








# Time Tracking

Operation	Keystroke	Function	Note
<b>Time Tracking with Emacs</b>	<p>Two main packages are used to track time with Emacs:</p> <ul style="list-style-type: none"> <li>Emacs built-in <b>timeclock</b> with <b>timelog</b> external library.</li> <li><b>Org-mode with clock time</b></li> </ul> <p>PEL currently explicitly supports the first set and currently provides non-documented support for Org-mode but no extra customization.</p> <p>Time information can be displayed on Emacs mode line when it is enabled. See <a href="#">» Mode Line</a> for more information.</p>		
Open this PDF file. See also: <a href="#">» Help/Info</a>	<b>&lt;f11&gt; T &lt;f1&gt;</b>	<b>(pel-help-pdf</b> &optional OPEN-WEB-PAGE)	Open the <a href="#">» Time Tracking</a> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around.
Open PEL abbreviation customization group. See also: <a href="#">» Customize</a>	<b>&lt;f11&gt; T &lt;f2&gt;</b>	<b>(pel-customize-pel</b> &optional OTHER-WINDOW)	Open the PEL customize group(s) for the current context. Use this to open to change PEL user option variables the activate and control the various abbreviations features. <ul style="list-style-type: none"> <li>When a prefix argument (like <b>C-u</b>) opens the buffer inside another window.</li> </ul>
Customize Emacs built-in abbreviation support See also: <a href="#">» Customize</a>	<b>&lt;f11&gt; T &lt;f3&gt;</b>	<b>(pel-customize-library</b> &optional OTHER-WINDOW)	Customize Emacs <i>time related</i> groups which includes: display-time, timeclock, timelog. <ul style="list-style-type: none"> <li>When a prefix argument (like <b>C-u</b>) opens the buffer inside another window.</li> <li>Group belonging to files that have not yet been loaded are normally not accessible in Emacs and via the customize-group command. PEL, however, attempts to locate the file that defines a non-loaded customization group and will prompt you for loading the file if it finds it.</li> </ul>
<b>Control timeclock display on Modeline</b>	 To activate the display of this information you can set the timeclock-mode-line-display user-option on. <ul style="list-style-type: none"> <li>Access the custom group buffer with <b>&lt;f11&gt; T &lt;f3&gt;</b></li> </ul> This can also be toggled using the following command.		
Toggle display of time left in todays' workday on the mode line  See also: <a href="#">» Mode Line</a>	<b>&lt;f11&gt; T M-d</b>	<b>(timeclock-mode-line-display</b> &optional ARG)	Toggle display of the amount of time left today in the mode line. <ul style="list-style-type: none"> <li>With prefix ARG, turn mode line display on if and onlyif ARG is positive. Returns the new status of timeclock mode line display (non-nil means on).</li> </ul>
	<ul style="list-style-type: none"> <li>If 'timeclock-use-display-time' is non-nil (the default), then the function 'display-time-mode' must be active, and the mode line will be updated whenever the time display is updated. Otherwise the timeclock will use its own sixty second timer to do its updating.</li> </ul>		
<b>World Time Zones</b>	Emacs provides the following functions to display time in different time zones.		
Display time in selected time zones	<b>&lt;f11&gt; T W</b>	<b>(display-time-world)</b>	Enable updating display of times in various time zones inside a *wclock* buffer. <ul style="list-style-type: none"> <li>The 'display-time-world-list' user-option specifies the zones.</li> <li>To turn off the world time display, go to that window and type 'q'.</li> </ul>
<b>Tracking Time Spent Using Emacs built-in timeclock</b>	<ul style="list-style-type: none"> <li>The simple <b>built-in timeclock package</b> with provides a set of commands to define a task name, start and stop timer. This logs into time into the <code>~/emacs.d/timelog</code> file by default. This can be modified by changing the <b>timeclock-file</b> user-option.</li> <li> Activated by <b>pel-use-timeclock</b> user-option. Use <b>&lt;f11&gt; T &lt;f2&gt;</b> to access the custom group to set its value.</li> <li>timeclock can display the following information on the modeline:               <ul style="list-style-type: none"> <li>time left in today's workday, as controlled by the user-option and the command described just above.</li> </ul> </li> </ul>		
Clock in specific activity	<b>C-x t i</b>	<b>(timeclock-in</b> &optional ARG PROJECT FIND-PROJECT)	Clock in, recording the current time moment in the timelog. <ul style="list-style-type: none"> <li>With a numeric prefix ARG, record the fact that today has only that many hours in it to be worked.</li> <li>If ARG is a non-numeric prefix argument (non-nil, but not a number), 0 is assumed (working on a holiday or weekend).</li> <li>This feature only has effect the first time this function is called within a day.</li> </ul> PROJECT is the project being clocked into. Prompt for project (activity) name.
Clock out	<b>C-x t o</b>	<b>(timeclock-out</b> &optional ARG REASON FIND-REASON)	Clock out, recording the current time moment in the timelog. <ul style="list-style-type: none"> <li>If a prefix ARG is given, the user has completed the project that was begun during the last time segment.</li> </ul> Prompt for the user's reason for clocking out.
Change activity	<b>C-x t c</b>	<b>(timeclock-change</b> &optional ARG PROJECT)	Change to working on a different project. <ul style="list-style-type: none"> <li>This clocks out of the current project, then clocks in on a new one.</li> <li>With a prefix ARG, consider the previous project as finished at the time of changeover.</li> <li>PROJECT is the name of the last project you were working on.</li> </ul>
Re-read timeclock file	<b>C-x t r</b>	<b>(timeclock-reread-log)</b>	Re-read the timeclock, to account for external changes inside the timelog file. <ul style="list-style-type: none"> <li>Returns the new value of 'timeclock-discrepancy'.</li> </ul>  If you modify the content of the timelog file, run this command to update all data.
Update timeclock info shown on the mode line	<b>C-x t u</b>	<b>(timeclock-update-mode-line)</b>	Update the 'timeclock-mode-string' displayed in the mode line. <ul style="list-style-type: none"> <li>The value of 'timeclock-relative' user-option affects the display as described in that variable's documentation: Whether to make reported time relative to 'timeclock-workday'. For example, if the length of a normal workday is eight hours, and you work four hours on Monday, then the amount of time "remaining" on Tuesday is twelve hours -- relative to an averaged work period of eight hours -- or eight hours, non-relative. So relative time takes into account any discrepancy of time under-worked or over-worked on previous days. This only affects the timeclock mode line display.</li> <li>To have anything show on the mode line, first do <b>M-x display-time</b> to activate time display.</li> </ul>
Display time of the end of today's workday	<b>C-x t w</b>	<b>(timeclock-when-to-leave-string</b> &optional SHOW-SECONDS TODAY-ONLY)	Return a string representing the end of today's workday. <ul style="list-style-type: none"> <li>This string is relative to the value of 'timeclock-workday' which defaults to 8 hours.</li> <li>If SHOW-SECONDS is non-nil, the value printed/returned will include seconds. If TODAY-ONLY is non-nil, the value returned will be relative only to the time worked today, and not to past time.</li> </ul>
<b>timelog extension for timeclock</b>	This external package complements the built-in timeclock, providing the ability to create time accumulation summaries, something that is lacking from timeclock.  Requires <b>timelog</b> external package  activated when <b>pel-use-timeclock-timelog</b> user-option is turned on.  Originally developed by <b>Markus Flambard</b> and saved as a gist, I cloned and modernized the file and stored it in Github.		
Print time summary for the specified date	<b>C-x t l d</b>	<b>(timelog-summarize-day</b> &optional DATE-STRING)	Print a time summary report for the specified day in the current buffer. <ul style="list-style-type: none"> <li>Interactively: prompts for the specified date in YYYYMMDD format.               <ul style="list-style-type: none"> <li>Use M-n to select today's date.</li> <li>Use M-p and then M-n to navigate through prompt history.</li> </ul> </li> </ul>
Print time summary for today	<b>C-x t l t</b>	<b>(timelog-summarize-today)</b>	Print a time summary report for today in the current buffer.
Print time summary for the specified month	<b>C-x t l m</b>	<b>(timelog-summarize-month</b> &optional MONTH-STRING)	Print a time summary report for the specified month in the current buffer. <ul style="list-style-type: none"> <li>Interactively: prompts for the specified moth in YYYYMM format.               <ul style="list-style-type: none"> <li>Use M-n to select this current month.</li> <li>Use M-p and then M-n to navigate through prompt history.</li> </ul> </li> </ul>
Print time summary for the specified period	<b>C-x t l r</b>	<b>(timelog-summarize-range</b> &optional FIRST-DAY LAST-DAY)	Print a summary for the period starting the first day and ending on the last day. <ul style="list-style-type: none"> <li>Interactively: prompts for the first and last (inclusive) date in YYYYMMDD format.               <ul style="list-style-type: none"> <li>Use M-n to select today's date for the last date.</li> <li>Use M-p and then M-n to navigate through prompt history.</li> </ul> </li> <li> Both dates <b>must</b> be inside the timelog file otherwise the operation fails; the function uses simple date string searches to locate the first and last entries inside the file.</li> </ul>

Operation	Keystroke	Function	Note
Print day-by-day time summary for the specified period	C-x t l D	(timelog-summarize-each-day-in-range &optional FIRST-DAY LAST-DAY)	Print a summary for the each days inside the period starting the first day and ending on the last day. <ul style="list-style-type: none"> <li>Interactively: prompts for the first and last (inclusive) date in YYYYMMDD format. <ul style="list-style-type: none"> <li>Use M-n to select today's date for the last date.</li> <li>Use M-p and then M-n to navigate through prompt history.</li> </ul> </li> <li>⚠ Both dates <b>must</b> be inside the timelog file otherwise the operation fails; the function uses simple date string searches to locate the first and last entries inside the file.</li> </ul>
Display time spent on current project	C-x t l p	(timelog-current-project)	Prints a summary of time spent in the current project on the echo area.
Display time worked today	C-x t l e	(timelog-workday-elapsed)	Prints the amount of time worked today on the echo area.
Open the timeclock-file in current buffer	C-x t l f	(timelog-open-file)	Open the timeclock-file in the current buffer. This is the file where all timeclock activity is stored.
Using Org-Mode to keep track of time	<p>Org-mode is most probably the most flexible, powerful and extensible way to track time via tasks. 🐛</p> <p>I will provide more information about it once I get the timelog to completely working and add some features. Org-mode is a huge topic and is very well documented. For the moment please refer to Org Mode documentation itself. See the references below.</p>		

### Time Tracking — References

Topic & Link	Notes
<a href="#">Clocking Work Time - The Org Manual</a>	
<a href="#">Time Tracking in Emacs with org-clock</a>	Short article written by David Charte, on November 2017.