

The `sesstime` Package

Niklas Beisert

Institut für Theoretische Physik
Eidgenössische Technische Hochschule Zürich
Wolfgang-Pauli-Strasse 27, 8093 Zürich, Switzerland

nbeisert@itp.phys.ethz.ch

17 January 2018, v1.11

Abstract

`sesstime` is a $\text{\LaTeX 2}\epsilon$ package to add timing marks to lecture notes in order to help managing the time available for presenting a given section of the document. It also provides tools to record and estimate the progress throughout the course.

Contents

1	Introduction	2
2	Usage	2
2.1	Blocks	2
2.2	Chapters	4
2.3	Sessions	4
2.4	Comments	5
2.5	Flags	6
2.6	Printing	6
2.7	Customisation	6
3	Command Reference	7
3.1	Blocks	7
3.2	Chapters	7
3.3	Sessions	8
3.4	Comments	8
3.5	Flags	8
3.6	Package Options	8
4	Information	9
4.1	Copyright	9
4.2	Files and Installation	10
4.3	Interaction with CTAN Packages	10
4.4	Feature Suggestions	10
4.5	Revision History	10
A	Sample File	11
B	Implementation	15

1 Introduction

A crucial step in designing a lecture course is to estimate the time needed for presenting the individual topics and to arrange them such that the overall teaching goals can be achieved within a predetermined amount of time. This step requires a lot of experience and the information is typically lost when (parts of) the lecture notes are recycled by different lecturer. While it is very useful to have timing information available for a number of purposes, it also takes patience and dedication to set it up and to keep it up to date when the presented material is changed (added, removed or shuffled).

This package offers an interface to store such timing information in the L^AT_EX source. It also offers some tools to process the raw data and display them in compiled form. The goals and features of this package include the following:

- Store timing information in the L^AT_EX source (as absolute or relative times).
- Show small timing marks in the margin of the text.
- Indicate where sessions begin and end in the L^AT_EX source and display session information in the text margin.
- Show the total time needed to present a given block, chapter or session at its start.
- Generate a table of sessions to track and estimate the progress throughout the course.
- Show brief comments regarding presentation, e.g. to skip certain parts or to pay particular attention.
- Maintain timing information in updates as conveniently as possible.
- Automatically generate the subdivision of sessions (based on the timing information provided).

2 Usage

This section describes the structures, markup and mechanisms provided by the package and some hints how to structure and maintain the information within the document. A complete reference of commands and configuration options is given in section 3.

To use the package `sesstime` add the command

```
\usepackage{sesstime}
```

to the preamble of your L^AT_EX document.

2.1 Blocks

The basic timing information is entered in a unit called *block*. It has the following generic structure:

```
\timingstart{15}  
abc  
def  
\timingsplit{30}  
ghi  
jkl  
\timingsplit{40}  
mno  
pqr  
\timingstop{1:00}
```

The above timing information is provided as absolute times (in [hours:]minutes), such that the block begins at 15 minutes and ends at 1 hour. It thus takes 45 minutes in total to present this block. Furthermore, two intermediate times are specified at 15 minutes and 25 minutes into the block (albeit in the form of absolute times). The timing information for the block will be displayed as:

90	
45	↓ abc
15	↕ def
	ghi
25	↕ jkl
	mno
45	↑ pqr
135	

Note that the displayed times are always relative to the block beginning. The bold numbers show the overall time outside the block (assuming that this block starts at 90 minutes). Furthermore, note that two L^AT_EX passes are needed to display all data correctly (as for labels).

Alternatively, the information can be entered as relative times with respect to the previous timing command:

```
\timingstart*{0}
...
\timingsplit*{15}
...
\timingsplit*{10}
...
\timingstop*{20}
```

This markup has the same effect as the above timing commands using absolute times. There are some benefits and shortcomings to the two modes of entering the information, and the choice is left to the user. Note that relative and absolute times can be mixed (one absolute reference point should be specified previously).

Some remarks and considerations are in order for setting up timing blocks and maintaining them when changing the presented material:

- If the duration of some presented material is estimated, it makes sense to specify relative times.
- If the duration is determined by periodically noting the present time displayed on a clock, it makes sense to specify these absolute times directly.
- When absolute times are specified, the duration of a block is determined by the final `\timingstop` (only). Split times given by `\timingsplit[*]` do not change the overall length. In order to change the duration of a block when some presented material is added or removed, all subsequent absolute times have to be adjusted. Alternatively, additional time can be specified by:

```
\timinglapse{5}
```

This command conveniently adds 5 minutes to the block without modifying the current absolute time. When using relative times throughout the block, this has the same effect as `\timingsplit*{5}`.

2.2 Chapters

The package provides an additional layer of structure called *chapters*. A chapter combines several blocks in order to display the overall required time as well as relative times within the chapter.

A chapter is enclosed by two commands

```
\timingchapter
abc
def
ghi
\timingchapterend
```

Closing a chapter is optional, it will be closed automatically by starting any subsequent chapter.

There are three uses to declaring chapters. The total time for presenting the chapter is computed and it is displayed at the start:

```
180 *|abc
      |def
      |ghi
```

Furthermore, the bold numbers shown in the above examples for blocks indicate the present time relative to the beginning of the chapter. Finally, one can also show the total time needed for any chapter anywhere within the document, e.g. for a customised table of contents or next to the chapter title in a customised form.

It will be useful to match the chapters and blocks to certain structures in the document. For example, chapters could correspond to chapters or sections depending on which documentclass is used. If the `\include` mechanism of \LaTeX is used to structure the document into several files, it makes sense to use these units as chapters for timing purposes. Blocks would typically correspond to the next level of structure (sections or subsections, respectively) and split times could be inserted naturally at any finer levels of structure (subsubsections, paragraphs).

Establishing some guidelines how to entering timing information will be useful because the package will then compute the total durations of these structures and display the times at their beginnings.

For maintenance purposes, a typical block could last around 30 minutes to an hour and be split at no more than a handful instances; more split times will take more efforts in maintaining across updates. There is no practical limitation to the length of individual chapters as times are computed automatically; typically it would consist of a couple of blocks.

2.3 Sessions

Chapters will normally not map one-to-one to the units of presentation (individual lectures) which we shall call *sessions*. It may be useful to keep track or estimate where each session will start (or end). There are two mechanisms to specify sessions. The manual declaration takes the form:

```
\timingsession
abc
def
ghi
\timingsessionend
```

Again, closing a session is optional, it will be closed automatically by starting any subsequent session. Alternatively, the subdivision of sessions can be automatically estimated by the command:

`\timingconfigure{autosession=duration}`

This declares the duration of a session to be *duration* minutes. New sessions will be started automatically when the available time for the previous session has elapsed (as far as possible with the timing information provided).

Sessions have two uses. The start of a session is indicated in the margin together with its duration:

abc	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; padding: 2px; display: inline-block;"> session 4 (35 min) </div>
def	
ghi	

Furthermore, a list of sessions can be displayed by the command:

`\timinglistofsessions`

The list of sessions can serve two purposes in planning a lecture course. First, it outlines the (approximate) overall schedule and allows one to find out conveniently whether there will be sufficiently many sessions to cover the intended material. Second, the preview can help coordinating the topics for exercises to accompany the course.

Note that the automatic assignment of sessions relies on the timing information provided in the blocks. On the one hand, the latter should be sufficiently fine-grained so that the session breaks can be located accurately. On the other hand, it will not be worth the effort to establish too detailed information. Providing one timing mark for around one third of the duration of a session will produce reasonable results. In the case of coarse-grained information, new sessions will be triggered such that the overall schedule remains approximately valid. The session marks will then be displayed some time after the session is expected to start, and the time difference is displayed.

Note also that the list of sessions and total times rely on information stored in the `.aux` files. In order to update the information, it may be necessary to compile the complete document (without `\includeonly` statements) twice.

Finally, it may be useful to emphasise that session marks can also be displayed without providing timing information. In that case, the subdivision of session evidently cannot be generated automatically, but must be stored manually.

2.4 Comments

Occasionally it is useful to write some brief comments or instructions in the margin of lecture notes. While there are native L^AT_EX tools to do so, the package provides an integrated mechanism:

`\timingcomment{comment}`

Such comments can serve many purposes, such as:

- ‘skipped’: The following part is skipped in the presentation.
- ‘brief!’: Be brief! Do not waste time!
- ‘explain!’: Explain well! Take your time!
- ‘exercise’: Hint at an accompanying exercise.
- ...

skipped

brief!

explain!

exercise

2.5 Flags

The package allows to specify flags to distinguish different versions of the document or to set it up for different purposes (e.g. a particular iteration of the lecture course or lecture notes vs. textbook):

```
\timingconfigure{flags={flags}}
```

Here *flags* is a comma-separated list of user-defined flags. Please avoid special characters and spaces (spaces next to comma separators are presently not ignored).

Most timing commands provided by the package accept an optional argument which can be `[if=flag]` or `[not=flag]` (or combinations of these separated by comma). These commands will be processed only if the specified condition(s) hold. For example, the commands

```
\timingcomment[if=2017]{skip}  
\timinglapse[if=2017]{-10}
```

can be used to indicate that the following part of the text is skipped in the 2017 iteration of the lecture course, and 10 minutes are saved by this omission.

2.6 Printing

Detailed timing information or comments may be intended for private purposes only, while these marks should be stripped in a public version of the document. The package supplies some global options to enable and disable the printing of timing marks. To turn off all marks, use the package option:

```
\timingconfigure{print=false}
```

Also individual types of timing marks can be suppressed by similar package options, see section 3.6. For a finer selection or suppression of comments, one could use an optional argument such as `[if=private]` (only in the private version) or `[not=public]` (only in the public version), see section 2.5.

Please note that the package uses a simple mechanism to place timing marks and comments in the margin next to the current location. This mechanism requires \TeX to be in so-called vertical mode (loosely speaking the space between paragraphs); marks cannot be set in horizontal mode (text within the paragraphs). As timing marks are naturally placed between the paragraphs and to avoid complications, a paragraph is automatically broken when a timing mark is placed in horizontal mode. This has the effect of splitting the current paragraph in two or occasionally inserting some extra vertical space between paragraphs. Note that this applies only if timing marks are actually printed, otherwise the structure of paragraphs is left untouched.

Relatedly, it makes no sense to place timing marks *after* a (sub)paragraph command which enforces horizontal mode to stay on the same line as the (sub)paragraph title; place any marks before the (sub)paragraph command. Also place timing marks at your own risk within manually designed boxes or minipages and avoid using them in mathematical mode.

2.7 Customisation

The package provides some default layout for the various pieces of information to be displayed. However, it makes sense to customise some of these.

For example, one could provide a list of session dates and customise the session display by changing the commands `\timingprintsession` and/or `\timingsessionline` to show the time instead of the session number. See appendix A for a sample implementation.

Please refer to the command reference in section 3 as well as the implementation in appendix B to find out about commands to be customised: Commands suitable for customisation are among the public ones (prefix `timing`) and they should not make reference to internal commands (prefix `sti@`).

3 Command Reference

This section describes the commands provided by the package *sesstime*.

3.1 Blocks

Timing information is entered using the block commands:

<code>\timingstart</code>	Timing information is entered using the block commands:	
<code>\timingsplit</code>		<code>\timingstart[cond]{abstime}</code>
<code>\timingstop</code>		<code>\timingsplit[cond]{abstime}</code>
<code>\timingnext</code>		<code>\timingstop[cond]{abstime}</code>
		<code>\timingnext[cond]{abstime}</code>

The optional argument *cond* is a comma-separated list of options described in section 3.5. The argument *abstime* represents the absolute time at this instant; it can be specified in minutes or in the format *hour:minutes*. The first three of the above commands start, split or stop a block, respectively. The last command stops a block and starts the next one at the same time.

Each of these commands has an alternative form:

```
\timingstart*[cond]{reltime}
\timingsplit*[cond]{reltime}
\timingstop*[cond]{reltime}
\timingnext*[cond]{reltime}
```

Here the argument *reltime* is a relative time (in minutes) based on the previous block command.

There are two further commands to tweak the timers in useful ways:

<code>\timinglapse</code>	There are two further commands to tweak the timers in use
<code>\timinggauge</code>	<code>\timinglapse[<i>cond</i>]{<i>reltime</i>}</code>

This command lets all timers evolve by *reltime* minutes while keeping the absolute time fixed. It can be used to add or remove time from a block which is otherwise specified by absolute times. Instead of adjusting all later times by the lapse, it suffices to insert this command between two absolute time commands.

The other command implements the opposite effect:

$$\backslash\mathrm{timinggauge}[cond]\{abstime\}$$

This command adjusts the current reference time to *abstime* (specified as minutes or in the format *hour:minutes*) without evolving any of the timers. This can be used if a block stretches over several sessions where the absolute time changes between sessions. Gauging the time allows one to enter absolute times as read off from a clock.

The total time of the current block (in minutes) can be obtained by the command:

$$\backslash \text{timingblocktotal}[*][error]$$

The duration is saved into the macro `\timingreturn`. The starred form returns the duration of the subsequent block instead of the current block. If the time cannot be determined the optional argument *error* (default value `'??'`) is returned.

3.2 Chapters

Chapters are started and ended by the commands:

<code>\timingchapter</code> <code>\timingchapterend</code>	Chapters are started and ended by the commands: <div style="text-align: right;"> <code>\timingchapter[cond]</code> <code>\timingchapterend[cond]</code> </div>
---	--

`\timingchapterlabel` The total duration of a chapter is determined by the commands:
`\timingchaptertotal`

```
\timingchapterlabel{label}
\timingchaptertotal[error]{label}
\timingchaptertotal*[error]{chapter}
```

The first one must be invoked within a chapter to assign the label *label*. The second one saves the duration for the chapter labelled by *label* into the macro `\timingreturn`. The third (starred) form selects the chapter by the number *chapter*. If undetermined, the optional argument *error* (default value ‘??’) is returned.

3.3 Sessions

`\timingsession` Sessions are started and ended by the commands:
`\timingsessionend`

```
\timingsession[cond]
\timingsessionend[cond]
```

`\timinglistofsessions` The list of sessions is printed by the command:

```
\timinglistofsessions
```

3.4 Comments

`\timingcomment` A comment is displayed by the command:

```
\timingsession[cond]{comment}
```

As space in the margin is typically very limited, it makes sense to be as concise as possible, e.g. use just a single word or two.

3.5 Flags

The optional argument *cond* for most of the above commands is a comma-separated list to specify the conditions under which the command is to be processed:

- `if=flag` – Process command only if *flag* is among the currently declared flags.
- `not=flag` – Process command unless *flag* is among the currently declared flags.

Note that the conditionals are processed in the same order as specified. After the first one is evaluated, all subsequent ones change the status only if *flag* is among the currently declared flags.

`\timingif` A command for optional processing is:

```
\timingif[cond]{code}
```

The code *code* is processed only if the conditions *cond* are met.

3.6 Package Options

`\timingconfigure` Options can be passed to the package by the commands:

```
\usepackage[opts]{sesstime}
or \PassOptionsToPackage{opts}{sesstime}
or \timingconfigure[cond]{opts}
```

`\PassOptionsToPackage` must be used before `\usepackage`; `\timingconfigure` must be used afterwards (for selected options). *opts* is a comma-separated list of options. Below we provide a complete list of available options:

- `print[=true|false]` (no value implies `true`, initially set to `true`) – Enable/disable printing all timing marks.
- `block[=true|false]` (no value implies `true`, initially set to `true`) – Enable/disable printing block marks.
- `chapter[=true|false]` (no value implies `true`, initially set to `true`) – Enable/disable printing chapter marks.
- `session[=true|false]` (no value implies `true`, initially set to `true`) – Enable/disable printing session marks.
- `comment[=true|false]` (no value implies `true`, initially set to `true`) – Enable/disable printing comments.
- `warning[=true|false]` (no value implies `true`, initially set to `true`) – Enable/disable printing warnings.
- `width=dimen` (initially set to `1cm`) – Specify width of timing marks in both margins.
- `widthl=dimen` (initially set to `1cm`) – Specify width of timing marks in the left margin which are typically related to blocks and chapters.
- `widthr=dimen` (initially set to `1cm`) – Specify width of timing marks in the right margin which are typically related to sessions and comments.
- `autosession=duration` (initially set to `0`) – Set duration of automatically determined sessions; `0` disables automatic determination.
- `depthsec=depth` (initially set to `3`) – Set maximum sectioning depth to be displayed along with their number in the list of sessions; deeper sections are considered as paragraphs and are displayed differently. Values `0` through `5` correspond to `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` and `\subparagraph`.
- `depthpar=depth` (initially set to `5`) – Set maximum sectioning depth to be displayed in list of sessions (similar to the standard L^AT_EX counter `secnumdepth`).
- `flags=flags` – Specify a comma-separated list of flags. Avoid spaces between flags.

Note that one can specify conditional evaluation for the command `\timingconfigure` to, e.g., conditionally turn on automatic determination of sessions.

4 Information

4.1 Copyright

Copyright © 2016–2018 Niklas Beisert

This work may be distributed and/or modified under the conditions of the L^AT_EX Project Public License, either version 1.3 of this license or (at your option) any later version. The latest version of this license is in <http://www.latex-project.org/lppl.txt> and version 1.3 or later is part of all distributions of L^AT_EX version 2005/12/01 or later.

This work has the LPPL maintenance status ‘maintained’.

The Current Maintainer of this work is Niklas Beisert.

This work consists of the files `README.txt`, `sesstime.ins` and `sesstime.dtx` as well as the derived files `sesstime.sty`, `stimsamp.tex` with `stimsmp3.tex`, `stimsmp4.tex` and `sesstime.pdf`.

4.2 Files and Installation

The package consists of the files:

<code>README.txt</code>	readme file
<code>sesstime.ins</code>	installation file
<code>sesstime.dtx</code>	source file
<code>sesstime.sty</code>	package file
<code>stimsamp.tex</code>	sample file
<code>stimsmp3.tex</code>	sample include file
<code>stimsmp4.tex</code>	sample include file
<code>sesstime.pdf</code>	manual

The distribution consists of the files `README.txt`, `sesstime.ins` and `sesstime.dtx`.

- Run (pdf)L^AT_EX on `sesstime.dtx` to compile the manual `sesstime.pdf` (this file).
- Run L^AT_EX on `sesstime.ins` to create the package `sesstime.sty` and the samples `stimsamp.tex` with include files `stimsmp3.tex` and `stimsmp4.tex`. Copy the file `sesstime.sty` to an appropriate directory of your L^AT_EX distribution, e.g. `texmf-root/tex/latex/sesstime`.

4.3 Interaction with CTAN Packages

The package relies on other packages:

- This package uses the package `keyval` from the `graphics` bundle to process the options for the package, environments and macros. Compatibility with the `keyval` package has been tested with v1.15 (2014/10/28).

4.4 Feature Suggestions

The following is a list of features which may be useful for future versions of this package:

- Option to display times as *hh:mm*.
- Warnings for unknown or multiply defined labels or times.
- Display position of timing marks in two-sided documents.
- Ignore leading and trailing spaces in the list of flags.

4.5 Revision History

v1.11: 2018/01/17

- bugfix for section titles including `\texorpdfstring`

v1.1: 2018/01/06

- grabbing of section titles improved
- options `depthsec` and `depthpar` to adjust depth in list of sessions

v1.0: 2017/05/01

- first version published on CTAN
- manual and install package
- bugfix chapter total
- minor display changes
- specify absolute times as *hour:minutes*
- block gauge
- conditional depending on flags

v0.7: 2017/04/17

- multiple flags
- chapter labels to access total times
- bugfix for open chapters and sessions at end of document (write immediately)
- display tuned, deactivate selected marks
- public interface streamlined

v0.55 / v0.6: 2017/04/01

- block display and preview
- made flags mechanism more flexible
- block start/split/stop/next uniformised; lapse added

v0.5: 2016/11/03

- basic functionality

A Sample File

In this section we provide an example of how to use some of the `sesstime` features. We also test the behaviour in some special cases.

Preamble. Standard document class:

```
1 \documentclass[12pt]{article}
```

Use package `geometry` to adjust the page layout, adjust the paragraph shape:

```
2 \usepackage{geometry}
3 \geometry{layout=a4paper}
4 \geometry{paper=a4paper}
5 \geometry{margin=3cm}
6 \parindent0pt
```

Include the `sesstime` package:

```
7 \RequirePackage{sesstime}
```

Enable automatic assignment of sessions at 30 minutes each. Optionally turn off all timing marks:

```
8 \timingconfigure{autosession=30}
9 %%\timingconfigure{print=false}
```

Declare schedule of lectures for a customised display:

```
10 \newcommand{\sessiondate}[1]{\ifcase#1\\
11 \or Mon 09:00%
12 \or Mon 10:00%
13 \or Tue 09:00%
14 \or Fri 11:00%
15 \or Fri 12:00%
16 \else END%
17 \fi}
```

Overwrite display of session marks:

```
18 \renewcommand{\timingprintsession}[2]{\timingprintremark{%
19 lecture #1\\
20 \sessiondate{#1}%
21 \timingsessiontotal[0]{#1}\ifnum\timingreturn>0%
22 \\\(\timingreturn\ min)%
23 \ifnum #2>0\\at #2 min\fi\fi\vphantom{g}}}
```

Overwrite depth of sections in session list:

```
24 \timingconfigure{depthsec=3,depthpar=4}
```

Overwrite display of sessions list:

```
25 \renewcommand{\timingsessionline}[5]{\ \ \%
26 \makebox[2em][r]{#1. }%
27 \makebox[5em][l]{\sessiondate{#1}:}%
28 \makebox[2.5em][r]{#3. }%
29 \makebox[10em][l]{\def\tmp{#5}\def\empty{}%
30 \ifx\tmp\empty #4\else\S\ #5\fi}%
31 \timingsessiontotal[0]{#1}\ifnum\timingreturn>0%
32 \ [\timingreturn\ min\ifnum #2>0; #2 min into\fi]\fi}
```

Define a macro \lorem to write out some paragraph of text:

```
33 \def\lorem{Lorem ipsum dolor sit amet, consectetur adipisicing elit,
34 sed eiusmod tempor incididunt ut labore et dolore magna aliqua.
35 Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris
36 nisi ut aliquid ex ea commodo consequat.
37 Quis aute iure reprehenderit in voluptate velit esse
38 cillum dolore eu fugiat nulla pariatur.
39 Excepteur sint obcaecat cupiditat non proident,
40 sunt in culpa qui officia deserunt mollit anim id est laborum.\par}
```

Start document body:

```
41 \begin{document}
```

Chapter 1. Basic test of functionality: First block before subsections provides several intermediate times. Both subsections are in individual blocks separated by `\timingnext`. Most times are specified in absolute terms, merely last block is stopped after a relative time. Manual display of total time in chapter:

```
42 \section{one}
```

```

43 \timingchapter
44 \timingchapterlabel{one}
45 \timingstart{0}
46 total chapter duration: \timingchaptertotal{one}\timingreturn\ min\par
47 \lorem
48 \lorem
49 \timingsplit{10}
50 \lorem
51 \timingsplit{15}
52 \lorem
53 \lorem
54 \timingstop{25}
55 \subsection{one.one}
56 \timingstart{0}
57 \lorem
58 \subsection{one.two}
59 \timingnext{15}
60 \lorem
61 \timingstop*{10}
62 \timingchapterend

```

Chapter 2. Test of some special cases. Block started without previous block closed. Specified absolute time leads to negative time evolved. Absolute time specified as *hour:minutes*. Manual display of time in block (before or after start). Behaviour within paragraph (display active/disabled):

```

63 \section{two}
64 \timingchapter
65 total following block: \timingblocktotal*\timingreturn\ min\par
66 \timingstart{0:15}
67 total current block: \timingblocktotal\timingreturn\ min\par
68 \lorem
69 \subsection{two.one}
70 \timingsplit*{10}
71 \lorem
72 \timingstart{0}
73 \subsection{two.two}
74 \timingsplit{40}
75 \lorem
76 \timingsplit{30}
77 \lorem
78 \timingstop{1:00}
79 \subsection{two.three}
80 \timingstart{0}
81 \lorem
82 some text broken
83 \timingsplit{10}
84 by active timing mark\par
85 \lorem
86 some text broken
87 {\timingconfigure{print=false}\timingsplit{15}}%
88 by inactive timing mark\par
89 \lorem
90 \timingstop{30}

```

Included Chapters 3 and 4. Test of interoperability with \include mechanism:

```

91 \include{stimsmp3}
92 \include{stimsmp4}

```

Chapter 5. Just another chapter with paragraphs:

```

93 \section{five}
94 \timingchapter
95 \paragraph{five.paragraph.}
96 \lorem
97 \timingstart{0}
98 \lorem
99 \timingsplit*{15}
100 \lorem
101 \timingsplit*{15}
102 \subsection{five.one}
103 \lorem
104 \timingsplit*{15}
105 \paragraph{five.one.paragraph.}
106 \lorem
107 \timingsplit*{15}
108 \subsection{five.two}
109 \paragraph{five.two.paragraph.}
110 \timingsession
111 \lorem
112 \timingsplit*{15}
113 \subparagraph{five.two.subparagraph.}
114 \timingsession
115 \lorem
116 \timingsplit*{15}
117 \lorem
118 \timingstop*{15}
119 \timingchapterend

```

Chapter 6. Test of \timinglapse and \timinggauge:

```

120 \section{six}
121 \timingchapter
122 \timinglapse{5}
123 \timingstart{10}
124 \lorem
125 \timinglapse{-5}
126 \lorem
127 \timingstop{20}
128 \lorem
129 \timinglapse{5}
130 \timingstart{20}
131 \lorem
132 \timingsplit{30}
133 \timinggauge{10}
134 \lorem
135 \timingstop{20}
136 \timingchapterend

```

Comments and Flags. Test comments and flags/conditionals:

```

137 \section{flags}
138 \timingcomment{test comments}

```

```

139 set flags ‘two’ and ‘three’: \timingconfigure{flags={two,three}}\par
140 one? \timingif[if=one]{yes}\par
141 two? \timingif[if=two]{yes}\par
142 one or two? \timingif[if=one,if=two]{yes}\par
143 not one? \timingif[not=one]{yes}\par
144 not two? \timingif[not=two]{yes}\par
145 not one or not two? \timingif[not=one,not=two]{yes}\par

```

List of Sessions. Test the List of Sessions.

```

146 \appendix
147 \section{list of sessions}
148 the list of sessions
149 \timinglistofsessions

```

End of document body:

```

150 \end{document}

```

Chapters 3 and 4 Include Files. A standard chapter in an included file `stimsmp3.tex`:

```

151 \section{three}
152 \timingchapter
153 \timingstart{30}
154 \lorem
155 \subsection{three.one}
156 \timingsplit{35}
157 \lorem
158 \subsection{three.two}
159 \timingsplit{40}
160 \lorem
161 \timingsplit{50}
162 \lorem
163 \timingstop{60}
164 \timingchapterend

```

Another chapter in an included file `stimsmp4.tex`. Test a very long interval to let automatic session generation overflow:

```

165 \section{four}
166 \timingchapter
167 \timingstart{0}
168 \lorem
169 \subsection{four.one}
170 \timingstop{25}
171 \lorem
172 \timingstart{-20}
173 \subsection{four.two}
174 \timingsplit{40}
175 \lorem
176 \timingsplit{50}
177 \lorem
178 \timingstop{60}
179 \timingchapterend

```

B Implementation

In this section we describe the package `sesstime.sty`.

Required Packages. The package loads the package `keyval` if not yet present. `keyval` is used for extended options processing:

```
180 \RequirePackage{keyval}
```

Flags. The following code compares flags for a command against a global list of flags for optional processing of the command.

`\sti@processflags` Declare variables for processing. `\sti@flags` stores the list of selected flags. `\ifsti@flag` is set by `\sti@processflags`. `\ifsti@flaginit` is used to detect the first specified key:

```
181 \def\sti@flags{}
182 \newif\ifsti@flag
183 \newif\ifsti@flaginit
```

Compare given flag to list of flags stored in `\sti@flags`; set `\ifsti@flag` accordingly. First key sets default to opposite of specified condition:

```
184 \define@key{sti@flag}{if}{%
185   \ifsti@flaginit\else\sti@flagfalse\fi\sti@flaginittrue%
186   \def\sti@tmpa{#1}%
187   \@for\sti@tmpb:=\sti@flags\do{\ifx\sti@tmpb\sti@tmpa\sti@flagtrue\fi}}
188 \define@key{sti@flag}{not}{%
189   \ifsti@flaginit\else\sti@flagtrue\fi\sti@flaginittrue%
190   \def\sti@tmpa{#1}%
191   \@for\sti@tmpb:=\sti@flags\do{\ifx\sti@tmpb\sti@tmpa\sti@flagfalse\fi}}
```

Initialise and process keys:

```
192 \newcommand{\sti@processflags}[1]{\sti@flagtrue\sti@flaginitfalse%
193   \setkeys{sti@flag}{#1}}
```

`\timingif` Conditional processing:

```
194 \newcommand{\timingif}[2][\]{\sti@processflags{#1}\ifsti@flag#2\fi}
```

Internal Definitions. The following definitions store package options. The conditional `\ifsti@print[...]` indicates whether timing marks (of particular kind) should be printed. The dimensions `\sti@widthl` and `\sti@widthr` store the width of timing marks on the left and right, respectively. `\sti@autosession` stores the duration of automatically generated sessions. `\sti@depthsec` and `\sti@depthpar` store the depth of sections to be included in the list of sessions:

```
195 \newif\ifsti@print\sti@printtrue
196 \newif\ifsti@printblock\sti@printblocktrue
197 \newif\ifsti@printsession\sti@printsessiontrue
198 \newif\ifsti@printchapter\sti@printchaptertrue
199 \newif\ifsti@printwarning\sti@printwarningtrue
200 \newif\ifsti@printcomment\sti@printcommenttrue
201 \newdimen\sti@widthl\setlength{\sti@widthl}{1cm}
202 \newdimen\sti@widthr\setlength{\sti@widthr}{1cm}
203 \newcounter{sti@autosession}
204 \def\sti@depthsec{3}
205 \def\sti@depthpar{5}
```

Package Options. Declare package options:

```
206 \define@key{sti@}{print}[true]{\csname sti@print#1\endcsname}
```



```

207 \define@key{sti@}{block}[true]{\csname sti@printblock#1\endcsname}
208 \define@key{sti@}{chapter}[true]{\csname sti@printchapter#1\endcsname}
209 \define@key{sti@}{session}[true]{\csname sti@printsession#1\endcsname}
210 \define@key{sti@}{warning}[true]{\csname sti@printwarning#1\endcsname}
211 \define@key{sti@}{comment}[true]{\csname sti@printcomment#1\endcsname}
212 \define@key{sti@}{width}{\setlength{sti@widthl}{#1}%
213   \setlength{sti@widthr}{#1}}
214 \define@key{sti@}{widthl}{\setlength{sti@widthl}{#1}}
215 \define@key{sti@}{widthr}{\setlength{sti@widthr}{#1}}
216 \define@key{sti@}{autosession}{\setcounter{sti@autosession}{#1}}
217 \define@key{sti@}{flags}{\def\sti@flags{#1}}
218 \define@key{sti@}{depthsec}{\def\sti@depthsec{#1}}
219 \define@key{sti@}{depthpar}{\def\sti@depthpar{#1}}

```

\timingconfigure Configure package options:

```

220 \newcommand{\timingconfigure}[2][{}]{%
221   \sti@processflags{#1}%
222   \ifsti@flag%
223     \setkeys{sti@}{#2}%
224   \fi}

```

Pass undeclared options on to keyval processing:

```

225 \DeclareOption*{\expandafter\timingconfigure\expandafter{\CurrentOption}}

```

Process package options:

```

226 \ProcessOptions

```

Internal Variables. Declare internal variables and counters. As they describe the status which may change in include files, they are declared as counters which are automatically stored in .aux files. Status variables for sessions (counter, within session, time elapsed):

```

227 \newcounter{sti@session}
228 \newcounter{sti@insession}
229 \newcounter{sti@sessiontime}

```

Status variables for chapters (counter, within session, time elapsed):

```

230 \newcounter{sti@chapter}
231 \newcounter{sti@inchapter}
232 \newcounter{sti@chaptertime}

```

Status variables for blocks (counter, within session, reference time, time elapsed):

```

233 \newcounter{sti@block}[sti@chapter]
234 \newcounter{sti@inblock}
235 \newcounter{sti@blockgauge}
236 \newcounter{sti@blocktime}

```

Temporary counter:

```

237 \newcount\c@sti@tmp

```

Print Marks.

\timingprint Print a timing mark into left or right margin. Leave horizontal mode and print only in vertical mode:

```

238 \newcommand{\timingprint}[2][1]{%
239   \ifsti@print%
240     \ifhmode\par\fi%
241     \ifvmode%
242       \@bsphack%
243       \dimen@ \prevdepth%
244       \nointerlineskip%
245       \if r#1%
246         \smash{\rlap{\hspace{\textwidth}\parbox[t]{\sti@widthr}%
247           {\raggedright\renewcommand{\bfdefault}{b}#2}}}%
248       \else%
249         \smash{\llap{\parbox[t]{\sti@widthl}%
250           {\raggedleft\renewcommand{\bfdefault}{b}#2}}}%
251       \fi%
252       \prevdepth \dimen@%
253       \@esphack%
254     \fi%
255   \else%
256     \ignorespaces%
257   \fi}

```

`\timingprintmark` Print a timing mark or a remark:

```

\timingprintremark
258 \providecommand{\timingprintmark}[1]{%
259   \timingprint{\footnotesize#1}}
260 \providecommand{\timingprintremark}[1]{%
261   \timingprint[r]{\tiny\sffamily\hrule#1\hrule}}

```

`\sti@warn` Print a warning message:

```

262 \newcommand{\sti@warn}[2][1]{\ifsti@printwarning%
263   \timingprint[#1]{\scriptsize\textbf{warn}\#2}\fi}

```

Comments.

`\timingprintcomment` Print a comment:

```

264 \newcommand{\timingcomment}[2][1]{%
265   \sti@processflags{#1}%
266   \ifsti@flag%
267     \ifsti@printcomment\timingprintremark{#2}\fi%
268   \fi}

```

Sessions.

`\sti@notesessiontotal` The following code remembers the duration of sessions. The duration is stored in the macro `\sti@sessiontotal@session`:

`\sti@writesessiontotal`

```

269 \newcommand{\sti@notesessiontotal}[2]{%
270   \expandafter\gdef\csname sti@sessiontotal@#1\endcsname{#2}}

```

A blank definition of the above macro is written to the `.aux` file to avoid compile errors in case the package is removed:

```

271 \AtBeginDocument{\immediate\write\@auxout{%
272   \string\providecommand{\string\sti@notesessiontotal}[2]{}}}

```

Writes the duration of the current session to the `.aux` file:

```

273 \newcommand{\sti@writesessiontotal}[1]
274 {\immediate\write\@auxout{\string\sti@notesessiontotal%
275   {\arabic{sti@session}}{#1}}}
```

`\timingsessiontotal` Save the duration of session number in the argument into `\timingreturn`. Return the optional argument if the duration is unavailable:

```

276 \newcommand\timingsessiontotal[2][??]{%
277   \expandafter\let\expandafter\timingreturn%
278   \csname sti@sessiontotal@#2\endcsname%
279   \ifx\timingreturn\relax\def\timingreturn{#1}\fi}
```

`\sti@recordtitle` Record the present sectioning title and number for usage in the list of sessions. The depth in the first parameter in comparison to `\sti@depthsec` and `\sti@depthpar` determines whether the information is stored as section or paragraph:

```

\sti@partitle
280 \newcommand{\sti@recordtitle}[4]{\ifnum#2>\sti@depthpar\else%
281   \ifnum#2>\sti@depthsec%
282     \gdef\sti@partitle{#4}%
283   \else%
284     \xdef\sti@secnum{#3}%
285     \gdef\sti@sectitle{#4}%
286     \gdef\sti@partitle{}%
287   \fi\fi}
```

`\@startsection` The following code hooks into the definition of `\@startsection` to record titles and numbers:

```

288 \let\sti@old@startsection\@startsection%
289 \def\@startsection#1#2#3#4#5#6{\@ifstar%
290   {\sti@ssect{#1}{#2}{#3}{#4}{#5}{#6}}%
291   {\@dblarg{\sti@sect{#1}{#2}{#3}{#4}{#5}{#6}}}}
292 \def\sti@ssect#1#2#3#4#5#6#7{\sti@recordtitle{#1}{#2}{#3}{#4}{#5}{#6}}%
293 \sti@old@startsection{#1}{#2}{#3}{#4}{#5}{#6}*{#7}%
294 \def\sti@sect#1#2#3#4#5#6[#7]#8{{\advance\csname c@#1\endcsname1%
295   \sti@recordtitle{#1}{#2}{\csname the#1\endcsname}{#7}}%
296   \sti@old@startsection{#1}{#2}{#3}{#4}{#5}{#6}[#7]{#8}}
```

`\chapter` The following code similarly hooks into the definitions of `\chapter` (if defined):

```

297 \ifdefined\chapter
298 \let\sti@old@chapter\chapter
299 \def\chapter{\@ifstar\sti@schapter{\@dblarg{\sti@chapter}}}
300 \def\sti@schapter#1{\sti@recordtitle{chapter}{0}{#1}\sti@old@chapter*{#1}}
301 \def\sti@chapter[#1]#2{{\advance\c@chapter1%
302   \sti@recordtitle{chapter}{0}{\thechapter}{#1}}\sti@old@chapter[#1]{#2}}
303 \fi}
```

`\timingprintsession` Print the mark for the current session. Indicate the number and duration of the session as well as the time into the session at which the label is printed (if applicable). The arguments are #1 session number, #2 time into session:

```

304 \providecommand{\timingprintsession}[2]{\timingprintremark{%
305   session #1%
306   \timingsessiontotal[0]{#1}\ifnum\timingreturn>0%
307   \\\(\timingreturn\ min)%
308   \ifnum #2>0\at #2 min\fi\fi\vphantom{g}}}
```

`\sti@startsession` Initialise a new session:

```

309 \newcommand{\sti@startsession}{%
310   \setcounter{sti@insession}{1}%
311   \stepcounter{sti@session}%
312   \ifsti@printsession%
313     \timingprintsession{\arabic{sti@session}}{\arabic{sti@sessiontime}}%
314   \fi%
315   \sti@addsessionline}

```

\sti@checksession Check whether a new session should be started in autosession mode based on the time elapsed since the start of the present session. Assumes that the session has a fixed duration so that the total number of sessions approximates well the total duration. In case of coarse graining of timing information, the actual start of a session may be earlier. Recurse if necessary:

```

316 \newcommand{\sti@checksession}{%
317   \ifnum\value{sti@autosession}>0%
318     \ifnum\value{sti@insession}=0%
319       \setcounter{sti@sessiontime}{0}%
320       \timingsession%
321     \else%
322       \ifnum\value{sti@sessiontime}<\value{sti@autosession}\else%
323         \sti@writesessiontotal{\arabic{sti@autosession}}%
324         \addtocounter{sti@sessiontime}{-\value{sti@autosession}}%
325         \sti@startsession%
326         \sti@checksession%
327       \fi%
328     \fi%
329   \fi}

```

\timingsession Start a new session. End old session if necessary:

```

330 \newcommand{\timingsession}[1] [] {%
331   \timingsessionend[#1]%
332   \sti@processflags{#1}%
333   \ifsti@flag%
334     \setcounter{sti@sessiontime}{0}%
335     \sti@startsession%
336   \fi\ignorespaces}

```

\timingsessionend End the session (if started). Remember session duration:

```

337 \newcommand{\timingsessionend}[1] [] {%
338   \sti@processflags{#1}%
339   \ifsti@flag%
340     \ifnum\value{sti@insession}=1%
341       \sti@writesessiontotal{\the sti@sessiontime}%
342       \setcounter{sti@insession}{0}%
343       \setcounter{sti@sessiontime}{0}%
344     \fi%
345   \fi\ignorespaces}

```

End the session at the end of the document:

```

346 \AtEndDocument{\timingsessionend}

```

iminglistofsessions Display the list of sessions (without a section title):

```

\timingsessionline
\sti@addsessionline
347 \newcommand\timinglistofsessions{\@starttoc{los}}

```

Display one line within the list of session. The arguments are #1 session number, #2 time into session, #3 section identifier, #4 section title, #5 paragraph title:

```

348 \providecommand{\timingsessionline}[5]{\ \ \%
349 \makebox[3em][r]{#1: }\makebox[2.5em][r]{#3. }\%
350 \def\sti@tmp{#5}\ifx\sti@tmp\@empty #4\else\S\ #5\fi%
351 \timingsessiontotal[0]{#1}\ifnum\timingreturn>0%
352 \ [\timingreturn\ min\ifnum #2>0; #2 min into\fi]\fi}

```

Add a line to the list of sessions:

```

353 \newcommand{\sti@addsessionline}{%
354 \addtocontents{los}{\protect\timingsessionline%
355 {\thesti@session}{\thesti@sessiontime}%
356 {\sti@secnum}{\sti@sectitle}{\sti@parttitle}}}%

```

Chapters.

`\sti@notechaptertotal` The following code remembers the duration of chapters. Store the duration in the macro `\sti@chaptertotal@chapter`:

```

357 \newcommand{\sti@notechaptertotal}[2]{%
358 \expandafter\gdef\csname sti@chaptertotal@#1\endcsname{#2}}

```

A blank definition of the above macro is written to the .aux file to avoid compile errors in case the package is removed:

```

359 \AtBeginDocument{\immediate\write\@auxout{%
360 \string\providecommand{\string\sti@notechaptertotal}[2]{}}}

```

Write the duration of the current chapter to the .aux file:

```

361 \newcommand{\sti@writechaptertotal}{
362 {\immediate\write\@auxout{\string\sti@notechaptertotal%
363 {\arabic{sti@chapter}}{\thesti@chaptertime}}}

```

`\timingchapterlabel` The following code remembers labels for chapters. Store the chapter number in the macro `\sti@chapterlabel@label`:

```

364 \newcommand{\sti@notechapterlabel}[2]{%
365 \expandafter\gdef\csname sti@chapterlabel@#1\endcsname{#2}}

```

A blank definition of the above macro is written to the .aux file to avoid compile errors in case the package is removed:

```

366 \AtBeginDocument{\immediate\write\@auxout{%
367 \string\providecommand{\string\sti@notechapterlabel}[2]{}}}

```

Write a label for the current chapter to the .aux file:

```

368 \newcommand{\timingchapterlabel}[1]
369 {\immediate\write\@auxout{\string\sti@notechapterlabel%
370 {#1}{\arabic{sti@chapter}}}}

```

`\timingchaptertotal` Save the total duration of a chapter into the macro `\timingreturn`. Return the optional argument if unavailable. Version specifying the chapter by its number:

```

371 \newcommand\sti@chaptertotalnumber[2][?]{%
372 \expandafter\let\expandafter\timingreturn%
373 \csname sti@chaptertotal@#2\endcsname%
374 \ifx\timingreturn\relax\def\timingreturn{#1}\fi}

```

Version specifying the chapter by its label:

```

375 \newcommand\sti@chaptertotallabel[2][??]{%
376   \expandafter\let\expandafter\timingreturn%
377     \csname sti@chapterlabel@#2\endcsname%
378   \ifx\timingreturn\relax\else%
379     \expandafter\let\expandafter\timingreturn%
380       \csname sti@chaptertotal@\timingreturn\endcsname\fi%
381   \ifx\timingreturn\relax\def\timingreturn{#1}\fi}

```

Version selection:

```

382 \newcommand\timingchaptertotal{%
383   \@ifstar{\sti@chaptertotalnumber}{\sti@chaptertotallabel}}

```

`\timingprintchapter` Print the mark for the present chapter:

```

384 \providecommand\timingprintchapter[1]{\timingprintmark{\raisebox{2.8ex}{%
385   \textbf{\timingchaptertotal*{#1}\timingreturn}%
386   \rlap{\kern0.4em$\star$}\kern1.2em}}}

```

`\timingchapter` Start a new chapter. End old chapter if necessary:

```

387 \newcommand{\timingchapter}[1][ ]{%
388   \sti@processflags{#1}%
389   \ifsti@flag%
390     \ifnum\value{sti@inblock}=1%
391       \PackageWarning{sesstime}{chapter: still in block; stopped}%
392       \sti@warn{chap}%
393       \sti@blockstopby{0}%
394     \fi%
395     \timingchapterend%
396     \setcounter{sti@inchapter}{1}%
397     \setcounter{sti@chaptertime}{0}%
398     \stepcounter{sti@chapter}%
399     \ifsti@printchapter\timingprintchapter{\arabic{sti@chapter}}\fi%
400   \fi\ignorespaces}

```

`\timingchapterend` End the chapter (if started). Remember chapter duration:

```

401 \newcommand{\timingchapterend}[1][ ]{%
402   \sti@processflags{#1}%
403   \ifsti@flag%
404     \ifnum\value{sti@inblock}=1%
405       \PackageWarning{sesstime}{chapterend: still in block; stopped}%
406       \sti@warn{chap}%
407       \sti@blockstopby{0}%
408     \fi%
409     \ifnum\value{sti@inchapter}=1%
410       \sti@writechaptertotal%
411       \setcounter{sti@inchapter}{0}%
412       \setcounter{sti@chaptertime}{0}%
413     \fi%
414   \fi\ignorespaces}

```

End the chapter at the end of document:

```

415 \AtEndDocument{\timingchapterend}

```

Blocks.

`\sti@writeblocktotal` The following code remembers the duration of blocks. Store the duration in the macro
`\sti@noteblocktotal` `\sti@blocktotal@chapter@block`:

```
416 \newcommand{\sti@noteblocktotal}[3]{%
417   \expandafter\gdef\csname sti@blocktotal@#1@#2\endcsname{#3}}
```

A blank definition of the above macro is written to the .aux file to avoid compile errors in case the package is removed:

```
418 \AtBeginDocument{\immediate\write\@auxout%
419   {\string\providecommand{\string\sti@noteblocktotal}[3]{}}}
```

Write the duration of the current block to the .aux file:

```
420 \newcommand{\sti@writeblocktotal}
421   {\immediate\write\@auxout{\string\sti@noteblocktotal%
422     {\arabic{sti@chapter}}{\arabic{sti@block}}{\theststi@blocktime}}}
```

`\timingblocktotal` Save the duration of the current block (in starred form: subsequent block) into the macro `\timingreturn`. Return optional argument if the duration is unavailable. Version for current block:

```
423 \newcommand\sti@blocktotalcurrent[1][??]{%
424   \expandafter\let\expandafter\timingreturn%
425     \csname sti@blocktotal@arabic{sti@chapter}@arabic{sti@block}\endcsname%
426   \ifx\timingreturn\relax\def\timingreturn{#1}\fi}
```

Version for subsequent block:

```
427 \newcommand\sti@blocktotalnext[1][??]{%
428   \setcounter{sti@tmp}{\arabic{sti@block}}\stepcounter{sti@tmp}%
429   \expandafter\let\expandafter\timingreturn%
430     \csname sti@blocktotal@arabic{sti@chapter}@arabic{sti@tmp}\endcsname%
431   \ifx\timingreturn\relax\def\timingreturn{#1}\fi}
```

Version selection:

```
432 \newcommand\timingblocktotal{\@ifstar%
433   {\sti@blocktotalnext}{\sti@blocktotalcurrent}}
```

`\timingprintblock` Print block timing mark:

```
434 \newcommand{\timingprintblock}[3]{\timingprintmark{%
435   \def\sti@tmp{#2}\ifx\sti@tmp\@empty%
436     \raisebox{-1ex}{#1\rlap{\kern0.4em$\Uppdownarrow$}}\kern1.2em%
437   \else%
438     \def\sti@tmp{#1}\ifx\sti@tmp\@empty\else%
439       #1\rlap{\kern0.4em$\Upparrow$}%
440       \vphantom{$\Uppdownarrow$}}\kern1.2em\hrule\fi%
441   \textbf{#2}\rlap{\vphantom{$\Uppdownarrow$}}\kern1.2em%
442   \def\sti@tmp{#3}\ifx\sti@tmp\@empty\else%
443     \hrule#3\rlap{\kern0.4em$\Downarrow$}%
444     \vphantom{$\Uppdownarrow$}}\kern1.2em\fi%
445   \fi}}
```

`\sti@advanceby` Advance all running clocks by some amount of time:

```
446 \newcommand{\sti@advanceby}[1]{%
447   \ifnum\value{sti@inblock}=1\addtocounter{sti@blocktime}{#1}\fi%
448   \ifnum\value{sti@inchapter}=1\addtocounter{sti@chaptertime}{#1}\fi%
449   \ifnum\value{sti@insession}=1\addtocounter{sti@sessiontime}{#1}\fi}
```

`\sti@hoursminutes` Convert hours:minutes format to minutes:

```
450 \def\sti@hoursminutes#1:#2!{%
451   \multiply\value{sti@tmp}60\addtocounter{sti@tmp}{#1}%
452   \def\sti@tmp{#2}\ifx\sti@tmp\@empty\else\sti@hoursminutes#2!\fi}
```

`\sti@computedelta` Convert an absolute time to a relative time:

```
453 \newcommand{\sti@computedelta}[1]{%
454   \setcounter{sti@tmp}{0}%
455   \sti@hoursminutes#1: !%
456   \addtocounter{sti@tmp}{-\value{sti@blockgauge}}%
457   \addtocounter{sti@tmp}{-\value{sti@blocktime}}}
```

`\sti@warnneg` Warn if negative time elapsed:

```
458 \newcommand{\sti@warnneg}[1]{%
459   \ifnum #1<0%
460     \PackageWarning{sesstime}{block: negative time elapsed}%
461     \sti@warn{neg}%
462   \fi}
```

`\timingstart` Start new block. Version specifying the absolute time:

```
463 \newcommand{\sti@blockstartat}[2][]{%
464   \sti@computedelta{#2}%
465   \sti@blockstartby[#1]{\value{sti@tmp}}}
```

Version specifying the relative time:

```
466 \newcommand{\sti@blockstartby}[2][]{%
467   \sti@processflags{#1}%
468   \ifsti@flag%
469     \ifnum\value{sti@inblock}=1%
470       \PackageWarning{sesstime}{already in block}%
471       \sti@warn{block}%
472     \else%
473       \setcounter{sti@inblock}{1}%
474       \addtocounter{sti@blockgauge}{\arabic{sti@blocktime}}%
475       \addtocounter{sti@blockgauge}{#2}%
476       \setcounter{sti@blocktime}{0}%
477       \stepcounter{sti@block}%
478       \sti@checksession%
479       \ifsti@printblock\timingblocktotal%
480         \timingprintblock{\the\sti@chaptertime}{\timingreturn}%
481       \fi%
482     \fi%
483   \fi\nopagebreak}
```

Version selection:

```
484 \newcommand{\timingstart}{\@ifstar{\sti@blockstartby}{\sti@blockstartat}}
```

`\timingsplit` Split current block. Version specifying the absolute time:

```
485 \newcommand{\sti@blocksplitat}[2][]{%
486   \sti@computedelta{#2}%
487   \sti@blocksplitby[#1]{\value{sti@tmp}}}
```

Version specifying the relative time:


```

488 \newcommand{\sti@blocksplitby}[2][]{%
489   \sti@processflags{#1}%
490   \ifsti@flag%
491     \ifnum\value{sti@inblock}=0%
492       \PackageWarning{sesstime}{not in block}%
493       \sti@warn{block}%
494     \else%
495       \sti@warnneg{#2}%
496       \sti@advanceby{#2}%
497       \sti@checksession%
498       \ifsti@printblock%
499         \timingprintblock{\the sti@blocktime}{}%
500       \fi%
501     \fi%
502 \fi}

```

Version selection:

```

503 \newcommand{\timingsplit}{\@ifstar{\sti@blocksplitby}{\sti@blocksplitat}}

```

\timingstop Stop current block. Version specifying the absolute time:

```

504 \newcommand{\sti@blockstopat}[2][]{%
505   \sti@computedelta{#2}%
506   \sti@blockstopby{#1}{\value{sti@tmp}}}

```

Version specifying the relative time:

```

507 \newcommand{\sti@blockstopby}[2][]{%
508   \sti@processflags{#1}%
509   \ifsti@flag%
510     \ifnum\value{sti@inblock}=0%
511       \PackageWarning{sesstime}{not in block}%
512       \sti@warn{block}%
513     \else%
514       \sti@warnneg{#2}%
515       \sti@advanceby{#2}%
516       \sti@writeblocktotal%
517       \ifsti@printblock%
518         \timingprintblock{\the sti@blocktime}{\the sti@chaptertime}{}%
519       \fi%
520       \setcounter{sti@inblock}{0}%
521       \setcounter{sti@blocktime}{0}%
522     \fi%
523 \fi}

```

Version selection:

```

524 \newcommand{\timingstop}{\@ifstar{\sti@blockstopby}{\sti@blockstopat}}

```

\timingnext Stop current block and start next block. Version specifying the absolute time:

```

525 \newcommand{\sti@blocknextat}[2][]{%
526   \sti@computedelta{#2}%
527   \sti@blocknextby{#1}{\value{sti@tmp}}}

```

Version specifying the relative time:

```

528 \newcommand{\sti@blocknextby}[2][]{%
529   \sti@processflags{#1}%
530   \ifsti@flag%

```

```

531 \ifnum\value{sti@inblock}=0%
532 \PackageWarning{sesstime}{not in block}%
533 \sti@warn{block}%
534 \else%
535 \sti@warnneg{#2}%
536 \sti@advanceby{#2}%
537 \sti@writeblocktotal%
538 \stepcounter{sti@block}%
539 \ifsti@printblock\timingblocktotal%
540 \timingprintblock{\thesti@blocktime}%
541 {\thesti@chaptertime}{\timingreturn}%
542 \fi%
543 \addtocounter{sti@blockgauge}{\arabic{sti@blocktime}}%
544 \setcounter{sti@blocktime}{0}%
545 \sti@checksession%
546 \fi%
547 \fi}

```

Version selection:

```

548 \newcommand{\timingnext}{\@ifstar{\sti@blocknextby}{\sti@blocknextat}}

```

`\timinglapse` Insert some time to the current block without changing the current absolute time:

```

549 \newcommand{\timinglapse}[2][0]{%
550 \sti@processflags{#1}%
551 \ifsti@flag%
552 \ifnum\value{sti@inblock}=1%
553 \addtocounter{sti@blockgauge}{-#2}%
554 \fi%
555 \sti@advanceby{#2}%
556 \fi}

```

`\timinggauge` Change the current absolute time without advancing any timers:

```

557 \newcommand{\timinggauge}[2][0]{%
558 \sti@processflags{#1}%
559 \ifsti@flag%
560 \sti@computedelta{#2}%
561 \addtocounter{sti@blockgauge}{\value{sti@tmp}}%
562 \fi}

```