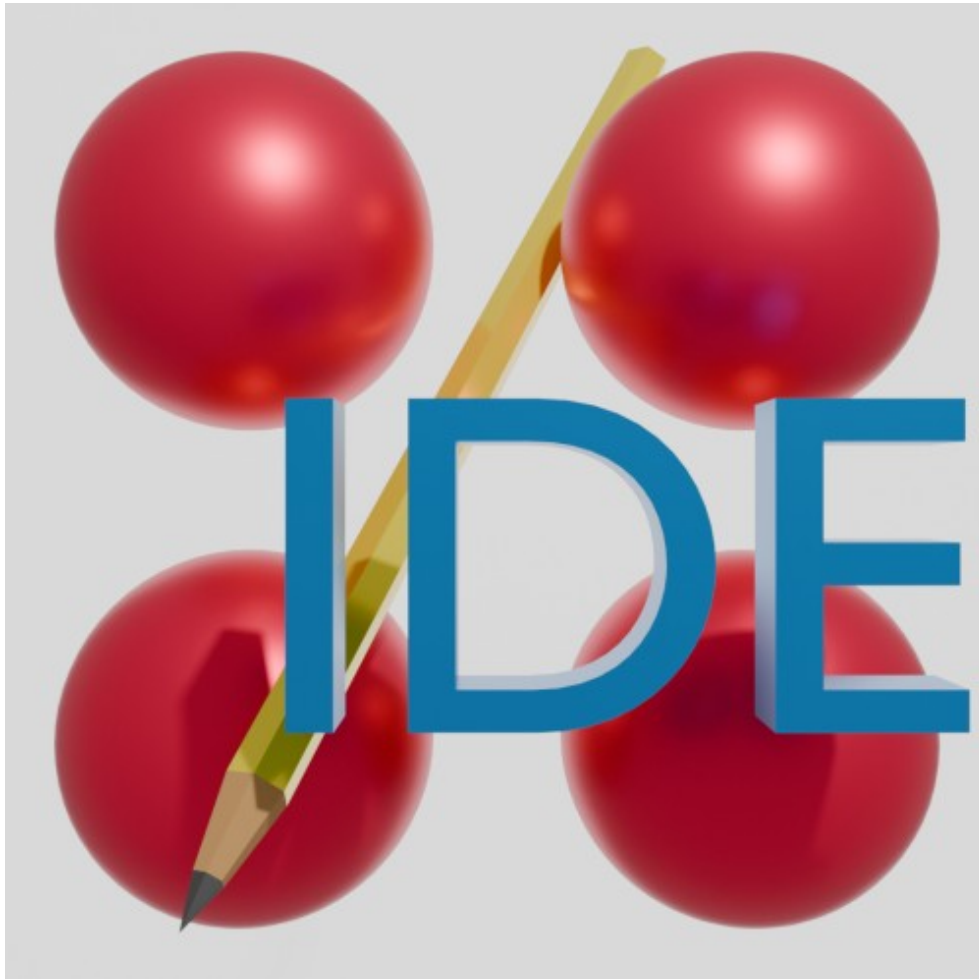# Codit

A versatile text editor / IDE aimed at the

Perl programming language

Manual

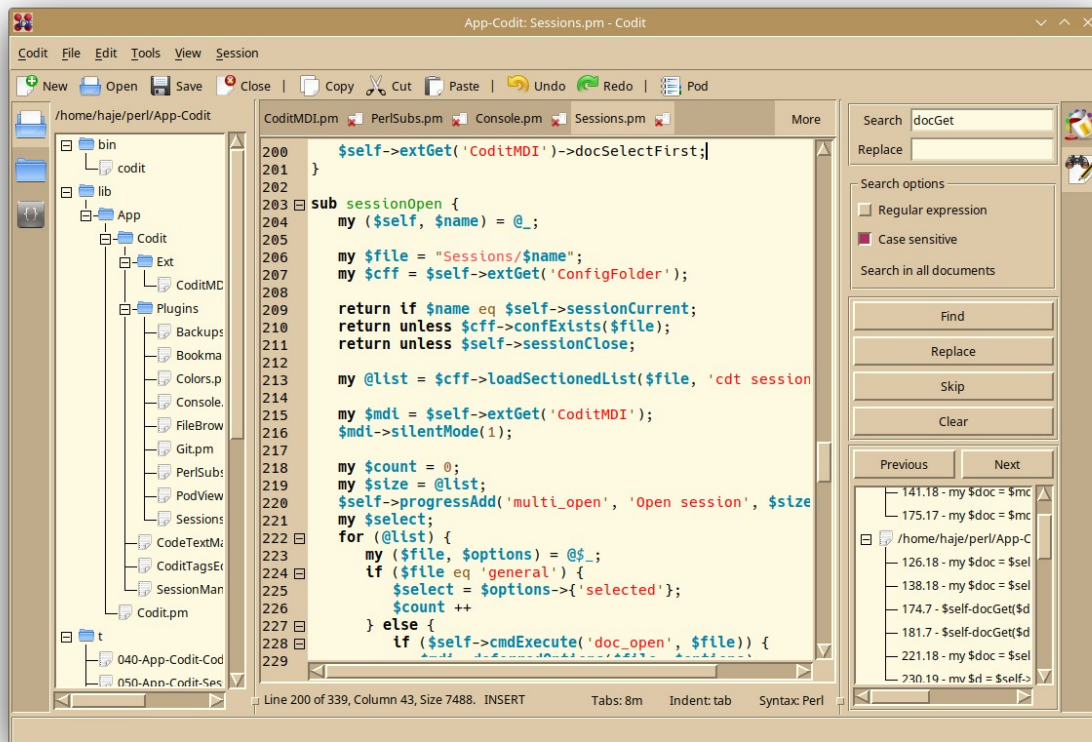Version

0.07

Hans Jeuken

July 3, 2024

Table of Contents

1. Introduction

Codit is a versatile text editor / integrated development environment aimed at the Perl programming language.

It is written in Perl/Tk and based on the Tk::AppWindow application framework.

It uses the Tk::CodeText text widget for editing.

Codit has been under development for about one year now. And even though it is considered alpha software, it already has gone quite some miles on our systems.

It features a multi document interface that can hold an unlimited number of documents, navigable through the tab bar at the top and a document list in the left side panel.

It has a plugin system designed to invite users to write their own plugins.

It is fully configurable through a configuration window, allowing you to set defaults for editing, the graphical user interface, syntax highlighting and (un)loading plugins.

Tk::CodeText offers syntax highlighting and code folding in plenty formats and languages. It has and advanced word based undo/redo stack that keeps track of selections and save points. It does auto indent, comment, uncomment, indent and unindent. Tab size and indent style are fully user configurable.

Enjoy playing!

2. Installation

Codit can simply be installed through CPAN:

>*sudo perl -MCPAN -e "install App::Codit"*

Alternatively you can manually download the package from here:

>*https://github.com/haje61/App-Codit*

After unpacking you do the following:

>*perl Makefile.PL*
>
>*make*
>
>*make test*
>
>*sudo make install*

After make you can do the following for visual inspection:

>*perl -Mblib t/060-App-Codit-CoditTagsEditor.t show*
>
>*perl -Mblib t/100-App-Codit.t show*
>
>*perl -Mblib bin/codit*

Unless you are running Windows, we strongly recommend you also install the Perl modules:

Tk::GtkSettings

Run the following commands each time you login;

>*tkgtk*
>
>*xrdb .Xdefaults*

This will make the look and feel of all your Tk applications conform to your desktop settings and helps Codit locate the correct icon library. The screenshots in this manual are taken from a KDE/Plasma desktop with the Golden Honey Oak color theme.

Image::LibRSVG

This will allow you to load vector graphics based themes like *Breeze*. We did not include it as a prerequisite since it does not respond well to unattended install. It requires the gnome library *librsvg-2* and its development files to be installed.

Codit on Windows

If you use the windows operating system please make sure you have an icon library installed. Preferably the *Oxygen* theme. Download it from here. Extract the file and rename the folder

oxygen-icons-master to *Oxygen*. Create a folder *Icons* in *C:\ProgramData* and move the *Oxygen* folder into it.

You can install any other icon theme. However, you then need to start Codit with the *-icontheme* command line option. SVG based icon themes do not work on Windows.

3. Running Codit

You can launch Codit from the command line as follows:

>	*codit [options] [files]*

The following command line options are available:

- *-c or -config*

	Specifies the configfolder to use. If the path does not exist it will be created.

- *-h or -help*

	Displays a help message on the command line and exits.

- *-i or -iconpath*

	Point to the folders where your icon libraries are located.[*]

- -t or -icontheme

	Icon theme to load.

- *-P or -noplugins*

	Launch without any plugins loaded. This supersedes the -plugins option.

- *-p or -plugins*

	Launch with only these plugins .[*]

- *-s or -session*

	Loads a session at launch. The plugin Sessions must be loaded for this to work.

- *-y or -syntax*

	Specify the default syntax to use for syntax highlighting. Codit will determine the syntax of documents by their extension. This options comes in handy when the file you are loading does not have an extension.

- *-v or -version*

	Displays the version number on the command line and exits.

[*] You can specify a list of items by separating them with a ':'.

4. Using Codit

4.1 The menu system

The main menu learns us a lot about the functionalities of *Codit*. Let's go through it.
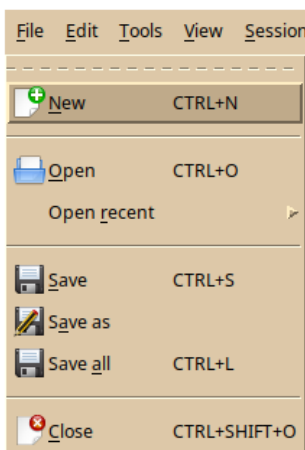
4.1.1 Codit

About opens a dialog box with the Codit logo and some information.

Help opens this manual in your default document viewer.

Settings launches the settings dialog. See chapter 5, Settings.

Quit checks for unsaved documents and exits Codit.
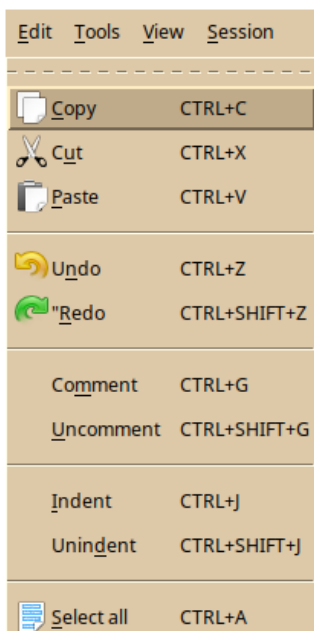
4.1.2 File

The file menu with all its keyboard shortcuts.

Whenever a file is closed it will be added to the recent documents menu. Whenever a file is opened it will be removed from the recent documents menu.

The Save and Close options work on the currently selected file.
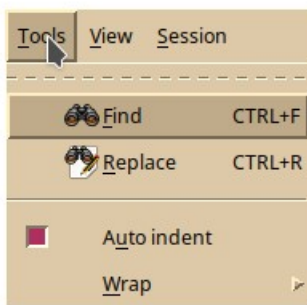
4.1.3 Edit

Rules for commenting and uncommenting are derived from the syntax definition in the current document. If no syntax is selected these commands do nothing.

Commenting, un-commenting, indenting and un-indenting work on single lines and selected blocks of text.
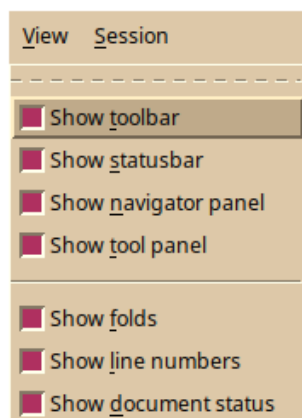
6

4.1.4 Tools

All options in this menu affect only the current selected document.

Auto indent will use the indent style of the current line to generate the same indentation on the nest line when enter is pressed.

Wrap can be set to *char, word* or *none.*

4.1.5 View

The options *toolbar, statusbar, navigator panel,* and *tool panel* show or hide the corresponding panels on the interface.

The options *folds, line numbers* and *document status* show or hide the corresponding panels on the current selected document. Other documents are not affected.

4.2 The tool bar

The toolbar gives you quick access to commands like opening, saving and closing files, cut, copy and past, undo and redo. You can hide it through the *View* menu or in the settings dialog.

4.3 The status bar

The status bar is probably the most boring feature of Codit, or maybe not. It displays information about the last action taken. Small help texts appear when hoovering over toolbar and side panel items. It displays a progress bar when the Sessions plugin loads a session. That's it.

4.4 The document list
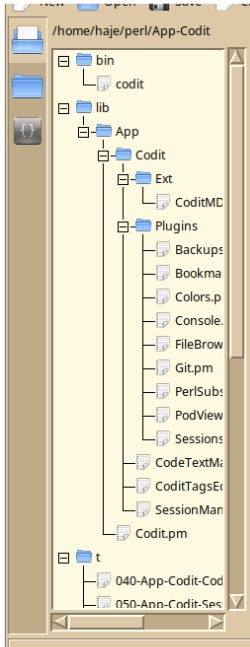
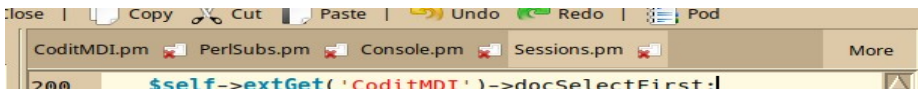The document list displays all open documents in alphanumerical order.

First it displays all un-tracked entries. An entry is un-tracked when it is a newly created document that does not yet exist as a file.

Second it displays all folders containing open files.

Last it displays all open files.

The label in the top of the list shows the folder that all open files have in common.

4.5 Document tabs

The document tab bar can hold an infinite number of open documents. You can move the tabs that are visible by dragging them left or right. When more documents are open than will fit in the *More* button is displayed. When you click it a list of available documents pops up. If you select one it will be shown as the first document in the document bar.

4.6 Indenting lines and selections

Indenting and un-indenting is done on the line where the insert cursor is or on a selection of text if a selection exists. You can specify the indent style in the settings dialog or in the status bar of the text widget.

If you set Indent style to *tab*, indentation is done using tabs. If you set it to a number indentation is done using that amount of spaces. You can use the keyboard shortcut *CTRL+J* for indenting and *CTRL+SHIFT+J* for unindenting.

4.7 Commenting lines and selections

Rules for commenting and uncommenting are derived from the syntax definition in the current document. If no syntax is selected commenting and uncommenting do not work. You can use the keyboard shortcut *CTRL+J* for indenting and *CTRL+SHIFT+J* for unindenting.
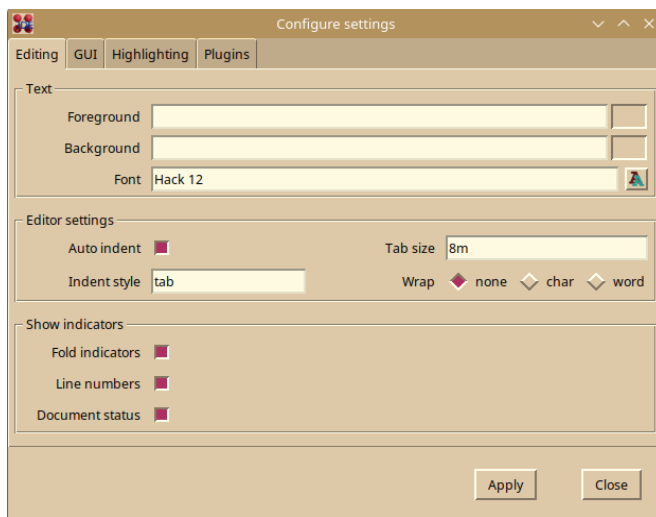
5. Settings

You can launch the settings dialog by pressing *F10 or through the menu:*

> *Codit → Settings*

It can occur that the *Apply* button is grayed out. This happens if one of the settings that is currently loaded is not valid. This happens especially with icon sizes after changing the icon library of your desktop. You can look for entries colored in red and modify them.

You can reset some options to their factory defaults by clearing out their entry. After you hit Apply you have to restart Codit to see the changes.

## 5.1 Editor settings

These are the settings for the text widget.

You may reset the *Foreground, Background, Font, Tab size* and *Indent style* fields to their factory defaults.

Tab sizes are measured in millimeters (m), centimeters (c), inches (i) or pixels (p).

If you set Indent style to *tab*, indentation is done using tabs. If you set it to a number indentation is done using that amount of spaces.

When you hit *Apply* all currently open documents will be updated.

5.2 GUI settings
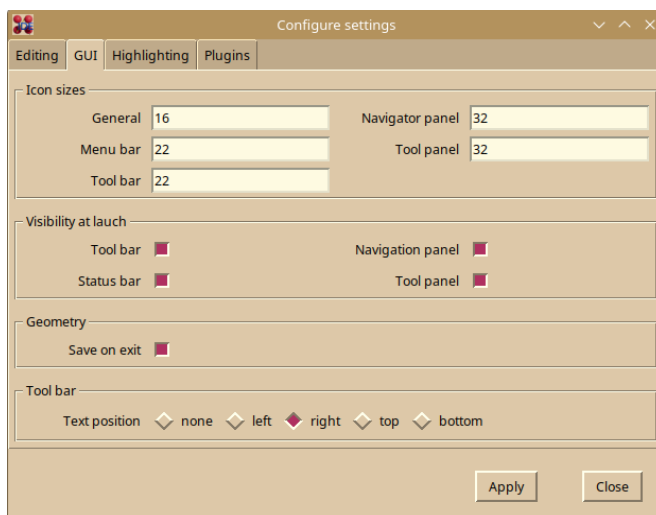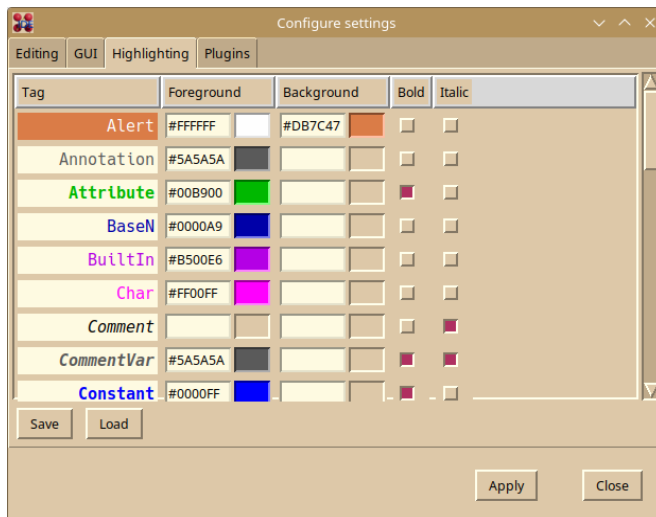

You may reset all icon sizes to their factory defaults by clearing out their entry.

If you set *Save on exit* Codit will remember it's window position and size and will launch next time at the same position and size.

When you hit *Apply* the interface of Codit is updated.
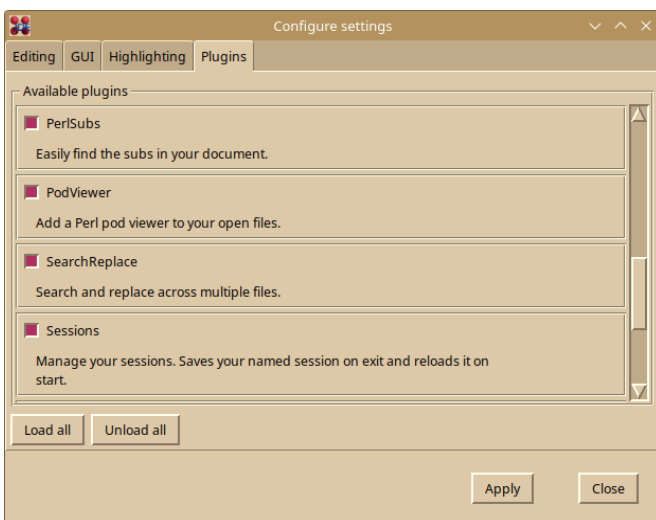
5.3 Highlighting panel

The Highlighting panel allows you to set foreground, background and font options for all highlight attributes.

After you hit apply the highlight tags in all open documents are updated.

You can save and load a highlight theme.

5.4 Plugins panel

The plugins panel allows you to load and unload plugins.

There is no need to hit the *Apply* button. The moment you click a checkbox the plugin is loaded or unloaded.

See also the next chapter, Plugins.

6. Plugins

Plugins add functionality to Codit at the command of the user. They can be loaded or unloaded by you through the settings dialog. This way you can leave out functionality you do not want.

The following plugins are at your disposal.

6.1 Backups

The Backups plugin protects you against crashes of all kinds. It silently does it's job in the background and only reports when it finds an existing backup of a file you open. It keeps backups

of all open and unsaved files. Whenever a file is saved or closed the backup is removed. It keeps the backups in the configuration folder, it does not pollute your working folders.
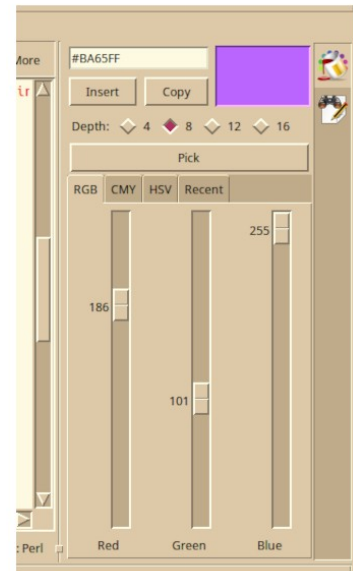
6.2 Colors

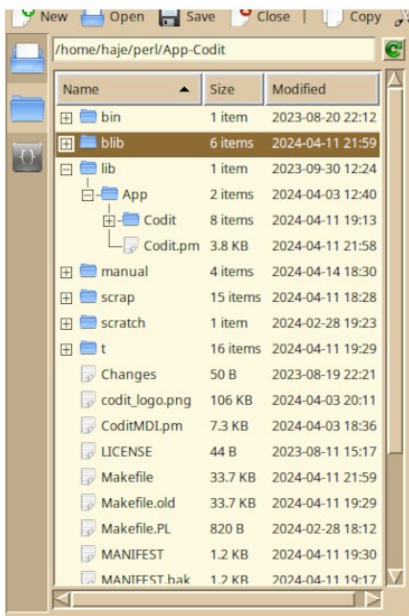The Colors plugin lets you choose a color and insert it's hex value into your document.

You can select a color in RGB, CMY and HSV space. Whenever you select a color it is added to the Recent tab.

It allows you to specify color depths 4, 8, 12 and 16 bits per color.

You can pick a color from any place on the screen with the pick button. This does not work on Windows.

6.3 FileBrowser

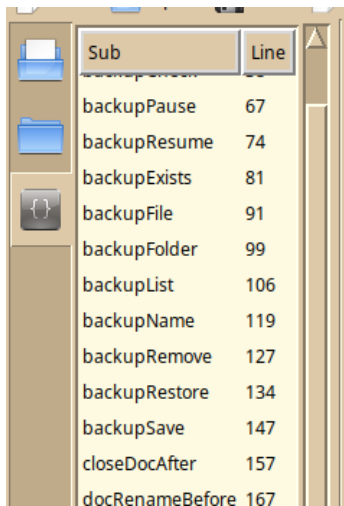The FileBrowser plugin lets you browse your harddrive and open multiple documents at once through its context menu (left-click).

All columns are sortable and sizable. If you left-click the header it will give you options to display hidden files (that start with a '.'), Sort case dependant or not and directories first.

Pressing CTRL+F when the file browser has the focus invokes a filter entry at the bottom.

6.4 PerlSubs

PerlSubs scans the current selected document for lines that begin with '*sub someName*' and displays it in a list with the line number. The list is refreshed after an edit.

When you click on and item in the list, the insert cursor is moved to that line and it is scrolled into visibility.

6.5 PodViewer

PodViewer adds a 'Pod' button to the toolbar. When you click it the frame of the current selected document will split and the bottom half will show the pod documentation in your document. The viewer is refreshed after you make an edit.

6.6 SearchReplace

This plugin allows you to do a search and replace across multiple or just one file. After filling out the search and replace fields you first click *Find*. The list box will fill with the search results. When click *Replace* the first item in the list is replaced and then removed from the list. You can skip replaces by pressing *Skip*. *Clear* deletes all search results.
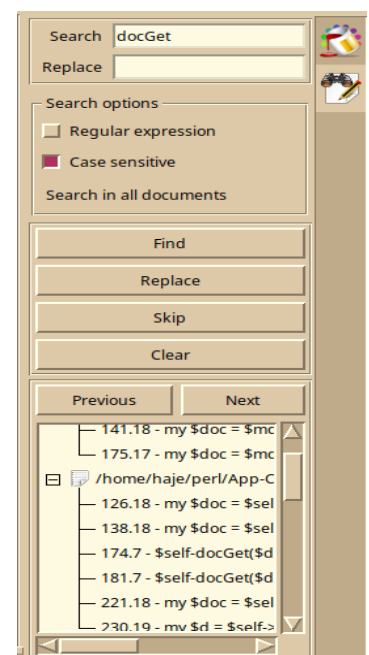
6.7 Sessions
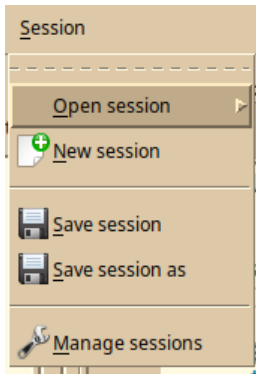
The sessions plugin allows you to save a collection of documents as a session. When re-opening the session the documents are loaded in the exact order as they were when the session was closed. Also the syntax option, tab size, indent style and insert cursor position are saved.

The session manager allows you to keep all your sessions orderly.
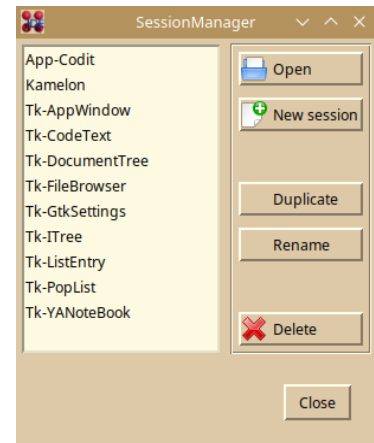
6.8 Snippets

Snippets are shorts pieces of code that you find yourself writing over and over again. The top side of the panel allows you to manage your collection of snippets. The bottom side allows you to insert a selected snippet into your document, copy the snippet to the clipboard or create a new document based on the snippet. A file dialog is launched and the snippet is saved to the selected file name. Then it is opened in a new tab.
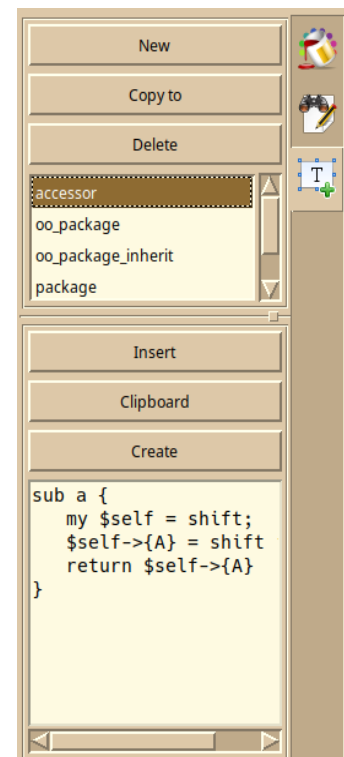
6.9 Console

The Console plugin allows you to run system commands and tests inside Codit. It works a bit as a standard command console. If a command produces errors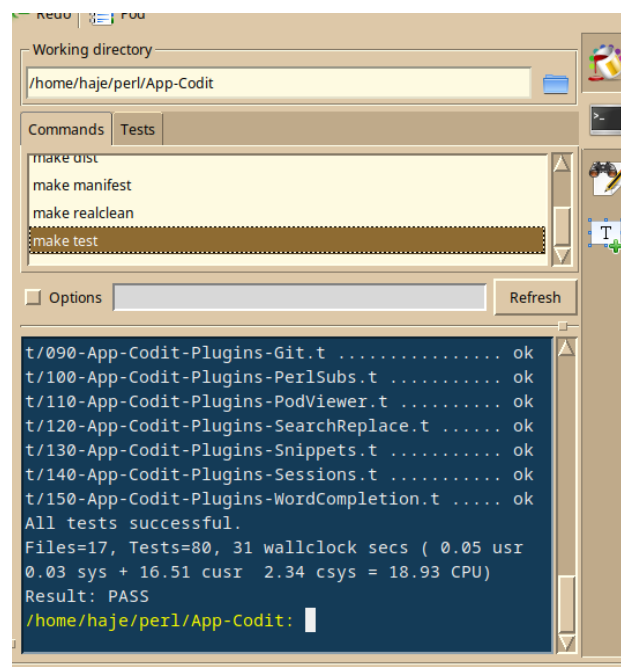, the output is scanned for document names and line numbers. Clickable links are created that bring you directly to the place where the error occured.

The command console has three keybindings:

- <CTRL+U>, Toggle buffering on or off.

- <CTRL-W>, Clear the screen.

- <CTRL+Z>, Kill the currently running process.

This plugin does not work under Windows.

6.10 Writing your own plugin

The plugins system of Codit is part of the Tk::AppWindow application framework. Here are some places to get started:

- The plugins cookbook.

- The plugins base class.

- Plugins base class with background jobs.

6.11 Third party plugins

Other users (you?) may have uploaded their plugin to CPAN. You can always do a search here.

6.12 Plugins under development

In the settings dialog you will find some plugins that are not listed in this document. They are under development and not yet usable. These are:

- Bookmarks

- Git

- WordCompletion

We expect them to be functional in a future version.

7. Keyboard shortcuts

7.1 Handling documents

CTRL+N	Create a new document.
CTRL+O	Open a document.
CTRL+SHIFT+O	Close current document.
CTRL+S	Save current document when modified.
CTRL+L	Save all open documents when modified.

7.2 Navigating text

END	Go to end of current line.
HOME	Go to begin of current line.
CTRL+END	Go to end of current document.
CTRL+HOME	Go to begin of current document.

7.3 Selecting text

CTRL+A	Select all.
SHIFT+DOWN	Increase or decrease selection by one line downward.
SHIFT+END	Increase or decrease from insert cursor position to the end of the line.
SHIFT+HOME	Increase or decrease from insert cursor position to the start of the line.
SHIFT+LEFT	Increase or decrease selection by one character to the left.
SHIFT+PGDOWN	Increase or decrease selection by screen size downward.
SHIFT+PGUP	Increase or decrease selection by screen size upward.
SHIFT+RIGHT	Increase or decrease selection by one character to the right.
SHIFT+UP	Increase or decrease selection by one line upward.

7.4 Modifying text

CTRL+C	Copy selection to clipboard.
CTRL+X	Move selection to clipboard.
CTRL+V	Paste clipboard into document.
CTRL+G	Comment line or selection.
CTRL+SHIFT+G	Uncomment line or selection.

CTRL+J	Indent line or selection.
CTRL+SHIFT+G	Unindent line or selection.
CTRL+Z	Undo.
CTRL+SHIFT+Z	Redo.

7.5 Help, settings and such

CTRL+Q	Quit.
F1	Help.
SHIFT+F1	About.
F9	Settings.

8. Troubleshooting

Just hoping you never need this

8.1 General troubleshooting

If you encounter problems and error messages using *Codit,* here are some general troubleshooting steps:

- Use the *-config* command line option to point to a new, preferably fresh settingsfolder.

- Use the *-noplugins* command line option to launch *Codit* without any plugins loaded.

- Use the *-plugins* command line option to launch *Codit* with only the plugins loaded you specify here.

8.2 No icons

If *Codit* launches without any icons do one or more of the following:

- Check if your icon theme is based on scalable vectors. Install *Icons::LibRSVG if so*. See Chapter 2, Installation.

- Locate where your icons are located on your system and use the *-iconpath* command line option to point there.

- Select an icon library by using the *-icontheme* command line option.

8.3 Session will not load

Sometimes it happens that a session file gets corrupted. You solve it like this:

- Launch the session manager. *Session → Manage sessions.*

- Remove the affected session.

- Rebuild it from scratch.

Sorry, that is all we have to offer.

8.4 Report a bug

If all fails you are welcome to open a ticket here: https://github.com/haje61/App-Codit/issues.

9. License

Codit is published under the same license as Perl itself.

10. Disclaimer

We have done our best to make Codit an excellent Perl/IDE. However, you use it at your own risk and expense. If it breaks, you own both parts.

This document may be incomplete and may contain errors. We apologize.