

Commit User Manual

Last Updated: December 21, 2023

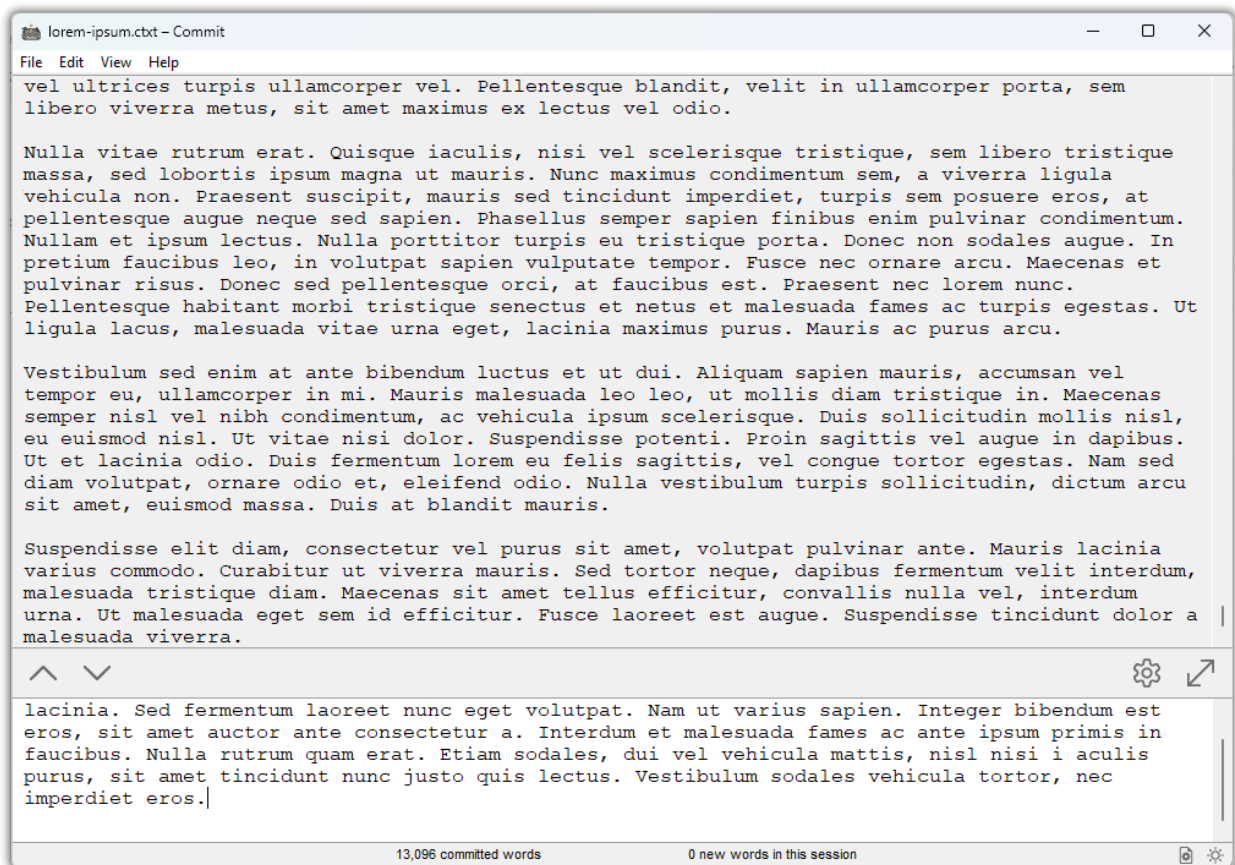
Introduction

Commit is a tool for writing. It stands at the half-way point between a typewriter and a modern word processor program. If you find yourself constantly distracted by an urge to edit and refine the text you have already written instead of writing more text, *Commit* might be able to help.

Only the most recent paragraph is editable. Once you hit “enter” at the end of the paragraph, it is “committed.” Move forward. Write the next paragraph. You can worry about editing and refinement later. For now, your job is to get the words on the page.

User Interface

Main Window



Commit's main interface is simple. The larger field at the top is the text you have already committed; it is visible but not editable. The smaller field at the bottom is your working paragraph, which is editable.

When you press "enter" or click the "commit" button, everything in the working text field moves up to the committed section and is no longer editable. If you make a mistake, you can pull back a single paragraph at a time by clicking the "pull back" icon or pressing "backspace" from the first position in the working field.

Toolbar

A toolbar is positioned between the committed and working text fields. It is split into left and right sections.

- **Left toolbar**
 - "Commit" – commit the text in the working field.
 - "Pull Back" – pull back the most recent paragraph from the committed field.
 - "Saved/Unsaved Status" (shown in "Full Screen" mode if the status bar is disabled) – a solid circle indicates that there are unsaved changes.
- **Right toolbar**
 - "Settings" – open the settings window.
 - "Metadata" (shown if the status bar is disabled) – edit the metadata for the open file.
 - "Full Screen" toggle – switch between "Full Screen" and "Windowed" modes.
 - "Menu" (shown in "Full Screen" mode if the status bar is disabled) – access the limited application menu.
 - "Quit" (shown in "Full Screen" mode) – quit the application.

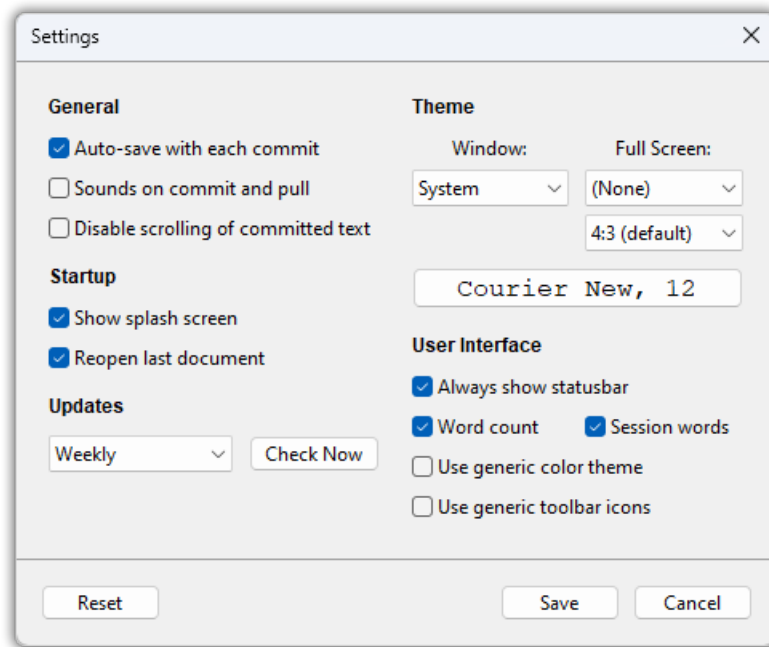
Status Bar

By default, a status bar is shown across the bottom of the *Commit* window. It can be disabled in the settings. The following items are shown on the status bar:

- "Saved/Unsaved Status" (shown in "Full Screen" mode) – a solid circle indicates that there are unsaved changes.
- "Status Information" – textual status information is shown when you mouse-over tools and menu items. If the status bar is disabled, this will still display as a transient overlay.
- "Committed Word Count" – the number of committed words in the document (unless disabled in the settings).

- “Session Word Count” – the number of committed words added during this writing session (unless disabled in the settings).
- “Metadata” – edit the metadata for the open file.
- “Theme Selector” – Select the theme mode (“System,” “Light,” or “Dark”). This setting will apply to the current display mode (“Full Screen” or “Windowed”).
- “Aspect Ratio” (shown in “Full Screen” mode) – choose the aspect ratio setting.
- “Menu” (shown in “Full Screen” mode) – access the limited application menu.

Settings



To access the settings window, click the “Settings” icon in the toolbar or go to “Preferences” in the menus (under “File” on Windows and Linux, and under “Commit” on MacOS).

- **General Settings**
 - “Auto-save with each commit” – default “on” – every time you press “enter” or click “commit,” the file will be saved. If you disable this setting, files will only be saved on request.
 - “Sounds on commit and pull” – default “off” – some writers want positive reinforcement. Enabling this setting will give you a nice “ding” when you commit text.
 - “Disable scrolling of committed text” – default “off” – enabling this setting will “lock” the committed field so you can’t scroll around and distract yourself.

- **Startup Settings**

- “Show splash screen” – default “on” – disabling this setting will get rid of the splash screen when the app starts.
- “Reopen last document” – default “on” – *Commit* normally opens the last document you were working on at startup. Disabling this setting will default to a blank document instead.

- **Updates**

- By default, *Commit* checks for updates weekly. You can switch between “Never,” “Daily,” “Weekly,” or “Monthly.”

- **Theme**

- “Window” – default “System” – *Commit* can run in light or dark mode. By default, it follows your system settings. You can force “Light” or “Dark” modes if you prefer.
- “Full Screen” – default “(None)” – In “Full Screen” mode, *Commit* normally follows your “Window” theme setting. If you want it to look different in “Full Screen” mode, you can select “System,” “Light,” or “Dark” modes separately.
- Full screen ratio – default “4:3” – In “Full Screen” mode, *Commit* defaults the working space to a comfortable 4:3 aspect ratio. You can choose “9:16,” “2:3,” “3:4,” “1:1,” “4:3,” “3:2,” “16:9,” or true “Full Screen” (i.e., use all available space).
- Fonts – *Commit* tries to set a reasonable default font face and size, but you can change it to whatever you like.

- **User Interface**

- “Show status bar” – default “on” – by default, *Commit* shows a status bar across the bottom of the window. You can disable it (essential functions are moved to the toolbar).
- “Word count” – default “on” – the count of committed words in the document is shown center-left. It can be disabled. (Option unavailable if status bar is disabled.)
- “Session words” – default “on” – the count of words committed during the current session is shown center-right. It can also be disabled. (Option unavailable if status bar is disabled.)
- “Use generic color theme” – by default, *Commit* attempts to set interface colors that match your operating system. This selector enables a generic color theme based on the default theme for the Bootstrap web toolkit.
- “Use generic toolbar icons” – by default, *Commit* attempts to use system icons from your operating system. This selector enables a generic icon set based on the Bootstrap Icons package.

- **Reset**

- The “reset” button at the bottom left will reset all options to their defaults and close the application. When you restart it, it will act the same way it did the first time you opened it.

Full Screen View

Clicking the “Full Screen” icon in the toolbar (or “Full Screen” in the “View” menu) will toggle the full screen mode. This mode takes over the whole screen so you can really immerse yourself in your writing.

In the full screen mode, the window’s title bar and menu bar are hidden. A limited menu is available from the “hamburger” icon in the status bar, or, if the status bar is disabled, the toolbar. You can choose a different theme in the full screen mode (i.e., follow system settings in windowed mode, but always use “Dark” mode for full screen). You can also set your desired aspect ratio for the working area.

Commit will remember the position of the window between launches. If it’s in full screen mode when you quit, it’ll be in full screen mode again when you start it next. If it’s maximized when you quit, it’ll be maximized when you restart. If it’s in a windowed mode, it will remember the size and position.

Import and Export

You can import or export files from *Commit* using “Import” and “Export” in the “File” menu. You can also export the entire document to your system clipboard to be pasted into other applications.

Commit supports importing and exporting the following formats:

- [HyperText Markup Language \(HTML\)](#) (*.html) — *recommended interchange format*
- [CommonMark \(Markdown\)](#) (*.md)
- [DocBook](#) (*.dbk)
- [EPUB e-book](#) (*.epub)
- [LibreOffice Text \(OpenDocument Text\)](#) (*.odt)
- [Microsoft Word \(Office Open XML Document\)](#) (*.docx)
- [Plain Text](#) (*.txt)
- [Rich Text Format](#) (*.rtf)

CommonMark (Markdown) and Plain Text, which are simple text formats, are exported using the default line endings on the system where you are running *Commit*: `/n` on Mac and Linux, and `/r/n` on Windows.

Only a sub-set of the available formats can be exported to the clipboard: HyperText Markup Language (HTML), CommonMark (Markdown), and Plain Text. Most word processors and content management systems will accept HTML formatted clipboard content, so that is usually the best option and is preselected by default.

Commit's import and export features rely on an open-source document conversion tool called [Pandoc](#). It is bundled in the MacOS, Windows, and Flatpak packages, and listed as a required dependency in the DEB and RPM packages.

Exporting — Technical Details

First, *Commit* translates all single line breaks in the CTXT formatted document (details below) into CommonMark (Markdown) compliant line breaks — i.e., it adds two spaces before line breaks. This makes CTXT files fully compatible with Pandoc's CommonMark parser.

After normalizing to the CommonMark format, *Commit* will run the document through Pandoc with the CommonMark input format and the desired output format.

The intermediate step of translation to CommonMark occurs even if the chosen output format is CommonMark. This step helps ensure wide compatibility with other Markdown parsers.

The *Commit* CTXT Format

The native CTXT file format for *Commit* is a variant of the CommonMark version of the Markdown format. The file extension is `*.ctxt` and the MIME type is `text/com.intersanity.commit`.

Fundamentally, CTXT is a plain text format. Your content is yours. There's no reason to obscure or complicate it.

If you open a CTXT file in BBEdit, Bluefish, CudaText, E, ed, EDLIN, Emacs, FeatherPad, FreeDOS Edit, Geany, GNOME Text Editor, Kate, Leafpad, mcedit, Mousepad, MS-DOS Editor, Nano, Norton Commander Editor, Notepad, Notepad++, Pe, Pico, Pluma, PSPad, Quick Menu III Editor, SimpleText, StyledEdit, TeachText, TextEdit, Tilde, vi, vim, Visual Studio Code, or Xed, it will be comprehensible. There might be some line-break oddities if you are viewing it on Windows, but that's all.

Generally, you should not edit CTXT files outside of *Commit*. Use *Commit*'s export tools to convert your document for editing in other applications, and then use the import tools to bring it back in.

HTML is a good intermediary format that is understood by most word processors, editors, and content management systems. If you use Microsoft Word or LibreOffice, their native formats might be a better choice. The Rich Text Format (RTF) is also supported by many applications.

There are some small but important differences between CTXT and other Markdown-based formats like CommonMark:

- CTXT files are *always* saved using UNIX-style `/n` line endings, even on Windows.
- A line break is a line break in CTXT. In CommonMark and most other Markdown dialects, single line breaks are treated as a soft break (i.e., a space) unless they are preceded by two spaces.

Advanced Formatting

Generally, *Commit* is meant for **plain text**. But if you really can't live with that, you *can* use [CommonMark \(Markdown\) formatting](#). When you export from *Commit* to other formats, your document will be passed through Pandoc's CommonMark parser and any compatible syntax you have used will be retained.

- Text you wrap in `*asterisks*` or `_underscores_` will be translated into *italics*.
- Text you wrap in `**double asterisks**` or `__double underscores__` will be translated into **bold**.

You can also use link syntax, list syntax, and a lot of other stuff. Review the [CommonMark documentation](#) and [Markdown Guide](#) if you want this kind of control. But you really ought to be focused on writing, not formatting.

The main area where CTXT format differs from CommonMark is **line breaks**. A single line break (which you can insert in *Commit* with shift + enter) is a line break. A normal break is a paragraph break. We do **not** treat single line breaks as a space, as is the case in most Markdown dialects.