

# timechart — A package for drawing chronological charts\*

Alan J. Cain<sup>†</sup>

Released 2026-07-10

## 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>5</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	Tagging support . . . . .	9
8.3	Grid configuration . . . . .	9
8.4	Axis configuration . . . . .	9

---

\*This file describes v0.57.1, last revised 2026-07-10.

<sup>†</sup>Email: `a.j.cain (AT) gmail.com`

<b>9</b>	<b>Within the timechart environment</b>	<b>10</b>
9.1	Intervals . . . . .	10
	Interval configuration . . . . .	10
9.2	Text . . . . .	12
	Text configuration . . . . .	12
9.3	Space . . . . .	12
9.4	Positioning . . . . .	12
9.5	Completion . . . . .	13
9.6	Shortcut keys . . . . .	13
<b>10</b>	<b>Era indicators</b>	<b>14</b>
<b>11</b>	<b>Legend</b>	<b>14</b>
<b>12</b>	<b>Usage notes</b>	<b>15</b>
12.1	Additional TikZ code . . . . .	15
12.2	‘year zero’ . . . . .	15
<b>13</b>	<b>Feature requests and bug reports</b>	<b>15</b>
<b>14</b>	<b>Appendix: full example source</b>	<b>15</b>
14.1	Setup source . . . . .	16
14.2	Timechart source . . . . .	17
14.3	Legend source . . . . .	20
<b>15</b>	<b>Implementation</b>	<b>20</b>
15.1	Coding standard . . . . .	20
15.2	Initial set-up . . . . .	20
15.3	Debugging macro . . . . .	21
15.4	Load TikZ . . . . .	21
15.5	Scratch variables . . . . .	21
15.6	Generic auxiliary functions . . . . .	21
15.7	PGF auxiliary functions . . . . .	22
15.8	PGF keys . . . . .	23
15.9	Main environment . . . . .	27
15.10	Grid drawing . . . . .	30
15.11	Axis drawing . . . . .	31
15.12	Bounding box . . . . .	37
15.13	Positioning . . . . .	37
15.14	Bounds checking . . . . .	39
15.15	Date and date range parsing . . . . .	40
15.16	Interval drawing . . . . .	44
	15.16.1 Preliminaries . . . . .	44
	15.16.2 Error message definition . . . . .	45
	15.16.3 Main macros . . . . .	45
	15.16.4 Computation . . . . .	50
	15.16.5 Drawing . . . . .	58
	15.16.6 Clipping . . . . .	63
	15.16.7 Marks . . . . .	73
	15.16.8 Bar node . . . . .	74
	15.16.9 Label . . . . .	74

15.17 Text . . . . .	77
15.18 Space . . . . .	78
15.19 Legends . . . . .	79

**Index** **81**

## 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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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](#).

<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>

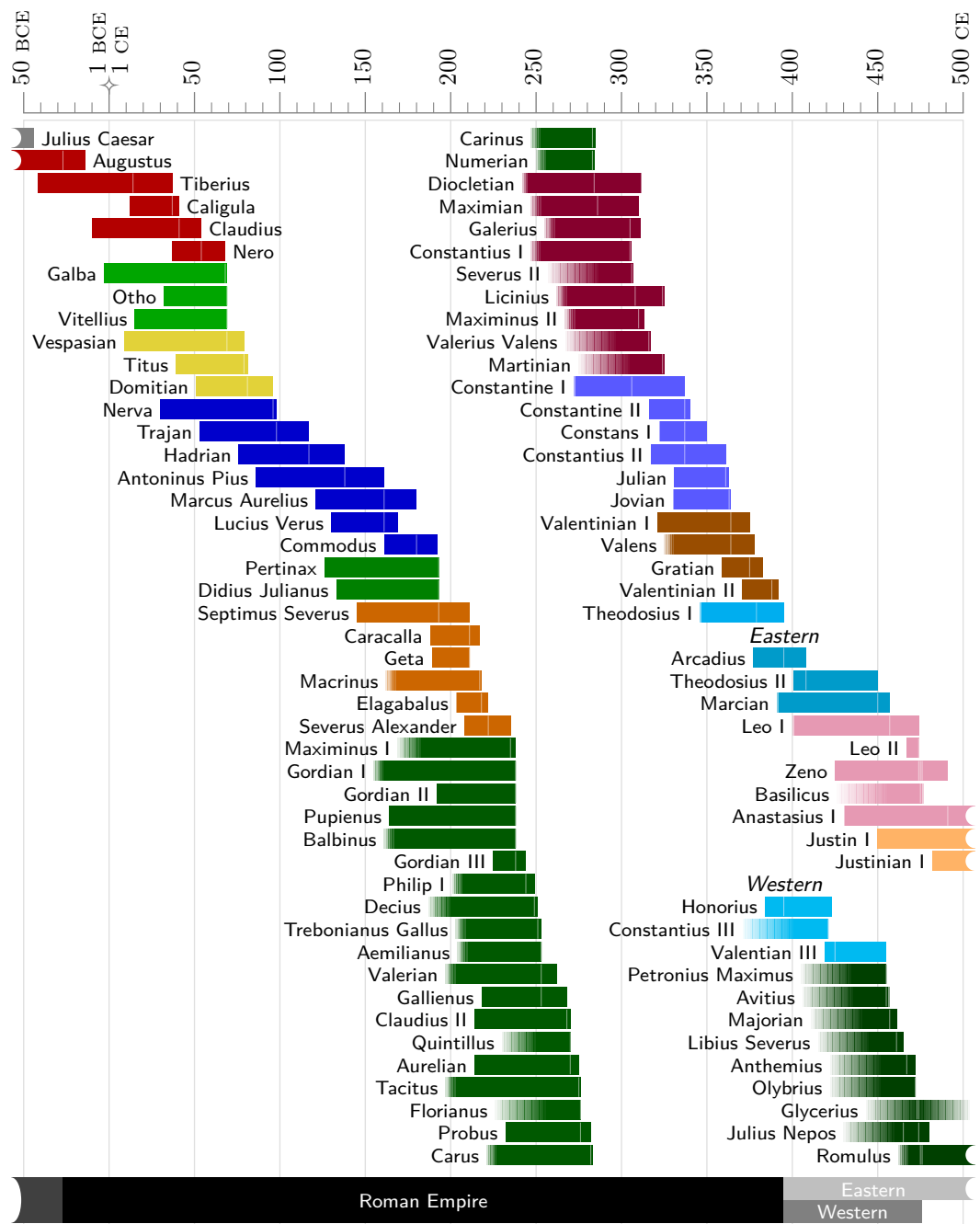


Figure 1: Timechart showing the lifetimes of Roman emperors from 50 BCE to 500 CE. Marks on each lifetime indicate the beginning (and, where relevant, the end) of that emperor's reign. Colours generally indicate dynasties, with shades of green indicating periods when the imperial power shifted between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

## 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 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}

```

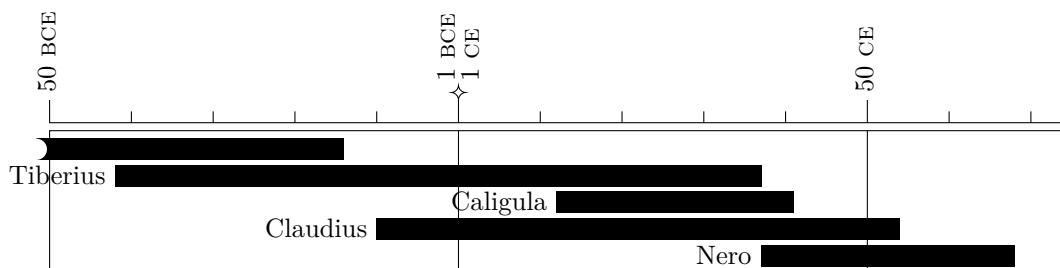


Figure 2: Flawed preliminary version of a chart showing the lifetimes of Roman emperors of the Julio-Claudian dynasty. (The final version is shown in [Figure 3](#).)

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}

```

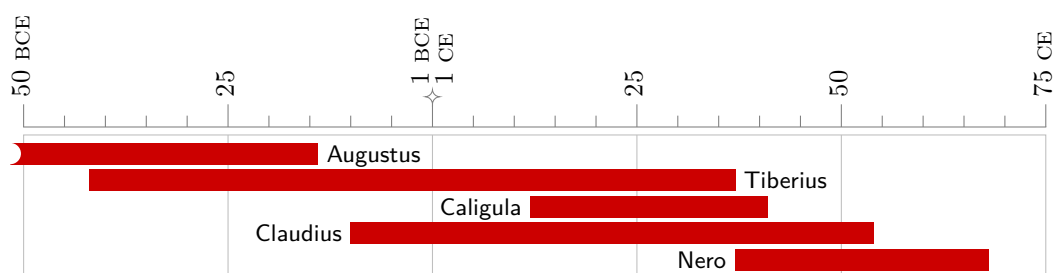


Figure 3: Chart showing the lifetimes of Roman emperors of the Julio-Claudian dynasty.

## 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]{<birth>}{<death>}{<name>}` 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;cdater-or-crangle&gt;</code>	::=	<code>&lt;cdater&gt;</code>   <code>&lt;crangle&gt;</code>
<code>&lt;cdater&gt;</code>	::=	<code>&lt;dater&gt;</code>   <code>c&lt;dater&gt;</code>
<code>&lt;dater&gt;</code>	::=	<code>&lt;pdater&gt;</code>   <code>-&lt;pdater&gt;</code>
<code>&lt;pdater&gt;</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>&lt;dater<sub>12   <code>c&lt;dater<sub>12   <code>&lt;dater<sub>12   <code>c&lt;dater<sub>12 (with <math>date_1 \leq date_2</math>)</sub></code></sub></code></sub></code></sub></code>

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

*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}[\langle options \rangle]{\langle start \rangle}{\langle finish \rangle}
\langle content \rangle
\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 `<cdater>` 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 Tagging support

The following four keys are used for tagging the chart. They simply forward to the corresponding TikZ keys, which are only defined if `\DocumentMetadata` is used. Using these keys when `\DocumentMetadata` is not used will cause an error. See the tagging documentation for further details of what these keys do.

- `/timechart/alt=<text>` (default [None])  
Sets alt text for the chart.
- `/timechart/actualtext=<text>` (default [None])  
Sets actualtext text for the chart.
- `/timechart/artifact`  
Tag the chart as an artifact.
- `/timechart/tagging-setup` (key-value pairs, initially empty)  
Configure tagging for the chart.

## 8.3 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.

## 8.4 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  $\langle dimension \rangle$ . (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= $\langle color \rangle$`  (default black)

Fill the interval bar with  $\langle color \rangle$ .

`/timechart/interval bar thickness= $\langle dimension \rangle$`  (default 8pt)

Set the vertical thickness of the interval bar to  $\langle dimension \rangle$ .

`/timechart/interval bar node name= $\langle string \rangle$`  (default bar node)

Set the name of the node containing the interval bar to  $\langle string \rangle$ .

`/timechart/interval mark color= $\langle color \rangle$`  (default gray)

Draw marks using  $\langle color \rangle$ .

`/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= $\langle dimension \rangle$`  (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= $\langle position \rangle$`  (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= $\langle string \rangle$`  (default label node)

Set the name of the node containing the interval label to  $\langle string \rangle$ .

`/timechart/start range= $\langle range-type \rangle$`  (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= $\langle range-type \rangle$`  (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` `\timecharttext[<options>]{<year>}{<text>}`

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

The optional argument *<options>* 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 *<string>*.

`/timechart/text` (style, initially empty)

Style to apply to the text.

`/timechart/text baseline=<dimension>` (default `-3pt`)

Position the baseline of the text *<dimension>* below the current *y* coordinate.

`/timechart/text pos=<position>` (default `left`)

Specify where to place the label relative to the given *<year>*: *<position>* 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`.



Figure 4: Example legend for the timechart shown in [Figure 1](#).

## 10 Era indicators

---

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

---

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

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.  $\langle options \rangle$  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.

---

<code>\timechartlegendstartrange</code>	<code>\timechartlegendstartrange[<i>&lt;options&gt;</i>]</code>
<code>\timechartlegendfinishrange</code>	<code>\timechartlegendfinishrange[<i>&lt;options&gt;</i>]</code>

---

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[fouemperors,mark=68]{-3}{69}{Galba}
25 \timechartinterval[fouemperors,mark=69]{32}{69}{Otho}
26 \timechartinterval[fouemperors,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}

```

36 \timechartinterval[nervaantonine,mark=161]{130}{169}{Lucius Verus}  
37 \timechartinterval[nervaantonine,mark=180]{161}{192}{Commodus}  
38  
39 \timechartinterval[fiveemperors,mark=193]{126}{193}{Pertinax}  
40 \timechartinterval[fiveemperors,mark=193]{133}{193}{Didius Julianus}  
41  
42 \timechartinterval[severan,mark=193]{145}{211}{Septimus Severus}  
43 \timechartinterval[severan,mark=211]{188}{217}{Caracalla}  
44 \timechartinterval[severan,mark=211]{189}{211}{Geta}  
45 \timechartinterval[severan,mark=217]{c165}{218}{Macrinus}  
46 \timechartinterval[severan,mark=218]{203/204}{222}{Elagabalus}  
47 \timechartinterval[severan,mark=222]{208}{235}{Severus Alexander}  
48  
49 \timechartinterval[thirdcentury,mark=235]{c172/c180}{238}{Maximinus I}  
50 \timechartinterval[thirdcentury,mark=238]{c158}{238}{Gordian I}  
51 \timechartinterval[thirdcentury,mark=238]{192}{238}{Gordian II}  
52 \timechartinterval[thirdcentury,mark=238]{164}{238}{Pupienus}  
53 \timechartinterval[thirdcentury,mark=238]{c164}{238}{Balbinus}  
54 \timechartinterval[thirdcentury,mark=238]{225}{244}{Gordian III}  
55 \timechartinterval[thirdcentury,mark=244]{c204}{249}{Philip I}  
56 \timechartinterval[thirdcentury,mark=249]{c190/200}{251}{Decius}  
57 \timechartinterval[thirdcentury,mark=251]{c206}{253}{Trebonianus Gallus}  
58 \timechartinterval[thirdcentury,mark=253]{c207}{253}{Aemilianus}  
59 \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}{2026-07-10}{0.57.1}
5   {Typesetting chronological charts}

```

### 15.3 Debugging macro

```

\__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

```

\l__timechart_tmpa_bool Scratch boolean variables.
\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
\l__timechart_tmpc_dim
\l__timechart_tmpd_dim
16 \cs_set_eq:NN\l__timechart_tmpa_dim\l_tmpa_dim
17 \cs_set_eq:NN\l__timechart_tmpb_dim\l_tmpb_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_pgfextractxy:nnn` Extract coordinates of `#3` (a PGF point) to dimension variables `#1` and `#2`. This macro simply combines the functionality of `\pgfextractx` and `\pgfextracty`.

```

35 \cs_new:Npn\__timechart_pgfextractxy: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_pgfextractxy:nnn`.)*

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

```

41 \cs_new:Npn\__timechart_hsmash_pgfnode:nmnn #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_pgfextractxy:nnn
66     {\l__timechart_tmpa_dim}{\l__timechart_tmpb_dim}{#1}
67     \__timechart_pgfextractxy: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     \begingroup
84     \tikzset{#2}
85     \pgfmathsetlengthmacro{#1}{\pgflinewidth}
86     \pgfmathsmuggle #1
87     \endgroup
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 for tagging.

```
97 alt/.forward~to = /tikz/alt,  
98 actualtext/.forward~to = /tikz/actualtext,  
99 artifact/.forward~to = /tikz/artifact,  
100 tagging-setup/.forward~to = /tikz/tagging-setup,
```

Keys applicable to whole chart.

```
101 width/.initial=\textwidth,  
102 ystep/.initial=-10pt,  
103 minor~tick~interval/.initial=10,  
104 major~tick~interval/.initial=50,  
105 tolerance~start/.initial=5pt,  
106 tolerance~finish/.initial=5pt,  
107 beyond~length~start/.initial=5pt,  
108 beyond~length~finish/.initial=5pt,  
109 beyond~x~radius~start/.initial=4pt,  
110 beyond~x~radius~finish/.initial=4pt,
```

Shortcuts for tolerance and ‘beyond’ indicators.

```
111 tolerance/.code = {  
112   \pgfkeyssetvalue{/timechart/tolerance~start}{#1}  
113   \pgfkeyssetvalue{/timechart/tolerance~finish}{#1}  
114 },  
115 beyond~length/.code = {  
116   \pgfkeyssetvalue{/timechart/beyond~length~start}{#1}  
117   \pgfkeyssetvalue{/timechart/beyond~length~finish}{#1}  
118 },  
119 beyond~x~radius/.code = {  
120   \pgfkeyssetvalue{/timechart/beyond~x~radius~start}{#1}  
121   \pgfkeyssetvalue{/timechart/beyond~x~radius~finish}{#1}  
122 },
```

Keys applicable to the grid.

```
123 no~grid/.code = { \bool_set_false:N\l__timechart_grid_bool },  
124 grid~top~ysep/.initial={3pt},  
125 grid~bottom~ysep/.initial={3pt},  
126 grid/.style={},
```

Keys applicable to the axis.

```
127 axis~line/.style={  
128   line~cap=rect,  
129 },  
130 axis~ysep/.initial=3pt,  
131 axis/.is~choice,  
132 axis/none/.code  
133   = { \int_set:Nn\l__timechart_axis_int{0} },  
134 axis/above/.code  
135   = { \int_set:Nn\l__timechart_axis_int{1} },  
136 axis/below/.code  
137   = { \int_set:Nn\l__timechart_axis_int{2} },  
138 no~axis/.code = { \int_set:Nn\l__timechart_axis_int{0} },  
139 minor~tick/.style={},  
140 minor~tick~length/.initial=1.5mm,  
141 major~tick/.style={},  
142 major~tick~length/.initial=3mm,
```

```

143 major~tick~label/.style={
144     inner~sep=0,
145     outer~sep=0,
146     anchor=mid~west,
147     rotate=90,
148 },
149 major~tick~eras/.is~choice,
150 major~tick~eras/none/.code
151     = { \int_set:Nn\l__timechart_major_tick_eras_int{0} },
152 major~tick~eras/all/.code
153     = { \int_set:Nn\l__timechart_major_tick_eras_int{1} },
154 major~tick~eras/outer/.code
155     = { \int_set:Nn\l__timechart_major_tick_eras_int{2} },

```

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

```

156 no~autostep/.code = { \bool_set_false:Nz\l__timechart_autostep_bool },
157 ref/.initial={},
158 mark/.initial={},
159 marks/.forward~to=/timechart/mark,
160 circa~uncertainty/.initial=3,
161 interval~minimum~width/.initial=1pt,
162 interval~bar~color/.initial=black,
163 interval~bar~thickness/.initial=8pt,
164 interval~bar~node~name/.initial = {bar~node},
165 interval~mark~color/.initial=gray,
166 interval~label/.style={},
167 interval~label~centered/.style={/timechart/interval~label,text=white},
168 interval~label~centered~background/.style={/timechart/interval~label},
169 interval~label~baseline/.initial=-3pt,
170 interval~label~pos/.is~choice,
171 interval~label~pos/left/.code
172     = { \int_set:Nn\l__timechart_label_pos_int{0} },
173 interval~label~pos/center/.code
174     = { \int_set:Nn\l__timechart_label_pos_int{1} },
175 interval~label~pos/right/.code
176     = { \int_set:Nn\l__timechart_label_pos_int{2} },
177 interval~label~node~name/.initial = {label~node},
178 start~range/.is~choice,
179 start~range/fade/.code
180     = { \int_set:Nn\l__timechart_start_range_type_int{0} },
181 start~range/slant/.code
182     = { \int_set:Nn\l__timechart_start_range_type_int{1} },
183 finish~range/.is~choice,
184 finish~range/fade/.code
185     = { \int_set:Nn\l__timechart_finish_range_type_int{0} },
186 finish~range/slant/.code
187     = { \int_set:Nn\l__timechart_finish_range_type_int{1} },

```

Keys applicable only to texts.

```

188 text~node~name/.initial = {text~node},
189 text/.style={},
190 text~baseline/.initial=-3pt,
191 text~pos/.is~choice,
192 text~pos/left/.code = { \int_set:Nn\l__timechart_text_pos_int{0} },
193 text~pos/center/.code = { \int_set:Nn\l__timechart_text_pos_int{1} },

```

```
194 text~pos/right/.code = { \int_set:Nn\l__timechart_text_pos_int{2} },
```

Keys applicable only to legends.

```
195 legend~item~width/.initial=9mm,
196 legend~item~range~width/.initial=3mm,
```

Shortcuts for positioning.

```
197 left/.code = {
198   \int_set:Nn\l__timechart_label_pos_int{0}
199   \int_set:Nn\l__timechart_text_pos_int{0}
200 },
201 center/.code = {
202   \int_set:Nn\l__timechart_label_pos_int{1}
203   \int_set:Nn\l__timechart_text_pos_int{1}
204 },
205 right/.code = {
206   \int_set:Nn\l__timechart_label_pos_int{2}
207   \int_set:Nn\l__timechart_text_pos_int{2}
208 },
209 }
```

`\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.

```
210 \bool_new:N\l__timechart_grid_bool
211 \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.

```
212 \int_new:N\l__timechart_axis_int
213 \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.

```
214 \int_new:N\l__timechart_major_tick_eras_int
215 \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.

```
216 \bool_new:N\l__timechart_autostep_bool
217 \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.

```
218 \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.

```
219 \int_new:N \l__timechart_text_pos_int
```

(End of definition for `\l__timechart_text_pos_int`.)

`\l__timechart_start_range_type_int` `\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.

```
220 \int_new:N \l__timechart_start_range_type_int
```

```
221 \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.

```
222 \NewDocumentEnvironment{timechart}{0}{m m }
```

```
223 { \__timechart_main_begin:nnn{#1}{#2}{#3} }
```

```
224 { \__timechart_main_end: }
```

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

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

```
225 \cs_new:Npn \__timechart_main_begin:nnn #1#2#3
```

```
226 {
```

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

```
227 \pgfkeys{
```

```
228 /timechart/.cd,
```

```
229 #1,
```

```
230 width/.get=\l__timechart_width_pgf,
```

```
231 ystep/.get=\l__timechart_ystep_pgf,
```

```
232 grid~top~ysep/.get=\l__timechart_grid_top_ysep_pgf,
```

```
233 grid~bottom~ysep/.get=\l__timechart_grid_bottom_ysep_pgf,
```

```
234 minor~tick~interval/.get=\l__timechart_minor_tick_interval_year,
```

```
235 major~tick~interval/.get=\l__timechart_major_tick_interval_year,
```

```
236 }
```

Start the TikZ picture and set up the necessary layers.

```
237 \tikzpicture
```

```
238 \pgfdeclarelayer{grid}
```

```
239 \pgfdeclarelayer{labelbg}
```

```
240 \pgfsetlayers{grid,labelbg,main}
```

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

```
241 \bool_if:NTF\l__timechart_grid_bool
```

```
242 {
```

```
243 \__timechart_set_style_line_width:nn
```

```
244 {\l__timechart_grid_line_width}
```

```
245 {/timechart/grid}
```

```
246 }
```

```

247     { \pgfmathsetlengthmacro{\l__timechart_grid_line_width}{0} }
248 \int_if_zero:nTF{ \l__timechart_axis_int }
249     {
250     \pgfmathsetlengthmacro{\l__timechart_axis_line_width}{0}
251     \pgfmathsetlengthmacro{\l__timechart_major_tick_line_width}{0}
252     \pgfmathsetlengthmacro{\l__timechart_minor_tick_line_width}{0}
253     }
254     {
255     \__timechart_set_style_line_width:nn
256     {\l__timechart_axis_line_width}
257     {/timechart/axis~line}
258     \__timechart_set_style_line_width:nn
259     {\l__timechart_major_tick_line_width}
260     {/timechart/major~tick}
261     \__timechart_set_style_line_width:nn
262     {\l__timechart_minor_tick_line_width}
263     {/timechart/minor~tick}
264     }

```

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.

```

265     \__timechart_parse_date:NNn\l_tmpa_bool\l__timechart_start_year{#2}
266     \__timechart_parse_date:NNn\l_tmpa_bool\l__timechart_finish_year{#3}
267     \pgfmathsetmacro{\l__timechart_start_year}
268     {floor(\l__timechart_start_year)}
269     \pgfmathsetmacro{\l__timechart_finish_year}
270     {floor(\l__timechart_finish_year)}
271     \pgfmathsetmacro{\l__timechart_x}
272     {
273     (
274     \l__timechart_width_pgf
275     - max(
276     \l__timechart_grid_line_width,
277     \l__timechart_axis_line_width,
278     \l__timechart_major_tick_line_width,
279     \l__timechart_minor_tick_line_width
280     )
281     )/(\l__timechart_finish_year-\l__timechart_start_year)
282     }
283     \pgfkeys{
284     /pgf/declare-function={
285     yeartox(\n)=\l__timechart_x*(\n-\l__timechart_start_year);
286     },
287     }

```

Calculate the start and finish  $x$  coordinates.

```

288     \pgfmathsetmacro{\l__timechart_start_x}
289     {yeartox(\l__timechart_start_year)}
290     \pgfmathsetmacro{\l__timechart_finish_x}
291     {yeartox(\l__timechart_finish_year)}

```

Set up tracking of current  $y$  coordinate.

```

292     \pgfmathsetmacro{\l__timechart_current_y}{0}
293     \pgfmathsetmacro{\l__timechart_saved_y}{0}

```

```

294 \pgfmathsetmacro{\l__timechart_auto_reset_minimum_y}{-16000pt}
295 \pgfmathsetmacro{\l__timechart_auto_reset_maximum_y}{16000pt}

```

Calculate some years used in loops.

```

296 \pgfmathsetmacro{\l__timechart_start_plus_year}{
297   \l__timechart_start_year+\l__timechart_minor_tick_interval_year
298 }
299 \pgfmathsetmacro{\l__timechart_start_plusplus_year}{
300   \l__timechart_start_year+(2*\l__timechart_minor_tick_interval_year)
301 }
302 \pgfmathsetmacro{\l__timechart_end_minus_year}{
303   \l__timechart_finish_year-\l__timechart_minor_tick_interval_year
304 }

```

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

```

305 \group_begin:
306 \cs_set_eq:NN\timechartinterval\__timechart_interval_user:Ommm
307 \cs_set_eq:NN\timecharttext\__timechart_text_user:Omm
308 \cs_set_eq:NN\timechartspace\__timechart_space_user:O
309 \cs_set_eq:NN\timechartsety\__timechart_set_y_user:m
310 \cs_set_eq:NN\timechartsavey\__timechart_save_y_user:
311 \cs_set_eq:NN\timechartresety\__timechart_reset_y_user:
312 \cs_set_eq:NN\timechartsetyminimumautoreset
313   \__timechart_set_y_minimum_auto_reset_user:m
314 \cs_set_eq:NN\timechartsetymaximumautoreset
315   \__timechart_set_y_maximum_auto_reset_user:m
316 \cs_set_eq:NN\timechartstepy\__timechart_step_y_user:O
317 \cs_set_eq:NN\timechartfinish\__timechart_main_end_user:
318 }

```

(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`.

```

319 \cs_new:Npn\__timechart_main_end:
320 {
321   \cs_if_eq:NNT\timechartfinish\__timechart_main_end_user:
322     { \__timechart_main_end_user: }
323   \endtikzpicture
324 }

```

(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.

```

325 \cs_new:Npn\__timechart_main_end_user:
326 {

```

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.

```

327 \pgfextracty{\l__timechart_tmpa_dim}
328 { \pgfpointanchor{current~bounding~box}{south} }
329 \pgfextracty{\l__timechart_tmpb_dim}
330 { \pgfpointanchor{current~bounding~box}{north} }
331 \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.

```

332 \dim_compare:nNnTF{\l__timechart_tmpa_dim}>{\l__timechart_tmpb_dim}
333 {
334   \pgfmathsetmacro{\l__timechart_content_bottom_y}{0pt}
335   \pgfmathsetmacro{\l__timechart_content_top_y}{0pt}
336 }
337 {
338   \pgfmathsetmacro{\l__timechart_content_bottom_y}
339   {\l__timechart_tmpa_dim}
340   \pgfmathsetmacro{\l__timechart_content_top_y}
341   {\l__timechart_tmpb_dim}
342 }

```

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

```

343 \bool_if:NTF{\l__timechart_grid_bool}
344 { \__timechart_grid_draw: }
345 { \__timechart_nogrid_bounding_box_set: }
346 \__timechart_axis_draw:

```

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

```

347 \group_end:
348 }

```

*(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.

```

349 \cs_new:Npn\__timechart_grid_draw:
350 {
351   \pgfmathsetmacro{\l__timechart_content_bottom_y}{
352     \l__timechart_content_bottom_y-\l__timechart_grid_bottom_ysep_pgf
353   }
354   \pgfmathsetmacro{\l__timechart_content_top_y}{
355     \l__timechart_content_top_y+\l__timechart_grid_top_ysep_pgf
356   }
357   \pgfonlayer{ grid }
358   \scope[/timechart/grid]
359   \foreach \year in {
360     \l__timechart_start_plus_year,
361     \l__timechart_start_plusplus_year,
362     ...,
363     \l__timechart_end_minus_year
364   } {

```

```

365     \group_begin:
Only draw gridlines at major ticks.
366     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
367     { Mod(\year,\l__timechart_major_tick_interval_year)==0 }
368     \bool_if:NT\l__timechart_tmpa_bool
369     {
370         \pgftransformshift{ \pgfpoint{yeartox(\year)}{0} }
371         \pgfpathmoveto{ \pgfpoint{0}{\l__timechart_content_top_y} }
372         \pgfpathlineto{ \pgfpoint{0}{\l__timechart_content_bottom_y} }
373         \pgfusepath{ draw }
374     }
375     \group_end:
376 }

```

Define and draw the grid node.

```

377     \__timechart_make_rectangle_node:nnnn
378     { \pgfpoint{\l__timechart_start_x}{\l__timechart_content_bottom_y} }
379     { \pgfpoint{\l__timechart_finish_x}{\l__timechart_content_top_y} }
380     { grid }
381     { \c_true_bool }
382     \endscope
383     \endpgfonlayer
384 }

```

*(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.

```

385 \cs_new:Npn\__timechart_axis_draw:
386 {
387     \group_begin:
388     \int_case:nn{ \l__timechart_axis_int }
389     {
390         {0}
391         { \prg_do_nothing: }
392         {1}
393         {
394             \pgftransformshift{
395                 \pgfpoint{0}{
396                     \l__timechart_content_top_y
397                     +\pgfkeysvalueof{/timechart/axis~ysep}
398                 }
399             }
400             \pgfmathsetmacro{\l__timechart_tick_orientation_pgf}{1}
401             \cs_set:Npn\l__timechart_tick_label_anchor_text
402             { mid~west }
403             \cs_set:Npn\l__timechart_zero_tick_before_label_anchor_text
404             { base~west }
405             \cs_set:Npn\l__timechart_zero_tick_after_label_anchor_text
406             { north~west }
407             \__timechart_axis_draw_aux:
408         }
409     {2}

```

```

410     {
411         \pgftransformshift{
412             \pgfpoint{0}{
413                 \l__timechart_content_bottom_y
414                 -\pgfkeysvalueof{/timechart/axis~ysep}
415             }
416         }
417         \pgfmathsetmacro{\l__timechart_tick_orientation_pgf}{-1}
418         \cs_set:Npn\l__timechart_tick_label_anchor_text
419             { mid~east }
420         \cs_set:Npn\l__timechart_zero_tick_before_label_anchor_text
421             { base~east }
422         \cs_set:Npn\l__timechart_zero_tick_after_label_anchor_text
423             { north~east }
424         \__timechart_axis_draw_aux:
425     }
426 }
427 \group_end:
428 }

```

(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.

```

429 \cs_new:Npn\__timechart_axis_draw_aux:
430 {
431     \pgfkeys{
432         /timechart/minor~tick~length/.get=\__timechart_minor_tick_length_pgf,
433         /timechart/major~tick~length/.get=\__timechart_major_tick_length_pgf,
434     }

```

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

```

435     \pgfmathsetmacro{\l__timechart_start_major_tick_year}
436     {
437         \l__timechart_start_year
438         -Mod(
439             \l__timechart_start_year,
440             \l__timechart_major_tick_interval_year
441         )
442     }
443     \pgfmathsetmacro{\l__timechart_start_major_tick_year}
444     {
445         ifthenelse(
446             \l__timechart_start_major_tick_year<\l__timechart_start_year,
447             \l__timechart_start_major_tick_year
448             +\l__timechart_major_tick_interval_year,
449             \l__timechart_start_major_tick_year
450         )
451     }
452     \pgfmathsetmacro{\l__timechart_start_plus_major_tick_year}
453     {
454         \l__timechart_start_major_tick_year
455         +\l__timechart_major_tick_interval_year
456     }

```

```

457 \pgfmathsetmacro{\l__timechart_finish_major_tick_year}
458 {
459   \l__timechart_finish_year
460   -Mod(
461     \l__timechart_finish_year,
462     \l__timechart_major_tick_interval_year
463   )
464 }

```

Loop over years and draw the minor ticks.

```

465 \foreach \year in {
466   \l__timechart_start_year,
467   \l__timechart_start_plus_year,
468   ...,
469   \l__timechart_finish_year
470 } {
471   \pgfmathsetmacro{\x}{yeartox(\year)}
472   \_timechart_pgfmthsetbool:nn
473   {\l__timechart_tmpa_bool}
474   {
475     Mod(
476       \year-\l__timechart_start_major_tick_year,
477       \l__timechart_major_tick_interval_year
478     )=0
479   }
480   \bool_if:NF\l__timechart_tmpa_bool
481   { \_timechart_axis_draw_minor_tick:N\x }
482 }

```

Loop over years and draw the major ticks.

```

483 \foreach \year in {
484   \l__timechart_start_major_tick_year,
485   \l__timechart_start_plus_major_tick_year,
486   ...,
487   \l__timechart_finish_major_tick_year
488 } {
489   \pgfmathsetmacro{\x}{yeartox(\year)}
490   \_timechart_axis_draw_labelled_major_tick:NN\x\year
491 }

```

Define the axis line and define the axis node.

```

492 \_timechart_draw_axis_line
493 \int_case:nn{\l__timechart_axis_int}
494 {
495   {1}
496   {
497     \pgfextracty{\l__timechart_tmpa_dim}
498     { \pgfpointanchor{current~bounding~box}{north} }
499     \pgfmathsetmacro{\l__timechart_axis_top_y}{\l__timechart_tmpa_dim}
500     \pgfmathsetmacro{\l__timechart_axis_bottom_y}{0}
501   }
502   {2}
503   {
504     \pgfextracty{\l__timechart_tmpa_dim}
505     { \pgfpointanchor{current~bounding~box}{south} }

```

```

506     \pgfmathsetmacro{\l__timechart_axis_top_y}{0}
507     \pgfmathsetmacro{\l__timechart_axis_bottom_y}{\l__timechart_tmpa_dim}
508   }
509 }
510 \__timechart_make_rectangle_node:nnnn
511 { \pgfpoint{\l__timechart_start_x}{\l__timechart_axis_bottom_y} }
512 { \pgfpoint{\l__timechart_finish_x}{\l__timechart_axis_top_y} }
513 { axis }
514 { \c_false_bool }
515 }

```

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

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

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

```

516 \cs_new:Npn\__timechart_axis_draw_minor_tick:N #1
517 {
518   \scope[/timechart/minor-tick]
519   \pgfpathmoveto{ \pgfpoint{#1}{0} }
520   \pgfpathlineto{
521     \pgfpoint{#1}{
522       \l__timechart_tick_orientation_pgf
523       *\__timechart_minor_tick_length_pgf
524     }
525   }
526   \pgfusepath{draw}
527   \endscope
528 }

```

(End of definition for `\__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.

```

529 \cs_new:Npn\__timechart_axis_draw_labelled_major_tick:NN #1#2
530 {
531   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{#2==0}
532   \bool_if:NTF \l__timechart_tmpa_bool
533     { \__timechart_axis_draw_zero_tick:N #1 }
534     {
535       \int_case:nn { \l__timechart_major_tick_eras_int }
536       {
537         {0}{ \bool_set_false:N\l__timechart_tmpd_bool }
538         {1}{ \bool_set_true:N\l__timechart_tmpd_bool }
539         {2}
540       }
541       {
542         \__timechart_pgfmathsetbool:nn{\l__timechart_tmpb_bool}
543         { #2==\l__timechart_start_major_tick_year }
544         \__timechart_pgfmathsetbool:nn{\l__timechart_tmpc_bool}
545         { #2==\l__timechart_finish_major_tick_year }
546         \bool_set:Nn\l__timechart_tmpd_bool
547         { \l__timechart_tmpb_bool || \l__timechart_tmpc_bool }

```

```

547     }
548   }
549   \__timechart_axis_draw_major_tick:N #1
550   \__timechart_axis_draw_year_label:nnnnn
551   { #1 }
552   { #2 }
553   { \l__timechart_tmpd_bool }
554   { \l__timechart_tick_label_anchor_text }
555   { 0 }
556   {
557     \l__timechart_tick_orientation_pgf
558     *(\__timechart_major_tick_length_pgf+1mm)
559   }
560 }
561 }

```

(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.

```

562 \cs_new:Npn\__timechart_axis_draw_major_tick:N #1
563 {
564   \scope[/timechart/major~tick]
565   \pgfpathmoveto{ \pgfpoint{#1}{0} }
566   \pgfpathlineto{
567     \pgfpoint{#1}{
568       \l__timechart_tick_orientation_pgf
569       *\__timechart_major_tick_length_pgf
570     }
571   }
572   \pgfusepath{draw}
573   \endscope
574 }

```

(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  $x = #1$ .

```

575 \cs_new:Npn\__timechart_axis_draw_zero_tick:N #1
576 {
577   \group_begin:
578   \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  $\a$ .

```

579   \pgfmathsetmacro{\a}{5}
580   \pgfmathsetlengthmacro{\r}{1mm}
581   \pgfmathsetlengthmacro{\t}{\r*(\cos(\a)-\cos(90-\a))/(\cos(90-\a)-1)}

```

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.

```

582   \scope[/timechart/major~tick,line~join=miter]
583   \group_begin:
584   \pgftransformyscale{\l__timechart_tick_orientation_pgf}

```

```

585 \pgfpathmoveto{\pgfpointorigin}
586 \pgfpathlineto{\pgfpoint{0}{\__timechart_major_tick_length_pgf}}
587 \pgfpatharc{0}{90-\a}{\t-and-\r}
588 \pgfpatharc{270+\a}{360-\a}{\r}
589 \pgfpatharc{180+\a}{270-\a}{\r}
590 \pgfpatharc{90+\a}{180-\a}{\t-and-\r}
591 \pgfpathlineto{\pgfpoint{0}{\__timechart_major_tick_length_pgf-1pt}}
592 \pgfusepath{draw}
593 \group_end:
594 \endscope

```

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

```

595 \__timechart_axis_draw_year_label:nnnnnn
596 { 0 }
597 { -1 }
598 { \c_true_bool }
599 { \l__timechart_zero_tick_before_label_anchor_text }
600 { -.5mm }
601 { \l__timechart_tick_orientation_pgf*5.5mm }
602 \__timechart_axis_draw_year_label:nnnnnn
603 { 0 }
604 { 1 }
605 { \c_true_bool }
606 { \l__timechart_zero_tick_after_label_anchor_text }
607 { .5mm }
608 { \l__timechart_tick_orientation_pgf*5.5mm }
609 \group_end:
610 }

```

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

`\__timechart_axis_draw_line` Draw the axis line itself.

```

611 \cs_new:Npn\__timechart_draw_axis_line
612 {
613   \scope[/timechart/axis~line]
614   \pgfpathmoveto{\pgfpoint{\l__timechart_start_x}{0}}
615   \pgfpathlineto{\pgfpoint{\l__timechart_finish_x}{0}}
616   \pgfusepath{draw}
617   \endscope
618 }

```

*(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).

```

619 \cs_new:Npn\__timechart_axis_draw_year_label:nnnnnn #1#2#3#4#5#6
620 {
621   \group_begin:
622   \pgftransformshift{\pgfpoint{#1+#5}{#6}}

```

```

623 \pgfmathtruncatemacro{\absyear}{ abs(#2) }
624 \scope[/timechart/major-tick-label]
625 \bool_if:NTF #3
626 {
627   \__timechart_pgfmthsetbool:nn{\l__timechart_tmpa_bool}{#2<0}
628   \bool_if:NTF\l__timechart_tmpa_bool
629   { \cs_set_eq:NN\__timechart_make_year:n\timechartmakebeforeyear }
630   { \cs_set_eq:NN\__timechart_make_year:n\timechartmakeafteryear }
631 }
632 { \cs_set_eq:NN\__timechart_make_year:n\use:n }
633 \__timechart_hsmash_pgfnode:nnnnn
634 {rectangle}
635 {#4}
636 {\__timechart_make_year:n{\absyear}}
637 {}
638 {}
639 \endscope
640 \group_end:
641 }

```

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

`\timechartmakebeforeyear` `\timechartmakeafteryear` User-redefineable macros to format a year as before or after the epoch.

```

642 \cs_new:Npn\timechartmakebeforeyear #1
643 {
644   #1\nobreakspace\textsc{bce}
645 }
646 \cs_new:Npn\timechartmakeafteryear #1
647 {
648   #1\nobreakspace\textsc{ce}
649 }

```

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

## 15.12 Bounding box

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

```

650 \cs_new:Npn\__timechart_nogrid_bounding_box_set:
651 {
652   \pgfpathmoveto
653   { \pgfpoint{\l__timechart_start_x}{\l__timechart_content_bottom_y} }
654   \pgfpathmoveto
655   { \pgfpoint{\l__timechart_finish_x}{\l__timechart_content_top_y} }
656   \pgfusepath{discard}
657 }

```

(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.

```

658 \cs_new:Npn\__timechart_set_y_user:m #1

```

```

659 {
660   \pgfmathsetmacro{\l__timechart_current_y}{#1}
661 }

```

(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.

```

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

```

(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.

```

666 \cs_new:Npn\__timechart_reset_y_user:
667 {
668   \pgfmathsetmacro{\l__timechart_current_y}{\l__timechart_saved_y}
669 }

```

(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.

```

670 \cs_new:Npn\__timechart_set_y_minimum_auto_reset_user:m #1
671 {
672   \pgfmathsetmacro{\l__timechart_auto_reset_minimum_y}{#1}
673 }

```

(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.

```

674 \cs_new:Npn\__timechart_set_y_maximum_auto_reset_user:m #1
675 {
676   \pgfmathsetmacro{\l__timechart_auto_reset_maximum_y}{#1}
677 }

```

(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.

```

678 \NewDocumentCommand{\__timechart_step_y_user:0}{ 0{1} }
679 {
680   \pgfmathsetmacro{\l__timechart_current_y}
681     {\l__timechart_current_y+#1*\l__timechart_ystep_pgf}
682   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
683     {
684     or(
685       \l__timechart_current_y<\l__timechart_auto_reset_minimum_y,
686       \l__timechart_current_y>\l__timechart_auto_reset_maximum_y

```

```

687     )
688   }
689   \bool_if:nT{\l__timechart_tmpa_bool}
690   { \pgfmathsetmacro{\l__timechart_current_y}{\l__timechart_saved_y} }
691 }

```

(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.

```

692 \cs_new:Npn\__timechart_if_x_in_bounds:nT #1#2
693 {
694   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{
695     and(
696       #1>=\l__timechart_start_x,
697       #1<=\l__timechart_finish_x
698     )
699   }
700   \bool_if:NT\l__timechart_tmpa_bool
701   {#2}
702 }
703

```

(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.

```

704 \cs_new:Npn\__timechart_if_x_range_intersect_bounds:nnT #1#2#3
705 {
706   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}{
707     or(
708       or(
709         and(
710           #2>=\l__timechart_start_x,
711           #2<=\l__timechart_finish_x
712         ),
713         and(
714           #1>=\l__timechart_start_x,
715           #1<=\l__timechart_finish_x
716         )
717       ),
718       and(
719         #1<\l__timechart_start_x,
720         #2>\l__timechart_finish_x
721       )
722     )
723   }
724   \bool_if:NT\l__timechart_tmpa_bool
725   {#3}
726 }

```

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

## 15.15 Date and date range parsing

`\_timechart_parse_date_or_daterange:NNNNn` 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.

```

727 \cs_new:Npn\_timechart_parse_date_or_daterange:NNNNn #1#2#3#4#5#6
728 {
729   \bool_set:Nn #1 {\_timechart_is_nondaterange_p:w #6/\q_stop}
730   \bool_set_inverse:N #1
731   \bool_if:nTF #1
732     { \_timechart_parse_range:w #2#3#4#5\q_mark #6\q_stop }
733     {
734       \_timechart_parse_date:NNn #2#3{#6}
735       \bool_set_eq:NN #4#2
736       \pgfmathsetmacro{#5}{#3}
737     }
738 }

```

*(End of definition for \\_timechart\_parse\_date\_or\_daterange:NNNNn.)*

`\_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.

```

739 \cs_new:Npn\_timechart_is_nondaterange_p:w #1/#2\q_stop
740 {
741   \tl_if_empty_p:n{#2}
742 }

```

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

`\_timechart_parse_range:w` To be called in the form `\_timechart_parse_range:w⟨cmin⟩⟨min⟩⟨cmax⟩⟨max⟩\q-mark⟨text⟩\q_stop`. Parse `⟨text⟩` (known to represent a date range) into minimum circa indicator boolean variable `⟨cmin⟩`, minimum variable `⟨min⟩`, maximum circa indicator boolean variable `⟨cmax⟩`, maximum variable `⟨max⟩`.

```

743 \cs_new:Npn\_timechart_parse_range:w #1#2#3#4\q_mark #5/#6\q_stop
744 {
745   \_timechart_parse_date:NNn #1#2{#5}
746   \_timechart_parse_date:NNn #3#4{#6}
747 }

```

*(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.

```

748 \cs_new:Npn\__timechart_parse_date:NNn #1#2#3
749 {
750   \bool_set:Nn #1 { \__timechart_is_circa_p:w #3c\q_stop }
751   \bool_if:NTF #1
752     { \__timechart_parse_circa_date:w #2\q_mark #3\q_stop }
753     { \__timechart_parse_noncirca_date:Nn #2{#3} }
754 }

```

*(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.

```

755 \cs_new:Npn\__timechart_is_circa_p:w #1c#2\q_stop
756 {
757   \tl_if_empty_p:n{#1}
758 }

```

*(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.

```

759 \cs_new:Npn\__timechart_parse_circa_date:w #1\q_mark c#2\q_stop
760 {
761   \__timechart_parse_noncirca_date:Nn #1{#2}
762 }

```

*(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.

```

763 \cs_new:Npn\__timechart_parse_noncirca_date:Nn #1#2
764 {
765   \bool_if:NTF { \__timechart_is_before_p:w #2-\q_stop }
766     { \__timechart_parse_before_date:w #1\q_mark #2\q_stop }
767     { \__timechart_parse_signed_date:w #1\q_mark #2-0-0\q_stop }
768 }

```

*(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 `-`.

```

769 \cs_new:Npn\__timechart_is_before_p:w #1-#2\q_stop
770 {
771   \tl_if_empty_p:n{#1}
772 }

```

*(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.

```

773 \cs_new:Npn\_timechart_parse_before_date:w #1\q_mark-#2\q_stop
774 {
775   \_timechart_parse_signed_date:w #1-\q_mark #2-0-0\q_stop
776 }

```

*(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.

```

777 \cs_new:cpn{c__timechart_year_days_pgf}{365}
778 \cs_new:cpn{c__timechart_month_days_1_pgf}{31}
779 \cs_new:cpn{c__timechart_month_days_2_pgf}{28}
780 \cs_new:cpn{c__timechart_month_days_3_pgf}{31}
781 \cs_new:cpn{c__timechart_month_days_4_pgf}{30}
782 \cs_new:cpn{c__timechart_month_days_5_pgf}{31}
783 \cs_new:cpn{c__timechart_month_days_6_pgf}{30}
784 \cs_new:cpn{c__timechart_month_days_7_pgf}{31}
785 \cs_new:cpn{c__timechart_month_days_8_pgf}{31}
786 \cs_new:cpn{c__timechart_month_days_9_pgf}{30}
787 \cs_new:cpn{c__timechart_month_days_10_pgf}{31}
788 \cs_new:cpn{c__timechart_month_days_11_pgf}{30}
789 \cs_new:cpn{c__timechart_month_days_12_pgf}{31}
790 \cs_new:cpn{c__timechart_cumulative_days_1_pgf}{0}
791 \cs_new:cpn{c__timechart_cumulative_days_2_pgf}{31}
792 \cs_new:cpn{c__timechart_cumulative_days_3_pgf}{59}
793 \cs_new:cpn{c__timechart_cumulative_days_4_pgf}{90}
794 \cs_new:cpn{c__timechart_cumulative_days_5_pgf}{120}
795 \cs_new:cpn{c__timechart_cumulative_days_6_pgf}{151}
796 \cs_new:cpn{c__timechart_cumulative_days_7_pgf}{181}
797 \cs_new:cpn{c__timechart_cumulative_days_8_pgf}{212}
798 \cs_new:cpn{c__timechart_cumulative_days_9_pgf}{243}
799 \cs_new:cpn{c__timechart_cumulative_days_10_pgf}{273}
800 \cs_new:cpn{c__timechart_cumulative_days_11_pgf}{304}
801 \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`.

```

802 \cs_new:Npn\__timechart_parse_signed_date:w #1#2\q_mark #3-#4-#5\q_stop
803 {
804   \pgfmathtruncatemacro{\__timechart_parsed_year_pgf}{#2#3}
805   \pgfmathtruncatemacro{\__timechart_parsed_month_pgf}{#4}
806   \pgfmathtruncatemacro{\__timechart_parsed_day_pgf}{#5}
807   \__timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
808     or(
809       \__timechart_parsed_month_pgf < 1,
810       \__timechart_parsed_month_pgf > 12,
811     )
812   }
813   \bool_if:NTF\l_tmpa_bool
814   {

```

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

```

815     \pgfmathsetmacro{#1}{#2#3}
816   }
817 {

```

*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.*

```

818   \cs_set_eq:NN\l__timechart_year_days_pgf\c__timechart_year_days_pgf
819   \cs_set_eq:Nc\l__timechart_month_days_pgf
820     { c__timechart_month_days_\__timechart_parsed_month_pgf _pgf }
821   \cs_set_eq:Nc\l__timechart_cumulative_days_pgf
822     { c__timechart_cumulative_days_\__timechart_parsed_month_pgf _pgf }
823   \__timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
824     or(
825       Mod(\__timechart_parsed_year_pgf,400) == 0,
826       and(
827         Mod(\__timechart_parsed_year_pgf,4) == 0,
828         Mod(\__timechart_parsed_year_pgf,100) != 0
829       )
830     )
831   }
832   \bool_if:NT\l_tmpa_bool
833   {
834     \pgfmathsetmacro{\l__timechart_year_days_pgf}
835       { \l__timechart_year_days_pgf+1 }
836     \__timechart_pgfmathsetbool:nn{\l_tmpb_bool}
837       { \__timechart_parsed_month == 1 }
838     \bool_if:NF\l_tmpb_bool
839     {
840       \__timechart_pgfmathsetbool:nn{\l_tmpb_bool}
841       { \__timechart_parsed_month == 2 }
842       \bool_if:NF\l_tmpb_bool
843       {
844         \pgfmathsetmacro{\l__timechart_month_days_pgf}
845           { \l__timechart_month_days_pgf + 1 }
846       }
847       {
848         \pgfmathsetmacro{\l__timechart_cumulative_days_pgf}
849           { \l__timechart_cumulative_days_pgf + 1 }
850       }

```

```

851     }
852   }
853   \_timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
854     or(
855       \_timechart_parsed_day_pgf < 1,
856       \_timechart_parsed_day_pgf > \l__timechart_month_days_pgf,
857     )
858   }
859   \bool_if:NTF\l_tmpa_bool
860   {

```

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

```

861     \pgfmathsetmacro{#1}
862     {
863       #2#3
864       + \l__timechart_cumulative_days_pgf/\l__timechart_year_days_pgf
865     }
866   }
867   {

```

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

```

868     \pgfmathsetmacro{#1}
869     {
870       #2#3
871       + (
872         \l__timechart_cumulative_days_pgf
873         + \_timechart_parsed_day_pgf
874       )/\l__timechart_year_days_pgf
875     }
876   }
877 }
878 }

```

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

## 15.16 Interval drawing

### 15.16.1 Preliminaries

\l\_\_timechart\_start\_is\_range\_bool  
 \l\_\_timechart\_interval\_start\_min\_circa\_bool  
 \l\_\_timechart\_interval\_start\_max\_circa\_bool  
 \l\_\_timechart\_finish\_is\_range\_bool  
 \l\_\_timechart\_interval\_finish\_min\_circa\_bool  
 \l\_\_timechart\_interval\_finish\_max\_circa\_bool

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’.

```

879 \bool_new:N\l__timechart_start_is_range_bool
880 \bool_new:N\l__timechart_interval_start_min_circa_bool
881 \bool_new:N\l__timechart_interval_start_max_circa_bool
882 \bool_new:N\l__timechart_finish_is_range_bool
883 \bool_new:N\l__timechart_interval_finish_min_circa_bool
884 \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

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

### 15.16.3 Main macros

`\_timechart_interval_user:0mmm` 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.

```
887 \NewDocumentCommand{\_timechart_interval_user:0mmm}{ O{} m m m }
888 {
889   \_timechart_interval:nnnn{#1}{#2}{#3}{#4}
890 }
```

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

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

```
891 \cs_new:Npn \_timechart_interval:nnnn #1#2#3#4
892 {
893   \_timechart_debug:n{
894     *****
895   }
896   \_timechart_debug:n{Interval~#2~to~#3~"#4"}
```

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

```
897   \group_begin:
898   \_timechart_parse_date_or_daterange:NNNNNn
899   \l__timechart_start_is_range_bool
900   \l__timechart_interval_start_min_circa_bool
901   \l__timechart_interval_start_min_year
902   \l__timechart_interval_start_max_circa_bool
903   \l__timechart_interval_start_max_year
904   {#2}
905   \_timechart_parse_date_or_daterange:NNNNNn
906   \l__timechart_finish_is_range_bool
907   \l__timechart_interval_finish_min_circa_bool
908   \l__timechart_interval_finish_min_year
909   \l__timechart_interval_finish_max_circa_bool
910   \l__timechart_interval_finish_max_year
911   {#3}
```

Check the results of parsing.

```
912   \_timechart_pgfmathsetbool:nn{\l_tmpa_bool}{
913     and(
914       \l__timechart_interval_start_min_year
915       <= \l__timechart_interval_start_max_year,
916     and(
917       \l__timechart_interval_start_max_year
918       <= \l__timechart_interval_finish_min_year,
919     \l__timechart_interval_finish_min_year
```

```

920         <= \l__timechart_interval_finish_max_year
921     )
922 )
923 }

```

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.

```

924 \bool_if:NTF\l_tmpa_bool
925 {
926     \__timechart_interval_checked:nn{#1}{#4}
927     \group_end:
928     \bool_if:NT\l__timechart_autostep_bool
929     { \__timechart_step_y_user:0 }
930 }
931 {
932     \msg_error:nnnn{timechart}{interval_dates_invalid}{#2}{#3}
933     \group_end:
934 }
935 }

```

(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.

```

936 \cs_new:Npn \__timechart_interval_checked:nn #1#2
937 {

```

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

```

938     \pgfqkeys{/timechart}{
939         #1,
940         circa-uncertainty/.get=\l__timechart_circa_uncertainty_year
941     }

```

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’).

```

942     \bool_if:NTF\l__timechart_interval_start_min_circa_bool
943     {
944         \pgfmathsetmacro{\l__timechart_interval_start_min_x}
945         { yeartox(\l__timechart_interval_start_min_year
946             - \l__timechart_circa_uncertainty_year) }

```

```

947     }
948     {
949         \pgfmathsetmacro{\l__timechart_interval_start_min_x}
950         { yeartox(\l__timechart_interval_start_min_year) }
951     }
952     \bool_if:NTF\l__timechart_interval_finish_max_circa_bool
953     {
954         \pgfmathsetmacro{\l__timechart_interval_finish_max_x}
955         { yeartox(\l__timechart_interval_finish_max_year
956             + \l__timechart_circa_uncertainty_year) }
957     }
958     {
959         \pgfmathsetmacro{\l__timechart_interval_finish_max_x}
960         { yeartox(\l__timechart_interval_finish_max_year) }
961     }

```

Proceed if some part of the interval is visible.

```

962     \__timechart_if_x_range_intersect_bounds:nnT
963     {\l__timechart_interval_start_min_x}
964     {\l__timechart_interval_finish_max_x}
965     { \__timechart_draw_visible_interval:nn{#1}{#2} }
966 }

```

*(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.

```

967 \bool_new:N\l__timechart_interval_start_beyond_bool
968 \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

```

969 \cs_new:Npn\__timechart_draw_visible_interval:nn #1#2
970 {

```

Retrieve PGF key values.

```

971     \pgfqkeys{/timechart}{
972         tolerance~start/.get=\l__timechart_start_tolerance_pgf,
973         tolerance~finish/.get=\l__timechart_finish_tolerance_pgf,
974         beyond~length~start/.get=\l__timechart_start_beyond_length_pgf,
975         beyond~length~finish/.get=\l__timechart_finish_beyond_length_pgf,
976         beyond~x~radius~start/.get=\l__timechart_start_beyond_x_radius_pgf,
977         beyond~x~radius~finish/.get=\l__timechart_finish_beyond_x_radius_pgf,
978         ref/.get=\l__timechart_ref_text,
979         mark/.get=\l__timechart_mark_text,
980         interval~minimum~width/.get=\l__timechart_minimum_width_pgf,
981         interval~bar~color/.get=\l__timechart_bar_color,
982         interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
983         interval~mark~color/.get=\l__timechart_mark_color,
984         interval~label~baseline/.get=\l__timechart_text_baseline_pgf,
985         interval~label~node~name/.get=\l__timechart_interval_label_node_name,
986     }

```

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

```

987 \pgfmathsetmacro{\l__timechart_start_tolerance_x}{
988 \l__timechart_start_x-(\l__timechart_start_tolerance_pgf)
989 }
990 \pgfmathsetmacro{\l__timechart_finish_tolerance_x}{
991 \l__timechart_finish_x+(\l__timechart_finish_tolerance_pgf)
992 }
993 \pgfmathsetmacro{\l__timechart_start_beyond_x}{
994 \l__timechart_start_x-(\l__timechart_start_beyond_length_pgf)
995 }
996 \pgfmathsetmacro{\l__timechart_finish_beyond_x}{
997 \l__timechart_finish_x+(\l__timechart_finish_beyond_length_pgf)
998 }

```

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

```

999 \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
1000 { .5*\l__timechart_bar_thickness_pgf }
1001 \cs_set:Npn\l__timechart_label_text{#2}
1002 \bool_if:NTF\l__timechart_interval_start_max_circa_bool
1003 {
1004 \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1005 { yeartox(\l__timechart_interval_start_max_year
1006 + \l__timechart_circa_uncertainty_year) }
1007 }
1008 {
1009 \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1010 { yeartox(\l__timechart_interval_start_max_year) }
1011 }
1012 \bool_if:NTF\l__timechart_interval_finish_min_circa_bool
1013 {
1014 \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1015 { yeartox(\l__timechart_interval_finish_min_year
1016 - \l__timechart_circa_uncertainty_year) }
1017 }
1018 {
1019 \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1020 { yeartox(\l__timechart_interval_finish_min_year) }
1021 }

```

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.

```

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

```

```

1034     )
1035   }
1036   \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1037     { \l__timechart_interval_start_max_x }
1038 }

```

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

```

1039   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1040   {
1041     (\l__timechart_interval_finish_max_x
1042     -\l__timechart_interval_start_min_x)
1043     > \l__timechart_minimum_width_pgf
1044   }
1045   \bool_if:NF \l__timechart_tmpa_bool
1046   {
1047     \pgfmathsetmacro{\l__timechart_width_adjust}
1048       { .5*\l__timechart_minimum_width_pgf }
1049     \pgfmathsetmacro{\l__timechart_interval_start_max_x}
1050       { \l__timechart_interval_start_max_x-\l__timechart_width_adjust }
1051     \pgfmathsetmacro{\l__timechart_interval_finish_min_x}
1052       { \l__timechart_interval_finish_min_x+\l__timechart_width_adjust }
1053     \pgfmathsetmacro{\l__timechart_interval_start_min_x}
1054     {
1055       min(
1056         \l__timechart_interval_start_min_x,
1057         \l__timechart_interval_start_max_x
1058       )
1059     }
1060     \pgfmathsetmacro{\l__timechart_interval_finish_max_x}
1061     {
1062       max(
1063         \l__timechart_interval_finish_max_x,
1064         \l__timechart_interval_finish_min_x
1065       )
1066     }
1067   }
1068   \__timechart_debug:n{Calculated+adjusted~x~coords:}
1069   \__timechart_debug:n{~
1070     \l__timechart_interval_start_min_x/\l__timechart_interval_start_max_x~
1071     to~
1072     \l__timechart_interval_finish_min_x/\l__timechart_interval_finish_max_x
1073   }

```

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).

```

1074   \__timechart_pgfmathsetbool:nn{\l__timechart_interval_start_beyond_bool}
1075     {\l__timechart_interval_start_min_x<=\l__timechart_start_tolerance_x}
1076   \bool_if:NT\l__timechart_interval_start_beyond_bool
1077   {
1078     \cs_set_eq:NN
1079     \__timechart_interval_start_clip_path:
1080     \__timechart_interval_start_clip_path_beyond:
1081   }
1082   \__timechart_pgfmathsetbool:nn{\l__timechart_interval_finish_beyond_bool}

```

```

1083     {\l__timechart_interval_finish_max_x>=\l__timechart_finish_tolerance_x}
1084     \bool_if:NT\l__timechart_interval_finish_beyond_bool
1085     {
1086         \cs_set_eq:NN
1087         \__timechart_interval_finish_clip_path:
1088         \__timechart_interval_finish_clip_path_beyond:
1089     }

```

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.

```

1090     \__timechart_interval_definite_compute:
1091     \__timechart_interval_startrange_compute:
1092     \__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.

```

1093     \pgftransformshift{ \pgfpoint{0}{\l__timechart_current_y} }
1094     \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.

```

1095     \__timechart_interval_clip:
1096     \__timechart_interval_startrange_draw:
1097     \__timechart_interval_finishrange_draw:
1098     \__timechart_interval_solid_draw:
1099     \__timechart_interval_mark:
1100     \__timechart_interval_define_bar_node:
1101     \endpgfscope
1102     \__timechart_interval_label:
1103 }

```

(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_iii:` 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.

```

1104 \cs_new:Npn\_timechart_interval_definite_compute:
1105 {
1106   \_timechart_debug:n{Computing:~definite-part}
1107   \_timechart_pgfmathsetbool:nn{\l_timechart_tmpa_bool}
1108   { \l_timechart_interval_finish_min_x < \l_timechart_start_x }
1109   \bool_if:nTF{
1110     \l_timechart_tmpa_bool && \l_timechart_interval_start_beyond_bool
1111   }
1112   {
1113     \cs_set_eq:NN
1114       \_timechart_interval_solid_draw:\prg_do_nothing:
1115     \cs_set_eq:NN
1116       \_timechart_interval_startrange_compute:\prg_do_nothing:
1117     \cs_set_eq:NN
1118       \_timechart_interval_startrange_draw:\prg_do_nothing:
1119   }
1120   {
1121     \_timechart_interval_definite_compute_ii:
1122   }
1123 }

```

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.

```

1124 \cs_new:Npn\_timechart_interval_definite_compute_ii:
1125 {
1126   \_timechart_debug:n{~(Stage-II)}
1127   \_timechart_pgfmathsetbool:nn{\l_timechart_tmpa_bool}
1128   { \l_timechart_interval_start_max_x > \l_timechart_finish_x }
1129   \bool_if:nTF{
1130     \l_timechart_tmpa_bool && \l_timechart_interval_finish_beyond_bool
1131   }
1132   {
1133     \cs_set_eq:NN
1134       \_timechart_interval_solid_draw:\prg_do_nothing:
1135     \cs_set_eq:NN
1136       \_timechart_interval_finishrange_compute:\prg_do_nothing:
1137     \cs_set_eq:NN
1138       \_timechart_interval_finishrange_draw:\prg_do_nothing:
1139   }
1140   {
1141     \_timechart_interval_definite_compute_iii:

```

```

1142     }
1143 }

```

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

```

1144 \cs_new:Npn\__timechart_interval_definite_compute_iii:
1145 {
1146   \__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.

```

1147   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1148   { \l__timechart_interval_start_max_x <= \l__timechart_start_x }
1149   \bool_if:nTF{
1150     \l__timechart_tmpa_bool && \l__timechart_interval_start_beyond_bool
1151   }
1152   {
1153     \cs_set_eq:NN
1154     \l__timechart_interval_solid_start_x\l__timechart_start_beyond_x
1155     \cs_set_eq:NN
1156     \__timechart_interval_startrange_compute:\prg_do_nothing:
1157     \cs_set_eq:NN
1158     \__timechart_interval_startrange_draw:\prg_do_nothing:
1159   }
1160   {
1161     \cs_set_eq:NN
1162     \l__timechart_interval_solid_start_x
1163     \l__timechart_interval_start_max_x
1164   }

```

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.

```

1165   \__timechart_pgfmathsetbool:nn{\l__timechart_tmpb_bool}
1166   { \l__timechart_interval_finish_min_x >= \l__timechart_finish_x }
1167   \bool_if:nTF{
1168     \l__timechart_tmpb_bool && \l__timechart_interval_finish_beyond_bool
1169   }
1170   {
1171     \cs_set_eq:NN
1172     \l__timechart_interval_solid_finish_x\l__timechart_finish_beyond_x
1173     \cs_set_eq:NN
1174     \__timechart_interval_finishrange_compute:\prg_do_nothing:
1175     \cs_set_eq:NN
1176     \__timechart_interval_finishrange_draw:\prg_do_nothing:
1177   }
1178   {
1179     \cs_set_eq:NN
1180     \l__timechart_interval_solid_finish_x
1181     \l__timechart_interval_finish_min_x
1182   }
1183 }

```

*(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.

```

1184 \cs_new:Npn\_timechart_interval_startrange_compute:
1185 {
1186   \_timechart_debug:n{Computing:~startrange}
1187   \_timechart_pgfmathsetbool:nn{\_timechart_tmpa_bool}
1188   {
1189     \l_timechart_interval_start_min_x
1190     == \l_timechart_interval_start_max_x
1191   }
1192   \bool_if:NTF\_timechart_tmpa_bool
1193   {
1194     \cs_set_eq:NN\_timechart_interval_startrange_draw:\prg_do_nothing:
1195   }
1196   {
1197     \_timechart_interval_startrange_compute_ii:
1198   }
1199 }

```

(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.

```

1200 \bool_new:N\_timechart_interval_startrange_finish_beyond_bool
1201 \cs_new:Npn\_timechart_interval_startrange_compute_ii:
1202 {
1203   \_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.

```

1204   \bool_if:NTF\_timechart_interval_start_beyond_bool
1205   {
1206     \cs_set_eq:NN
1207     \l_timechart_interval_startrange_start_x\l_timechart_start_x
1208     \pgfmathsetmacro{\_timechart_interval_startrange_start_level_pgf}
1209     {
1210       (\l_timechart_interval_startrange_start_x
1211        -\l_timechart_interval_start_min_x)
1212       /
1213       (\l_timechart_interval_start_max_x
1214        -\l_timechart_interval_start_min_x)
1215     }
1216   }
1217   {
1218     \cs_set_eq:NN
1219     \l_timechart_interval_startrange_start_x
1220     \l_timechart_interval_start_min_x
1221     \pgfmathsetmacro
1222     {\_timechart_interval_startrange_start_level_pgf}{0}
1223   }

```

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.

```

1224   \_timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1225   { \l__timechart_interval_start_max_x > \l__timechart_finish_x }
1226   \bool_set:Nn\l__timechart_interval_startrange_finish_beyond_bool
1227   {
1228     \l__timechart_tmpa_bool
1229     && \l__timechart_interval_finish_beyond_bool
1230   }
1231   \bool_if:NTF\l__timechart_interval_startrange_finish_beyond_bool
1232   {
1233     \cs_set_eq:NN
1234     \l__timechart_interval_startrange_finish_x
1235     \l__timechart_finish_x
1236     \pgfmathsetmacro{\l__timechart_interval_startrange_finish_level_pgf}
1237     {
1238       (\l__timechart_interval_startrange_finish_x
1239        -\l__timechart_interval_start_min_x)
1240       /
1241       (\l__timechart_interval_start_max_x
1242        -\l__timechart_interval_start_min_x)
1243     }
1244   }
1245   {
1246     \cs_set_eq:NN
1247     \l__timechart_interval_startrange_finish_x
1248     \l__timechart_interval_start_max_x
1249     \pgfmathsetmacro
1250     {\l__timechart_interval_startrange_finish_level_pgf}{1}
1251   }
1252   \int_case:nn {\l__timechart_start_range_type_int}
1253   {
1254     {0}{
1255       \cs_set_eq:NN
1256       \l__timechart_interval_startrange_draw:
1257       \l__timechart_interval_startrange_draw_pseudofade:
1258     }
1259     {1}{
1260       \l__timechart_interval_startrange_slant_compute:
1261     }
1262   }
1263 }

```

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

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

```

1264 \cs_new:Npn \_timechart_interval_startrange_slant_compute:
1265 {
1266   \_timechart_debug:n{~(Startrange~slant)}
1267   \bool_if:NTF\l__timechart_interval_start_beyond_bool
1268   {

```

```

1269     \bool_if:NTF\l__timechart_interval_startrange_finish_beyond_bool
1270     {
1271         \cs_set_eq:NN
1272             \__timechart_interval_both_clip_path:
1273             \__timechart_interval_startrange_clip_path_beyond_both:
1274     }
1275     {
1276         \cs_set_eq:NN
1277             \__timechart_interval_start_clip_path:
1278             \__timechart_interval_startrange_clip_path_beyond_start:
1279     }
1280 }
1281 {
1282     \bool_if:NTF\l__timechart_interval_startrange_finish_beyond_bool
1283     {
1284         \cs_set_eq:NN
1285             \__timechart_interval_both_clip_path:
1286             \__timechart_interval_startrange_clip_path_beyond_finish:
1287     }
1288     {
1289         \cs_set_eq:NN
1290             \__timechart_interval_start_clip_path:
1291             \__timechart_interval_startrange_clip_path_start:
1292     }
1293 }
1294 \cs_if_eq:NNTF\__timechart_interval_solid_draw:\prg_do_nothing:
1295 {
1296     \cs_set_eq:NN
1297         \__timechart_interval_startrange_draw:
1298         \__timechart_interval_startfinishrange_draw_slant:
1299 }
1300 {
1301     \cs_set_eq:NN
1302         \l__timechart_interval_solid_start_x
1303         \l__timechart_start_beyond_x
1304     \cs_set_eq:NN
1305         \__timechart_interval_startrange_draw:
1306         \prg_do_nothing:
1307 }
1308 }

```

(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.

```

1309 \cs_new:Npn\__timechart_interval_finishrange_compute:
1310 {
1311     \__timechart_debug:n{Computing:~finishrange}
1312     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}
1313     {
1314         \l__timechart_interval_finish_max_x
1315         == \l__timechart_interval_finish_min_x

```

```

1316     }
1317     \bool_if:NTF\l__timechart_tmpa_bool
1318     {
1319         \cs_set_eq:NN\__timechart_interval_finishrange_draw:\prg_do_nothing:
1320     }
1321     {
1322         \__timechart_interval_finishrange_compute_ii:
1323     }
1324 }

```

(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.

```

1325 \bool_new:N\l__timechart_interval_finishrange_start_beyond_bool
1326 \cs_new:Npn\__timechart_interval_finishrange_compute_ii:
1327 {
1328     \__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.

```

1329     \bool_if:NTF\l__timechart_interval_finish_beyond_bool
1330     {
1331         \cs_set_eq:NN
1332         \l__timechart_interval_finishrange_finish_x
1333         \l__timechart_finish_x
1334         \pgfmathsetmacro{\l__timechart_interval_finishrange_finish_level_pgf}
1335         {
1336             (\l__timechart_interval_finishrange_finish_x
1337              -\l__timechart_interval_finish_max_x)
1338             /
1339             (\l__timechart_interval_finish_min_x
1340              -\l__timechart_interval_finish_max_x)
1341         }
1342     }
1343     {
1344         \cs_set_eq:NN
1345         \l__timechart_interval_finishrange_finish_x
1346         \l__timechart_interval_finish_max_x
1347         \pgfmathsetmacro
1348         {\l__timechart_interval_finishrange_finish_level_pgf}{0}
1349     }
1350

```

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.

```

1351     \__timechart_pgfmathsetbool:nn{\l__timechart_tmpa_bool}

```

```

1352     { \l__timechart_interval_finish_min_x < \l__timechart_start_x }
1353 \bool_set:Nn\l__timechart_interval_finishrange_start_beyond_bool
1354     {
1355     \l__timechart_tmpa_bool
1356     && \l__timechart_interval_start_beyond_bool
1357     }
1358 \bool_if:NTF\l__timechart_interval_finishrange_start_beyond_bool
1359     {
1360     \cs_set_eq:NN
1361     \l__timechart_interval_finishrange_start_x
1362     \l__timechart_start_x
1363     \pgfmathsetmacro{\l__timechart_interval_finishrange_start_level_pgf}
1364     {
1365     (\l__timechart_interval_finishrange_start_x
1366     -\l__timechart_interval_finish_max_x)
1367     /
1368     (\l__timechart_interval_finish_min_x
1369     -\l__timechart_interval_finish_max_x)
1370     }
1371     }
1372     {
1373     \cs_set_eq:NN
1374     \l__timechart_interval_finishrange_start_x
1375     \l__timechart_interval_finish_min_x
1376     \pgfmathsetmacro
1377     {\l__timechart_interval_finishrange_start_level_pgf}{1}
1378     }
1379 \int_case:mn {\l__timechart_finish_range_type_int}
1380     {
1381     {0}{
1382     \cs_set_eq:NN
1383     \__timechart_interval_finishrange_draw:
1384     \__timechart_interval_finishrange_draw_pseudofade:
1385     }
1386     {1}{
1387     \__timechart_interval_finishrange_slant_compute:
1388     }
1389     }
1390 }

```

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

mechart\_interval\_finishrange\_slant\_compute:

Compute the clipping and drawing necessary for a slant finishrange.

```

1391 \cs_new:Npn \__timechart_interval_finishrange_slant_compute:
1392     {
1393     \__timechart_debug:n{~(Finishrange-slant)}
1394     \bool_if:NTF\l__timechart_interval_finish_beyond_bool
1395     {
1396     \bool_if:NTF\l__timechart_interval_finishrange_start_beyond_bool
1397     {
1398     \cs_set_eq:NN
1399     \__timechart_interval_both_clip_path:
1400     \__timechart_interval_finishrange_clip_path_beyond_both:
1401     }

```

```

1402     {
1403       \cs_set_eq:NN
1404         \__timechart_interval_finish_clip_path:
1405         \__timechart_interval_finishrange_clip_path_beyond_finish:
1406     }
1407 }
1408 {
1409   \bool_if:NTF\l__timechart_interval_finishrange_start_beyond_bool
1410   {
1411     \cs_set_eq:NN
1412       \__timechart_interval_both_clip_path:
1413       \__timechart_interval_finishrange_clip_path_beyond_start:
1414   }
1415   {
1416     \cs_set_eq:NN
1417       \__timechart_interval_finish_clip_path:
1418       \__timechart_interval_finishrange_clip_path_finish:
1419   }
1420 }
1421 \cs_if_eq:NNTF\__timechart_interval_solid_draw:\prg_do_nothing:
1422 {
1423   \cs_set_eq:NN
1424     \__timechart_interval_finishrange_draw:
1425     \__timechart_interval_startfinishrange_draw_slant:
1426 }
1427 {
1428   \cs_set_eq:NN
1429     \l__timechart_interval_solid_finish_x
1430     \l__timechart_finish_beyond_x
1431   \cs_set_eq:NN
1432     \__timechart_interval_finishrange_draw:
1433     \prg_do_nothing:
1434 }
1435 }

```

(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.

```

1436 \cs_new:Npn\__timechart_interval_solid_draw:
1437 {
1438   \__timechart_debug:n{
1439     Drawing:~solid~\l__timechart_interval_solid_start_x~
1440     to~\l__timechart_interval_solid_finish_x
1441   }
1442   \pgfpathrectanglecorners{
1443     \pgfpoint
1444       {\l__timechart_interval_solid_start_x}
1445       {-\l__timechart_bar_half_thickness_pgf}
1446   }{
1447     \pgfpoint

```

```

1448         {\l__timechart_interval_solid_finish_x}
1449         {\l__timechart_bar_half_thickness_pgf}
1450     }
1451     \pgfsetfillcolor{\l__timechart_bar_color}
1452     \pgfusepath{fill}
1453 }

```

(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

```

1454 \cs_new:Npn\__timechart_interval_startrange_draw_pseudofade:
1455 {
1456     \__timechart_debug:n{Drawing:~startrange,~pseudofade}
1457     \__timechart_debug:n{~
1458         \l__timechart_interval_startrange_start_x~
1459         (\l__timechart_interval_startrange_start_level_pgf)~
1460         to~\l__timechart_interval_startrange_finish_x~
1461         (\l__timechart_interval_startrange_finish_level_pgf)
1462     }
1463     \__timechart_interval_draw_pseudofade:nnnnn
1464     {\l__timechart_interval_startrange_start_x}
1465     {\l__timechart_interval_startrange_finish_x}
1466     {\l__timechart_bar_color}
1467     {\l__timechart_interval_startrange_start_level_pgf}
1468     {\l__timechart_interval_startrange_finish_level_pgf}
1469     \bool_if:NT\l__timechart_interval_start_beyond_bool
1470     { \__timechart_interval_startrange_start_beyond_draw_fade: }
1471     \bool_if:NT\l__timechart_interval_startrange_finish_beyond_bool
1472     { \__timechart_interval_startrange_finish_beyond_draw_fade: }
1473 }

```

(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.

```

1474 \cs_new:Npn\__timechart_interval_startrange_start_beyond_draw_fade:
1475 {
1476     \__timechart_debug:n{Drawing:~startrange,~pseudofade-beyond~start}
1477     \__timechart_debug:n{~
1478         \l__timechart_start_beyond_x~to~\l__timechart_start_x
1479     }
1480     \pgfscope
1481     \pgfpathrectanglecorners{
1482         \pgfpoint
1483         {\l__timechart_start_beyond_x}
1484         {-\l__timechart_bar_half_thickness_pgf}
1485     }{
1486         \pgfpoint
1487         {\l__timechart_start_x}
1488         {\l__timechart_bar_half_thickness_pgf}
1489     }
1490     \pgfsetfillcolor{\l__timechart_bar_color}
1491     \pgfsetfillopacity
1492     {\l__timechart_interval_startrange_start_level_pgf}

```

```

1493   \pgfusepath{fill}
1494   \endpgfscope
1495 }

```

(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.

```

1496 \cs_new:Npn\__timechart_interval_startrange_finish_beyond_draw_fade:
1497 {
1498   \__timechart_debug:n{Drawing:~startrange,~pseudofade~beyond~finish}
1499   \__timechart_debug:n{~
1500     \l__timechart_finish_beyond_x~to~\l__timechart_finish_x
1501   }
1502   \pgfscope
1503   \pgfpathrectanglecorners{
1504     \pgfpoint
1505       {\l__timechart_finish_x}
1506       {-\l__timechart_bar_half_thickness_pgf}
1507   }{
1508     \pgfpoint
1509       {\l__timechart_finish_beyond_x}
1510       {\l__timechart_bar_half_thickness_pgf}
1511   }
1512   \pgfsetfillcolor{\l__timechart_bar_color}
1513   \pgfsetfillopacity
1514     {\l__timechart_interval_startrange_finish_level_pgf}
1515   \pgfusepath{fill}
1516   \endpgfscope
1517 }

```

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

`chart_interval_finishrange_draw_pseudofade:` Draw a fading startrange and excess indicators

```

1518 \cs_new:Npn\__timechart_interval_finishrange_draw_pseudofade:
1519 {
1520   \__timechart_debug:n{Drawing:~finishrange,~pseudofade}
1521   \__timechart_debug:n{~
1522     \l__timechart_interval_finishrange_start_x~
1523     (\l__timechart_interval_finishrange_start_level_pgf)~
1524     to~\l__timechart_interval_finishrange_finish_x~
1525     (\l__timechart_interval_finishrange_finish_level_pgf)
1526   }
1527   \__timechart_interval_draw_pseudofade:nnnnn
1528     {\l__timechart_interval_finishrange_start_x}
1529     {\l__timechart_interval_finishrange_finish_x}
1530     {\l__timechart_bar_color}
1531     {\l__timechart_interval_finishrange_start_level_pgf}
1532     {\l__timechart_interval_finishrange_finish_level_pgf}
1533   \bool_if:NT\l__timechart_interval_finish_beyond_bool
1534     { \__timechart_interval_finishrange_finish_beyond_draw_fade: }
1535   \bool_if:NT\l__timechart_interval_finishrange_start_beyond_bool
1536     { \__timechart_interval_finishrange_start_beyond_draw_fade: }
1537 }

```

(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.

```
1538
1539 \cs_new:Npn\__timechart_interval_finishrange_finish_beyond_draw_fade:
1540 {
1541   \__timechart_debug:n{Drawing:~finishrange,~pseudofade~beyond~finish}
1542   \__timechart_debug:n{~
1543     \l__timechart_finish_beyond_x~to~\l__timechart_finish_x
1544   }
1545   \pgfscope
1546   \pgfpathrectanglecorners{
1547     \pgfpoint
1548       {\l__timechart_finish_beyond_x}
1549     {-\l__timechart_bar_half_thickness_pgf}
1550   }{
1551     \pgfpoint
1552       {\l__timechart_finish_x}
1553     {\l__timechart_bar_half_thickness_pgf}
1554   }
1555   \pgfsetfillcolor{\l__timechart_bar_color}
1556   \pgfsetfillopacity
1557     {\l__timechart_interval_finishrange_finish_level_pgf}
1558   \pgfusepath{fill}
1559   \endpgfscope
1560 }
```

(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.

```
1561 \cs_new:Npn\__timechart_interval_finishrange_start_beyond_draw_fade:
1562 {
1563   \__timechart_debug:n{Drawing:~finishrange,~pseudofade~beyond~start}
1564   \__timechart_debug:n{~
1565     \l__timechart_start_beyond_x~to~\l__timechart_start_x
1566   }
1567   \pgfscope
1568   \pgfpathrectanglecorners{
1569     \pgfpoint
1570       {\l__timechart_start_x}
1571     {-\l__timechart_bar_half_thickness_pgf}
1572   }{
1573     \pgfpoint
1574       {\l__timechart_start_beyond_x}
1575     {\l__timechart_bar_half_thickness_pgf}
1576   }
1577   \pgfsetfillcolor{\l__timechart_bar_color}
1578   \pgfsetfillopacity
1579     {\l__timechart_interval_finishrange_start_level_pgf}
1580   \pgfusepath{fill}
1581   \endpgfscope
1582 }
```

(End of definition for `\_timechart_interval_finishrange_start_beyond_draw_fade:`)

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

```
1583 \pgfmathsetmacro{\l_timechart_fadestep_pgf}{1}
1584 \cs_new:Npn\_timechart_interval_draw_pseudofade:nmmnn #1#2#3#4#5
1585 {
1586   \_timechart_debug:n{Pseudofade:~#1~to~#2,~color~#3,~opacity~#4~to~#5}
1587   \group_begin:
1588   \pgfmathsetmacro{\start}{#1}
1589   \pgfmathsetmacro{\stop}{#2}
1590   \pgfmathsetmacro{\fadestep}
1591     {(\stop-\start)/(\floor((\stop-\start)/\l_timechart_fadestep_pgf)+1)}
1592   \pgfmathsetmacro{\startnext}{\start+\fadestep}
1593   \pgfmathsetmacro{\stopprev}{\stop-\fadestep}
1594   \pgfmathsetmacro{\denominator}{abs(\stop-\start)+\fadestep}
1595   \foreach \xa in {\start,\startnext,...,\stopprev} {
1596     \pgfmathsetlengthmacro{\o}
1597       {((\stop-\xa)*#4+(\xa-\start+\fadestep)*#5)/\denominator}
1598     \pgfscope
1599     \pgfpathrectanglecorners{
1600       \pgfpoint{\xa}{-\l_timechart_bar_half_thickness_pgf}
1601     }{
1602       \pgfpoint{\xa+\fadestep}{\l_timechart_bar_half_thickness_pgf}
1603     }
1604     \pgfsetfillcolor{#3}
1605     \pgfsetfillopaicity{\o}
1606     \pgfusepath{fill}
1607     \endpgfscope
1608   }
1609   \group_end:
1610 }
```

(End of definition for `\_timechart_interval_draw_pseudofade:nmmnn`.)

### 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 draw (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

```
1611 \cs_new:Npn\_timechart_interval_startfinishrange_draw_slant:
1612 {
1613   \pgfpathrectanglecorners{
1614     \pgfpoint
1615       {\l_timechart_start_beyond_x}
1616       {-\l_timechart_bar_half_thickness_pgf}
1617   }{
1618     \pgfpoint
1619       {\l_timechart_finish_beyond_x}
1620       {\l_timechart_bar_half_thickness_pgf}
1621   }
1622   \pgfsetfillcolor{\l_timechart_bar_color}
1623   \pgfusepath{fill}
```

```
1624 }
(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` and `\c__timechart_right_far_x` No-clipping paths actually use the following  $x$ -coordinates, which are large in the positive and negative direction.

```
1625 \pgfmathsetmacro{\c__timechart_left_far_x}{-16000pt}
1626 \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.

```
1627 \cs_new:Npn\__timechart_interval_start_clip_path_none:
1628 {
1629   \__timechart_debug:n{~Start~clip~path:~none}
1630   \pgfpathlineto{
1631     \pgfpoint
1632     {\c__timechart_left_far_x}
1633     {-\l__timechart_bar_thickness_pgf}
1634   }
1635   \pgfpathlineto{
1636     \pgfpoint
1637     {\c__timechart_left_far_x}
1638     {\l__timechart_bar_thickness_pgf}
1639   }
1640 }
(End of definition for \__timechart_interval_start_clip_path_none:.)
```

`\__timechart_interval_finish_clip_path_none:` No clip path for finish.

```
1641 \cs_new:Npn\__timechart_interval_finish_clip_path_none:
1642 {
1643   \__timechart_debug:n{~Finish~clip~path:~none}
1644   \pgfpathmoveto{
1645     \pgfpoint
1646     {\c__timechart_right_far_x}
1647     {\l__timechart_bar_thickness_pgf}
1648   }
1649   \pgfpathlineto{
1650     \pgfpoint
1651     {\c__timechart_right_far_x}
1652     {-\l__timechart_bar_thickness_pgf}
1653   }
1654 }
(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.

```
1655 \cs_new:Npn\__timechart_interval_start_clip_path_beyond:
1656 {
1657   \__timechart_debug:n{~Start~clip~path:~beyond}
1658   \pgfpathlineto{
1659     \pgfpoint
1660     {\l__timechart_start_beyond_x}
1661     {-\l__timechart_bar_thickness_pgf}
1662   }
1663   \pgfpathlineto{
1664     \pgfpoint
1665     {\l__timechart_start_beyond_x}
1666     {-\l__timechart_bar_half_thickness_pgf}
1667   }
1668   \pgfpatharc
1669     {-90}
1670     {90}
1671     {\l__timechart_start_beyond_x_radius_pgf
1672     ~and~\l__timechart_bar_half_thickness_pgf}
1673   \pgfpathlineto{
1674     \pgfpoint
1675     {\l__timechart_start_beyond_x}
1676     {\l__timechart_bar_thickness_pgf}
1677   }
1678 }
```

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

`timechart_interval_finish_clip_path_beyond:` Clip path for finish beyond indicator.

```
1679 \cs_new:Npn\__timechart_interval_finish_clip_path_beyond:
1680 {
1681   \__timechart_debug:n{~Finish~clip~path:~beyond}
1682   \pgfpathmoveto{
1683     \pgfpoint
1684     {\l__timechart_finish_beyond_x}
1685     {\l__timechart_bar_thickness_pgf}
1686   }
1687   \pgfpathlineto{
1688     \pgfpoint
1689     {\l__timechart_finish_beyond_x}
1690     {\l__timechart_bar_half_thickness_pgf}
1691   }
1692   \pgfpatharc
1693     {90}
1694     {270}
1695     {\l__timechart_finish_beyond_x_radius_pgf
1696     ~and~\l__timechart_bar_half_thickness_pgf}
1697   \pgfpathlineto{
1698     \pgfpoint
1699     {\l__timechart_finish_beyond_x}
1700     {-\l__timechart_bar_thickness_pgf}
1701   }
1702 }
```

(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.

```
1703 \cs_new:Npn\__timechart_interval_startrange_clip_path_start:
1704 {
1705   \__timechart_debug:n{~Start~clip~path:~startrange}
1706   \pgfpathlineto{
1707     \pgfpoint
1708     {\l__timechart_interval_startrange_start_x}
1709     {-\l__timechart_bar_thickness_pgf}
1710   }
1711   \pgfpathlineto{
1712     \pgfpoint
1713     {\l__timechart_interval_startrange_start_x}
1714     {-\l__timechart_bar_half_thickness_pgf}
1715   }
1716   \pgfpathlineto{
1717     \pgfpoint
1718     {\l__timechart_interval_startrange_finish_x}
1719     {\l__timechart_bar_half_thickness_pgf}
1720   }
1721   \pgfpathlineto{
1722     \pgfpoint
1723     {\l__timechart_interval_startrange_finish_x}
1724     {\l__timechart_bar_thickness_pgf}
1725   }
1726 }
```

(End of definition for `\__timechart_interval_startrange_clip_path_start:`)

`\__timechart_interval_finishrange_clip_path_finish:` Clip path for finishrange.

```
1727 \cs_new:Npn\__timechart_interval_finishrange_clip_path_finish:
1728 {
1729   \__timechart_debug:n{~Finish~clip~path:~finishrange}
1730   \pgfpathmoveto{
1731     \pgfpoint
1732     {\l__timechart_interval_finishrange_finish_x}
1733     {\l__timechart_bar_thickness_pgf}
1734   }
1735   \pgfpathlineto{
1736     \pgfpoint
1737     {\l__timechart_interval_finishrange_finish_x}
1738     {\l__timechart_bar_half_thickness_pgf}
1739   }
1740   \pgfpathlineto{
1741     \pgfpoint
1742     {\l__timechart_interval_finishrange_start_x}
1743     {-\l__timechart_bar_half_thickness_pgf}
1744   }
1745   \pgfpathlineto{
1746     \pgfpoint
1747     {\l__timechart_interval_finishrange_start_x}
```

```

1748     {-\l__timechart_bar_thickness_pgf}
1749   }
1750 }

```

(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.

```

1751 \cs_new:Npn \__timechart_interval_startrange_clip_path_beyond_start:
1752 {
1753   \__timechart_debug:n{~Start~clip~path:~startrange,~beyond~start}
1754   \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
1755     \l__timechart_bar_thickness_pgf
1756     *\l__timechart_interval_startrange_start_level_pgf
1757   }
1758   \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
1759     .5*\l__timechart_tmpa_pgf
1760   }
1761   \pgfpathlineto{
1762     \pgfpoint
1763       {\l__timechart_start_beyond_x}
1764       {-\l__timechart_bar_thickness_pgf}
1765   }
1766   \pgfpathlineto{
1767     \pgfpoint
1768       {\l__timechart_start_beyond_x}
1769       {-\l__timechart_bar_half_thickness_pgf}
1770   }
1771   \pgfpatharc
1772     {-90}
1773     {90}
1774     {\l__timechart_start_beyond_x_radius_pgf
1775     ~and~\l__timechart_tmpb_pgf}
1776   \pgfpathlineto{
1777     \pgfpointadd{
1778       \pgfpoint
1779         {\l__timechart_start_x}
1780         {-\l__timechart_bar_half_thickness_pgf}
1781     }{
1782       \pgfpoint{0}{\l__timechart_tmpa_pgf}
1783     }
1784   }
1785   \pgfpathlineto{
1786     \pgfpoint
1787       {\l__timechart_interval_start_max_x}
1788       {\l__timechart_bar_half_thickness_pgf}
1789   }
1790   \pgfpathlineto{
1791     \pgfpoint
1792       {\l__timechart_interval_start_max_x}
1793       {\l__timechart_bar_thickness_pgf}
1794   }

```

1795 }  
}

(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.

```
1796 \cs_new:Npn\__timechart_interval_finishrange_clip_path_beyond_finish:
1797 {
1798   \__timechart_debug:n{~Finish~clip~path:~finishrange,~beyond~finish}
1799   \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
1800     \l__timechart_bar_thickness_pgf
1801     *\l__timechart_interval_finishrange_finish_level_pgf
1802   }
1803   \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
1804     .5*\l__timechart_tmpa_pgf
1805   }
1806   \pgfpathmoveto{
1807     \pgfpoint
1808       {\l__timechart_finish_beyond_x}
1809       {\l__timechart_bar_thickness_pgf}
1810   }
1811   \pgfpathlineto{
1812     \pgfpoint
1813       {\l__timechart_finish_beyond_x}
1814       {\l__timechart_bar_half_thickness_pgf}
1815   }
1816   \pgfpatharc
1817     {90}
1818     {270}
1819     {\l__timechart_finish_beyond_x_radius_pgf
1820     ~and~\l__timechart_tmpb_pgf}
1821   \pgfpathlineto{
1822     \pgfpointadd{
1823       \pgfpoint
1824         {\l__timechart_finish_x}
1825         {\l__timechart_bar_half_thickness_pgf}
1826     }{
1827       \pgfpoint{0}{-\l__timechart_tmpa_pgf}
1828     }
1829   }
1830   \pgfpathlineto{
1831     \pgfpoint
1832       {\l__timechart_interval_finish_min_x}
1833       {-\l__timechart_bar_half_thickness_pgf}
1834   }
1835   \pgfpathlineto{
1836     \pgfpoint
1837       {\l__timechart_interval_finish_min_x}
1838       {-\l__timechart_bar_thickness_pgf}
1839   }
1840 }
```

(End of definition for `\__timechart_interval_finishrange_clip_path_beyond_finish:`)

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

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.

```

1841 \cs_new:Npn\__timechart_interval_startrange_clip_path_beyond_finish:
1842 {
1843   \__timechart_debug:n{~Two-sided~clip~path:~startrange,~beyond~finish}
1844   \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
1845     \l__timechart_bar_thickness_pgf
1846     *\l__timechart_interval_startrange_finish_level_pgf
1847   }
1848   \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
1849     .5*\l__timechart_tmpc_pgf
1850   }
1851   \pgfpathmoveto{
1852     \pgfpointadd{
1853       \pgfpoint
1854       {\l__timechart_finish_beyond_x}
1855       {-\l__timechart_bar_half_thickness_pgf}
1856     }{
1857       \pgfpoint{0}{\l__timechart_tmpc_pgf}
1858     }
1859   }
1860   \pgfpatharc
1861     {90}
1862     {270}
1863     {\l__timechart_start_beyond_x_radius_pgf
1864     ~and~\l__timechart_tmpd_pgf}
1865   \pgfpathlineto{
1866     \pgfpoint
1867     {\l__timechart_finish_beyond_x}
1868     {-\l__timechart_bar_thickness_pgf}
1869   }
1870   \pgfpathlineto{
1871     \pgfpoint
1872     {\l__timechart_interval_start_min_x}
1873     {-\l__timechart_bar_thickness_pgf}
1874   }
1875   \pgfpathlineto{
1876     \pgfpoint
1877     {\l__timechart_interval_start_min_x}
1878     {-\l__timechart_bar_half_thickness_pgf}
1879   }
1880   \pgfpathlineto{
1881     \pgfpointadd{
1882       \pgfpoint
1883       {\l__timechart_finish_x}
1884       {-\l__timechart_bar_half_thickness_pgf}
1885     }{
1886       \pgfpoint{0}{\l__timechart_tmpc_pgf}
1887     }
1888   }
1889 }

```

(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.

```
1890 \cs_new:Npn\__timechart_interval_finishrange_clip_path_beyond_start:
1891 {
1892   \__timechart_debug:n{~Two-sided~clip~path:~finishrange,~beyond-start}
1893   \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
1894     \l__timechart_bar_thickness_pgf
1895     *\l__timechart_interval_finishrange_start_level_pgf
1896   }
1897   \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
1898     .5*\l__timechart_tmpc_pgf
1899   }
1900   \pgfpathmoveto{
1901     \pgfpointadd{
1902       \pgfpoint
1903         {\l__timechart_start_beyond_x}
1904         {\l__timechart_bar_half_thickness_pgf}
1905     }{
1906       \pgfpoint{0}{-\l__timechart_tmpc_pgf}
1907     }
1908   }
1909   \pgfpatharc
1910     {-90}
1911     {90}
1912     {\l__timechart_start_beyond_x_radius_pgf
1913     ~and~\l__timechart_tmpd_pgf}
1914   \pgfpathlineto{
1915     \pgfpoint
1916       {\l__timechart_start_beyond_x}
1917       {\l__timechart_bar_thickness_pgf}
1918   }
1919   \pgfpathlineto{
1920     \pgfpoint
1921       {\l__timechart_interval_finish_max_x}
1922       {\l__timechart_bar_thickness_pgf}
1923   }
1924   \pgfpathlineto{
1925     \pgfpoint
1926       {\l__timechart_interval_finish_max_x}
1927       {\l__timechart_bar_half_thickness_pgf}
1928   }
1929   \pgfpathlineto{
1930     \pgfpointadd{
1931       \pgfpoint
1932         {\l__timechart_start_x}
1933         {\l__timechart_bar_half_thickness_pgf}
1934     }{
1935       \pgfpoint{0}{-\l__timechart_tmpc_pgf}
1936     }
1937   }
1938 }
```

*(End of definition for `\__timechart_interval_finishrange_clip_path_beyond_start:`)*

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

`_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.

```

1939 \cs_new:Npn\__timechart_interval_startrange_clip_path_beyond_both:
1940 {
1941   \__timechart_debug:n{~Two-sided~clip~path:~startrange,~beyond~both}
1942   \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
1943     \l__timechart_bar_thickness_pgf
1944     *\l__timechart_interval_startrange_start_level_pgf
1945   }
1946   \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
1947     .5*\l__timechart_tmpa_pgf
1948   }
1949   \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
1950     \l__timechart_bar_thickness_pgf
1951     *\l__timechart_interval_startrange_finish_level_pgf
1952   }
1953   \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
1954     .5*\l__timechart_tmpc_pgf
1955   }
1956   \pgfpathmoveto{
1957     \pgfpointadd{
1958       \pgfpoint
1959       {\l__timechart_finish_beyond_x}
1960       {-\l__timechart_bar_half_thickness_pgf}
1961     }{
1962       \pgfpoint{0}{\l__timechart_tmpc_pgf}
1963     }
1964   }
1965   \pgfpatharc
1966     {90}
1967     {270}
1968     {\l__timechart_finish_beyond_x_radius_pgf
1969     ~and~\l__timechart_tmpd_pgf}
1970   \pgfpathlineto{
1971     \pgfpoint
1972     {\l__timechart_finish_beyond_x}
1973     {-\l__timechart_bar_thickness_pgf}
1974   }
1975   \pgfpathlineto{
1976     \pgfpoint
1977     {\l__timechart_start_beyond_x}
1978     {-\l__timechart_bar_thickness_pgf}
1979   }
1980   \pgfpathlineto{
1981     \pgfpoint
1982     {\l__timechart_start_beyond_x}
1983     {-\l__timechart_bar_half_thickness_pgf}
1984   }
1985   \pgfpatharc
1986     {-90}

```

```

1987     {90}
1988     {\l__timechart_start_beyond_x_radius_pgf
1989     ~and~\l__timechart_tmpb_pgf}
1990 \pgfpathlineto{
1991   \pgfpointadd{
1992     \pgfpoint
1993     {\l__timechart_start_x}
1994     {-\l__timechart_bar_half_thickness_pgf}
1995   }{
1996     \pgfpoint{0}{\l__timechart_tmpa_pgf}
1997   }
1998 }
1999 \pgfpathlineto{
2000   \pgfpointadd{
2001     \pgfpoint
2002     {\l__timechart_finish_x}
2003     {-\l__timechart_bar_half_thickness_pgf}
2004   }{
2005     \pgfpoint{0}{\l__timechart_tmpc_pgf}
2006   }
2007 }
2008 }

```

(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.

```

2009 \cs_new:Npn\__timechart_interval_finishrange_clip_path_beyond_both:
2010 {
2011   \__timechart_debug:n{~Two-sided~clip~path:~finishrange,~beyond-both}
2012   \pgfmathsetmacro{\l__timechart_tmpa_pgf}{
2013     \l__timechart_bar_thickness_pgf
2014     * \l__timechart_interval_finishrange_finish_level_pgf
2015   }
2016   \pgfmathsetmacro{\l__timechart_tmpb_pgf}{
2017     .5 * \l__timechart_tmpa_pgf
2018   }
2019   \pgfmathsetmacro{\l__timechart_tmpc_pgf}{
2020     \l__timechart_bar_thickness_pgf
2021     * \l__timechart_interval_finishrange_start_level_pgf
2022   }
2023   \pgfmathsetmacro{\l__timechart_tmpd_pgf}{
2024     .5 * \l__timechart_tmpc_pgf
2025   }
2026   \pgfpathmoveto{
2027     \pgfpointadd{
2028       \pgfpoint
2029       {\l__timechart_start_beyond_x}
2030       {\l__timechart_bar_half_thickness_pgf}
2031     }{
2032       \pgfpoint{0}{-\l__timechart_tmpc_pgf}
2033     }
2034   }

```

```

2035 \pgfpatharc
2036   {-90}
2037   {90}
2038   {\l__timechart_start_beyond_x_radius_pgf
2039     ~and~\l__timechart_tmpd_pgf}
2040 \pgfpathlineto{
2041   \pgfpoint
2042     {\l__timechart_start_beyond_x}
2043     {\l__timechart_bar_thickness_pgf}
2044 }
2045 \pgfpathlineto{
2046   \pgfpoint
2047     {\l__timechart_finish_beyond_x}
2048     {\l__timechart_bar_thickness_pgf}
2049 }
2050 \pgfpathlineto{
2051   \pgfpoint
2052     {\l__timechart_finish_beyond_x}
2053     {\l__timechart_bar_half_thickness_pgf}
2054 }
2055 \pgfpatharc
2056   {90}
2057   {270}
2058   {\l__timechart_finish_beyond_x_radius_pgf
2059     ~and~\l__timechart_tmpb_pgf}
2060 \pgfpathlineto{
2061   \pgfpointadd{
2062     \pgfpoint
2063       {\l__timechart_finish_x}
2064       {\l__timechart_bar_half_thickness_pgf}
2065   }{
2066     \pgfpoint{0}{-\l__timechart_tmpe_pgf}
2067   }
2068 }
2069 \pgfpathlineto{
2070   \pgfpointadd{
2071     \pgfpoint
2072       {\l__timechart_start_x}
2073       {\l__timechart_bar_half_thickness_pgf}
2074   }{
2075     \pgfpoint{0}{-\l__timechart_tmpe_pgf}
2076   }
2077 }
2078 }

```

(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.

```

2079 \cs_new:Npn\__timechart_interval_clip:
2080   {
2081     \__timechart_debug:n{Clipping:}

```

```

2082 \pgfinterruptboundingbox
2083 \_timechart_interval_both_clip_path:
2084 \pgfpathclose
2085 \pgfusepath{clip}
2086 \endpgfinterruptboundingbox
2087 }

```

(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.

```

2088 \cs_new:Npn\_timechart_interval_both_clip_path:
2089 {
2090   \cs_if_exist_use:NF\_timechart_interval_finish_clip_path:
2091     { \_timechart_interval_finish_clip_path_none: }
2092   \cs_if_exist_use:NF\_timechart_interval_start_clip_path:
2093     { \_timechart_interval_start_clip_path_none: }
2094 }

```

(End of definition for `\_timechart_interval_both_clip_path:`)

### 15.16.7 Marks

```

2095 \msg_new:nnn{timechart}{interval_mark_outside}
2096 { Ignoring-mark-outside-interval-definite-part-at-date-#1 }

```

`\_timechart_interval_mark:` Draw marks on an interval.

```

2097 \cs_new:Npn\_timechart_interval_mark:
2098 {
2099   \pgfscope
2100   \bool_if:NTF\l_timechart_interval_start_beyond_bool
2101     {
2102       \pgfmathsetmacro{\l_timechart_interval_mark_min_x}
2103         { max(\l_timechart_interval_start_max_x,\l_timechart_start_x) }
2104     }
2105     {
2106       \cs_set_eq:NN\l_timechart_interval_mark_min_x\l_timechart_interval_start_max_x
2107     }
2108   \bool_if:NTF\l_timechart_interval_finish_beyond_bool
2109     {
2110       \pgfmathsetmacro{\l_timechart_interval_mark_max_x}
2111         { min(\l_timechart_interval_finish_min_x,\l_timechart_finish_x) }
2112     }
2113     {
2114       \cs_set_eq:NN\l_timechart_interval_mark_max_x\l_timechart_interval_finish_min_x
2115     }
2116   \pgfsetstrokecolor{\l_timechart_mark_color}
2117   \foreach \year in \l_timechart_mark_text
2118     {
2119       \pgfmathsetmacro{\l_timechart_mark_x}{yeartox(\year)}
2120       \_timechart_pgfmathsetbool:nn{\l_timechart_tmpa_bool}
2121         {
2122           and (
2123             \l_timechart_mark_x >= \l_timechart_interval_mark_min_x,

```

```

2124         \l__timechart_mark_x <= \l__timechart_interval_mark_max_x
2125     )
2126 }
2127 \bool_if:NTF\l__timechart_tmpa_bool
2128 {
2129     \pgfpathmoveto{
2130         \pgfpoint
2131             {\l__timechart_mark_x}
2132             {-\l__timechart_bar_half_thickness_pgf}
2133     }
2134     \pgfpathlineto{
2135         \pgfpoint
2136             {\l__timechart_mark_x}
2137             {\l__timechart_bar_half_thickness_pgf}
2138     }
2139     \pgfusepath{draw}
2140 }
2141 { \msg_warning:nne{timechart}{interval_mark_outside}{\year} }
2142 }
2143 \endpgfscope
2144 }

```

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

### 15.16.8 Bar node

`\__timechart_interval_define_bar_node:` Define a node with into which the bar fits exactly.

```

2145 \cs_new:Npn\__timechart_interval_define_bar_node:
2146 {
2147     \__timechart_make_rectangle_node:nmmn
2148     {
2149         \pgfpoint
2150             {\l__timechart_interval_start_min_x}
2151             {-\l__timechart_bar_half_thickness_pgf}
2152     }{
2153         \pgfpoint
2154             {\l__timechart_interval_finish_max_x}
2155             {\l__timechart_bar_half_thickness_pgf}
2156     }
2157     {\pgfkeysvalueof{/timechart/interval-bar-node-name}}
2158     {\c_false_bool}
2159 }

```

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

### 15.16.9 Label

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

```

2160 \cs_new:Npn\__timechart_interval_label:
2161 {
2162     \str_if_empty:NF \l__timechart_label_text
2163     {
2164         \pgfinterruptboundingbox
2165         \int_case:nn {\l__timechart_label_pos_int}

```

```

2166     {
2167     {0}{ \_timechart_interval_label_left: }
2168     {1}{ \_timechart_interval_label_center: }
2169     {2}{ \_timechart_interval_label_right: }
2170     }
2171     \endpgfinterruptboundingbox
2172   }
2173 }

```

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

\\_timechart\_interval\_label\_left: Place the label on the left of the interval.

```

2174 \cs_new:Npn\_timechart_interval_label_left:
2175 {
2176   \_timechart_if_x_in_bounds:nT{\l\_timechart_interval_start_min_x}
2177   {
2178     \group_begin:
2179     \pgftransformshift{
2180       \pgfpoint
2181       {\l\_timechart_interval_start_min_x}
2182       {\l\_timechart_text_baseline_pgf}
2183     }
2184     \node[/timechart/interval-label,anchor=base-east]
2185     (\l\_timechart_interval_label_node_name)
2186     at (0,0)
2187     {
2188       \_timechart_make_ref:NN
2189       \l\_timechart_ref_text
2190       \l\_timechart_label_text
2191     };
2192     \group_end:
2193   }
2194 }

```

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

\\_timechart\_interval\_label\_right: Place the label on the right of the interval.

```

2195 \cs_new:Npn\_timechart_interval_label_right:
2196 {
2197   \_timechart_if_x_in_bounds:nT{\l\_timechart_interval_finish_max_x}
2198   {
2199     \group_begin:
2200     \pgftransformshift{
2201       \pgfpoint
2202       {\l\_timechart_interval_finish_max_x}
2203       {\l\_timechart_text_baseline_pgf}
2204     }
2205     \node[/timechart/interval-label,anchor=base-west]
2206     (\l\_timechart_interval_label_node_name)
2207     at (0,0)
2208     {
2209       \_timechart_make_ref:NN
2210       \l\_timechart_ref_text
2211       \l\_timechart_label_text
2212     };

```

```

2213     \group_end:
2214   }
2215 }

```

(End of definition for `\_timechart_interval_label_right:`)

`\_timechart_interval_label_center:` Place the label at the center of the interval.

```

2216 \cs_new:Npn\_timechart_interval_label_center:
2217 {
2218   \pgfmathsetlengthmacro{\l_timechart_label_anchor_x}
2219   {
2220     .5*max(\l_timechart_interval_start_max_x,\l_timechart_start_x)
2221     + .5*min(\l_timechart_interval_finish_min_x,\l_timechart_finish_x)
2222   }

```

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

```

2223   \group_begin:
2224   \pgftransformshift{
2225     \pgfpoint
2226     {\l_timechart_label_anchor_x}
2227     {\l_timechart_text_baseline_pgf}
2228   }
2229   \pgfonlayer{labelbg}
2230   \node[
2231     /timechart/interval~label~centered~background,
2232     anchor=base
2233   ]
2234     at (0,0)
2235     { \l_timechart_label_text };
2236   \endpgfonlayer
2237   \group_end:

```

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

```

2238   \pgfscope
2239   \pgfpathrectanglecorners{
2240     \pgfpoint
2241     {\l_timechart_interval_start_min_x}
2242     {-\l_timechart_bar_half_thickness_pgf}
2243   }{
2244     \pgfpoint
2245     {\l_timechart_interval_finish_max_x}
2246     {\l_timechart_bar_half_thickness_pgf}
2247   }
2248   \pgfusepath{clip}
2249   \group_begin:
2250   \pgftransformshift{
2251     \pgfpoint
2252     {\l_timechart_label_anchor_x}
2253     {\l_timechart_text_baseline_pgf}
2254   }
2255   \node[
2256     /timechart/interval~label~centered,
2257     anchor=base
2258   ]
2259     (\l_timechart_interval_label_node_name)

```

```

2260     at (0,0)
2261     {
2262         \__timechart_make_ref:NN
2263         \l__timechart_ref_text
2264         \l__timechart_label_text
2265     };
2266     \group_end:
2267     \endpgfscope
2268 }

```

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

## 15.17 Text

\\_\_timechart\_text\_user:Omm 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.

```

2269 \NewDocumentCommand{\__timechart_text_user:Omm}{ O{} m m }
2270 {
2271     \str_if_empty:nF{#3}{
2272         \group_begin:
2273         \pgfmathsetmacro{\l__timechart_text_x}{yeartox(#2)}

```

Process keys supplied locally and retrieve initially needed keys.

```

2274     \pgfqkeys{/timechart}{
2275         #1,
2276         tolerance~start/.get=\l__timechart_start_tolerance_pgf,
2277         tolerance~finish/.get=\l__timechart_finish_tolerance_pgf,
2278     }
2279     \__timechart_if_x_in_bounds:nT{\l__timechart_text_x}
2280     {

```

Retrieve other needed keys.

```

2281         \pgfqkeys{/timechart}{
2282             ref/.get=\l__timechart_ref_text,
2283             text~node~name/.get=\l__timechart_node_name_text,
2284             text~baseline/.get=\l__timechart_text_baseline_pgf,
2285         }

```

Shift to the correct vertical coordinate and place the text.

```

2286         \pgftransformshift{
2287             \pgfpoint{0}{\l__timechart_current_y}
2288         }
2289         \pgfinterruptboundingbox
2290         \group_begin:
2291         \pgftransformshift{
2292             \pgfpoint{\l__timechart_text_x}{\l__timechart_text_baseline_pgf}
2293         }
2294         \cs_set:Npn\l__timechart_text_{#3}
2295         \int_case:nn {\l__timechart_text_pos_int}
2296         {
2297             {0}{ \cs_set:Npn\l__timechart_node_anchor_text{base~east} }

```

```

2298         {1}{ \cs_set:Npn\l__timechart_node_anchor_text{base} }
2299         {2}{ \cs_set:Npn\l__timechart_node_anchor_text{base-west} }
2300     }
2301     \node[/timechart/text,anchor=\l__timechart_node_anchor_text]
2302     (\l__timechart_node_name_text)
2303     at (0,0)
2304     {
2305         \__timechart_make_ref:NN
2306         \l__timechart_ref_text
2307         \l__timechart_text
2308     };
2309     \group_end:
2310     \endpgfinterruptboundingbox
2311 }
2312 \group_end:
2313 }

```

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

```

2314     \__timechart_space_user:0[#1]
2315 }

```

*(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.

```

2316 \NewDocumentCommand{\__timechart_space_user:0}{ 0{} }
2317 {
2318     \group_begin:

```

Process keys supplied locally and retrieve the one needed value.

```

2319     \pgfqkeys{/timechart}{
2320         #1,
2321         interval-bar-thickness/.get=\l__timechart_bar_thickness_pgf,
2322     }
2323     \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
2324     { .5*\l__timechart_bar_thickness_pgf }

```

Shift to the correct vertical coordinate and create the space.

```

2325     \pgftransformshift{
2326         \pgfpoint{0}{\l__timechart_current_y}
2327     }
2328     \pgfpathmoveto{
2329         \pgfpoint{0}{-\l__timechart_bar_half_thickness_pgf}
2330     }
2331     \pgfpathmoveto{
2332         \pgfpoint{0}{\l__timechart_bar_half_thickness_pgf}
2333     }
2334     \pgfusepath{discard}
2335     \group_end:
2336     \bool_if:NT\l__timechart_autostep_bool{
2337         \__timechart_step_y_user:0
2338     }

```

```
2339 }

```

(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.

```
2340 \NewDocumentCommand{\timechartlegenditem}{ 0{} }
2341 {
2342   \__timechart_legend_aux:nn{#1}{
2343     \pgfmathsetlengthmacro{\l__timechart_interval_solid_start_x}{0}
2344     \pgfmathsetlengthmacro{\l__timechart_interval_solid_finish_x}
2345       {\l__timechart_legenditem_width_pgf}
2346     \__timechart_interval_solid_draw:
2347   }
2348 }
```

(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`

```
2349 \NewDocumentCommand{\timechartlegendstartrange}{ 0{} }
2350 {
2351   \__timechart_legend_aux:nn{#1}{
2352     \pgfmathsetlengthmacro
2353       {\l__timechart_interval_startrange_start_x}{0}
2354     \pgfmathsetlengthmacro{\l__timechart_interval_startrange_finish_x}
2355       {\l__timechart_legenditem_range_width_pgf}
2356     \pgfmathsetlengthmacro
2357       {\l__timechart_interval_startrange_start_level_pgf}{0}
2358     \pgfmathsetlengthmacro
2359       {\l__timechart_interval_startrange_finish_level_pgf}{1}
2360     \pgfmathsetlengthmacro{\l__timechart_interval_solid_finish_x}
2361       {\l__timechart_legenditem_width_pgf}
2362     \int_case:nn {\l__timechart_start_range_type_int}
2363       {
2364         {0}{
2365           \pgfmathsetlengthmacro{\l__timechart_interval_solid_start_x}
2366             {\l__timechart_legenditem_range_width_pgf}
2367           \__timechart_interval_startrange_draw_pseudofade:
2368         }
2369         {1}{
2370           \pgfmathsetlengthmacro{\l__timechart_interval_solid_start_x}
2371             {\l__timechart_interval_startrange_start_x}
2372           \pgfinterruptboundingbox
2373           \__timechart_interval_finish_clip_path_none:
2374           \__timechart_interval_startrange_clip_path_start:
2375           \pgfpathclose
2376           \pgfusepath{clip}
2377           \endpgfinterruptboundingbox
2378         }
2379       }
2380     \__timechart_interval_solid_draw:
2381   }
2382 }
```

```

2383 \NewDocumentCommand{\timechartlegendfinishrange}{ 0{ } }
2384 {
2385   \__timechart_legend_aux:nn{#1}{
2386     \pgfmathsetlengthmacro{\l__timechart_interval_finishrange_start_x}
2387     {
2388       \l__timechart_legenditem_width_pgf
2389       -\l__timechart_legenditem_range_width_pgf
2390     }
2391     \pgfmathsetlengthmacro{\l__timechart_interval_finishrange_finish_x}
2392     {\l__timechart_legenditem_width_pgf}
2393     \pgfmathsetlengthmacro
2394     {\l__timechart_interval_finishrange_start_level_pgf}{1}
2395     \pgfmathsetlengthmacro
2396     {\l__timechart_interval_finishrange_finish_level_pgf}{0}
2397     \pgfmathsetlengthmacro
2398     {\l__timechart_interval_solid_start_x}{0}
2399     \pgfmathsetlengthmacro{\l__timechart_interval_finish_max_x}
2400     {\l__timechart_legenditem_width_pgf}
2401     \int_case:nn {\l__timechart_finish_range_type_int}
2402     {
2403       {0}{
2404         \pgfmathsetlengthmacro{\l__timechart_interval_solid_finish_x}
2405         {
2406           \l__timechart_legenditem_width_pgf
2407           -\l__timechart_legenditem_range_width_pgf
2408         }
2409         \__timechart_interval_finishrange_draw_pseudofade:
2410       }
2411       {1}{
2412         \pgfmathsetlengthmacro{\l__timechart_interval_solid_finish_x}
2413         {\l__timechart_interval_finishrange_finish_x}
2414         \pgfinterruptboundingbox
2415         \__timechart_interval_finishrange_clip_path_finish:
2416         \__timechart_interval_start_clip_path_none:
2417         \pgfpathclose
2418         \pgfusepath{clip}
2419         \endpgfinterruptboundingbox
2420       }
2421     }
2422     \__timechart_interval_solid_draw:
2423   }
2424 }

```

(End of definition for `\timechartlegendstartrange` and `\timechartlegendfinishrange`. These functions are documented on page 15.)

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

```

2425 \cs_new:Npn\__timechart_legend_aux:nn #1#2
2426 {
2427   \__timechart_debug:n{Legend~auxiliary}
2428   \tikzpicture

```

Process supplied PGF keys and retrieve only those necessary.

```

2429   \pgfkeys{

```

```

2430     /timechart/.cd,
2431     #1,
2432     interval~bar~thickness/.get=\l__timechart_bar_thickness_pgf,
2433     interval~bar~color/.get=\l__timechart_bar_color,
2434     interval~minimum~width/.get=\l__timechart_minimum_width_pgf,
2435     beyond~length~start/.get=\l__timechart_start_beyond_length_pgf,
2436     beyond~length~finish/.get=\l__timechart_finish_beyond_length_pgf,
2437     legend~item~width/.get=\l__timechart_legenditem_width_pgf,
2438     legend~item~range~width/.get=\l__timechart_legenditem_range_width_pgf,
2439   }
2440   \pgfmathsetlengthmacro{\l__timechart_bar_half_thickness_pgf}
2441     {.5*\l__timechart_bar_thickness_pgf}
2442   \pgfmathsetmacro{\l__timechart_start_x}{0}
2443   \pgfmathsetmacro{\l__timechart_finish_x}
2444     {\l__timechart_legenditem_width_pgf}
2445   \pgfmathsetlengthmacro{\l__timechart_start_beyond_x}
2446     {\l__timechart_start_x-\l__timechart_start_beyond_length_pgf}
2447   \pgfmathsetlengthmacro{\l__timechart_finish_beyond_x}
2448     {\l__timechart_finish_x+\l__timechart_finish_beyond_length_pgf}
2449   \pgfmathsetmacro{\l__timechart_current_y}{0}

```

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

```

2450   \pgfmathsetmacro{\l__timechart_start_tolerance_x}
2451     {\l__timechart_start_x-10mm}
2452   \pgfmathsetmacro{\l__timechart_finish_tolerance_x}
2453     {\l__timechart_finish_x+10mm}
2454   \pgfscope
2455   #2
2456   \endpgfscope
2457   \endtikzpicture%
2458 }

```

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

```

2459 </package>

```

## 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.

	<b>A</b>	
<code>\a</code> .....	<i>35, 579, 581, 587, 588, 589, 590</i>	1027, 1045, 1076, 1084, 1192, 1204, 1231, 1267, 1269, 1282, 1317, 1329,
<code>\absyear</code> .....	623, 636	1358, 1394, 1396, 1409, 1469, 1471, 1533, 1535, 2100, 2108, 2127, 2336
	<b>B</b>	
<code>\begin</code> .....	<i>8</i>	<code>\bool_if:nTF</code> .....
<code>\begingroup</code> .....	<i>23, 83</i>	689, 731, 765, 1109, 1129, 1149, 1167
bool commands:		<code>\bool_new:N</code> .....
<code>\bool_if:NTF</code> .....	<i>33, 76,</i>	13, 14, 15, 210, 216, 879, 880, 881, 882, 883, 884, 967, 968, 1200, 1325
241, 343, 368, 480, 532, 625, 628,		<code>\bool_set:Nn</code> 545, 729, 750, 1226, 1353
700, 724, 751, 813, 832, 838, 842,		<code>\bool_set_eq:NN</code> .....
859, 924, 928, 942, 952, 1002, 1012,		735

<code>\bool_set_false:N</code> . . . . .	123, 537	<b>E</b>
<code>\bool_set_false:Nz</code> . . . . .	156	<code>\end</code> . . . . .
<code>\bool_set_inverse:N</code> . . . . .	730	<code>\endgroup</code> . . . . .
<code>\bool_set_true:N</code> . . . . .	211, 217, 538	<code>\endpgfinterruptboundingbox</code> . . . . .
<code>\c_false_bool</code> . . . . .	28, 514, 2158	. . . . .
<code>\l_tmpa_bool</code> . . . . .	265, 266,	. . . . .
807, 813, 823, 832, 853, 859, 912, 924		<code>\endpgfonlayer</code> . . . . .
<code>\l_tmpb_bool</code> . . . . .	836, 838, 840, 842	<code>\endpgfscope</code> . . . . .
<code>\c_true_bool</code> . . . . .	28, 381, 598, 605	1101, 1494, 1516,
		1559, 1581, 1607, 2143, 2267, 2456
		<code>\endscope</code> . . . . .
		382, 527, 573, 594, 617, 639
		<code>\endtikzpicture</code> . . . . .
		323, 2457
<b>C</b>		<b>F</b>
cs commands:		<code>\fadestep</code> 1590, 1592, 1593, 1594, 1597, 1602
<code>\cs_if_eq:NNTF</code> . . . . .	321, 1294, 1421	<code>\foreach</code> . . . . .
<code>\cs_if_exist_use:NTF</code> . . . . .	2090, 2092	359, 465, 483, 1595, 2117
<code>\cs_new:Npn</code> . . . . .	6,	<b>G</b>
20, 26, 30, 35, 41, 62, 81, 225,		group commands:
319, 325, 349, 385, 429, 516, 529,		<code>\group_begin:</code> . . . . .
562, 575, 611, 619, 642, 646, 650,		64, 305, 365,
658, 662, 666, 670, 674, 692, 704,		387, 577, 583, 621, 897, 1587, 2178,
727, 739, 743, 748, 755, 759, 763,		2199, 2223, 2249, 2272, 2290, 2318
769, 773, 777, 778, 779, 780, 781,		<code>\group_end:</code> . . . . .
782, 783, 784, 785, 786, 787, 788,		79, 347, 375, 427,
789, 790, 791, 792, 793, 794, 795,		593, 609, 640, 927, 933, 1609, 2192,
796, 797, 798, 799, 800, 801, 802,		2213, 2237, 2266, 2309, 2312, 2335
891, 936, 969, 1104, 1124, 1144,		<b>H</b>
1184, 1201, 1264, 1309, 1326, 1391,		<code>\hyperref</code> . . . . .
1436, 1454, 1474, 1496, 1518, 1539,		24
1561, 1584, 1611, 1627, 1641, 1655,		<b>I</b>
1679, 1703, 1727, 1751, 1796, 1841,		int commands:
1890, 1939, 2009, 2079, 2088, 2097,		<code>\int_case:nn</code> . . . . .
2145, 2160, 2174, 2195, 2216, 2425		388, 493,
<code>\cs_set:Npn</code> . . . . .	401, 403, 405, 418,	535, 1252, 1379, 2165, 2295, 2362, 2401
420, 422, 1001, 2294, 2297, 2298, 2299		<code>\int_if_zero:nTF</code> . . . . .
<code>\cs_set_eq:NN</code> . . . . .	10,	248
16, 17, 92, 95, 306, 307, 308, 309,		<code>\int_new:N</code> . . . . .
310, 311, 312, 314, 316, 317, 629,		212, 214, 218, 219, 220, 221
630, 632, 818, 819, 821, 1078, 1086,		<code>\int_set:Nn</code> . . . . .
1113, 1115, 1117, 1133, 1135, 1137,		133, 135, 137,
1153, 1155, 1157, 1161, 1171, 1173,		138, 151, 153, 155, 172, 174, 176,
1175, 1179, 1194, 1206, 1218, 1233,		180, 182, 185, 187, 192, 193, 194,
1246, 1255, 1271, 1276, 1284, 1289,		198, 199, 202, 203, 206, 207, 213, 215
1296, 1301, 1304, 1319, 1331, 1344,		<b>M</b>
1360, 1373, 1382, 1398, 1403, 1411,		msg commands:
1416, 1423, 1428, 1431, 2106, 2114		<code>\msg_error:nnnn</code> . . . . .
		932
		<code>\msg_new:nnn</code> . . . . .
		885, 2095
		<code>\msg_warning:nnn</code> . . . . .
		2141
<b>D</b>		<b>N</b>
<code>\denominator</code> . . . . .	1594, 1597	<code>\n</code> . . . . .
dim commands:		285
<code>\dim_compare:nNnTF</code> . . . . .	332	<code>\NeedsTeXFormat</code> . . . . .
<code>\dim_new:N</code> . . . . .	18, 19	3
<code>\l_tmpa_dim</code> . . . . .	21, 16	<code>\NewDocumentCommand</code> . . . . .
<code>\l_tmpb_dim</code> . . . . .	21, 17	678, 887, 2269, 2316, 2340, 2349, 2383
<code>\DocumentMetadata</code> . . . . .	9	<code>\NewDocumentEnvironment</code> . . . . .
		222
		<code>\nobreakspace</code> . . . . .
		644, 648
		<code>\node</code> . . . . .
		2184, 2205, 2230, 2255, 2301
		<b>O</b>
		<code>\o</code> . . . . .
		1596, 1605

P

<code>\pgfcoordinate</code> .....	45, 48	654, 1644, 1682, 1730, 1806, 1851, 1900, 1956, 2026, 2129, 2328, 2331
<code>\pgfdeclarelayer</code> .....	238, 239	<code>\pgfpathrectanglecorners</code> . 1442, 1481, 1503, 1546, 1568, 1599, 1613, 2239
<code>\pgfextractx</code> .....	22, 52, 55	<code>\pgfpoint</code> .....
<code>\pgfextracty</code> .....	22, 327, 329, 497, 504	58, 59, 370, 371, 372, 378, 379, 395, 412, 511, 512, 519, 521, 565, 567, 578, 586, 591, 614, 615, 622, 653, 655, 1093, 1443, 1447, 1482, 1486, 1504, 1508, 1547, 1551, 1569, 1573, 1600, 1602, 1614, 1618, 1631, 1636, 1645, 1650, 1659, 1664, 1674, 1683, 1688, 1698, 1707, 1712, 1717, 1722, 1731, 1736, 1741, 1746, 1762, 1767, 1778, 1782, 1786, 1791, 1807, 1812, 1823, 1827, 1831, 1836, 1853, 1857, 1866, 1871, 1876, 1882, 1886, 1902, 1906, 1915, 1920, 1925, 1931, 1935, 1958, 1962, 1971, 1976, 1981, 1992, 1996, 2001, 2005, 2028, 2032, 2041, 2046, 2051, 2062, 2066, 2071, 2075, 2130, 2135, 2149, 2153, 2180, 2201, 2225, 2240, 2244, 2251, 2287, 2292, 2326, 2329, 2332
<code>\pgfinterruptboundingbox</code> .....	43, 2082, 2164, 2289, 2372, 2414	<code>\pgfpointadd</code> .....
<code>\pgfkeys</code> .....	89, 227, 283, 431, 2429	1777, 1822, 1852, 1881, 1901, 1930, 1957, 1991, 2000, 2027, 2061, 2070
<code>\pgfkeyssetvalue</code> 112, 113, 116, 117, 120, 121		<code>\pgfpointanchor</code> .....
<code>\pgfkeysvalueof</code> .....	397, 414, 2157	47, 50, 54, 57, 328, 330, 498, 505
<code>\pgflinewidth</code> .....	85	<code>\pgfpointorigin</code> .....
<code>\pgfmathsetlengthmacro</code> .....	85, 247, 250, 251, 252, 580, 581, 999, 1596, 2218, 2323, 2343, 2344, 2352, 2354, 2356, 2358, 2360, 2365, 2370, 2386, 2391, 2393, 2395, 2397, 2399, 2404, 2412, 2440, 2445, 2447	585
<code>\pgfmathsetmacro</code> 22, 28, 267, 269, 271, 288, 290, 292, 293, 294, 295, 296, 299, 302, 334, 335, 338, 340, 351, 354, 400, 417, 435, 443, 452, 457, 471, 489, 499, 500, 506, 507, 579, 660, 664, 668, 672, 676, 680, 690, 736, 815, 834, 844, 848, 861, 868, 944, 949, 954, 959, 987, 990, 993, 996, 1004, 1009, 1014, 1019, 1029, 1036, 1047, 1049, 1051, 1053, 1060, 1208, 1221, 1236, 1249, 1334, 1347, 1363, 1376, 1583, 1588, 1589, 1590, 1592, 1593, 1594, 1625, 1626, 1754, 1758, 1799, 1803, 1844, 1848, 1893, 1897, 1942, 1946, 1949, 1953, 2012, 2016, 2019, 2023, 2102, 2110, 2119, 2273, 2442, 2443, 2449, 2450, 2452		<code>\pgfresetboundingbox</code> .....
<code>\pgfmathsmuggle</code> .....	23, 86	331
<code>\pgfmathtruncatemacro</code> . 623, 804, 805, 806		<code>\pgfscope</code> .....
<code>\pgfnode</code> .....	22, 44, 77, 78	1094, 1480, 1502, 1545, 1567, 1598, 2099, 2238, 2454
<code>\pgfonlayer</code> .....	357, 2229	<code>\pgfset</code> .....
<code>\pgfpatharc</code> .....	587, 588, 589, 590, 1668, 1692, 1771, 1816, 1860, 1909, 1965, 1985, 2035, 2055	70
<code>\pgfpathclose</code> .....	2084, 2375, 2417	<code>\pgfsetfillcolor</code> .....
<code>\pgfpathlineto</code> .....	372, 520, 566, 586, 591, 615, 1630, 1635, 1649, 1658, 1663, 1673, 1687, 1697, 1706, 1711, 1716, 1721, 1735, 1740, 1745, 1761, 1766, 1776, 1785, 1790, 1811, 1821, 1830, 1835, 1865, 1870, 1875, 1880, 1914, 1919, 1924, 1929, 1970, 1975, 1980, 1990, 1999, 2040, 2045, 2050, 2060, 2069, 2134	1451, 1490, 1512, 1555, 1577, 1604, 1622
<code>\pgfpathmoveto</code> .....	58, 59, 371, 519, 565, 585, 614, 652,	<code>\pgfsetfillopacity</code> .....
		1491, 1513, 1556, 1578, 1605
		<code>\pgfsetlayers</code> .....
		240
		<code>\pgfsetstrokecolor</code> .....
		2116
		<code>\pgftransformshift</code> .....
		69, 370, 394, 411, 578, 622, 1093, 2179, 2200, 2224, 2250, 2286, 2291, 2325
		<code>\pgftransformyscale</code> .....
		584
		<code>\pgfusepath</code> .....
		60, 77, 78, 373, 526, 572, 592, 616, 656, 1452, 1493, 1515, 1558, 1580, 1606, 1623, 2085, 2139, 2248, 2334, 2376, 2418
		prg commands:
		<code>\prg_do_nothing</code> : .....
		391, 1114, 1116, 1118, 1134, 1136, 1138, 1156, 1158, 1174, 1176, 1194, 1294, 1306, 1319, 1421, 1433
		<code>\ProvidesExplPackage</code> .....
		4

<b>Q</b>	
quark commands:	
\q_mark	40–42, 732, 743, 752, 759, 766, 767, 773, 775, 802
\q_stop	40–42, 729, 732, 739, 743, 750, 752, 755, 759, 765, 766, 767, 769, 773, 775, 802
<b>R</b>	
\r	35, 580, 581, 587, 588, 589, 590
\relax	38, 39
\RequirePackage	11
<b>S</b>	
\scope	358, 518, 564, 582, 613, 624
\sffamily	6
\small	6
\start	1588, 1591, 1592, 1594, 1595, 1597
\startnext	1592, 1595
\stop	1589, 1591, 1593, 1594, 1597
\stopprev	1593, 1595
str commands:	
\str_if_empty:NTF	22, 2162
\str_if_empty:nTF	2271
<b>T</b>	
\t	581, 587, 590
$\TeX$ and $\LaTeX$ commands:	
\pgf@process	37
\pgf@x	38
\pgf@y	39
\textsc	644, 648
\textwidth	8, 101
\tikzpicture	237, 2428
\tikzset	84
\timechart	13, 15, 29
timechart	8, 222
timechart internal commands:	
\l__timechart_auto_reset_-	
maximum_y	295, 676, 686
\l__timechart_auto_reset_-	
minimum_y	294, 672, 685
\l__timechart_autostep_bool	156, 216, 928, 2336
\l__timechart_axis_bottom_y	500, 507, 511
\__timechart_axis_draw:	346, 385, 385
\__timechart_axis_draw_	34
\__timechart_axis_draw_aux:	407, 424, 429, 429
\__timechart_axis_draw_labelled_-	
major_tick:NN	490, 529, 529
\__timechart_axis_draw_line	611
\__timechart_axis_draw_major_-	
tick:N	549, 562, 562
\__timechart_axis_draw_minor_-	
tick:N	481, 516, 516
\__timechart_axis_draw_year_-	
label:nmnnnn	550, 595, 602, 619, 619
\__timechart_axis_draw_zero_-	
tick:N	533, 575, 575
\l__timechart_axis_int	133, 135, 137, 138, 212, 248, 388, 493
\l__timechart_axis_line_width	250, 256, 277
\l__timechart_axis_top_y	499, 506, 512
\l__timechart_axis_y	32
\l__timechart_bar_color	981, 1451, 1466, 1490, 1512, 1530, 1555, 1577, 1622, 2433
\l__timechart_bar_half_thickness_-	
pgf	999, 1445, 1449, 1484, 1488, 1506, 1510, 1549, 1553, 1571, 1575, 1600, 1602, 1616, 1620, 1666, 1672, 1690, 1696, 1714, 1719, 1738, 1743, 1769, 1780, 1788, 1814, 1825, 1833, 1855, 1878, 1884, 1904, 1927, 1933, 1960, 1983, 1994, 2003, 2030, 2053, 2064, 2073, 2132, 2137, 2151, 2155, 2242, 2246, 2323, 2329, 2332, 2440
\l__timechart_bar_thickness_pgf	982, 1000, 1633, 1638, 1647, 1652, 1661, 1676, 1685, 1700, 1709, 1724, 1733, 1748, 1755, 1764, 1793, 1800, 1809, 1838, 1845, 1868, 1873, 1894, 1917, 1922, 1943, 1950, 1973, 1978, 2013, 2020, 2043, 2048, 2321, 2324, 2432, 2441
\l__timechart_circa_uncertainty_-	
year	940, 946, 956, 1006, 1016
\l__timechart_content_bottom_y	30, 334, 338, 351, 352, 372, 378, 413, 653
\l__timechart_content_top_y	30, 335, 340, 354, 355, 371, 379, 396, 655
\l__timechart_cumulative_days_-	
pgf	821, 848, 849, 864, 872
\l__timechart_current_y	292, 660, 664, 668, 680, 681, 685, 686, 690, 1093, 2287, 2326, 2449
\__timechart_debug:n	6, 10, 92, 95, 893, 896, 1068, 1069, 1106, 1126, 1146, 1186, 1203, 1266, 1311, 1328, 1393, 1438, 1456, 1457, 1476, 1477, 1498, 1499, 1520, 1521, 1541, 1542, 1563, 1564, 1586, 1629, 1643, 1657, 1681, 1705, 1729, 1753, 1798, 1843, 1892, 1941, 2011, 2081, 2427
\__timechart_debug_real:n	6, 6, 92

<code>\__timechart_draw_axis_line</code>	492, 611	<code>\__timechart_if_x_range_intersect_-</code>	
<code>__timechart_draw_visible_-</code>		<code>bounds:nnTF</code>	704, 962
<code>interval:nn</code>	969	<code>\__timechart_if_x_range_intersect_-</code>	
<code>\__timechart_draw_visible_-</code>		<code>bounds_x:nnTF</code>	704
<code>interval:nn</code>	965, 969	<code>\__timechart_interval:nnnn</code>	45, 889, 891, 891
<code>\l_timechart_end_minus_year</code>	302, 363	<code>\__timechart_interval_both_clip_-</code>	
<code>\l_timechart_fadestep_pgf</code>	1583, 1591	<code>path:</code>	68–72, 1272, 1285, 1399, 1412, 2083, 2088, 2088
<code>\__timechart_finish_beyond_bool</code>	967	<code>\__timechart_interval_checked:nn</code>	926, 936, 936
<code>\l__timechart_finish_beyond_-</code>		<code>\__timechart_interval_clip:</code>	1095, 2079, 2079
<code>length_pgf</code>	975, 997, 2436, 2448	<code>\__timechart_interval_define_-</code>	
<code>\l__timechart_finish_beyond_x</code>	996, 1172, 1430, 1500, 1509, 1543, 1548, 1619, 1684, 1689, 1699, 1808, 1813, 1854, 1867, 1959, 1972, 2047, 2052, 2447	<code>bar_node:</code>	1100, 2145, 2145
<code>\l__timechart_finish_beyond_x_-</code>		<code>\__timechart_interval_definite_-</code>	
<code>radius_pgf</code>	977, 1695, 1819, 1968, 2058	<code>compute:</code>	1090, 1104, 1104
<code>\l__timechart_finish_is_range_-</code>		<code>\__timechart_interval_definite_-</code>	
<code>bool</code>	879, 906	<code>compute_ii:</code>	51, 1121, 1124
<code>\__timechart_finish_major_tick_-</code>		<code>\__timechart_interval_definite_-</code>	
<code>year</code>	34	<code>compute_iii:</code>	1141, 1144
<code>\l__timechart_finish_major_tick_-</code>		<code>\__timechart_interval_draw_-</code>	
<code>year</code>	457, 487, 544	<code>pseudofade:nnnnn</code>	1463, 1527, 1583, 1584
<code>\l__timechart_finish_min_x</code>	48	<code>\l__timechart_interval_finish_-</code>	
<code>\l__timechart_finish_range_type_-</code>		<code>beyond_bool</code>	968, 1082, 1084, 1130, 1168, 1229, 1329, 1394, 1533, 2108
<code>int</code>	185, 187, 220, 1379, 2401	<code>\__timechart_interval_finish_-</code>	
<code>\l__timechart_finish_tolerance_-</code>		<code>clip_path:</code>	73, 1087, 1404, 1417, 2090
<code>pgf</code>	973, 991, 2277	<code>\__timechart_interval_finish_-</code>	
<code>\l__timechart_finish_tolerance_x</code>	990, 1083, 2452	<code>clip_path_beyond:</code>	1088, 1679, 1679
<code>\l__timechart_finish_x</code>	290, 379, 512, 615, 655, 697, 711, 715, 720, 991, 997, 1128, 1166, 1225, 1235, 1333, 1500, 1505, 1543, 1552, 1824, 1883, 2002, 2063, 2111, 2221, 2443, 2448, 2453	<code>\__timechart_interval_finish_-</code>	
<code>\l__timechart_finish_year</code>	266, 269, 270, 281, 291, 303, 459, 461, 469	<code>clip_path_none:</code>	1641, 1641, 2091, 2373
<code>\l__timechart_grid_bool</code>	123, 210, 241, 343	<code>\l__timechart_interval_finish_-</code>	
<code>\l__timechart_grid_bottom_ysep_-</code>		<code>max_circa_bool</code>	879, 909, 952
<code>pgf</code>	233, 352	<code>\l__timechart_interval_finish_-</code>	
<code>\__timechart_grid_draw:</code>	344, 349, 349	<code>max_x</code>	46, 50, 954, 959, 964, 1041, 1060, 1063, 1072, 1083, 1314, 1337, 1340, 1346, 1366, 1369, 1921, 1926, 2154, 2197, 2202, 2245, 2399
<code>\l__timechart_grid_line_width</code>	244, 247, 276	<code>\l__timechart_interval_finish_-</code>	
<code>\l__timechart_grid_top_ysep_pgf</code>	232, 355	<code>max_year</code>	46, 910, 920, 955, 960
<code>\__timechart_hsmash_pgfnode:nnnnn</code>	41, 41, 633	<code>\l__timechart_interval_finish_-</code>	
<code>\__timechart_if_equal:nnTF</code>	30, 30	<code>min_circa_bool</code>	879, 907, 1012
<code>\__timechart_if_x_in_bounds:nTF</code>	692, 692, 2176, 2197, 2279	<code>\l__timechart_interval_finish_-</code>	
		<code>min_x</code>	48, 50, 1014, 1019, 1025, 1033, 1036, 1051, 1052, 1064, 1072, 1108, 1166, 1181, 1315, 1339, 1352, 1368, 1375, 1832, 1837, 2111, 2114, 2221
		<code>\l__timechart_interval_finish_-</code>	
		<code>min_year</code>	46, 908, 918, 919, 1015, 1020

```

\__timechart_interval_finishrange_
clip_path_beyond_both: . . . . .
. . . . . 1400, 2009, 2009
\__timechart_interval_finishrange_
clip_path_beyond_finish: . . . . .
. . . . . 1405, 1796, 1796
\__timechart_interval_finishrange_
clip_path_beyond_start: . . . . .
. . . . . 1413, 1890, 1890
\__timechart_interval_finishrange_
clip_path_finish: . . . . .
. . . . . 1418, 1727, 1727, 2415
\__timechart_interval_finishrange_
compute: 1092, 1136, 1174, 1309, 1309
\__timechart_interval_finishrange_
compute_ii: . . . . . 1322, 1325, 1326
\__timechart_interval_finishrange_
draw: . . . . . 1097,
1138, 1176, 1319, 1383, 1424, 1432
\__timechart_interval_finishrange_
draw_pseudofade: . . . . .
. . . . . 1384, 1518, 1518, 2409
\__timechart_interval_finishrange_
finish_beyond_draw_fade: . . . . .
. . . . . 1534, 1538, 1539
\l__timechart_interval_finishrange_
finish_level_pgf . 1334, 1348,
1525, 1532, 1557, 1801, 2014, 2396
\l__timechart_interval_finishrange_
finish_x . . . . 1332, 1336, 1345,
1524, 1529, 1732, 1737, 2391, 2413
\__timechart_interval_finishrange_
slant_compute: . . 1387, 1391, 1391
\l__timechart_interval_finishrange_
start_beyond_bool . . . . .
. . 1325, 1353, 1358, 1396, 1409, 1535
\__timechart_interval_finishrange_
start_beyond_draw_fade: . . . . .
. . . . . 1536, 1561, 1561
\l__timechart_interval_finishrange_
start_level_pgf . . 1363, 1377,
1523, 1531, 1579, 1895, 2021, 2394
\l__timechart_interval_finishrange_
start_x . . . . . 1361, 1365,
1374, 1522, 1528, 1742, 1747, 2386
\__timechart_interval_label: . . . . .
. . . . . 1102, 2160, 2160
\__timechart_interval_label_
center: . . . . . 2168, 2216, 2216
\__timechart_interval_label_
left: . . . . . 2167, 2174, 2174
\l__timechart_interval_label_
node_name . . . 985, 2185, 2206, 2259
\__timechart_interval_label_
right: . . . . . 2169, 2195, 2195
\__timechart_interval_mark: . . . . .
. . . . . 1099, 2097, 2097
\l__timechart_interval_mark_max_
x . . . . . 2110, 2114, 2124
\l__timechart_interval_mark_min_
x . . . . . 2102, 2106, 2123
\__timechart_interval_solid_
draw: . . . 1098, 1114, 1134, 1294,
1421, 1436, 1436, 2346, 2380, 2422
\l__timechart_interval_solid_
finish_x . . . . 1172, 1180, 1429,
1440, 1448, 2344, 2360, 2404, 2412
\l__timechart_interval_solid_
start_x . . . . 1154, 1162, 1302,
1439, 1444, 2343, 2365, 2370, 2398
\l__timechart_interval_start_
beyond_bool . . . . .
. . . . . 967, 1074, 1076, 1110,
1150, 1204, 1267, 1356, 1469, 2100
\__timechart_interval_start_
clip_path: 73, 1079, 1277, 1290, 2092
\__timechart_interval_start_
clip_path_beyond: 1080, 1655, 1655
\__timechart_interval_start_
clip_path_none: . . . . .
. . . . . 1627, 1627, 2093, 2416
\l__timechart_interval_start_
max_circa_bool . . . . 879, 902, 1002
\l__timechart_interval_start_
max_x . . . . . 48,
50, 1004, 1009, 1024, 1029, 1032,
1037, 1049, 1050, 1057, 1070, 1128,
1148, 1163, 1190, 1213, 1225, 1241,
1248, 1787, 1792, 2103, 2106, 2220
\l__timechart_interval_start_
max_year 46, 903, 915, 917, 1005, 1010
\l__timechart_interval_start_
min_circa_bool . . . . 879, 900, 942
\l__timechart_interval_start_
min_x . . . . . 46, 50, 944, 949,
963, 1042, 1053, 1056, 1070, 1075,
1189, 1211, 1214, 1220, 1239, 1242,
1872, 1877, 2150, 2176, 2181, 2241
\l__timechart_interval_start_
min_year . . . . 46, 901, 914, 945, 950
\__timechart_interval_startfinishrange_
draw_slant: . 1298, 1425, 1611, 1611
\__timechart_interval_startrange_
clip_path_beyond_both: . . . . .
. . . . . 1273, 1939, 1939
\__timechart_interval_startrange_
clip_path_beyond_finish: . . . . .

```

..... 1286, 1841, 1841  
 \\_timechart\_interval\_startrange\_  
 clip\_path\_beyond\_start: ....  
 ..... 1278, 1751, 1751  
 \\_timechart\_interval\_startrange\_  
 clip\_path\_start: .....  
 ..... 1291, 1703, 1703, 2374  
 \\_timechart\_interval\_startrange\_  
 compute: 1091, 1116, 1156, 1184, 1184  
 \\_timechart\_interval\_startrange\_  
 compute\_ii: ..... 1197, 1200, 1201  
 \\_timechart\_interval\_startrange\_  
 draw: ..... 1096,  
 1118, 1158, 1194, 1256, 1297, 1305  
 \\_timechart\_interval\_startrange\_  
 draw\_pseudofade: .....  
 ..... 1257, 1454, 1454, 2367  
 \l\_timechart\_interval\_startrange\_  
 finish\_beyond\_bool .....  
 .. 1200, 1226, 1231, 1269, 1282, 1471  
 \\_timechart\_interval\_startrange\_  
 finish\_beyond\_draw\_fade: ....  
 ..... 1472, 1496, 1496  
 \l\_timechart\_interval\_startrange\_  
 finish\_level\_pgf . 1236, 1250,  
 1461, 1468, 1514, 1846, 1951, 2359  
 \l\_timechart\_interval\_startrange\_  
 finish\_x ..... 1234, 1238,  
 1247, 1460, 1465, 1718, 1723, 2354  
 \\_timechart\_interval\_startrange\_  
 slant\_compute: .. 1260, 1264, 1264  
 \\_timechart\_interval\_startrange\_  
 start\_beyond\_draw\_fade: ....  
 ..... 1470, 1474, 1474  
 \l\_timechart\_interval\_startrange\_  
 start\_level\_pgf .. 1208, 1222,  
 1459, 1467, 1492, 1756, 1944, 2357  
 \l\_timechart\_interval\_startrange\_  
 start\_x ..... 1207, 1210, 1219,  
 1458, 1464, 1708, 1713, 2353, 2371  
 \\_timechart\_interval\_user:Ommm .  
 ..... 306, 887, 887  
 \\_timechart\_interval\_user\_  
 aux:nnnn ..... 46  
 \\_timechart\_is\_before\_p:w .....  
 ..... 41, 765, 769, 769  
 \\_timechart\_is\_circa\_p:w .....  
 ..... 41, 750, 755, 755  
 \\_timechart\_is\_nondaterange\_p:w  
 ..... 40, 729, 739, 739  
 \l\_timechart\_label\_anchor\_x ...  
 ..... 2218, 2226, 2252  
 \l\_timechart\_label\_pos\_int ....  
 172, 174, 176, 198, 202, 206, 218, 2165  
 \l\_timechart\_label\_text .....  
 .. 1001, 2162, 2190, 2211, 2235, 2264  
 \c\_timechart\_left\_far\_x .....  
 ..... 1625, 1632, 1637  
 \\_timechart\_legend\_aux:nn .....  
 ..... 2342, 2351, 2385, 2425, 2425  
 \l\_timechart\_legenditem\_range\_  
 width\_pgf .....  
 ..... 2355, 2366, 2389, 2407, 2438  
 \l\_timechart\_legenditem\_width\_  
 pgf ..... 2345, 2361,  
 2388, 2392, 2400, 2406, 2437, 2444  
 \\_timechart\_main\_begin:nnn ....  
 ..... 29, 30, 223, 225, 225  
 \\_timechart\_main\_end: . 224, 319, 319  
 \\_timechart\_main\_end\_user: ....  
 ..... 29, 317, 321, 322, 325, 325  
 \l\_timechart\_major\_tick\_eras\_  
 int ..... 151, 153, 155, 214, 535  
 \l\_timechart\_major\_tick\_  
 interval\_year .....  
 .... 235, 367, 440, 448, 455, 462, 477  
 \\_timechart\_major\_tick\_length\_  
 pgf ..... 433, 558, 569, 586, 591  
 \l\_timechart\_major\_tick\_line\_  
 width ..... 251, 259, 278  
 \\_timechart\_make\_rectangle\_  
 node:nnnn ... 62, 62, 377, 510, 2147  
 \\_timechart\_make\_ref:NN .....  
 ..... 20, 20, 2188, 2209, 2262, 2305  
 \\_timechart\_make\_year:n .....  
 ..... 629, 630, 632, 636  
 \l\_timechart\_mark\_color .. 983, 2116  
 \l\_timechart\_mark\_text ... 979, 2117  
 \l\_timechart\_mark\_x .....  
 ..... 2119, 2123, 2124, 2131, 2136  
 \l\_timechart\_minimum\_width\_pgf .  
 ..... 980, 1043, 1048, 2434  
 \l\_timechart\_minor\_tick\_  
 interval\_year .. 234, 297, 300, 303  
 \\_timechart\_minor\_tick\_length\_  
 pgf ..... 432, 523  
 \l\_timechart\_minor\_tick\_line\_  
 width ..... 252, 262, 279  
 \l\_timechart\_month\_days\_pgf ...  
 ..... 819, 844, 845, 856  
 \l\_timechart\_node\_anchor\_text ..  
 ..... 2297, 2298, 2299, 2301  
 \l\_timechart\_node\_name\_text ...  
 ..... 2283, 2302  
 \\_timechart\_nogrid\_bounding\_  
 box\_set: ..... 345, 650, 650  
 \\_timechart\_parse\_before\_date:w  
 ..... 42, 766, 773, 773

`\__timechart_parse_circa_date:w` . . . . . 41, 752, 759, 759  
`\__timechart_parse_date:NNn` . . . . . 265, 266, 734, 745, 746, 748, 748  
`\__timechart_parse_date_or_date:NNNNn` 727, 727, 898, 905  
`\__timechart_parse_noncirca_date:Nn` . . . . . 753, 761, 763  
`\__timechart_parse_noncirca_date:w` . . . . . 41, 763  
`\__timechart_parse_positive_date:w` . . . . . 42  
`\__timechart_parse_range:w` . . . . . 40, 732, 743, 743  
`\__timechart_parse_signed_date:w` . . . . . 767, 775, 802, 802  
`\__timechart_parsed_day_pgf` . . . . . 806, 855, 856, 873  
`\__timechart_parsed_month` . . . . . 837, 841  
`\__timechart_parsed_month_pgf` . . . . . 805, 809, 810, 820, 822  
`\__timechart_parsed_year_pgf` . . . . . 804, 825, 827, 828  
`\__timechart_pgfextractxy:nnn` . . . . . 35, 35, 65, 67  
`\__timechart_pgfmathsetbool:nn` . . . . . 26  
`\__timechart_pgfmathsetbool:nn` . . . . . 26, 32, 366, 472, 531, 541, 543, 627, 682, 694, 706, 807, 823, 836, 840, 853, 912, 1022, 1039, 1074, 1082, 1107, 1127, 1147, 1165, 1187, 1224, 1312, 1351, 2120  
`\l__timechart_ref_text` . . . . . 978, 2189, 2210, 2263, 2282, 2306  
`\__timechart_reset_y_user:` . . . . . 311, 666, 666  
`\c__timechart_right_far_x` . . . . . 1625, 1646, 1651  
`\__timechart_save_y_user:` . . . . . 310, 662, 662  
`\l__timechart_saved_y` . . . . . 293, 664, 668, 690  
`\__timechart_set_style_line_width:nn` . 81, 81, 243, 255, 258, 261  
`\__timechart_set_y_maximum_auto_reset_user:m` . . . . . 315, 674, 674  
`\__timechart_set_y_minimum_auto_reset_user:m` . . . . . 313, 670, 670  
`\__timechart_set_y_user:m` . . . . . 309, 658, 658  
`\__timechart_space_user:0` . . . . . 308, 2314, 2316, 2316  
`\__timechart_start_beyond_bool` . 967  
`\l__timechart_start_beyond_length_pgf` . . . . . 974, 994, 2435, 2446  
`\l__timechart_start_beyond_x` . . . . . 993, 1154, 1303, 1478, 1483, 1565, 1574, 1615, 1660, 1665, 1675, 1763, 1768, 1903, 1916, 1977, 1982, 2029, 2042, 2445  
`\l__timechart_start_beyond_x_radius_pgf` . . . . . 976, 1671, 1774, 1863, 1912, 1988, 2038  
`\l__timechart_start_is_range_bool` . . . . . 879, 899  
`\__timechart_start_major_tick_year` . . . . . 34  
`\l__timechart_start_major_tick_year` . . . . . 435, 443, 446, 447, 449, 454, 476, 484, 542  
`\l__timechart_start_max_x` . . . . . 48  
`\l__timechart_start_plus_major_tick_year` . . . . . 452, 485  
`\l__timechart_start_plus_year` . . . . . 296, 360, 467  
`\l__timechart_start_plusplus_year` . . . . . 299, 361  
`\l__timechart_start_range_type_int` . . . . . 180, 182, 220, 1252, 2362  
`\l__timechart_start_tolerance_pgf` . . . . . 972, 988, 2276  
`\l__timechart_start_tolerance_x` . . . . . 987, 1075, 2450  
`\l__timechart_start_x` . . . . . 288, 378, 511, 614, 653, 696, 710, 714, 719, 988, 994, 1108, 1148, 1207, 1352, 1362, 1478, 1487, 1565, 1570, 1779, 1932, 1993, 2072, 2103, 2220, 2442, 2446, 2451  
`\l__timechart_start_year` . . . . . 265, 267, 268, 281, 285, 289, 297, 300, 437, 439, 446, 466  
`\__timechart_step_y_user:` . . . . . 38  
`\__timechart_step_y_user:0` . . . . . 316, 678, 678, 929, 2337  
`\l__timechart_text` . . . . . 2294, 2307  
`\l__timechart_text_baseline_pgf` . . . . . 984, 2182, 2203, 2227, 2253, 2284, 2292  
`\l__timechart_text_pos_int` . . . . . 192, 193, 194, 199, 203, 207, 219, 2295  
`\__timechart_text_user:Omm` . . . . . 307, 2269, 2269  
`\l__timechart_text_x` 2273, 2279, 2292  
`\l__timechart_tick_label_anchor_text` . . . . . 401, 418, 554  
`\l__timechart_tick_orientation_pgf` . . . . . 400,

417, 522, 557, 568, 584, 601, 608	<code>\l__timechart_zero_tick_before_</code>	<code>label_anchor_text</code> .. 403, 420, 599
<code>\l__timechart_tmpa_bool</code> .....	<code>\timechartfinish</code> .... 13, 15, 29, 317, 321	<code>\timechartinterval</code> .....
..... 12, 32, 33, 366, 368,	..... 5, 7, 8, 10, 12, 15, 45, 306	<code>\timechartlegendfinishrange</code> ... 15, 2349
473, 480, 531, 532, 627, 628, 682,	<code>\timechartlegenditem</code> .....	14, 15, 2340
689, 694, 700, 706, 724, 1022, 1027,	<code>\timechartlegendstartrange</code> ....	15, 2349
1039, 1045, 1107, 1110, 1127, 1130,	<code>\timechartlength</code> .....	14
1147, 1150, 1187, 1192, 1224, 1228,	<code>\timechartmakeafteryear</code> ...	14, 630, 642
1312, 1317, 1351, 1355, 2120, 2127	<code>\timechartmakebeforeyear</code> ...	14, 629, 642
<code>\l__timechart_tmpa_dim</code> . 16, 53, 58,	<code>\timechartresety</code> .....	12, 13, 311
66, 71, 327, 332, 339, 497, 499, 504, 507	<code>\timechartsavey</code> .....	12, 13, 38, 310
<code>\l__timechart_tmpa_pgf</code> .....	<code>\timechartsety</code> .....	12, 37, 309
1754, 1759, 1782, 1799, 1804, 1827,	<code>\timechartsetymaximumautoreset</code> ....	
1942, 1947, 1996, 2012, 2017, 2066	..... 12, 13, 314	<code>\timechartsetyminimumautoreset</code> ....
<code>\l__timechart_tmpb_bool</code> .....	..... 12, 13, 312	<code>\timechartspace</code> .....
..... 12, 541, 546, 1165, 1168	<code>\timechartsresety</code> .....	7, 8, 12, 78, 308
<code>\l__timechart_tmpb_dim</code> .....	<code>\timechartsresety</code> .....	38
..... 16, 56, 59, 66, 72, 329, 332, 341	<code>\timechartsresetymaximumautoreset</code> ....	38
<code>\l__timechart_tmpb_pgf</code> 1758, 1775,	<code>\timechartsresetyminimumautoreset</code> ....	38
1803, 1820, 1946, 1989, 2016, 2059	<code>\timechartstepy</code> .....	13, 38, 316
<code>\l__timechart_tmpc_bool</code> . 12, 543, 546	<code>\timecharttext</code> .....	7, 8, 12, 15, 77, 307
<code>\l__timechart_tmpc_dim</code> .... 16, 68, 71	tl commands:	
<code>\l__timechart_tmpc_pgf</code> .....	<code>\tl_if_empty_p:n</code> .....	741, 757, 771
..... 1844, 1849, 1857, 1886,	<code>\typeout</code> .....	8
1893, 1898, 1906, 1935, 1949, 1954,		
1962, 2005, 2019, 2024, 2032, 2075		
<code>\l__timechart_tmpd_bool</code> .....		
..... 12, 537, 538, 545, 553		
<code>\l__timechart_tmpd_dim</code> .... 16, 68, 72		
<code>\l__timechart_tmpd_pgf</code> 1848, 1864,		
1897, 1913, 1953, 1969, 2023, 2039		
<code>\l__timechart_width_adjust</code> .....		
..... 1047, 1050, 1052		
<code>\l__timechart_width_pgf</code> .... 230, 274		
<code>\l__timechart_x</code> .....		
28, 271, 285		
<code>\c__timechart_year_days_pgf</code> ... 818		
<code>\l__timechart_year_days_pgf</code> .....		
..... 818, 834, 835, 864, 874		
<code>\l__timechart_ystep_pgf</code> .... 231, 681		
<code>\l__timechart_zero_tick_after_</code>		
<code>label_anchor_text</code> .. 405, 422, 606		

	<b>U</b>
use commands:	
<code>\use:n</code> .....	632
<code>\use_none:n</code> .....	10, 95
<code>\usepackage</code> .....	3
	<b>X</b>
<code>\x</code> .....	471, 481, 489, 490
<code>\xa</code> .....	1595, 1597, 1600, 1602
	<b>Y</b>
<code>\year</code> .....	359, 367, 370, 465, 471,
	476, 483, 489, 490, 2117, 2119, 2141