l__timechart_circa_uncertainty_year) }
1003     }
1004     {
1005     \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1006       { yeartox(\l__timechart_interval_start_max_year) }
1007     }
1008   \bool_if:NTF\l__timechart_interval_finish_min_circa_bool
1009     {
1010     \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1011       { yeartox(\l__timechart_interval_finish_min_year
1012         - \l__timechart_circa_uncertainty_year) }
1013     }
1014     {
1015     \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1016       { yeartox(\l__timechart_interval_finish_min_year) }
1017     }

```

It is possible that circa indicators have made  $\l__timechart\_start\_max\_x$  greater than  $\l__timechart\_interval\_finish\_min\_x$ . Check for this; if so, set them both to their average.

```

1018   \l__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1019   {
1020     \l__timechart_interval_start_max_x

```

```

1021         > \l__timechart_interval_finish_min_x
1022     }
1023 \bool_if:NT\l__timechart_tmpa_bool
1024 {
1025     \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1026     {
1027         .5*(
1028             \l__timechart_interval_start_max_x
1029             + \l__timechart_interval_finish_min_x
1030         )
1031     }
1032     \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1033     { \l__timechart_interval_start_max_x }
1034 }

```

Check whether the minimum width requirement is satisfied; if not, make an adjustment.

```

1035 \__timechart_pgfmthsetbool:nn{\l__timechart_tmpa_bool}
1036 {
1037     (\l__timechart_interval_finish_max_x
1038     -\l__timechart_interval_start_min_x)
1039     > \l__timechart_minimum_width_pgf
1040 }
1041 \bool_if:NF \l__timechart_tmpa_bool
1042 {
1043     \pgfmathsetmacro{\l__timechart_width_adjust}
1044     { .5*\l__timechart_minimum_width_pgf }
1045     \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1046     { \l__timechart_interval_start_max_x-\l__timechart_width_adjust }
1047     \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1048     { \l__timechart_interval_finish_min_x+\l__timechart_width_adjust }
1049     \pgfmathsetmacro{\l__timechart_interval_start_min_x}
1050     {
1051         min(
1052             \l__timechart_interval_start_min_x,
1053             \l__timechart_interval_start_max_x
1054         )
1055     }
1056     \pgfmathsetmacro{\l__timechart_interval_finish_max_x}
1057     {
1058         max(
1059             \l__timechart_interval_finish_max_x,
1060             \l__timechart_interval_finish_min_x
1061         )
1062     }
1063 }
1064 \__timechart_debug:n{Calculated+adjusted~x~coords:}
1065 \__timechart_debug:n{~
1066     \l__timechart_interval_start_min_x/\l__timechart_interval_start_max_x~
1067     to~
1068     \l__timechart_interval_finish_min_x/\l__timechart_interval_finish_max_x
1069 }

```

At this point, the  $x$ -coordinates of the interval have been established. First, decide whether the interval exceeds the chart by more than the tolerance at the start and the finish. If so, set the appropriate clipping path (this may be changed again later).



```

1070 \__timechart_pgfmathsetbool:nn{\l__timechart_interval_start_beyond_bool}
1071   {\l__timechart_interval_start_min_x<=\l__timechart_start_tolerance_x}
1072 \bool_if:NT\l__timechart_interval_start_beyond_bool
1073   {
1074     \cs_set_eq:NN
1075       \__timechart_interval_start_clip_path:
1076       \__timechart_interval_start_clip_path_beyond:
1077   }
1078 \__timechart_pgfmathsetbool:nn{\l__timechart_interval_finish_beyond_bool}
1079   {\l__timechart_interval_finish_max_x>=\l__timechart_finish_tolerance_x}
1080 \bool_if:NT\l__timechart_interval_finish_beyond_bool
1081   {
1082     \cs_set_eq:NN
1083       \__timechart_interval_finish_clip_path:
1084       \__timechart_interval_finish_clip_path_beyond:
1085   }

```

Compute how the interval will be drawn. There are three components:

- the *startrange*, comprising the part between `\l__timechart_interval_start_min_x` to `\l__timechart_interval_start_max_x`;
- the definite part: `\l__timechart_interval_start_max_x` to `\l__timechart_interval_finish_min_x`;
- the *finishrange*: `\l__timechart_interval_finish_min_x` to `\l__timechart_interval_finish_max_x`.

It is necessary to decide which of these will be drawn. The logic is complicated because any part may lie outside the timechart or be involved in the beyond indicator. Further, clipping is used for a ‘slant’ startrange or finishrange and for beyond indicators.

First of all, consider the definite part. Depending on where this is located, it may be unnecessary to compute or draw the startrange or finishrange, so some of the following macros may be set to do nothing.

```

1086 \__timechart_interval_definite_compute:
1087 \__timechart_interval_startrange_compute:
1088 \__timechart_interval_finishrange_compute:

```

All the data needed to draw the interval are now ready. Shift to the correct vertical coordinate and open a scope for drawing.

```

1089 \pgftransformshift{ \pgfpoint{0}{\l__timechart_current_y} }
1090 \pgfscope

```

First, do the computed clipping, for a slant startrange and/or a slant finishrange, or beyond indicators. Then draw the startrange, finish range, and then the ‘solid’ part, which may be only the definite part or may include a slant startrange or a slant finishrange. Finally, define the node for the bar, then do the labelling.

```

1091 \__timechart_interval_clip:
1092 \__timechart_interval_startrange_draw:
1093 \__timechart_interval_finishrange_draw:
1094 \__timechart_interval_solid_draw:
1095 \__timechart_interval_mark:
1096 \__timechart_interval_define_bar_node:
1097 \endpgfscope
1098 \__timechart_interval_label:
1099 }

```

(End of definition for `__timechart_draw_visible_interval:nn`.)

## 15.16.4 Computation

These macros determine precisely what is to be drawn. The aim is to minimize the number of drawing commands by avoiding drawing where areas that would be clipped.

### 15.16.4.1 Definite part of the interval

`\_timechart_interval_definite_compute:` Compute whether and where to draw the definite part of the interval. Two auxiliary macros, `\_timechart_interval_definite_compute_ii:` and `\_timechart_interval_definite_compute_ii:` are used, simply to avoid deep nesting of code.

First, check whether the entire definite part of the interval is located to the left of the start of the chart and that the interval does exceed the start tolerance. In this case, the only thing that might be drawn to appear to the left of the start will be the excess indicator drawn as part of the finish range, so there is no need to draw the definite part or do anything for the start range.

```

1100 \cs_new:Npn\_timechart_interval_definite_compute:
1101 {
1102   \_timechart_debug:n{Computing:~definite~part}
1103   \_timechart_pgfmathsetbool:nn{\l\_timechart_tmpa_bool}
1104   { \l\_timechart_interval_finish_min_x < \l\_timechart_start_x }
1105   \bool_if:nTF{
1106     \l\_timechart_tmpa_bool && \l\_timechart_interval_start_beyond_bool
1107   }
1108   {
1109     \cs_set_eq:NN
1110     \_timechart_interval_solid_draw:\prg_do_nothing:
1111     \cs_set_eq:NN
1112     \_timechart_interval_startrange_compute:\prg_do_nothing:
1113     \cs_set_eq:NN
1114     \_timechart_interval_startrange_draw:\prg_do_nothing:
1115   }
1116   {
1117     \_timechart_interval_definite_compute_ii:
1118   }
1119 }

```

Second, check whether the entire definite part of the interval is located to the right of the finish of the chart and that the interval does exceed the finish tolerance. In this case, the only thing that might be drawn to appear to the right of the finish will be the excess indicator drawn as part of the start range, so there is no need to draw the definite part or do anything for the finish range.

```

1120 \cs_new:Npn\_timechart_interval_definite_compute_ii:
1121 {
1122   \_timechart_debug:n{~(Stage~II)}
1123   \_timechart_pgfmathsetbool:nn{\l\_timechart_tmpa_bool}
1124   { \l\_timechart_interval_start_max_x > \l\_timechart_finish_x }
1125   \bool_if:nTF{
1126     \l\_timechart_tmpa_bool && \l\_timechart_interval_finish_beyond_bool
1127   }
1128   {
1129     \cs_set_eq:NN

```

```

1130     \_timechart_interval_solid_draw:\prg_do_nothing:
1131     \cs_set_eq:NN
1132     \_timechart_interval_finishrange_compute:\prg_do_nothing:
1133     \cs_set_eq:NN
1134     \_timechart_interval_finishrange_draw:\prg_do_nothing:
1135   }
1136   {
1137     \_timechart_interval_definite_compute_iii:
1138   }
1139 }

```

At this point, some part of the definite part of the interval has to be drawn.

```

1140 \cs_new:Npn\_timechart_interval_definite_compute_iii:
1141 {
1142   \_timechart_debug:n{~(Stage-III)}

```

If the start of the definite part of the interval is at or to the left of the start of the chart and the interval exceeds the start tolerance, set the start of the solid part to the position of the start excess indicator. In this case, nothing needs to be done for the start range. Otherwise, set the start of the solid part to the start of the definite part.

```

1143   \_timechart_pgfmathsetbool:nn{\l_timechart_tmpa_bool}
1144   { \l_timechart_interval_start_max_x <= \l_timechart_start_x }
1145   \bool_if:nTF{
1146     \l_timechart_tmpa_bool && \l_timechart_interval_start_beyond_bool
1147   }
1148   {
1149     \cs_set_eq:NN
1150     \l_timechart_interval_solid_start_x\l_timechart_start_beyond_x
1151     \cs_set_eq:NN
1152     \_timechart_interval_startrange_compute:\prg_do_nothing:
1153     \cs_set_eq:NN
1154     \_timechart_interval_startrange_draw:\prg_do_nothing:
1155   }
1156   {
1157     \cs_set_eq:NN
1158     \l_timechart_interval_solid_start_x
1159     \l_timechart_interval_start_max_x
1160   }

```

If the finish of the definite part of the interval is at or to the right of the finish of the chart and the interval exceeds the finish tolerance, set the finish of the solid part to the position of the finish excess indicator. In this case, nothing needs to be done for the start range. Otherwise, set the start of the solid part to the start of the definite part.

```

1161   \_timechart_pgfmathsetbool:nn{\l_timechart_tmpb_bool}
1162   { \l_timechart_interval_finish_min_x >= \l_timechart_finish_x }
1163   \bool_if:nTF{
1164     \l_timechart_tmpb_bool && \l_timechart_interval_finish_beyond_bool
1165   }
1166   {
1167     \cs_set_eq:NN
1168     \l_timechart_interval_solid_finish_x\l_timechart_finish_beyond_x
1169     \cs_set_eq:NN
1170     \_timechart_interval_finishrange_compute:\prg_do_nothing:
1171     \cs_set_eq:NN
1172     \_timechart_interval_finishrange_draw:\prg_do_nothing:

```

```

1173     }
1174     {
1175     \cs_set_eq:NN
1176     \l__timechart_interval_solid_finish_x
1177     \l__timechart_interval_finish_min_x
1178     }
1179 }

```

(End of definition for `\__timechart_interval_definite_compute:.`)

#### 15.16.4.2 Startrange

`\__timechart_interval_startrange_compute:` Compute how to draw the startrange of an interval. This macro will only be called if it is necessary to draw some part of the startrange.

```

1180 \cs_new:Npn\__timechart_interval_startrange_compute:
1181 {
1182   \__timechart_debug:n{Computing:~startrange}
1183   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1184   {
1185     \l__timechart_interval_start_min_x
1186     == \l__timechart_interval_start_max_x
1187   }
1188   \bool_if:NTF\l__timechart_tmpa_bool
1189   {
1190     \cs_set_eq:NN\__timechart_interval_startrange_draw:\prg_do_nothing:
1191   }
1192   {
1193     \__timechart_interval_startrange_compute_ii:
1194   }
1195 }

```

(End of definition for `\__timechart_interval_startrange_compute:.`)

`\__timechart_interval_startrange_compute_ii:` Compute how to draw the startrange of an interval. This macro will only be called if the start range is non-empty.

```

1196 \bool_new:N\l__timechart_interval_startrange_finish_beyond_bool
1197 \cs_new:Npn\__timechart_interval_startrange_compute_ii:
1198 {
1199   \__timechart_debug:n{~(Stage-II)}

```

First, if the interval extends beyond the start tolerance then (since some part of the startrange) is visible, the start excess indicator forms part of the range. In this case, the drawn range starts at the start of the timechart and the start excess indicator mirrors this ‘level’ of the range. Otherwise the drawn range starts at the extreme start of the interval at ‘level’ 0 and a start excess indicator is not drawn.

```

1200   \bool_if:NTF\l__timechart_interval_start_beyond_bool
1201   {
1202     \cs_set_eq:NN
1203     \l__timechart_interval_startrange_start_x\l__timechart_start_x
1204     \pgfmathsetmacro{\l__timechart_interval_startrange_start_level_pgf}
1205     {
1206       (\l__timechart_interval_startrange_start_x
1207       -\l__timechart_interval_start_min_x)
1208     /

```

```

1209         (\l__timechart_interval_start_max_x
1210         -\l__timechart_interval_start_min_x)
1211     }
1212 }
1213 {
1214     \cs_set_eq:NN
1215     \l__timechart_interval_startrange_start_x
1216     \l__timechart_interval_start_min_x
1217     \pgfmathsetmacro
1218     {\l__timechart_interval_startrange_start_level_pgf}{0}
1219 }

```

Second, if the interval extends beyond the finish tolerance and the start of the definite part of the interval is to the right of the finish of the timechart, the finish excess indicator forms part of the range. In this case, the drawn range finishes at the finish of the timechart and the excess indicator mirrors this ‘level’ of the range. Otherwise the drawn range finishes at the extreme finish of the interval at ‘level’ 0 and a finish excess indicator is not drawn.

```

1220     \l__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1221     { \l__timechart_interval_start_max_x > \l__timechart_finish_x }
1222     \bool_set:Nn\l__timechart_interval_startrange_finish_beyond_bool
1223     {
1224         \l__timechart_tmpa_bool
1225         && \l__timechart_interval_finish_beyond_bool
1226     }
1227     \bool_if:NTF\l__timechart_interval_startrange_finish_beyond_bool
1228     {
1229         \cs_set_eq:NN
1230         \l__timechart_interval_startrange_finish_x
1231         \l__timechart_finish_x
1232         \pgfmathsetmacro{\l__timechart_interval_startrange_finish_level_pgf}
1233         {
1234             (\l__timechart_interval_startrange_finish_x
1235             -\l__timechart_interval_start_min_x)
1236             /
1237             (\l__timechart_interval_start_max_x
1238             -\l__timechart_interval_start_min_x)
1239         }
1240     }
1241     {
1242         \cs_set_eq:NN
1243         \l__timechart_interval_startrange_finish_x
1244         \l__timechart_interval_start_max_x
1245         \pgfmathsetmacro
1246         {\l__timechart_interval_startrange_finish_level_pgf}{1}
1247     }
1248     \int_case:nn {\l__timechart_start_range_type_int}
1249     {
1250         {0}{
1251             \cs_set_eq:NN
1252             \l__timechart_interval_startrange_draw:
1253             \l__timechart_interval_startrange_draw_pseudofade:
1254         }
1255         {1}{
1256             \l__timechart_interval_startrange_slant_compute:

```

```

1257     }
1258   }
1259 }

```

(End of definition for `\_timechart_interval_startrange_compute_ii:`)

`\timechart_interval_startrange_slant_compute:` Compute the clipping and drawing necessary for a slant startrange.

```

1260 \cs_new:Npn \_timechart_interval_startrange_slant_compute:
1261 {
1262   \_timechart_debug:n{~(Startrange-slant)}
1263   \bool_if:NTF\l__timechart_interval_start_beyond_bool
1264   {
1265     \bool_if:NTF\l__timechart_interval_startrange_finish_beyond_bool
1266     {
1267       \cs_set_eq:NN
1268         \_timechart_interval_both_clip_path:
1269         \_timechart_interval_startrange_clip_path_beyond_both:
1270     }
1271     {
1272       \cs_set_eq:NN
1273         \_timechart_interval_start_clip_path:
1274         \_timechart_interval_startrange_clip_path_beyond_start:
1275     }
1276   }
1277   {
1278     \bool_if:NTF\l__timechart_interval_startrange_finish_beyond_bool
1279     {
1280       \cs_set_eq:NN
1281         \_timechart_interval_both_clip_path:
1282         \_timechart_interval_startrange_clip_path_beyond_finish:
1283     }
1284     {
1285       \cs_set_eq:NN
1286         \_timechart_interval_start_clip_path:
1287         \_timechart_interval_startrange_clip_path_start:
1288     }
1289   }
1290   \cs_if_eq:NNTF\_timechart_interval_solid_draw:\prg_do_nothing:
1291   {
1292     \cs_set_eq:NN
1293       \_timechart_interval_startrange_draw:
1294       \_timechart_interval_startfinishrange_draw_slant:
1295   }
1296   {
1297     \cs_set_eq:NN
1298       \l__timechart_interval_solid_start_x
1299       \l__timechart_start_beyond_x
1300     \cs_set_eq:NN
1301       \_timechart_interval_startrange_draw:
1302       \prg_do_nothing:
1303   }
1304 }

```

(End of definition for `\_timechart_interval_startrange_slant_compute:`)

### 15.16.4.3 Finishrange

`\_timechart_interval_finishrange_compute:` Compute how to draw the finishrange of an interval. This macro will only be called if it is necessary to draw some part of the finishrange.

```

1305 \cs_new:Npn\_timechart_interval_finishrange_compute:
1306 {
1307   \_timechart_debug:n{Computing:~finishrange}
1308   \_timechart_pgfmathsetbool:nn{\_timechart_tmpa_bool}
1309   {
1310     \_timechart_interval_finish_max_x
1311     == \_timechart_interval_finish_min_x
1312   }
1313   \bool_if:NTF\_timechart_tmpa_bool
1314   {
1315     \cs_set_eq:NN\_timechart_interval_finishrange_draw:\prg_do_nothing:
1316   }
1317   {
1318     \_timechart_interval_finishrange_compute_ii:
1319   }
1320 }

```

*(End of definition for \\_timechart\_interval\_finishrange\_compute:.)*

`\_timechart_interval_finishrange_compute_ii:` Compute how to draw the finishrange of an interval. This macro will only be called if the finishrange is non-empty.

```

1321 \bool_new:N\_timechart_interval_finishrange_start_beyond_bool
1322 \cs_new:Npn\_timechart_interval_finishrange_compute_ii:
1323 {
1324   \_timechart_debug:n{~(Stage-II)}

```

First, if the interval extends beyond the finish tolerance then (since some part of the finishrange) is visible, the finish excess indicator forms part of the range. In this case, the drawn range finishes at the finish of the timechart and the finish excess indicator mirrors this ‘level’ of the range. Otherwise the drawn range finishes at the extreme finish of the interval at ‘level’ 0 and a finish excess indicator is not drawn.

```

1325   \bool_if:NTF\_timechart_interval_finish_beyond_bool
1326   {
1327     \cs_set_eq:NN
1328     \_timechart_interval_finishrange_finish_x
1329     \_timechart_finish_x
1330     \pgfmathsetmacro{\_timechart_interval_finishrange_finish_level_pgf}
1331     {
1332       (\_timechart_interval_finishrange_finish_x
1333       -\_timechart_interval_finish_max_x)
1334       /
1335       (\_timechart_interval_finish_min_x
1336       -\_timechart_interval_finish_max_x)
1337     }
1338   }
1339   {
1340     \cs_set_eq:NN
1341     \_timechart_interval_finishrange_finish_x
1342     \_timechart_interval_finish_max_x
1343     \pgfmathsetmacro

```

```

1344     {\l__timechart_interval_finishrange_finish_level_pgf}{0}
1345   }
1346

```

Second, if the interval extends beyond the start tolerance and the finish of the definite part of the interval is to the left of the start of the timechart, the start excess indicator forms part of the range. In this case, the drawn range starts at the start of the timechart and the excess indicator mirrors this ‘level’ of the range. Otherwise the drawn range starts at the extreme start of the interval at ‘level’ 0 and a start excess indicator is not drawn.

```

1347   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1348   { \l__timechart_interval_finish_min_x < \l__timechart_start_x }
1349   \bool_set:Nn\l__timechart_interval_finishrange_start_beyond_bool
1350   {
1351     \l__timechart_tmpa_bool
1352     && \l__timechart_interval_start_beyond_bool
1353   }
1354   \bool_if:NTF\l__timechart_interval_finishrange_start_beyond_bool
1355   {
1356     \cs_set_eq:NN
1357     \l__timechart_interval_finishrange_start_x
1358     \l__timechart_start_x
1359     \pgfmathsetmacro{\l__timechart_interval_finishrange_start_level_pgf}
1360     {
1361       (\l__timechart_interval_finishrange_start_x
1362        -\l__timechart_interval_finish_max_x)
1363       /
1364       (\l__timechart_interval_finish_min_x
1365        -\l__timechart_interval_finish_max_x)
1366     }
1367   }
1368   {
1369     \cs_set_eq:NN
1370     \l__timechart_interval_finishrange_start_x
1371     \l__timechart_interval_finish_min_x
1372     \pgfmathsetmacro
1373     {\l__timechart_interval_finishrange_start_level_pgf}{1}
1374   }
1375   \int_case:nn {\l__timechart_finish_range_type_int}
1376   {
1377     {0}{
1378       \cs_set_eq:NN
1379       \__timechart_interval_finishrange_draw:
1380       \__timechart_interval_finishrange_draw_pseudofade:
1381     }
1382     {1}{
1383       \__timechart_interval_finishrange_slant_compute:
1384     }
1385   }
1386 }

```

*(End of definition for \\_\_timechart\_interval\_finishrange\_compute\_ii:.)*

`mechart_interval_finishrange_slant_compute:` Compute the clipping and drawing necessary for a slant finishrange.



```

1387 \cs_new:Npn \__timechart_interval_finishrange_slant_compute:
1388 {
1389   \__timechart_debug:n{~(Finishrange-slant)}
1390   \bool_if:NTF\l__timechart_interval_finish_beyond_bool
1391   {
1392     \bool_if:NTF\l__timechart_interval_finishrange_start_beyond_bool
1393     {
1394       \cs_set_eq:NN
1395         \__timechart_interval_both_clip_path:
1396         \__timechart_interval_finishrange_clip_path_beyond_both:
1397     }
1398     {
1399       \cs_set_eq:NN
1400         \__timechart_interval_finish_clip_path:
1401         \__timechart_interval_finishrange_clip_path_beyond_finish:
1402     }
1403   }
1404   {
1405     \bool_if:NTF\l__timechart_interval_finishrange_start_beyond_bool
1406     {
1407       \cs_set_eq:NN
1408         \__timechart_interval_both_clip_path:
1409         \__timechart_interval_finishrange_clip_path_beyond_start:
1410     }
1411     {
1412       \cs_set_eq:NN
1413         \__timechart_interval_finish_clip_path:
1414         \__timechart_interval_finishrange_clip_path_finish:
1415     }
1416   }
1417   \cs_if_eq:NNTF\__timechart_interval_solid_draw:\prg_do_nothing:
1418   {
1419     \cs_set_eq:NN
1420     \__timechart_interval_finishrange_draw:
1421     \__timechart_interval_startfinishrange_draw_slant:
1422   }
1423   {
1424     \cs_set_eq:NN
1425     \l__timechart_interval_solid_finish_x
1426     \l__timechart_finish_beyond_x
1427     \cs_set_eq:NN
1428     \__timechart_interval_finishrange_draw:
1429     \prg_do_nothing:
1430   }
1431 }

```

(End of definition for \\_\_timechart\_interval\_finishrange\_slant\_compute:.)

## 15.16.5 Drawing

### 15.16.5.1 Solid

\\_\_timechart\_interval\_solid\_draw: Draw the solid part of an interval.

```

1432 \cs_new:Npn \__timechart_interval_solid_draw:

```

```

1433 {
1434   \_timechart_debug:n{
1435     Drawing:~solid~\l__timechart_interval_solid_start_x~
1436     to~\l__timechart_interval_solid_finish_x
1437   }
1438   \pgfpathrectanglecorners{
1439     \pgfpoint
1440     {\l__timechart_interval_solid_start_x}
1441     {-\l__timechart_bar_half_thickness_pgf}
1442   }{
1443     \pgfpoint
1444     {\l__timechart_interval_solid_finish_x}
1445     {\l__timechart_bar_half_thickness_pgf}
1446   }
1447   \pgfsetfillcolor{\l__timechart_bar_color}
1448   \pgfusepath{fill}
1449 }

```

(End of definition for \\_timechart\_interval\_solid\_draw:.)

### 15.16.5.2 Start/finish range pseudofade

timechart\_interval\_startrange\_draw\_pseudofade: Draw a fading startrange and beyond indicators

```

1450 \cs_new:Npn\_timechart_interval_startrange_draw_pseudofade:
1451 {
1452   \_timechart_debug:n{Drawing:~startrange,~pseudofade}
1453   \_timechart_debug:n{~
1454     \l__timechart_interval_startrange_start_x~
1455     (\l__timechart_interval_startrange_start_level_pgf)~
1456     to~\l__timechart_interval_startrange_finish_x~
1457     (\l__timechart_interval_startrange_finish_level_pgf)
1458   }
1459   \_timechart_interval_draw_pseudofade:nnnnn
1460   {\l__timechart_interval_startrange_start_x}
1461   {\l__timechart_interval_startrange_finish_x}
1462   {\l__timechart_bar_color}
1463   {\l__timechart_interval_startrange_start_level_pgf}
1464   {\l__timechart_interval_startrange_finish_level_pgf}
1465   \bool_if:NT\l__timechart_interval_start_beyond_bool
1466   { \_timechart_interval_startrange_start_beyond_draw_fade: }
1467   \bool_if:NT\l__timechart_interval_startrange_finish_beyond_bool
1468   { \_timechart_interval_startrange_finish_beyond_draw_fade: }
1469 }

```

(End of definition for \\_timechart\_interval\_startrange\_draw\_pseudofade:.)

timechart\_interval\_startrange\_start\_beyond\_draw\_fade: Draw a 'faded' beyond indicator for a startrange that extends beyond the timechart start.

```

1470 \cs_new:Npn\_timechart_interval_startrange_start_beyond_draw_fade:
1471 {
1472   \_timechart_debug:n{Drawing:~startrange,~pseudofade-beyond~start}
1473   \_timechart_debug:n{~
1474     \l__timechart_start_beyond_x~to~\l__timechart_start_x
1475   }
1476   \pgfscope
1477   \pgfpathrectanglecorners{

```

```

1478     \pgfpoint
1479     {\l__timechart_start_beyond_x}
1480     {-\l__timechart_bar_half_thickness_pgf}
1481   }{
1482     \pgfpoint
1483     {\l__timechart_start_x}
1484     {\l__timechart_bar_half_thickness_pgf}
1485   }
1486   \pgfsetfillcolor{\l__timechart_bar_color}
1487   \pgfsetfillopacity
1488     {\l__timechart_interval_startrange_start_level_pgf}
1489   \pgfusepath{fill}
1490   \endpgfscope
1491 }

```

(End of definition for `\__timechart_interval_startrange_start_beyond_draw_fade:`.)

`interval_startrange_finish_beyond_draw_fade:` Draw a ‘faded’ beyond indicator for a startrange that extends beyond the timechart finish.

```

1492 \cs_new:Npn\__timechart_interval_startrange_finish_beyond_draw_fade:
1493 {
1494   \__timechart_debug:n{Drawing:~startrange,~pseudofade~beyond~finish}
1495   \__timechart_debug:n{~
1496     \l__timechart_finish_beyond_x~to~\l__timechart_finish_x
1497   }
1498   \pgfscope
1499   \pgfpathrectanglecorners{
1500     \pgfpoint
1501     {\l__timechart_finish_x}
1502     {-\l__timechart_bar_half_thickness_pgf}
1503   }{
1504     \pgfpoint
1505     {\l__timechart_finish_beyond_x}
1506     {\l__timechart_bar_half_thickness_pgf}
1507   }
1508   \pgfsetfillcolor{\l__timechart_bar_color}
1509   \pgfsetfillopacity
1510     {\l__timechart_interval_startrange_finish_level_pgf}
1511   \pgfusepath{fill}
1512   \endpgfscope
1513 }

```

(End of definition for `\__timechart_interval_startrange_finish_beyond_draw_fade:`.)

`chart_interval_finishrange_draw_pseudofade:` Draw a fading startrange and excess indicators

```

1514 \cs_new:Npn\__timechart_interval_finishrange_draw_pseudofade:
1515 {
1516   \__timechart_debug:n{Drawing:~finishrange,~pseudofade}
1517   \__timechart_debug:n{~
1518     \l__timechart_interval_finishrange_start_x~
1519     (\l__timechart_interval_finishrange_start_level_pgf)~
1520     to~\l__timechart_interval_finishrange_finish_x~
1521     (\l__timechart_interval_finishrange_finish_level_pgf)
1522   }

```

```

1523 \__timechart_interval_draw_pseudofade:nmmm
1524   {\l__timechart_interval_finishrange_start_x}
1525   {\l__timechart_interval_finishrange_finish_x}
1526   {\l__timechart_bar_color}
1527   {\l__timechart_interval_finishrange_start_level_pgf}
1528   {\l__timechart_interval_finishrange_finish_level_pgf}
1529 \bool_if:NT\l__timechart_interval_finish_beyond_bool
1530   { \__timechart_interval_finishrange_finish_beyond_draw_fade: }
1531 \bool_if:NT\l__timechart_interval_finishrange_start_beyond_bool
1532   { \__timechart_interval_finishrange_start_beyond_draw_fade: }
1533 }

```

(End of definition for \\_\_timechart\_interval\_finishrange\_draw\_pseudofade:.)

interval\_finishrange\_finish\_beyond\_draw\_fade: Draw a ‘faded’ beyond indicator for a finishrange that extends beyond the timechart finish.

```

1534
1535 \cs_new:Npn\__timechart_interval_finishrange_finish_beyond_draw_fade:
1536 {
1537   \__timechart_debug:n{Drawing:~finishrange,~pseudofade~beyond~finish}
1538   \__timechart_debug:n{~
1539     \l__timechart_finish_beyond_x~to~\l__timechart_finish_x
1540   }
1541   \pgfscope
1542   \pgfpathrectanglecorners{
1543     \pgfpoint
1544       {\l__timechart_finish_beyond_x}
1545       {-\l__timechart_bar_half_thickness_pgf}
1546   }{
1547     \pgfpoint
1548       {\l__timechart_finish_x}
1549       {\l__timechart_bar_half_thickness_pgf}
1550   }
1551   \pgfsetfillcolor{\l__timechart_bar_color}
1552   \pgfsetfillopacity
1553     {\l__timechart_interval_finishrange_finish_level_pgf}
1554   \pgfusepath{fill}
1555   \endpgfscope
1556 }

```

(End of definition for \\_\_timechart\_interval\_finishrange\_finish\_beyond\_draw\_fade:.)

interval\_finishrange\_start\_beyond\_draw\_fade: Draw a ‘faded’ beyond indicator for a finishrange that extends beyond the timechart start.

```

1557 \cs_new:Npn\__timechart_interval_finishrange_start_beyond_draw_fade:
1558 {
1559   \__timechart_debug:n{Drawing:~finishrange,~pseudofade~beyond~start}
1560   \__timechart_debug:n{~
1561     \l__timechart_start_beyond_x~to~\l__timechart_start_x
1562   }
1563   \pgfscope
1564   \pgfpathrectanglecorners{
1565     \pgfpoint
1566       {\l__timechart_start_x}

```

```

1567     {-\l__timechart_bar_half_thickness_pgf}
1568   }{
1569     \pgfpoint
1570     {\l__timechart_start_beyond_x}
1571     {\l__timechart_bar_half_thickness_pgf}
1572   }
1573   \pgfsetfillcolor{\l__timechart_bar_color}
1574   \pgfsetfillopacity
1575     {\l__timechart_interval_finishrange_start_level_pgf}
1576   \pgfusepath{fill}
1577   \endpgfscope
1578 }

```

(End of definition for `\__timechart_interval_finishrange_start_beyond_draw_fade:`.)

`\__timechart_interval_draw_pseudofade:nmmn` Draw an imitation fading from #1 to #2 with color #3, starting with opacity #4 and ending with opacity #5.

```

1579 \pgfmathsetmacro{\l__timechart_fadestep_pgf}{1}
1580 \cs_new:Npn\__timechart_interval_draw_pseudofade:nmmn #1#2#3#4#5
1581 {
1582   \__timechart_debug:n{Pseudofade:~#1~to~#2,~color~#3,~opacity~#4~to~#5}
1583   \group_begin:
1584   \pgfmathsetmacro{\start}{#1}
1585   \pgfmathsetmacro{\stop}{#2}
1586   \pgfmathsetmacro{\fadestep}
1587     {(\stop-\start)/(floor((\stop-\start)/\l__timechart_fadestep_pgf)+1)}
1588   \pgfmathsetmacro{\startnext}{\start+\fadestep}
1589   \pgfmathsetmacro{\stopprev}{\stop-\fadestep}
1590   \pgfmathsetmacro{\denominator}{abs(\stop-\start)+\fadestep}
1591   \foreach \xa in {\start,\startnext,...,\stopprev} {
1592     \pgfmathsetlengthmacro{\o}
1593       {((\stop-\xa)*#4+(\xa-\start+\fadestep)*#5)/\denominator}
1594     \pgfscope
1595     \pgfpathrectanglecorners{
1596       \pgfpoint{\xa}{-\l__timechart_bar_half_thickness_pgf}
1597     }{
1598       \pgfpoint{\xa+\fadestep}{\l__timechart_bar_half_thickness_pgf}
1599     }
1600     \pgfsetfillcolor{#3}
1601     \pgfsetfillopacity{\o}
1602     \pgfusepath{fill}
1603     \endpgfscope
1604   }
1605   \group_end:
1606 }

```

(End of definition for `\__timechart_interval_draw_pseudofade:nmmn`.)

### 15.16.5.3 Start/finish range slant

`chart_interval_startfinishrange_draw_slant:` Draw an already-clipped slanted start/finishrange and beyond indicators. This macro will only be called when there is no ‘solid’ part to drawn (for otherwise the solid part is extended to fill the start/finishrange), and so the only thing to draw is the start/finishrange. Hence the macro can simply fill everything

```

1607 \cs_new:Npn\__timechart_interval_startfinishrange_draw_slant:
1608 {
1609   \pgfpathrectanglecorners{
1610     \pgfpoint
1611       {\l__timechart_start_beyond_x}
1612       {-\l__timechart_bar_half_thickness_pgf}
1613   }{
1614     \pgfpoint
1615       {\l__timechart_finish_beyond_x}
1616       {\l__timechart_bar_half_thickness_pgf}
1617   }
1618   \pgfsetfillcolor{\l__timechart_bar_color}
1619   \pgfusepath{fill}
1620 }

```

*(End of definition for \\_\_timechart\_interval\_startfinishrange\_draw\_slant:.)*

## 15.16.6 Clipping

### 15.16.6.1 Start/finish no-clipping paths

\c\_\_timechart\_left\_far\_x  
 \c\_\_timechart\_right\_far\_x

No-clipping paths actually use the following  $x$ -coordinates, which are large in the positive and negative direction.

```

1621 \pgfmathsetmacro{\c__timechart_left_far_x}{-16000pt}
1622 \pgfmathsetmacro{\c__timechart_right_far_x}{16000pt}

```

*(End of definition for \c\_\_timechart\_left\_far\_x and \c\_\_timechart\_right\_far\_x.)*

\\_\_timechart\_interval\_start\_clip\_path\_none:

No clip path for start.

```

1623 \cs_new:Npn\__timechart_interval_start_clip_path_none:
1624 {
1625   \__timechart_debug:n{~Start~clip~path:~none}
1626   \pgfpathlineto{
1627     \pgfpoint
1628     {\c__timechart_left_far_x}
1629     {-\l__timechart_bar_thickness_pgf}
1630   }
1631   \pgfpathlineto{
1632     \pgfpoint
1633     {\c__timechart_left_far_x}
1634     {\l__timechart_bar_thickness_pgf}
1635   }
1636 }

```

*(End of definition for \\_\_timechart\_interval\_start\_clip\_path\_none:.)*

\\_\_timechart\_interval\_finish\_clip\_path\_none:

No clip path for finish.

```

1637 \cs_new:Npn\__timechart_interval_finish_clip_path_none:
1638 {
1639   \__timechart_debug:n{~Finish~clip~path:~none}
1640   \pgfpathmoveto{
1641     \pgfpoint
1642     {\c__timechart_right_far_x}
1643     {\l__timechart_bar_thickness_pgf}
1644   }

```

```

1645 \pgfpathlineto{
1646 \pgfpoint
1647 {\c__timechart_right_far_x}
1648 {-\l__timechart_bar_thickness_pgf}
1649 }
1650 }

```

(End of definition for \\_\_timechart\_interval\_finish\_clip\_path\_none:.)

### 15.16.6.2 Start/finish beyond indicator clipping paths

\_timechart\_interval\_start\_clip\_path\_beyond: Clip path for start beyond indicator.

```

1651 \cs_new:Npn\__timechart_interval_start_clip_path_beyond:
1652 {
1653 \__timechart_debug:n{~Start~clip~path:~beyond}
1654 \pgfpathlineto{
1655 \pgfpoint
1656 {\l__timechart_start_beyond_x}
1657 {-\l__timechart_bar_thickness_pgf}
1658 }
1659 \pgfpathlineto{
1660 \pgfpoint
1661 {\l__timechart_start_beyond_x}
1662 {-\l__timechart_bar_half_thickness_pgf}
1663 }
1664 \pgfpatharc
1665 {-90}
1666 {90}
1667 {\l__timechart_start_beyond_x_radius_pgf
1668 ~and~\l__timechart_bar_half_thickness_pgf}
1669 \pgfpathlineto{
1670 \pgfpoint
1671 {\l__timechart_start_beyond_x}
1672 {\l__timechart_bar_thickness_pgf}
1673 }
1674 }

```

(End of definition for \\_\_timechart\_interval\_start\_clip\_path\_beyond:.)

timechart\_interval\_finish\_clip\_path\_beyond: Clip path for finish beyond indicator.

```

1675 \cs_new:Npn\__timechart_interval_finish_clip_path_beyond:
1676 {
1677 \__timechart_debug:n{~Finish~clip~path:~beyond}
1678 \pgfpathmoveto{
1679 \pgfpoint
1680 {\l__timechart_finish_beyond_x}
1681 {\l__timechart_bar_thickness_pgf}
1682 }
1683 \pgfpathlineto{
1684 \pgfpoint
1685 {\l__timechart_finish_beyond_x}
1686 {\l__timechart_bar_half_thickness_pgf}
1687 }
1688 \pgfpatharc
1689 {90}

```

```

1690     {270}
1691     {\l__timechart_finish_beyond_x_radius_pgf
1692      ~and~\l__timechart_bar_half_thickness_pgf}
1693 \pgfpathlineto{
1694   \pgfpoint
1695     {\l__timechart_finish_beyond_x}
1696     {-\l__timechart_bar_thickness_pgf}
1697 }
1698 }

```

(End of definition for \\_\_timechart\_interval\_finish\_clip\_path\_beyond:.)

### 15.16.6.3 Startrange/finishrange clipping paths

\_\_timechart\_interval\_startrange\_clip\_path\_start: Clip path for startrange.

```

1699 \cs_new:Npn\__timechart_interval_startrange_clip_path_start:
1700 {
1701   \__timechart_debug:n{~Start~clip~path:~startrange}
1702   \pgfpathlineto{
1703     \pgfpoint
1704       {\l__timechart_interval_startrange_start_x}
1705       {-\l__timechart_bar_thickness_pgf}
1706   }
1707   \pgfpathlineto{
1708     \pgfpoint
1709       {\l__timechart_interval_startrange_start_x}
1710       {-\l__timechart_bar_half_thickness_pgf}
1711   }
1712   \pgfpathlineto{
1713     \pgfpoint
1714       {\l__timechart_interval_startrange_finish_x}
1715       {\l__timechart_bar_half_thickness_pgf}
1716   }
1717   \pgfpathlineto{
1718     \pgfpoint
1719       {\l__timechart_interval_startrange_finish_x}
1720       {\l__timechart_bar_thickness_pgf}
1721   }
1722 }

```

(End of definition for \\_\_timechart\_interval\_startrange\_clip\_path\_start:.)

\_\_timechart\_interval\_finishrange\_clip\_path\_finish: Clip path for finishrange.

```

1723 \cs_new:Npn\__timechart_interval_finishrange_clip_path_finish:
1724 {
1725   \__timechart_debug:n{~Finish~clip~path:~finishrange}
1726   \pgfpathmoveto{
1727     \pgfpoint
1728       {\l__timechart_interval_finishrange_finish_x}
1729       {\l__timechart_bar_thickness_pgf}
1730   }
1731   \pgfpathlineto{
1732     \pgfpoint
1733       {\l__timechart_interval_finishrange_finish_x}
1734       {\l__timechart_bar_half_thickness_pgf}

```



```

1735 }
1736 \pgfpathlineto{
1737   \pgfpoint
1738     {\l__timechart_interval_finishrange_start_x}
1739     {-\l__timechart_bar_half_thickness_pgf}
1740 }
1741 \pgfpathlineto{
1742   \pgfpoint
1743     {\l__timechart_interval_finishrange_start_x}
1744     {-\l__timechart_bar_thickness_pgf}
1745 }
1746 }

```

(End of definition for `\__timechart_interval_finishrange_clip_path_finish:`.)

#### 15.16.6.4 Startrange/finishrange clipping paths with ‘same side’ beyond indicator

`interval_startrange_clip_path_beyond_start:` Clip path for startrange with start beyond indicator.

```

1747 \cs_new:Npn \__timechart_interval_startrange_clip_path_beyond_start:
1748 {
1749   \__timechart_debug:n{~Start~clip~path:~startrange,~beyond~start}
1750   \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
1751     \l__timechart_bar_thickness_pgf
1752     * \l__timechart_interval_startrange_start_level_pgf
1753   }
1754   \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
1755     .5 * \l__timechart_tmpa_pgf
1756   }
1757   \pgfpathlineto{
1758     \pgfpoint
1759       {\l__timechart_start_beyond_x}
1760       {-\l__timechart_bar_thickness_pgf}
1761   }
1762   \pgfpathlineto{
1763     \pgfpoint
1764       {\l__timechart_start_beyond_x}
1765       {-\l__timechart_bar_half_thickness_pgf}
1766   }
1767   \pgfpatharc
1768     {-90}
1769     {90}
1770     {\l__timechart_start_beyond_x_radius_pgf
1771      ~and~\l__timechart_tmpb_pgf}
1772   \pgfpathlineto{
1773     \pgfpointadd{
1774       \pgfpoint
1775         {\l__timechart_start_x}
1776         {-\l__timechart_bar_half_thickness_pgf}
1777     }{
1778       \pgfpoint{0}{\l__timechart_tmpa_pgf}
1779     }
1780   }
1781   \pgfpathlineto{

```

```

1782     \pgfpoint
1783         {\l__timechart_interval_start_max_x}
1784         {\l__timechart_bar_half_thickness_pgf}
1785     }
1786     \pgfpathlineto{
1787         \pgfpoint
1788             {\l__timechart_interval_start_max_x}
1789             {\l__timechart_bar_thickness_pgf}
1790     }
1791 }

```

(End of definition for \\_\_timechart\_interval\_startrange\_clip\_path\_beyond\_start:.)

interval\_finishrange\_clip\_path\_beyond\_finish: Clip path for finishrange with finish beyond indicator.

```

1792 \cs_new:Npn\__timechart_interval_finishrange_clip_path_beyond_finish:
1793 {
1794     \__timechart_debug:n{~Finish-clip-path:~finishrange,~beyond~finish}
1795     \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
1796         \l__timechart_bar_thickness_pgf
1797         * \l__timechart_interval_finishrange_finish_level_pgf
1798     }
1799     \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
1800         .5 * \l__timechart_tmpa_pgf
1801     }
1802     \pgfpathmoveto{
1803         \pgfpoint
1804             {\l__timechart_finish_beyond_x}
1805             {\l__timechart_bar_thickness_pgf}
1806     }
1807     \pgfpathlineto{
1808         \pgfpoint
1809             {\l__timechart_finish_beyond_x}
1810             {\l__timechart_bar_half_thickness_pgf}
1811     }
1812     \pgfpatharc
1813         {90}
1814         {270}
1815         {\l__timechart_finish_beyond_x_radius_pgf
1816         ~and~\l__timechart_tmpb_pgf}
1817     \pgfpathlineto{
1818         \pgfpointadd{
1819             \pgfpoint
1820                 {\l__timechart_finish_x}
1821                 {\l__timechart_bar_half_thickness_pgf}
1822         }{
1823             \pgfpoint{0}{-\l__timechart_tmpa_pgf}
1824         }
1825     }
1826     \pgfpathlineto{
1827         \pgfpoint
1828             {\l__timechart_interval_finish_min_x}
1829             {-\l__timechart_bar_half_thickness_pgf}
1830     }
1831     \pgfpathlineto{
1832         \pgfpoint

```

```

1833         {\l__timechart_interval_finish_min_x}
1834         {-\l__timechart_bar_thickness_pgf}
1835     }
1836 }

```

(End of definition for `\__timechart_interval_finishrange_clip_path_beyond_finish:`)

### 15.16.6.5 Startrange/finishrange clipping paths with ‘opposite side’ beyond indicator

`\__timechart_interval_startrange_clip_path_beyond_finish:` Clip path for startrange with finish beyond indicator. This macro should only be executed as `\__timechart_interval_both_clip_path:` and specifies a complete path.

```

1837 \cs_new:Npn\__timechart_interval_startrange_clip_path_beyond_finish:
1838 {
1839     \__timechart_debug:n{~Two-sided~clip~path:~startrange,~beyond~finish}
1840     \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
1841         \l__timechart_bar_thickness_pgf
1842         *\l__timechart_interval_startrange_finish_level_pgf
1843     }
1844     \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
1845         .5*\l__timechart_tmpc_pgf
1846     }
1847     \pgfpathmoveto{
1848         \pgfpointadd{
1849             \pgfpoint
1850             {\l__timechart_finish_beyond_x}
1851             {-\l__timechart_bar_half_thickness_pgf}
1852         }{
1853             \pgfpoint{0}{\l__timechart_tmpc_pgf}
1854         }
1855     }
1856     \pgfpatharc
1857     {90}
1858     {270}
1859     {\l__timechart_start_beyond_x_radius_pgf
1860     ~and~\l__timechart_tmpd_pgf}
1861     \pgfpathlineto{
1862         \pgfpoint
1863         {\l__timechart_finish_beyond_x}
1864         {-\l__timechart_bar_thickness_pgf}
1865     }
1866     \pgfpathlineto{
1867         \pgfpoint
1868         {\l__timechart_interval_start_min_x}
1869         {-\l__timechart_bar_thickness_pgf}
1870     }
1871     \pgfpathlineto{
1872         \pgfpoint
1873         {\l__timechart_interval_start_min_x}
1874         {-\l__timechart_bar_half_thickness_pgf}
1875     }
1876     \pgfpathlineto{
1877         \pgfpointadd{
1878             \pgfpoint

```

```

1879         {\l__timechart_finish_x}
1880         {-\l__timechart_bar_half_thickness_pgf}
1881     }{
1882     \pgfpoint{0}{\l__timechart_tmpc_pgf}
1883     }
1884 }
1885 }

```

(End of definition for `\__timechart_interval_startrange_clip_path_beyond_finish:`)

`interval_finishrange_clip_path_beyond_start:` Clip path for finishrange with start beyond indicator. This macro should only be executed as `\__timechart_interval_both_clip_path:` and specifies a complete path.

```

1886 \cs_new:Npn\__timechart_interval_finishrange_clip_path_beyond_start:
1887 {
1888     \__timechart_debug:n{~Two-sided~clip~path:~finishrange,~beyond-start}
1889     \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
1890         \l__timechart_bar_thickness_pgf
1891         * \l__timechart_interval_finishrange_start_level_pgf
1892     }
1893     \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
1894         .5 * \l__timechart_tmpc_pgf
1895     }
1896     \pgfpathmoveto{
1897         \pgfpointadd{
1898             \pgfpoint
1899             {\l__timechart_start_beyond_x}
1900             {\l__timechart_bar_half_thickness_pgf}
1901         }{
1902             \pgfpoint{0}{-\l__timechart_tmpc_pgf}
1903         }
1904     }
1905     \pgfpatharc
1906     {-90}
1907     {90}
1908     {\l__timechart_start_beyond_x_radius_pgf
1909     ~and~\l__timechart_tmpd_pgf}
1910     \pgfpathlineto{
1911         \pgfpoint
1912         {\l__timechart_start_beyond_x}
1913         {\l__timechart_bar_thickness_pgf}
1914     }
1915     \pgfpathlineto{
1916         \pgfpoint
1917         {\l__timechart_interval_finish_max_x}
1918         {\l__timechart_bar_thickness_pgf}
1919     }
1920     \pgfpathlineto{
1921         \pgfpoint
1922         {\l__timechart_interval_finish_max_x}
1923         {\l__timechart_bar_half_thickness_pgf}
1924     }
1925     \pgfpathlineto{
1926         \pgfpointadd{
1927             \pgfpoint

```

```

1928         {\l__timechart_start_x}
1929         {\l__timechart_bar_half_thickness_pgf}
1930     }{
1931     \pgfpoint{0}{-\l__timechart_tmpc_pgf}
1932     }
1933 }
1934 }

```

(End of definition for `\__timechart_interval_finishrange_clip_path_beyond_start:`)

### 15.16.6.6 Startrange/finishrange clipping paths with ‘both sides’ beyond indicator

`\__timechart_interval_startrange_clip_path_beyond_both:`

Clip path for startrange with start and finish beyond indicators. This macro should only be executed as `\__timechart_interval_both_clip_path:` and specifies a complete path.

```

1935 \cs_new:Npn\__timechart_interval_startrange_clip_path_beyond_both:
1936 {
1937     \__timechart_debug:n{~Two-sided~clip~path:~startrange,~beyond~both}
1938     \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
1939     \l__timechart_bar_thickness_pgf
1940     *\l__timechart_interval_startrange_start_level_pgf
1941     }
1942     \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
1943     .5*\l__timechart_tmpa_pgf
1944     }
1945     \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
1946     \l__timechart_bar_thickness_pgf
1947     *\l__timechart_interval_startrange_finish_level_pgf
1948     }
1949     \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
1950     .5*\l__timechart_tmpc_pgf
1951     }
1952     \pgfpathmoveto{
1953     \pgfpointadd{
1954     \pgfpoint
1955     {\l__timechart_finish_beyond_x}
1956     {-\l__timechart_bar_half_thickness_pgf}
1957     }{
1958     \pgfpoint{0}{\l__timechart_tmpc_pgf}
1959     }
1960     }
1961     \pgfpatharc
1962     {90}
1963     {270}
1964     {\l__timechart_finish_beyond_x_radius_pgf
1965     ~and~\l__timechart_tmpd_pgf}
1966     \pgfpathlineto{
1967     \pgfpoint
1968     {\l__timechart_finish_beyond_x}
1969     {-\l__timechart_bar_thickness_pgf}
1970     }
1971     \pgfpathlineto{
1972     \pgfpoint

```

```

1973         {\l__timechart_start_beyond_x}
1974         {-\l__timechart_bar_thickness_pgf}
1975     }
1976     \pgfpathlineto{
1977         \pgfpoint
1978         {\l__timechart_start_beyond_x}
1979         {-\l__timechart_bar_half_thickness_pgf}
1980     }
1981     \pgfpatharc
1982     {-90}
1983     {90}
1984     {\l__timechart_start_beyond_x_radius_pgf
1985     ~and~\l__timechart_tmpb_pgf}
1986     \pgfpathlineto{
1987         \pgfpointadd{
1988             \pgfpoint
1989             {\l__timechart_start_x}
1990             {-\l__timechart_bar_half_thickness_pgf}
1991         }{
1992             \pgfpoint{0}{\l__timechart_tmpa_pgf}
1993         }
1994     }
1995     \pgfpathlineto{
1996         \pgfpointadd{
1997             \pgfpoint
1998             {\l__timechart_finish_x}
1999             {-\l__timechart_bar_half_thickness_pgf}
2000         }{
2001             \pgfpoint{0}{\l__timechart_tmpc_pgf}
2002         }
2003     }
2004 }

```

(End of definition for \\_\_timechart\_interval\_startrange\_clip\_path\_beyond\_both:.)

interval\_finishrange\_clip\_path\_beyond\_both:

Clip path for finishrange with start and finish beyond indicators. This macro should only be executed as \\_\_timechart\_interval\_both\_clip\_path: and specifies a complete path.

```

2005 \cs_new:Npn\__timechart_interval_finishrange_clip_path_beyond_both:
2006 {
2007     \__timechart_debug:n{~Two-sided~clip~path:~finishrange,~beyond~both}
2008     \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
2009         \l__timechart_bar_thickness_pgf
2010         *\l__timechart_interval_finishrange_finish_level_pgf
2011     }
2012     \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
2013         .5*\l__timechart_tmpa_pgf
2014     }
2015     \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
2016         \l__timechart_bar_thickness_pgf
2017         *\l__timechart_interval_finishrange_start_level_pgf
2018     }
2019     \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
2020         .5*\l__timechart_tmpc_pgf

```

```

2021 }
2022 \pgfpathmoveto{
2023   \pgfpointadd{
2024     \pgfpoint
2025       {\l__timechart_start_beyond_x}
2026       {\l__timechart_bar_half_thickness_pgf}
2027   }{
2028     \pgfpoint{0}{-\l__timechart_tmpc_pgf}
2029   }
2030 }
2031 \pgfpatharc
2032   {-90}
2033   {90}
2034   {\l__timechart_start_beyond_x_radius_pgf
2035     ~and~\l__timechart_tmpd_pgf}
2036 \pgfpathlineto{
2037   \pgfpoint
2038     {\l__timechart_start_beyond_x}
2039     {\l__timechart_bar_thickness_pgf}
2040 }
2041 \pgfpathlineto{
2042   \pgfpoint
2043     {\l__timechart_finish_beyond_x}
2044     {\l__timechart_bar_thickness_pgf}
2045 }
2046 \pgfpathlineto{
2047   \pgfpoint
2048     {\l__timechart_finish_beyond_x}
2049     {\l__timechart_bar_half_thickness_pgf}
2050 }
2051 \pgfpatharc
2052   {90}
2053   {270}
2054   {\l__timechart_finish_beyond_x_radius_pgf
2055     ~and~\l__timechart_tmpb_pgf}
2056 \pgfpathlineto{
2057   \pgfpointadd{
2058     \pgfpoint
2059       {\l__timechart_finish_x}
2060       {\l__timechart_bar_half_thickness_pgf}
2061   }{
2062     \pgfpoint{0}{-\l__timechart_tmpa_pgf}
2063   }
2064 }
2065 \pgfpathlineto{
2066   \pgfpointadd{
2067     \pgfpoint
2068       {\l__timechart_start_x}
2069       {\l__timechart_bar_half_thickness_pgf}
2070   }{
2071     \pgfpoint{0}{-\l__timechart_tmpc_pgf}
2072   }
2073 }
2074 }

```

(End of definition for `\__timechart_interval_finishrange_clip_path_beyond_both:`)

### 15.16.6.7 Core clipping macros

`\__timechart_interval_clip:` Clip the interval. The internal macro `\__timechart_interval_both_clip_path:` may be redefined from its default.

```
2075 \cs_new:Npn\__timechart_interval_clip:
2076 {
2077   \__timechart_debug:n{Clipping:}
2078   \pgfinterruptboundingbox
2079   \__timechart_interval_both_clip_path:
2080   \pgfpathclose
2081   \pgfusepath{clip}
2082   \endpgfinterruptboundingbox
2083 }
```

(End of definition for `\__timechart_interval_clip:`)

`\__timechart_interval_both_clip_path:` Specify a clipping path for the interval. If either of the macros `\__timechart_interval_start_clip_path:` is `\__timechart_interval_finish_clip_path:` is defined, it is used, otherwise the respective defaults are used.

```
2084 \cs_new:Npn\__timechart_interval_both_clip_path:
2085 {
2086   \cs_if_exist_use:NF\__timechart_interval_finish_clip_path:
2087   { \__timechart_interval_finish_clip_path_none: }
2088   \cs_if_exist_use:NF\__timechart_interval_start_clip_path:
2089   { \__timechart_interval_start_clip_path_none: }
2090 }
```

(End of definition for `\__timechart_interval_both_clip_path:`)

### 15.16.7 Marks

```
2091 \msg_new:nnn{timechart}{interval_mark_outside}
2092 { Ignoring~mark~outside~interval~definite~part~at~date~#1 }
```

`\__timechart_interval_mark:` Draw marks on an interval.

```
2093 \cs_new:Npn\__timechart_interval_mark:
2094 {
2095   \pgfscope
2096   \bool_if:NTF\__timechart_interval_start_beyond_bool
2097   {
2098     \pgfmathsetmacro{\__timechart_interval_mark_min_x}
2099     { max(\__timechart_interval_start_max_x,\__timechart_start_x) }
2100   }
2101   {
2102     \cs_set_eq:NN\__timechart_interval_mark_min_x\__timechart_interval_start_max_x
2103   }
2104   \bool_if:NTF\__timechart_interval_finish_beyond_bool
2105   {
2106     \pgfmathsetmacro{\__timechart_interval_mark_max_x}
2107     { min(\__timechart_interval_finish_min_x,\__timechart_finish_x) }
2108   }
2109   {
```



```

2110     \cs_set_eq:NN\l__timechart_interval_mark_max_x\l__timechart_interval_finish_min_x
2111   }
2112   \pgfsetstrokecolor{\l__timechart_mark_color}
2113   \foreach \year in \l__timechart_mark_text
2114   {
2115     \pgfmathsetmacro{\l__timechart_mark_x}{yeartox(\year)}
2116     \l__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
2117     {
2118       and (
2119         \l__timechart_mark_x >= \l__timechart_interval_mark_min_x,
2120         \l__timechart_mark_x <= \l__timechart_interval_mark_max_x
2121       )
2122     }
2123     \bool_if:NTF\l__timechart_tmpa_bool
2124     {
2125       \pgfpathmoveto{
2126         \pgfpoint
2127           {\l__timechart_mark_x}
2128           {-\l__timechart_bar_half_thickness_pgf}
2129       }
2130       \pgfpathlineto{
2131         \pgfpoint
2132           {\l__timechart_mark_x}
2133           {\l__timechart_bar_half_thickness_pgf}
2134       }
2135       \pgfusepath{draw}
2136     }
2137     { \msg_warning:nne{timechart}{interval_mark_outside}{\year} }
2138   }
2139   \endpgfscope
2140 }

```

(End of definition for \l\_\_timechart\_interval\_mark:.)

### 15.16.8 Bar node

\l\_\_timechart\_interval\_define\_bar\_node: Define a node with into which the bar fits exactly.

```

2141 \cs_new:Npn\l__timechart_interval_define_bar_node:
2142 {
2143   \l__timechart_make_rectangle_node:nnnn
2144   {
2145     \pgfpoint
2146       {\l__timechart_interval_start_min_x}
2147       {-\l__timechart_bar_half_thickness_pgf}
2148   }{
2149     \pgfpoint
2150       {\l__timechart_interval_finish_max_x}
2151       {\l__timechart_bar_half_thickness_pgf}
2152   }
2153   {\pgfkeysvalueof{/timechart/interval-bar-node-name}}
2154   {\c_false_bool}
2155 }

```

(End of definition for \l\_\_timechart\_interval\_define\_bar\_node:.)

### 15.16.9 Label

`\__timechart_interval_label:` Place the label for the item.

```
2156 \cs_new:Npn\__timechart_interval_label:
2157 {
2158   \str_if_empty:NF \l__timechart_label_text
2159   {
2160     \pgfinterruptboundingbox
2161     \int_case:nn {\l__timechart_label_pos_int}
2162     {
2163       {0}{ \__timechart_interval_label_left: }
2164       {1}{ \__timechart_interval_label_center: }
2165       {2}{ \__timechart_interval_label_right: }
2166     }
2167     \endpgfinterruptboundingbox
2168   }
2169 }
```

*(End of definition for \\_\_timechart\_interval\_label:.)*

`\__timechart_interval_label_left:` Place the label on the left of the interval.

```
2170 \cs_new:Npn\__timechart_interval_label_left:
2171 {
2172   \__timechart_if_x_in_bounds:nT{\l__timechart_interval_start_min_x}
2173   {
2174     \group_begin:
2175     \pgftransformshift{
2176       \pgfpoint
2177       {\l__timechart_interval_start_min_x}
2178       {\l__timechart_text_baseline_pgf}
2179     }
2180     \node[/timechart/interval-label,anchor=base~east]
2181     (\l__timechart_interval_label_node_name)
2182     at (0,0)
2183     {
2184       \__timechart_make_ref:NN
2185       \l__timechart_ref_text
2186       \l__timechart_label_text
2187     };
2188     \group_end:
2189   }
2190 }
```

*(End of definition for \\_\_timechart\_interval\_label\_left:.)*

`\__timechart_interval_label_right:` Place the label on the right of the interval.

```
2191 \cs_new:Npn\__timechart_interval_label_right:
2192 {
2193   \__timechart_if_x_in_bounds:nT{\l__timechart_interval_finish_max_x}
2194   {
2195     \group_begin:
2196     \pgftransformshift{
2197       \pgfpoint
2198       {\l__timechart_interval_finish_max_x}
2199       {\l__timechart_text_baseline_pgf}
```

```

2200     }
2201     \node[/timechart/interval~label,anchor=base~west]
2202         (\l__timechart_interval_label_node_name)
2203         at (0,0)
2204         {
2205             \__timechart_make_ref:NN
2206             \l__timechart_ref_text
2207             \l__timechart_label_text
2208         };
2209     \group_end:
2210 }
2211 }

```

(End of definition for \\_\_timechart\_interval\_label\_right:.)

\\_\_timechart\_interval\_label\_center: Place the label at the center of the interval.

```

2212 \cs_new:Npn\__timechart_interval_label_center:
2213 {
2214     \pgfmathsetlengthmacro{\l__timechart_label_anchor_x}
2215     {
2216         .5*max(\l__timechart_interval_start_max_x,\l__timechart_start_x)
2217         + .5*min(\l__timechart_interval_finish_min_x,\l__timechart_finish_x)
2218     }

```

First draw the ‘background label’ on the layer below the bar.

```

2219     \group_begin:
2220     \pgftransformshift{
2221         \pgfpoint
2222             {\l__timechart_label_anchor_x}
2223             {\l__timechart_text_baseline_pgf}
2224     }
2225     \pgfonlayer{labelbg}
2226     \node[
2227         /timechart/interval~label~centered~background,
2228         anchor=base
2229     ]
2230     at (0,0)
2231     { \l__timechart_label_text };
2232     \endpgfonlayer
2233     \group_end:

```

Then draw the label on top of the bar, clipping it to the bar outline.

```

2234     \pgfscope
2235     \pgfpathrectanglecorners{
2236         \pgfpoint
2237             {\l__timechart_interval_start_min_x}
2238             {-\l__timechart_bar_half_thickness_pgf}
2239     }{
2240         \pgfpoint
2241             {\l__timechart_interval_finish_max_x}
2242             {\l__timechart_bar_half_thickness_pgf}
2243     }
2244     \pgfusepath{clip}
2245     \group_begin:
2246     \pgftransformshift{

```

```

2247     \pgfpoint
2248       {\l__timechart_label_anchor_x}
2249       {\l__timechart_text_baseline_pgf}
2250   }
2251   \node[
2252     /timechart/interval-label~centered,
2253     anchor=base
2254   ]
2255     (\l__timechart_interval_label_node_name)
2256     at (0,0)
2257     {
2258       \__timechart_make_ref:NN
2259       \l__timechart_ref_text
2260       \l__timechart_label_text
2261     };
2262   \group_end:
2263   \endpgfscope
2264 }

```

(End of definition for \\_\_timechart\_interval\_label\_center:.)

## 15.17 Text

\\_\_timechart\_text\_user:0mm Place text. This macro will be made available as \timecharttext inside the timechart environment.

**#1** : PGF keys under /timechart/ to apply.

**#2** : Year at which to place text.

**#3** : Text.

```

2265 \NewDocumentCommand{\__timechart_text_user:0mm}{ O{} m m }
2266 {
2267   \str_if_empty:nF{#3}{
2268     \group_begin:
2269     \pgfmathsetmacro{\l__timechart_text_x}{yeartox(#2)}

```

Process keys supplied locally and retrieve initially needed keys.

```

2270   \pgfqkeys{/timechart}{
2271     #1,
2272     tolerance~start/.get=\l__timechart_start_tolerance_pgf,
2273     tolerance~finish/.get=\l__timechart_finish_tolerance_pgf,
2274   }
2275   \__timechart_if_x_in_bounds:nT{\l__timechart_text_x}
2276   {

```

Retrieve other needed keys.

```

2277     \pgfqkeys{/timechart}{
2278       ref/.get=\l__timechart_ref_text,
2279       text~node~name/.get=\l__timechart_node_name_text,
2280       text~baseline/.get=\l__timechart_text_baseline_pgf,
2281     }

```

Shift to the correct vertical coordinate and place the text.

```

2282     \pgftransformshift{
2283       \pgfpoint{0}{\l__timechart_current_y}
2284     }

```

```

2285     \pgfinterruptboundingbox
2286     \group_begin:
2287     \pgftransformshift{
2288       \pgfpoint{\l__timechart_text_x}{\l__timechart_text_baseline_pgf}
2289     }
2290     \cs_set:Npn\l__timechart_text{#3}
2291     \int_case:nn {\l__timechart_text_pos_int}
2292     {
2293       {0}{ \cs_set:Npn\l__timechart_node_anchor_text{base~east} }
2294       {1}{ \cs_set:Npn\l__timechart_node_anchor_text{base} }
2295       {2}{ \cs_set:Npn\l__timechart_node_anchor_text{base~west} }
2296     }
2297     \node[/timechart/text,anchor=\l__timechart_node_anchor_text]
2298     (\l__timechart_node_name_text)
2299     at (0,0)
2300     {
2301       \__timechart_make_ref:NN
2302       \l__timechart_ref_text
2303       \l__timechart_text
2304     };
2305     \group_end:
2306     \endpgfinterruptboundingbox
2307   }
2308   \group_end:
2309 }

```

Since the text itself does not affect the bounding box, create a space (which will handle the automatic step).

```

2310   \__timechart_space_user:0[#1]
2311 }

```

*(End of definition for \\_\_timechart\_text\_user:0mm.)*

## 15.18 Space

`\__timechart_space_user:0` Create a vertical space as if there were an interval at the current coordinate. This macro will be made available as `\timechartspace` inside the `timechart` environment.

```

2312 \NewDocumentCommand{\__timechart_space_user:0}{ 0{} }
2313 {
2314   \group_begin:

```

Process keys supplied locally and retrieve the one needed value.

```

2315   \pgfqkeys{/timechart}{
2316     #1,
2317     interval-bar-thickness/.get=\l__timechart_bar_thickness_pgf,
2318   }
2319   \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
2320   { .5*\l__timechart_bar_thickness_pgf }

```

Shift to the correct vertical coordinate and create the space.

```

2321   \pgftransformshift{
2322     \pgfpoint{0}{\l__timechart_current_y}
2323   }
2324   \pgfpathmoveto{
2325     \pgfpoint{0}{-\l__timechart_bar_half_thickness_pgf}

```

```

2326 }
2327 \pgfpathmoveto{
2328   \pgfpoint{0}{\l__timechart_bar_half_thickness_pgf}
2329 }
2330 \pgfusepath{discard}
2331 \group_end:
2332 \bool_if:NT\l__timechart_autostep_bool{
2333   \__timechart_step_y_user:0
2334 }
2335 }

```

(End of definition for `\__timechart_space_user:0`.)

## 15.19 Legends

`\timechartlegenditem` Draw a bar suitable for use in a legend, applying style in #1.

```

2336 \NewDocumentCommand{\timechartlegenditem}{ 0{} }
2337 {
2338   \__timechart_legend_aux:nn{#1}{
2339     \pgfmathsetlengthmacro{\l__timechart_interval_solid_start_x}{0}
2340     \pgfmathsetlengthmacro{\l__timechart_interval_solid_finish_x}
2341       {\l__timechart_legenditem_width_pgf}
2342     \__timechart_interval_solid_draw:
2343   }
2344 }

```

(End of definition for `\timechartlegenditem`. This function is documented on page 14.)

`\timechartlegendstartrange` Draw a bar with start/finish range, suitable for use in a legend, applying style in #1.  
`\timechartlegendfinishrange`

```

2345 \NewDocumentCommand{\timechartlegendstartrange}{ 0{} }
2346 {
2347   \__timechart_legend_aux:nn{#1}{
2348     \pgfmathsetlengthmacro
2349       {\l__timechart_interval_startrange_start_x}{0}
2350     \pgfmathsetlengthmacro{\l__timechart_interval_startrange_finish_x}
2351       {\l__timechart_legenditem_range_width_pgf}
2352     \pgfmathsetlengthmacro
2353       {\l__timechart_interval_startrange_start_level_pgf}{0}
2354     \pgfmathsetlengthmacro
2355       {\l__timechart_interval_startrange_finish_level_pgf}{1}
2356     \pgfmathsetlengthmacro{\l__timechart_interval_solid_finish_x}
2357       {\l__timechart_legenditem_width_pgf}
2358     \int_case:nn {\l__timechart_start_range_type_int}
2359       {
2360         {0}{
2361           \pgfmathsetlengthmacro{\l__timechart_interval_solid_start_x}
2362             {\l__timechart_legenditem_range_width_pgf}
2363           \__timechart_interval_startrange_draw_pseudofade:
2364         }
2365         {1}{
2366           \pgfmathsetlengthmacro{\l__timechart_interval_solid_start_x}
2367             {\l__timechart_interval_startrange_start_x}
2368           \pgfinterruptboundingbox
2369           \__timechart_interval_finish_clip_path_none:

```

```

2370         \_timechart_interval_startrange_clip_path_start:
2371         \pgfpathclose
2372         \pgfusepath{clip}
2373         \endpgfinterruptboundingbox
2374     }
2375 }
2376 \_timechart_interval_solid_draw:
2377 }
2378 }
2379 \NewDocumentCommand{\timechartlegendfinishrange}{ 0{ } }
2380 {
2381     \_timechart_legend_aux:nn{#1}{
2382         \pgfmathsetlengthmacro{\l_timechart_interval_finishrange_start_x}
2383         {
2384             \l_timechart_legenditem_width_pgf
2385             -\l_timechart_legenditem_range_width_pgf
2386         }
2387         \pgfmathsetlengthmacro{\l_timechart_interval_finishrange_finish_x}
2388         {\l_timechart_legenditem_width_pgf}
2389         \pgfmathsetlengthmacro
2390         {\l_timechart_interval_finishrange_start_level_pgf}{1}
2391         \pgfmathsetlengthmacro
2392         {\l_timechart_interval_finishrange_finish_level_pgf}{0}
2393         \pgfmathsetlengthmacro
2394         {\l_timechart_interval_solid_start_x}{0}
2395         \pgfmathsetlengthmacro{\l_timechart_interval_finish_max_x}
2396         {\l_timechart_legenditem_width_pgf}
2397         \int_case:nn {\l_timechart_finish_range_type_int}
2398         {
2399             {0}{
2400                 \pgfmathsetlengthmacro{\l_timechart_interval_solid_finish_x}
2401                 {
2402                     \l_timechart_legenditem_width_pgf
2403                     -\l_timechart_legenditem_range_width_pgf
2404                 }
2405                 \_timechart_interval_finishrange_draw_pseudofade:
2406             }
2407             {1}{
2408                 \pgfmathsetlengthmacro{\l_timechart_interval_solid_finish_x}
2409                 {\l_timechart_interval_finishrange_finish_x}
2410                 \pgfinterruptboundingbox
2411                 \_timechart_interval_finishrange_clip_path_finish:
2412                 \_timechart_interval_start_clip_path_none:
2413                 \pgfpathclose
2414                 \pgfusepath{clip}
2415                 \endpgfinterruptboundingbox
2416             }
2417         }
2418         \_timechart_interval_solid_draw:
2419     }
2420 }

```

(End of definition for `\timechartlegendstartrange` and `\timechartlegendfinishrange`. These functions are documented on page 14.)

`\__timechart_legend_aux:nn` Auxiliary command for legend items. Draw a *TikZ* picture, applying PGF keys #1 under */timechart/* and using *TikZ* code #2.

```

2421 \cs_new:Npn\__timechart_legend_aux:nn #1#2
2422 {
2423   \__timechart_debug:n{Legend~auxiliary}
2424   \tikzpicture

```

Process supplied PGF keys and retrieve only those necessary.

```

2425   \pgfkeys{
2426     /timechart/.cd,
2427     #1,
2428     interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
2429     interval~bar~color/.get=\l__timechart_bar_color,
2430     interval~minimum~width/.get=\l__timechart_minimum_width_pgf,
2431     beyond~length~start/.get=\l__timechart_start_beyond_length_pgf,
2432     beyond~length~finish/.get=\l__timechart_finish_beyond_length_pgf,
2433     legend~item~width/.get=\l__timechart_legenditem_width_pgf,
2434     legend~item~range~width/.get=\l__timechart_legenditem_range_width_pgf,
2435   }
2436   \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
2437     {.5*\l__timechart_bar_thickness_pgf}
2438   \pgfmathsetmacro{\l__timechart_start_x}{0}
2439   \pgfmathsetmacro{\l__timechart_finish_x}
2440     {\l__timechart_legenditem_width_pgf}
2441   \pgfmathsetlengthmacro{\l__timechart_start_beyond_x}
2442     {\l__timechart_start_x-\l__timechart_start_beyond_length_pgf}
2443   \pgfmathsetlengthmacro{\l__timechart_finish_beyond_x}
2444     {\l__timechart_finish_x+\l__timechart_finish_beyond_length_pgf}
2445   \pgfmathsetmacro{\l__timechart_current_y}{0}

```

Ensure that the legend is ‘visible’ from the perspective of the drawing macros.

```

2446   \pgfmathsetmacro{\l__timechart_start_tolerance_x}
2447     {\l__timechart_start_x-10mm}
2448   \pgfmathsetmacro{\l__timechart_finish_tolerance_x}
2449     {\l__timechart_finish_x+10mm}
2450   \pgfscope
2451     #2
2452   \endpgfscope
2453   \endtiktikpicture%
2454 }

```

*(End of definition for \\_\_timechart\_legend\_aux:nn.)*

```

2455 \endpackage

```

## Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

A

`\absyear` . . . . . 619, 632

`\a` . . . . . 35, 575, 577, 583, 584, 585, 586



<b>B</b>	
<code>\begin</code> .....	8
<code>\begingroup</code> .....	22, 83
bool commands:	
<code>\bool_if:NnTF</code> .....	33, 76,
237, 339, 364, 476, 528, 621, 624,	
696, 720, 747, 809, 828, 834, 838,	
855, 920, 924, 938, 948, 998, 1008,	
1023, 1041, 1072, 1080, 1188, 1200,	
1227, 1263, 1265, 1278, 1313, 1325,	
1354, 1390, 1392, 1405, 1465, 1467,	
1529, 1531, 2096, 2104, 2123, 2332	
<code>\bool_if:nTF</code> .....	
685, 727, 761, 1105, 1125, 1145, 1163	
<code>\bool_new:N</code> .....	12,
13, 14, 15, 206, 212, 875, 876, 877,	
878, 879, 880, 963, 964, 1196, 1321	
<code>\bool_set:Nn</code> .....	541, 725, 746, 1222, 1349
<code>\bool_set_eq:NN</code> .....	731
<code>\bool_set_false:N</code> .....	119, 533
<code>\bool_set_false:Nz</code> .....	152
<code>\bool_set_inverse:N</code> .....	726
<code>\bool_set_true:N</code> .....	207, 213, 534
<code>\c_false_bool</code> .....	28, 510, 2154
<code>\l_tmpa_bool</code> .....	261, 262,
803, 809, 819, 828, 849, 855, 908, 920	
<code>\l_tmpb_bool</code> .....	832, 834, 836, 838
<code>\c_true_bool</code> .....	28, 377, 594, 601
<b>C</b>	
cs commands:	
<code>\cs_if_eq:NNTF</code> .....	317, 1290, 1417
<code>\cs_if_exist_use:NnTF</code> .....	2086, 2088
<code>\cs_new:Npn</code> .....	6,
20, 26, 30, 35, 41, 62, 81, 221,	
315, 321, 345, 381, 425, 512, 525,	
558, 571, 607, 615, 638, 642, 646,	
654, 658, 662, 666, 670, 688, 700,	
723, 735, 739, 744, 751, 755, 759,	
765, 769, 773, 774, 775, 776, 777,	
778, 779, 780, 781, 782, 783, 784,	
785, 786, 787, 788, 789, 790, 791,	
792, 793, 794, 795, 796, 797, 798,	
887, 932, 965, 1100, 1120, 1140,	
1180, 1197, 1260, 1305, 1322, 1387,	
1432, 1450, 1470, 1492, 1514, 1535,	
1557, 1580, 1607, 1623, 1637, 1651,	
1675, 1699, 1723, 1747, 1792, 1837,	
1886, 1935, 2005, 2075, 2084, 2093,	
2141, 2156, 2170, 2191, 2212, 2421	
<code>\cs_set:Npn</code> .....	397, 399, 401, 414,
416, 418, 997, 2290, 2293, 2294, 2295	
<code>\cs_set_eq:NN</code> .....	10,
16, 17, 92, 95, 302, 303, 304, 305,	
306, 307, 308, 310, 312, 313, 625,	
626, 628, 814, 815, 817, 1074, 1082,	
1109, 1111, 1113, 1129, 1131, 1133,	
1149, 1151, 1153, 1157, 1167, 1169,	
1171, 1175, 1190, 1202, 1214, 1229,	
1242, 1251, 1267, 1272, 1280, 1285,	
1292, 1297, 1300, 1315, 1327, 1340,	
1356, 1369, 1378, 1394, 1399, 1407,	
1412, 1419, 1424, 1427, 2102, 2110	
<b>D</b>	
<code>\denominator</code> .....	1590, 1593
dim commands:	
<code>\dim_compare:nNnTF</code> .....	328
<code>\dim_new:N</code> .....	18, 19
<code>\l_tmpa_dim</code> .....	21, 16
<code>\l_tmpb_dim</code> .....	21, 17
<b>E</b>	
<code>\end</code> .....	8
<code>\endgroup</code> .....	22, 87
<code>\endpgfinterruptboundingbox</code> .....	
.....	51, 2082, 2167, 2306, 2373, 2415
<code>\endpgfonlayer</code> .....	379, 2232
<code>\endpgfscope</code> .....	1097, 1490, 1512,
1555, 1577, 1603, 2139, 2263, 2452	
<code>\endscope</code> .....	378, 523, 569, 590, 613, 635
<code>\endtikzpicture</code> .....	319, 2453
<b>F</b>	
<code>\fadestep</code> .....	1586, 1588, 1589, 1590, 1593, 1598
<code>\foreach</code> .....	355, 461, 479, 1591, 2113
<b>G</b>	
group commands:	
<code>\group_begin:</code> .....	64, 301, 361,
383, 573, 579, 617, 893, 1583, 2174,	
2195, 2219, 2245, 2268, 2286, 2314	
<code>\group_end:</code> .....	79, 343, 371, 423,
589, 605, 636, 923, 929, 1605, 2188,	
2209, 2233, 2262, 2305, 2308, 2331	
<b>H</b>	
<code>\hyperref</code> .....	24
<b>I</b>	
int commands:	
<code>\int_case:nn</code> .....	384, 489,
531, 1248, 1375, 2161, 2291, 2358, 2397	
<code>\int_if_zero:nTF</code> .....	244
<code>\int_new:N</code> .....	208, 210, 214, 215, 216, 217
<code>\int_set:Nn</code> .....	129, 131, 133,
134, 147, 149, 151, 168, 170, 172,	
176, 178, 181, 183, 188, 189, 190,	
194, 195, 198, 199, 202, 203, 209, 211	

<b>M</b>	
msg commands:	
\msg_error:nmmn	928
\msg_new:nmm	881, 2091
\msg_warning:nmm	2137
<b>N</b>	
\n	281
\NeedsTeXFormat	3
\NewDocumentCommand	674, 883, 2265, 2312, 2336, 2345, 2379
\NewDocumentEnvironment	218
\nobreakspace	640, 644
\node	2180, 2201, 2226, 2251, 2297
<b>O</b>	
\o	1592, 1601
<b>P</b>	
\pgfcoordinate	45, 48
\pgfdeclarelayer	234, 235
\pgfextractx	21, 52, 55
\pgfextracty	21, 323, 325, 493, 500
\pgfinterruptboundingbox	43, 2078, 2160, 2285, 2368, 2410
\pgfkeys	89, 223, 279, 427, 2425
\pgfkeyssetvalue	108, 109, 112, 113, 116, 117
\pgfkeysvalueof	393, 410, 2153
\pgflinewidth	85
\pgfmathsetlengthmacro	85, 243, 246, 247, 248, 576, 577, 995, 1592, 2214, 2319, 2339, 2340, 2348, 2350, 2352, 2354, 2356, 2361, 2366, 2382, 2387, 2389, 2391, 2393, 2395, 2400, 2408, 2436, 2441, 2443
\pgfmathsetmacro	21, 28, 263, 265, 267, 284, 286, 288, 289, 290, 291, 292, 295, 298, 330, 331, 334, 336, 347, 350, 396, 413, 431, 439, 448, 453, 467, 485, 495, 496, 502, 503, 575, 656, 660, 664, 668, 672, 676, 686, 732, 811, 830, 840, 844, 857, 864, 940, 945, 950, 955, 983, 986, 989, 992, 1000, 1005, 1010, 1015, 1025, 1032, 1043, 1045, 1047, 1049, 1056, 1204, 1217, 1232, 1245, 1330, 1343, 1359, 1372, 1579, 1584, 1585, 1586, 1588, 1589, 1590, 1621, 1622, 1750, 1754, 1795, 1799, 1840, 1844, 1889, 1893, 1938, 1942, 1945, 1949, 2008, 2012, 2015, 2019, 2098, 2106, 2115, 2269, 2438, 2439, 2445, 2446, 2448
\pgfmathsmuggle	22, 86
\pgfmathtruncatemacro	619, 800, 801, 802
\pgfnode	22, 44, 77, 78
\pgfonlayer	353, 2225
\pgfpatharc	583, 584, 585, 586, 1664, 1688, 1767, 1812, 1856, 1905, 1961, 1981, 2031, 2051
\pgfpathclose	2080, 2371, 2413
\pgfpathlineto	368, 516, 562, 582, 587, 611, 1626, 1631, 1645, 1654, 1659, 1669, 1683, 1693, 1702, 1707, 1712, 1717, 1731, 1736, 1741, 1757, 1762, 1772, 1781, 1786, 1807, 1817, 1826, 1831, 1861, 1866, 1871, 1876, 1910, 1915, 1920, 1925, 1966, 1971, 1976, 1986, 1995, 2036, 2041, 2046, 2056, 2065, 2130
\pgfpathmoveto	58, 59, 367, 515, 561, 581, 610, 648, 650, 1640, 1678, 1726, 1802, 1847, 1896, 1952, 2022, 2125, 2324, 2327
\pgfpathrectanglecorners	1438, 1477, 1499, 1542, 1564, 1595, 1609, 2235
\pgfpoint	58, 59, 366, 367, 368, 374, 375, 391, 408, 507, 508, 515, 517, 561, 563, 574, 582, 587, 610, 611, 618, 649, 651, 1089, 1439, 1443, 1478, 1482, 1500, 1504, 1543, 1547, 1565, 1569, 1596, 1598, 1610, 1614, 1627, 1632, 1641, 1646, 1655, 1660, 1670, 1679, 1684, 1694, 1703, 1708, 1713, 1718, 1727, 1732, 1737, 1742, 1758, 1763, 1774, 1778, 1782, 1787, 1803, 1808, 1819, 1823, 1827, 1832, 1849, 1853, 1862, 1867, 1872, 1878, 1882, 1898, 1902, 1911, 1916, 1921, 1927, 1931, 1954, 1958, 1967, 1972, 1977, 1988, 1992, 1997, 2001, 2024, 2028, 2037, 2042, 2047, 2058, 2062, 2067, 2071, 2126, 2131, 2145, 2149, 2176, 2197, 2221, 2236, 2240, 2247, 2283, 2288, 2322, 2325, 2328
\pgfpointadd	1773, 1818, 1848, 1877, 1897, 1926, 1953, 1987, 1996, 2023, 2057, 2066
\pgfpointanchor	47, 50, 54, 57, 324, 326, 494, 501
\pgfpointorigin	581
\pgfqkeys	934, 967, 2270, 2277, 2315
\pgfresetboundingbox	327
\pgfscope	1090, 1476, 1498, 1541, 1563, 1594, 2095, 2234, 2450
\pgfset	70
\pgfsetfillcolor	1447, 1486, 1508, 1551, 1573, 1600, 1618



`\l__timechart_content_bottom_y` . . . . . 30,  
330, 334, 347, 348, 368, 374, 409, 649  
`\l__timechart_content_top_y` . 30,  
331, 336, 350, 351, 367, 375, 392, 651  
`\l__timechart_cumulative_days_-`  
`pgf` . . . . . 817, 844, 845, 860, 868  
`\l__timechart_current_y` . . . . .  
. . . . . 288, 656, 660, 664, 676, 677,  
681, 682, 686, 1089, 2283, 2322, 2445  
`\__timechart_debug:n` . . . . . 6, 10,  
92, 95, 889, 892, 1064, 1065, 1102,  
1122, 1142, 1182, 1199, 1262, 1307,  
1324, 1389, 1434, 1452, 1453, 1472,  
1473, 1494, 1495, 1516, 1517, 1537,  
1538, 1559, 1560, 1582, 1625, 1639,  
1653, 1677, 1701, 1725, 1749, 1794,  
1839, 1888, 1937, 2007, 2077, 2423  
`\__timechart_debug_real:n` . . 6, 6, 92  
`\__timechart_draw_axis_line` 488, 607  
`\__timechart_draw_visible_-`  
`interval:nn` . . . . . 965  
`\__timechart_draw_visible_-`  
`interval:nn` . . . . . 961, 965  
`\l__timechart_end_minus_year` 298, 359  
`\l__timechart_fadestep_pgf` 1579, 1587  
`\__timechart_finish_beyond_bool` 963  
`\l__timechart_finish_beyond_-`  
`length_pgf` . . . 971, 993, 2432, 2444  
`\l__timechart_finish_beyond_x` . . . . .  
. . . . . 992, 1168,  
1426, 1496, 1505, 1539, 1544, 1615,  
1680, 1685, 1695, 1804, 1809, 1850,  
1863, 1955, 1968, 2043, 2048, 2443  
`\l__timechart_finish_beyond_x_-`  
`radius_pgf` . . . . .  
. . . . . 973, 1691, 1815, 1964, 2054  
`\l__timechart_finish_is_range_-`  
`bool` . . . . . 875, 902  
`\__timechart_finish_major_tick_-`  
`year` . . . . . 34  
`\l__timechart_finish_major_tick_-`  
`year` . . . . . 453, 483, 540  
`\l__timechart_finish_min_x` . . . . . 47  
`\l__timechart_finish_range_type_-`  
`int` . . . . . 181, 183, 216, 1375, 2397  
`\l__timechart_finish_tolerance_-`  
`pgf` . . . . . 969, 987, 2273  
`\l__timechart_finish_tolerance_x`  
. . . . . 986, 1079, 2448  
`\l__timechart_finish_x` . . . . .  
. . . . . 286, 375, 508, 611,  
651, 693, 707, 711, 716, 987, 993,  
1124, 1162, 1221, 1231, 1329, 1496,  
1501, 1539, 1548, 1820, 1879, 1998,  
2059, 2107, 2217, 2439, 2444, 2449  
`\l__timechart_finish_year` . . 262,  
265, 266, 277, 287, 299, 455, 457, 465  
`\l__timechart_grid_bool` . . . . .  
. . . . . 119, 206, 237, 339  
`\l__timechart_grid_bottom_ysep_-`  
`pgf` . . . . . 229, 348  
`\__timechart_grid_draw:` 340, 345, 345  
`\l__timechart_grid_line_width` . . . . .  
. . . . . 240, 243, 272  
`\l__timechart_grid_top_ysep_pgf` . . . . .  
. . . . . 228, 351  
`\__timechart_hsmash_pgfnode:nnnnn`  
. . . . . 41, 41, 629  
`\__timechart_if_equal:nnTF` . . 30, 30  
`\__timechart_if_x_in_bounds:nTF` . . . . .  
. . . . . 688, 688, 2172, 2193, 2275  
`\__timechart_if_x_range_intersect_-`  
`bounds:nnTF` . . . . . 700, 958  
`\__timechart_if_x_range_intersect_-`  
`bounds_x:nnTF` . . . . . 700  
`\__timechart_interval:nnnn` . . . . .  
. . . . . 44, 885, 887, 887  
`\__timechart_interval_both_clip_-`  
`path:` . . . . . 67-70, 72, 1268,  
1281, 1395, 1408, 2079, 2084, 2084  
`\__timechart_interval_checked:nn`  
. . . . . 922, 932, 932  
`\__timechart_interval_clip:` . . . . .  
. . . . . 1091, 2075, 2075  
`\__timechart_interval_define_-`  
`bar_node:` . . . . . 1096, 2141, 2141  
`\__timechart_interval_definite_-`  
`compute:` . . . . . 1086, 1100, 1100  
`\__timechart_interval_definite_-`  
`compute_ii:` . . . . . 50, 1117, 1120  
`\__timechart_interval_definite_-`  
`compute_iii:` . . . . . 1137, 1140  
`\__timechart_interval_draw_-`  
`pseudofade:nnnnn` . . . . .  
. . . . . 1459, 1523, 1579, 1580  
`\l__timechart_interval_finish_-`  
`beyond_bool` . . . . .  
. . . . . 964, 1078, 1080, 1126,  
1164, 1225, 1325, 1390, 1529, 2104  
`\__timechart_interval_finish_-`  
`clip_path:` 72, 1083, 1400, 1413, 2086  
`\__timechart_interval_finish_-`  
`clip_path_beyond:` 1084, 1675, 1675  
`\__timechart_interval_finish_-`  
`clip_path_none:` . . . . .  
. . . . . 1637, 1637, 2087, 2369

`\l_timechart_interval_finish_max_circa_bool` . . . . . 875, 905, 948  
`\l_timechart_interval_finish_max_x` . . . . . 46, 49, 950, 955, 960, 1037, 1056, 1059, 1068, 1079, 1310, 1333, 1336, 1342, 1362, 1365, 1917, 1922, 2150, 2193, 2198, 2241, 2395  
`\l_timechart_interval_finish_max_year` . . . . . 45, 906, 916, 951, 956  
`\l_timechart_interval_finish_min_circa_bool` . . . . . 875, 903, 1008  
`\l_timechart_interval_finish_min_x` . . . . . 47, 49, 1010, 1015, 1021, 1029, 1032, 1047, 1048, 1060, 1068, 1104, 1162, 1177, 1311, 1335, 1348, 1364, 1371, 1828, 1833, 2107, 2110, 2217  
`\l_timechart_interval_finish_min_year` 45, 904, 914, 915, 1011, 1016  
`\_timechart_interval_finishrange_clip_path_beyond_both`: . . . . . 1396, 2005, 2005  
`\_timechart_interval_finishrange_clip_path_beyond_finish`: . . . . . 1401, 1792, 1792  
`\_timechart_interval_finishrange_clip_path_beyond_start`: . . . . . 1409, 1886, 1886  
`\_timechart_interval_finishrange_clip_path_finish`: . . . . . 1414, 1723, 1723, 2411  
`\_timechart_interval_finishrange_compute`: 1088, 1132, 1170, 1305, 1305  
`\_timechart_interval_finishrange_compute_ii`: . . . . . 1318, 1321, 1322  
`\_timechart_interval_finishrange_draw`: . . . . . 1093, 1134, 1172, 1315, 1379, 1420, 1428  
`\_timechart_interval_finishrange_draw_pseudofade`: . . . . . 1380, 1514, 1514, 2405  
`\_timechart_interval_finishrange_finish_beyond_draw_fade`: . . . . . 1530, 1534, 1535  
`\l_timechart_interval_finishrange_finish_level_pgf` . 1330, 1344, 1521, 1528, 1553, 1797, 2010, 2392  
`\l_timechart_interval_finishrange_finish_x` . . . . . 1328, 1332, 1341, 1520, 1525, 1728, 1733, 2387, 2409  
`\_timechart_interval_finishrange_slant_compute`: . . 1383, 1387, 1387  
`\l_timechart_interval_finishrange_start_beyond_bool` . . . . . 1321, 1349, 1354, 1392, 1405, 1531  
`\_timechart_interval_finishrange_start_beyond_draw_fade`: . . . . . 1532, 1557, 1557  
`\l_timechart_interval_finishrange_start_level_pgf` . . 1359, 1373, 1519, 1527, 1575, 1891, 2017, 2390  
`\l_timechart_interval_finishrange_start_x` . . . . . 1357, 1361, 1370, 1518, 1524, 1738, 1743, 2382  
`\_timechart_interval_label`: . . . . . 1098, 2156, 2156  
`\_timechart_interval_label_center`: . . . . . 2164, 2212, 2212  
`\_timechart_interval_label_left`: . . . . . 2163, 2170, 2170  
`\l_timechart_interval_label_node_name` . . . . . 981, 2181, 2202, 2255  
`\_timechart_interval_label_right`: . . . . . 2165, 2191, 2191  
`\_timechart_interval_mark`: . . . . . 1095, 2093, 2093  
`\l_timechart_interval_mark_max_x` . . . . . 2106, 2110, 2120  
`\l_timechart_interval_mark_min_x` . . . . . 2098, 2102, 2119  
`\_timechart_interval_solid_draw`: . . . . . 1094, 1110, 1130, 1290, 1417, 1432, 1432, 2342, 2376, 2418  
`\l_timechart_interval_solid_finish_x` . . . . . 1168, 1176, 1425, 1436, 1444, 2340, 2356, 2400, 2408  
`\l_timechart_interval_solid_start_x` . . . . . 1150, 1158, 1298, 1435, 1440, 2339, 2361, 2366, 2394  
`\l_timechart_interval_start_beyond_bool` . . . . . 963, 1070, 1072, 1106, 1146, 1200, 1263, 1352, 1465, 2096  
`\_timechart_interval_start_clip_path`: 72, 1075, 1273, 1286, 2088  
`\_timechart_interval_start_clip_path_beyond`: 1076, 1651, 1651  
`\_timechart_interval_start_clip_path_none`: . . . . . 1623, 1623, 2089, 2412  
`\l_timechart_interval_start_max_circa_bool` . . . . . 875, 898, 998  
`\l_timechart_interval_start_max_x` . . . . . 47, 49, 1000, 1005, 1020, 1025, 1028, 1033, 1045, 1046, 1053, 1066, 1124, 1144, 1159, 1186, 1209, 1221, 1237, 1244, 1783, 1788, 2099, 2102, 2216

`\l_timechart_interval_start_-`  
`max_year` [45](#), [899](#), [911](#), [913](#), [1001](#), [1006](#)  
`\l_timechart_interval_start_-`  
`min_circa_bool` [875](#), [896](#), [938](#)  
`\l_timechart_interval_start_-`  
`min_x` [46](#), [49](#), [940](#), [945](#),  
[959](#), [1038](#), [1049](#), [1052](#), [1066](#), [1071](#),  
[1185](#), [1207](#), [1210](#), [1216](#), [1235](#), [1238](#),  
[1868](#), [1873](#), [2146](#), [2172](#), [2177](#), [2237](#)  
`\l_timechart_interval_start_-`  
`min_year` [45](#), [897](#), [910](#), [941](#), [946](#)  
`\_timechart_interval_startfinishrange_-`  
`draw_slant:` [1294](#), [1421](#), [1607](#), [1607](#)  
`\_timechart_interval_startrange_-`  
`clip_path_beyond_both:` [1269](#), [1935](#), [1935](#)  
`\_timechart_interval_startrange_-`  
`clip_path_beyond_finish:` [1282](#), [1837](#), [1837](#)  
`\_timechart_interval_startrange_-`  
`clip_path_beyond_start:` [1274](#), [1747](#), [1747](#)  
`\_timechart_interval_startrange_-`  
`clip_path_start:` [1287](#), [1699](#), [1699](#), [2370](#)  
`\_timechart_interval_startrange_-`  
`compute:` [1087](#), [1112](#), [1152](#), [1180](#), [1180](#)  
`\_timechart_interval_startrange_-`  
`compute_ii:` [1193](#), [1196](#), [1197](#)  
`\_timechart_interval_startrange_-`  
`draw:` [1092](#),  
[1114](#), [1154](#), [1190](#), [1252](#), [1293](#), [1301](#)  
`\_timechart_interval_startrange_-`  
`draw_pseudofade:` [1253](#), [1450](#), [1450](#), [2363](#)  
`\l_timechart_interval_startrange_-`  
`finish_beyond_bool` [1196](#), [1222](#), [1227](#), [1265](#), [1278](#), [1467](#)  
`\_timechart_interval_startrange_-`  
`finish_beyond_draw_fade:` [1468](#), [1492](#), [1492](#)  
`\l_timechart_interval_startrange_-`  
`finish_level_pgf` [1232](#), [1246](#),  
[1457](#), [1464](#), [1510](#), [1842](#), [1947](#), [2355](#)  
`\l_timechart_interval_startrange_-`  
`finish_x` [1230](#), [1234](#),  
[1243](#), [1456](#), [1461](#), [1714](#), [1719](#), [2350](#)  
`\_timechart_interval_startrange_-`  
`slant_compute:` [1256](#), [1260](#), [1260](#)  
`\_timechart_interval_startrange_-`  
`start_beyond_draw_fade:` [1466](#), [1470](#), [1470](#)  
`\l_timechart_interval_startrange_-`  
`start_level_pgf` [1204](#), [1218](#),  
[1455](#), [1463](#), [1488](#), [1752](#), [1940](#), [2353](#)  
`\l_timechart_interval_startrange_-`  
`start_x` [1203](#), [1206](#), [1215](#),  
[1454](#), [1460](#), [1704](#), [1709](#), [2349](#), [2367](#)  
`\_timechart_interval_user:Ommm` [302](#), [883](#), [883](#)  
`\_timechart_interval_user_-`  
`aux:nmm` [45](#)  
`\_timechart_is_before_p:w` [41](#), [761](#), [765](#), [765](#)  
`\_timechart_is_circa_p:w` [40](#), [746](#), [751](#), [751](#)  
`\_timechart_is_nondaterange_p:w` [39](#), [725](#), [735](#), [735](#)  
`\l_timechart_label_anchor_x` [2214](#), [2222](#), [2248](#)  
`\l_timechart_label_pos_int` [168](#), [170](#), [172](#), [194](#), [198](#), [202](#), [214](#), [2161](#)  
`\l_timechart_label_text` [997](#), [2158](#), [2186](#), [2207](#), [2231](#), [2260](#)  
`\c_timechart_left_far_x` [1621](#), [1628](#), [1633](#)  
`\_timechart_legend_aux:nn` [2338](#), [2347](#), [2381](#), [2421](#), [2421](#)  
`\l_timechart_legenditem_range_-`  
`width_pgf` [2351](#), [2362](#), [2385](#), [2403](#), [2434](#)  
`\l_timechart_legenditem_width_-`  
`pgf` [2341](#), [2357](#),  
[2384](#), [2388](#), [2396](#), [2402](#), [2433](#), [2440](#)  
`\_timechart_main_begin:nnn` [29](#), [219](#), [221](#), [221](#)  
`\_timechart_main_end:` [220](#), [315](#), [315](#)  
`\_timechart_main_end_user:` [28](#), [29](#), [313](#), [317](#), [318](#), [321](#), [321](#)  
`\l_timechart_major_tick_eras_-`  
`int` [147](#), [149](#), [151](#), [210](#), [531](#)  
`\l_timechart_major_tick_-`  
`interval_year` [231](#), [363](#), [436](#), [444](#), [451](#), [458](#), [473](#)  
`\_timechart_major_tick_length_-`  
`pgf` [429](#), [554](#), [565](#), [582](#), [587](#)  
`\l_timechart_major_tick_line_-`  
`width` [247](#), [255](#), [274](#)  
`\_timechart_make_rectangle_-`  
`node:nmm` [62](#), [62](#), [373](#), [506](#), [2143](#)  
`\_timechart_make_ref:NN` [20](#), [20](#), [2184](#), [2205](#), [2258](#), [2301](#)  
`\_timechart_make_year:n` [625](#), [626](#), [628](#), [632](#)  
`\l_timechart_mark_color` [979](#), [2112](#)  
`\l_timechart_mark_text` [975](#), [2113](#)  
`\l_timechart_mark_x` [2115](#), [2119](#), [2120](#), [2127](#), [2132](#)

<code>\l__timechart_minimum_width_pgf</code> . . . . .	<code>\c__timechart_right_far_x</code> . . . . .
. . . . . 976, 1039, 1044, 2430	. . . . . 1621, 1642, 1647
<code>\l__timechart_minor_tick_interval_year</code> . . . . .	<code>\__timechart_save_y_user:</code> . . . . .
. . . . . 230, 293, 296, 299	. . . . . 306, 658, 658
<code>\__timechart_minor_tick_length_pgf</code> . . . . .	<code>\l__timechart_saved_y</code> . . . . .
. . . . . 428, 519	. . . . . 289, 660, 664, 686
<code>\l__timechart_minor_tick_line_width</code> . . . . .	<code>\__timechart_set_style_line_width:mn</code> . . . . .
. . . . . 248, 258, 275	. . . . . 81, 81, 239, 251, 254, 257
<code>\l__timechart_month_days_pgf</code> . . . . .	<code>\__timechart_set_y_maximum_auto_reset_user:m</code> . . . . .
. . . . . 815, 840, 841, 852	. . . . . 311, 670, 670
<code>\l__timechart_node_anchor_text</code> . . . . .	<code>\__timechart_set_y_minimum_auto_reset_user:m</code> . . . . .
. . . . . 2293, 2294, 2295, 2297	. . . . . 309, 666, 666
<code>\l__timechart_node_name_text</code> . . . . .	<code>\__timechart_set_y_user:m</code> . . . . .
. . . . . 2279, 2298	. . . . . 305, 654, 654
<code>\__timechart_nogrid_bounding_box_set:</code> . . . . .	<code>\__timechart_space_user:0</code> . . . . .
. . . . . 341, 646, 646	. . . . . 304, 2310, 2312, 2312
<code>\__timechart_parse_before_date:w</code> . . . . .	<code>\__timechart_start_beyond_bool</code> . . . . .
. . . . . 41, 762, 769, 769	. . . . . 963
<code>\__timechart_parse_circa_date:w</code> . . . . .	<code>\l__timechart_start_beyond_length_pgf</code> . . . . .
. . . . . 40, 748, 755, 755	. . . . . 970, 990, 2431, 2442
<code>\__timechart_parse_date:NNn</code> . . . . .	<code>\l__timechart_start_beyond_x</code> . . . . .
. . . . . 261, 262, 730, 741, 742, 744, 744	. . . . . 989, 1150, 1299, 1474, 1479, 1561, 1570, 1611, 1656, 1661, 1671, 1759, 1764, 1899, 1912, 1973, 1978, 2025, 2038, 2441
<code>\__timechart_parse_date_or_daterange:NNNNn</code> 723, 723, 894, 901	<code>\l__timechart_start_beyond_x_radius_pgf</code> . . . . .
<code>\__timechart_parse_noncirca_date:Nn</code> . . . . .	. . . . . 972, 1667, 1770, 1859, 1908, 1984, 2034
. . . . . 749, 757, 759	<code>\l__timechart_start_is_range_bool</code> . . . . .
<code>\__timechart_parse_noncirca_date:w</code> . . . . .	. . . . . 875, 895
. . . . . 40, 759	<code>\__timechart_start_major_tick_year</code> . . . . .
<code>\__timechart_parse_positive_date:w</code> . . . . .	. . . . . 34
. . . . . 42	<code>\l__timechart_start_major_tick_year</code> . . . . .
<code>\__timechart_parse_range:w</code> . . . . .	. . . . . 431, 439, 442, 443, 445, 450, 472, 480, 538
. . . . . 40, 728, 739, 739	<code>\l__timechart_start_max_x</code> . . . . .
<code>\__timechart_parse_signed_date:w</code> . . . . .	. . . . . 47
. . . . . 763, 771, 798, 798	<code>\l__timechart_start_plus_major_tick_year</code> . . . . .
<code>\__timechart_parsed_day_pgf</code> . . . . .	. . . . . 448, 481
. . . . . 802, 851, 852, 869	<code>\l__timechart_start_plus_year</code> . . . . .
<code>\__timechart_parsed_month</code> . . . . .	. . . . . 292, 356, 463
. . . . . 833, 837	<code>\l__timechart_start_plusplus_year</code> . . . . .
<code>\__timechart_parsed_month_pgf</code> . . . . .	. . . . . 295, 357
. . . . . 801, 805, 806, 816, 818	<code>\l__timechart_start_range_type_int</code> . . . . .
<code>\__timechart_parsed_year_pgf</code> . . . . .	. . . . . 176, 178, 216, 1248, 2358
. . . . . 800, 821, 823, 824	<code>\l__timechart_start_tolerance_pgf</code> . . . . .
<code>\__timechart_pgfextractxy:nnn</code> . . . . .	. . . . . 968, 984, 2272
. . . . . 35, 35, 65, 67	<code>\l__timechart_start_tolerance_x</code> . . . . .
<code>\__timechart_pgfmathsetbool:nn</code> . . . . .	. . . . . 983, 1071, 2446
. . . . . 26, 32, 362, 468, 527, 537, 539, 623, 678, 690, 702, 803, 819, 832, 836, 849, 908, 1018, 1035, 1070, 1078, 1103, 1123, 1143, 1161, 1183, 1220, 1308, 1347, 2116	<code>\l__timechart_start_x</code> . . . . .
<code>\l__timechart_ref_text</code> . . . . .	. . . . . 284, 374, 507, 610, 649, 692, 706, 710, 715, 984, 990, 1104, 1144, 1203, 1348, 1358, 1474, 1483, 1561, 1566, 1775, 1928, 1989, 2068, 2099, 2216, 2438, 2442, 2447
. . . . . 974, 2185, 2206, 2259, 2278, 2302	
<code>\__timechart_reset_y_user:</code> . . . . .	
. . . . . 307, 662, 662	

<code>\l__timechart_start_year</code> . . . . .	<code>\l__timechart_width_adjust</code> . . . . .
. . . . . 261, 263, 264, 277,	. . . . . 1043, 1046, 1048
281, 285, 293, 296, 433, 435, 442, 462	<code>\l__timechart_width_pgf</code> . . . . 226, 270
<code>\__timechart_step_y_user:</code> . . . 37, 38	<code>\l__timechart_x</code> . . . . . 27, 267, 281
<code>\__timechart_step_y_user:0</code> . . . . .	<code>\c__timechart_year_days_pgf</code> . . . 814
. . . . . 312, 674, 674, 925, 2333	<code>\l__timechart_year_days_pgf</code> . . . . .
<code>\l__timechart_text</code> . . . . . 2290, 2303	. . . . . 814, 830, 831, 860, 870
<code>\l__timechart_text_baseline_pgf</code> .	<code>\l__timechart_ystep_pgf</code> . . . . 227, 677
980, 2178, 2199, 2223, 2249, 2280, 2288	<code>\l__timechart_zero_tick_after_-</code>
<code>\l__timechart_text_pos_int</code> . . . . .	label_anchor_text . . 401, 418, 602
188, 189, 190, 195, 199, 203, 215, 2291	<code>\l__timechart_zero_tick_before_-</code>
<code>\__timechart_text_user:Omm</code> . . . . .	label_anchor_text . . 399, 416, 595
. . . . . 303, 2265, 2265	<code>\timechartfinish</code> . . . . 13, 15, 29, 313, 317
<code>\l__timechart_text_x</code> 2269, 2275, 2288	<code>\timechartinterval</code> . . . . .
<code>\l__timechart_tick_label_anchor_-</code>	. . . . . 3, 5-9, 12, 14, 15, 44, 302
text . . . . . 397, 414, 550	<code>\timechartlegendfinishrange</code> . . . 14, 2345
<code>\l__timechart_tick_orientation_-</code>	<code>\timechartlegenditem</code> . . . . . 14, 2336
pgf . . . . . 396,	<code>\timechartlegendstartrange</code> . . . . 14, 2345
413, 518, 553, 564, 580, 597, 604	<code>\timechartlength</code> . . . . . 14
<code>\l__timechart_tmpa_bool</code> . . . . .	<code>\timechartmakeafteryear</code> . . . 13, 626, 638
. . . . . 12, 32, 33, 362, 364,	<code>\timechartmakebeforeyear</code> . . . 13, 625, 638
469, 476, 527, 528, 623, 624, 678,	<code>\timechartresety</code> . . . . . 12, 307
685, 690, 696, 702, 720, 1018, 1023,	<code>\timechartsavey</code> . . . . . 12, 37, 306
1035, 1041, 1103, 1106, 1123, 1126,	<code>\timechartsety</code> . . . . . 12, 37, 305
1143, 1146, 1183, 1188, 1220, 1224,	<code>\timechartsetymaximumautoreset</code> . 12, 310
1308, 1313, 1347, 1351, 2116, 2123	<code>\timechartsetyminimumautoreset</code> . 12, 308
<code>\l__timechart_tmpa_dim</code> . 16, 53, 58,	<code>\timechartspace</code> . . . . . 6, 8, 12, 77, 304
66, 71, 323, 328, 335, 493, 495, 500, 503	<code>\timechartsresety</code> . . . . . 37
<code>\l__timechart_tmpa_pgf</code> . . . . .	<code>\timechartssetymaximumautoreset</code> . . . . 38
1750, 1755, 1778, 1795, 1800, 1823,	<code>\timechartssetyminimumautoreset</code> . . . . 37
1938, 1943, 1992, 2008, 2013, 2062	<code>\timechartstepy</code> . . . . . 12, 38, 312
<code>\l__timechart_tmpb_bool</code> . . . . .	<code>\timechartstext</code> 6-8, 11, 12, 14, 15, 76, 303
. . . . . 12, 537, 542, 1161, 1164	tl commands:
<code>\l__timechart_tmpb_dim</code> . . . . .	<code>\tl_if_empty_p:n</code> . . . . . 737, 753, 767
. . . . . 16, 56, 59, 66, 72, 325, 328, 337	<code>\typeout</code> . . . . . 8
<code>\l__timechart_tmpb_pgf</code> 1754, 1771,	
1799, 1816, 1942, 1985, 2012, 2055	<b>U</b>
<code>\l__timechart_tmpc_bool</code> . 12, 539, 542	use commands:
<code>\l__timechart_tmpc_dim</code> . . . . 16, 68, 71	<code>\use:n</code> . . . . . 628
<code>\l__timechart_tmpc_pgf</code> . . . . .	<code>\use_none:n</code> . . . . . 10, 95
. . . . . 1840, 1845, 1853, 1882,	<code>\usepackage</code> . . . . . 3
1889, 1894, 1902, 1931, 1945, 1950,	
1958, 2001, 2015, 2020, 2028, 2071	<b>X</b>
<code>\l__timechart_tmpd_bool</code> . . . . .	<code>\x</code> . . . . . 467, 477, 485, 486
. . . . . 12, 533, 534, 541, 549	<code>\xa</code> . . . . . 1591, 1593, 1596, 1598
<code>\l__timechart_tmpd_dim</code> . . . . 16, 68, 72	
<code>\l__timechart_tmpd_pgf</code> 1844, 1860,	<b>Y</b>
1893, 1909, 1949, 1965, 2019, 2035	<code>\year</code> . . . . . 355, 363, 366, 461, 467,
	472, 479, 485, 486, 2113, 2115, 2137