

Tines Manual

Larry Kollar

This manual was exported from the default Tines tree using Export->XSLT (see extras/export_xslt.rc) and slightly cleaned up.

10 February 2016

1. Introduction to Tines

Tines is an outliner, a planner, an organizer, and a notebook. It is based on the old hnb (hierarchical notebook) software, originally written and maintained by Oyvind Kolas. Larry Kollar (lkollar@gmail.com) currently maintains Tines.

You can use Tines as a straight outliner, editing individual files in OPML, hnb (a simple XML format), or tab-indented text formats.

A more powerful way to use Tines is to keep all your working outlines, project notes, and calendar entries in a single file. If you do not specify a file name when starting Tines, it uses .tines in your home directory.

1.1. Installing Tines

1.1.1. Unix

1.1.1.1. Prepackaged Binaries

At this point, there are no packages for Tines. Prepackaged binaries for the older hnb exist for some operating-systems/distributions, these might not be the latest versions.

- Andras Bali maintains a Debian package of hnb, which can be installed by issuing:
apt-get install hnb
- Patrick Li maintains a Free-BSD ports package of hnb.
- A MacOS X install package is planned, although it will likely be behind the latest version.

1.1.1.2. Compiling from Source

Tines source is available from github. To get the latest source, use the following command at the shell prompt:

```
git clone https://github.com/larrykollar/tines
```

Once you have the source, installing it should be simple:

```
make  
sudo make install
```

Platform-specific workarounds:

SuSE Linux:

SuSE Linux doesn't install ncurses as curses, which ncurses actually does by default if it is allowed to. Changing the CFLAGS line in src/Makefile to -lncurses instead of -lcurses makes hnb compile on SuSE.

OS X / Darwin:

Check Makefile and src/Makefile for OSX-specific commands and options, and make adjustments accordingly.

The preprocessor included with earlier versions of OS X doesn't work properly. Adding `-no-cpp-precomp` to the CFLAGS definition in src/Makefile should make it compile properly.

This does not apply to Snow Leopard or later versions of OSX. It compiles without any problems on Yosemite.

Note that hnb compiles on 64-bit Macs but segfaults at startup because pointers were being cast as int (this probably applies to all 64-bit systems). Tines does not have this issue.

1.2. Starting Tines

When you start Tines without specifying a file name or other options, it automatically loads the default `~/tines` file. This is the best way to run Tines when using it as a planner.

If `~/tines` doesn't exist, Tines loads a starter database with basic information, including this manual.

Press ESC (or F9) to display the Tines menu. You can edit the `~/tinesrc` file to customize the menu.

The full commandline usage for Tines is:

```
tines [options] [filename] [-e 'command' ['command' ['command' ...]]]
```

Available options are:

- `-h` or `--help` show recognized options
- `-a` or `--ascii` specifies a tab-indented ASCII file.
Note: ASCII doesn't recognize or save attributes like to-do status, done etc.
- `-o` or `--opml` specifies an OPML file.
- `-x` or `--xml` specifies a generic XML file. This mode lets you use Tines as a basic XML editor, collapsing all but the portion you are working on.
Note: Like ASCII, generic XML files do not recognize attributes such as to-do items.
- `--hnb` specifies an hnb file. This is the current default; future versions of Tines will use OPML as the default version.
- `--rc=<file>` specifies a configuration file. The default is `~/tinesrc` (Tines creates it if it does not exist).
- `--ui=<type>` specifies the interface. The choices are: `curses` (default), `cli`, or `keygrab` (a test mode that outputs hex values for each keypress; press 'q' to quit).

1.3. Running CLI Commands

Tines has a CLI mode that exposes a great deal of the internal functionality. This can be useful for scripts. For example:

```
tines -e 'go_root' 'export_htmlcss test.html'
```

Tines loads the default file, moves to the start (uppermost node), and exports it with the HTML+CSS filter to a file called test.html.

While running Tines interactively, press F5 to change to CLI mode. This allows you to use advanced commands not exposed by the default menu. Enter a blank line to return to curses mode.

1.4. Quitting Tines

Press ^Q (Ctrl+Q) to quit. Tines asks if you want to save before quitting. Press Y to save, N to quit without saving, or ESC to continue working.

1.5. Importing data

To import the contents of another file:

- Press ^L (Ctrl+L) or choose File -> Import in the menu.
- Choose the desired format. You can choose between OPML, hnb, or tab-indented text formats.
- Enter the relative or full path of the filename you want to import.
- Tines imports the file contents under the current level.

1.6. Exporting data

To export the current file:

- Press ^E (Ctrl+E) or choose File -> Export from the menu.
- Choose the desired format: OPML, HTML, HTML+CSS, hnb, PostScript, or tab-indented text.
- Enter the relative or full path of the filename you want to write to.

Tines 1.10 provides a script in the "extras" directory that exports the selected subtree to a temporary file, then processes it using XSLT. This can be used to export an outline to a working format such as Markdown or Groff. See the extras/README file for setup details.

1.7. Printing

Tines does not support direct printing. You can export to PostScript and send that to your printer queue. You can also use HTML or HTML+CSS to print from a browser.

2. Using Outlines

2.1. Managing entries

2.1.1. Navigating

The arrow keys move among the entries. Up and Down move within the current level. Left moves to the parent entry. Right moves to the first child entries (or creates one).

To jump to an entry on the current level, enter the first few letters of that entry. Press Tab twice to enter the sub-level under the chosen entry.

PageUp and PageDown skip a large amount of entries up and down. Practically, they move you to the top or bottom of the current level.

Home jumps to the root (the first entry) of the tree.

2.1.2. Searching

To begin a search:

- Press ^F (Ctrl+F). Tines prompts you for a search term, then moves to the first entry containing the search term.
- To find the next occurrence of the search term, press Down or N.
- To find the previous occurrence press Up or P.
- To exit search mode at the current entry, press space or Return.
- To exit search mode and return to the node where you began the search, press ESC.

2.1.3. Moving entries

To move the current entry, hold Shift and press the arrow keys. This may not work on all computers or in UTF-8 terminal windows.

To 'grab' the current entry, press ^G (Ctrl+G). In Grab mode, use the arrow keys (without a modifier) to move the entry. Press Space or Return to confirm the move, or ESC to return the entry to its previous position.

To copy an entry, press ^C (Ctrl+C).

To cut an entry, press ^X (Ctrl+X).

To paste an entry below the current location, press ^V (Ctrl+V).

2.1.4. Adding entries

To add a new entry below the current one, press Insert or ^O (Ctrl+O). You will also immediately enter edit mode.

To add a new entry above the current one, press ^B (Ctrl+B).

To add a new entry at the bottom of the current level, type a unique string of characters and press Return.

2.1.5. Deleting entries

To delete the current entry, press Delete. If the entry has children, you will be asked to confirm the deletion.

To cut the current entry, press ^X (Ctrl+X). Tines does not confirm the action, but the data is available on the clipboard and can be pasted.

2.1.6. Indenting and Outdenting entries

Press > to move the current entry and all the following entries of the same level to be children of the preceding entry.

Press < to move the current entry and all the following entries of the same level to follow the parent of the preceding entry.

2.1.7. Expanding and Collapsing entries

To expand an entry to display its sub-entries, press the '+' key. This is a sticky expansion; Tines continues to display the sub-entry when you move away.

To hide expanded entries, press the '-' key.

To expand the entire outline, select File -> Expand All in the menu.

To collapse the entire outline, select File -> Collapse All in the menu.

To expand or collapse all entries under the current entry, select Entry -> Expand Subtree or Entry -> Collapse Subtree.

2.2. Editing entries

To edit the current entry, press Return.

In edit mode, the cursor keys move between the characters of the data in the entry instead among the entries.

The default editing shortcuts are:

- ^A (Ctrl+A) or Home: moves to the beginning of the entry.
- ^E (Ctrl+E) or End: moves to the end of the entry.
- ^W (Ctrl+W) moves forward one word.
- ^B (Ctrl+B) moves back one word.
- ^S (Ctrl+S) splits the entry at the cursor.
- ^J (Ctrl+J) joins the following entry to the current entry. Has no effect if the following entry has sub-entries.
- Return exits edit mode, keeping your changes.
- ESC exits edit mode, canceling your changes.

2.3. Spell-checking

Press F6 to spell-check the current entry. By default, tines uses ispell; you can specify a different spell-checker in ~/.tinesrc.

When the current entry is done:

- Press Space to spell check the next entry.

- Press Return to exit spell-checking and accept the changes.
- Press ESC to exit spell-checking and cancel changes.

2.4. Sorting entries

Use the Level menu to sort entries at the current level.

- Sort is a typical alphanumeric sort.
- Sort (desc.) sorts in reverse order.
- Shuffle randomizes the entries.

2.5. Setting the entry type

Press F3 to change the entry to a text type. A text entry has a "t" before the bullet. This can be used with XSLT stylesheets to transform an hnb or OPML file to a document with headings, body text, and bulleted lists. Press F3 again to revert the entry to a plain type.

Press ^T (Ctrl+T) to change the entry to a todo type. See "Planner Functions" below for more information.

3. Other features

3.1. Planner Functions

Besides its outlining capabilities, Tines has a number of features making it suitable as a planner. The Planner menu provides convenient access to these functions. You can customize and extend the Planner menu by editing ~/.tinesrc.

3.1.1. Todo-list mode

Press ^T (Ctrl+T) to toggle between a checkbox and a bullet for the current entry. When the entry has a checkbox, press ^D (Ctrl+D) to check or uncheck the box.

If an entry has a clear (not checked) box, and sub-entries with checkboxes, Tines shows a percentage of checked sub-entries instead of the empty box.

3.1.2. Inserting Calendar Entries

The Planner menu can insert entries for a specified month, or an entire year. The calendar entries are no different than other entries, besides being automatically inserted.

To insert a monthly calendar, select Planner -> Add Month... from the menu. You are prompted for the month (1-12) and year (four digits).

To insert a yearly calendar, select Planner -> Add Year... from the menu. You are prompted for the year (four digits).

3.1.3. Inserting Lists

The Planner menu also provides pre-formatted Actions, Waiting For, and Project lists. These types of lists are based on the 'Getting Things Done' (GTD) methodology, conceived by David Allen.

If you prefer a different methodology, you can modify the Planner menu (by editing `~/tinesrc`) as you see fit.

3.2. Using URLs

If an entry contains an URL, press `^A` (Ctrl+A) to open that URL in a browser.

The default browser is lynx. The `~/tinesrc` file has a list of supported browsers; you can select any of them by commenting out the current selected 'web_command' line and uncommenting another.

3.3. Using Email addresses

If an entry contains an email address, press `^A` (Ctrl+A) to open the configured mail client.

The default mail client is mutt. The `~/tinesrc` file has a short list of supported clients; you can select any of them by commenting out the current selected 'mail_command' line and uncommenting another.

4. Configuring Tines

When you start Tines for the first time, it creates a file named `.tinesrc` in your home directory. This file controls the default configuration. All the commandline functions are available for use in the configuration file.

For safety, make a copy of the configuration file and edit it as you see fit. If your copy is called `test.tines`, you can load it instead of the default configuration with the following command:

```
tines --rc=test.tines
```

If you edited your `.tinesrc`, and caused problems you cannot fix, delete or rename `.tinesrc`. Next time you start Tines, it will reload a default configuration.

You can edit the configuration file in any text editor. The file is tab-indented, so you can use Tines to edit it with the following command:

```
tines -a ~/tinesrc
```

Tines collapses each section; you can expand them as needed to focus only on the part you want.

4.1. Context

Contexts provide a convenient way to change keybindings, depending on what you are doing at any given moment. Tines provides four default contexts:

- main
- nodeedit
- lineedit
- confirm

You can define up to 12 custom contexts as well. To define or switch to a context, use the following command:

```
context <name>
```

4.2. Macros

Tines supports macros. Macros are named collections of CLI commands that can be executed as needed. See the 'Commandline Reference' below for a list of Tines CLI commands.

To define a macro, use the following sequence of commands:

```
macro define macro_name  
cmd  
cmd...  
end
```

To execute a macro, use 'macro macro_name'

4.3. Key Bindings

Key bindings allow you to assign one or more actions to a keypress in a given context. Valid keys include:

- Any single character, including punctuation and similar characters. Case is significant; a and A can be bound to two different functions.
- The sequence '^X' to indicate a control character (Ctrl+X in this example). Note that Ctrl+H is named backspace2, and Ctrl+M is named return.
- Function keys f1 through f12
- Arrow keys: up, down, left, right. With Shift modifier: sleft, sright, sup, sdown.
- Navigation and editing keys: npage (page down), ppage (page up), home, end, return, enter, ins (insert), del (delete), backspace (0x7f), backspace2 (0x08), tab, esc, space
- The special key 'any' to indicate any unbound key. This can be used to set up a default action. Its usual purpose is to return an error message using 'bind any unbound'.

To set up a key binding, use the 'bind' command. Arguments are the key and the action. Substitute '..' for the key if you want to bind more actions to a single keypress. For example, the following key binding substitutes for an 'insert_above' command and is used in the default .tinesrc configuration:

```
bind ^B insert_below  
bind .. "movenode up"
```

bind .. edit

Note that you have to enclose a command in quotes if it has an argument.

4.4. Helper Commands

These commands specify external utilities to invoke when the "action" command detects certain patterns in the current entry.

4.4.1. web_command <command>

Runs the specified command in response to an http:// action. This usually opens a web browser, but could be used with curl or wget to save a copy instead. Use the * character to substitute the URL.

4.4.2. mail_command <command>

Runs the specified command in response to an "@" action. This usually opens a mail client, but could be used to trigger other actions. Use the * character to substitute the email address.

4.4.3. spell_command <command>

Runs the specified command in response to the "spell" command. This usually opens a spell checker.

4.4.4. Related commands

The "system" command opens a shell and runs its argument as its command line. While it has no effect on the current database, it can be used to process an exported file.

If the "action" command runs on an entry that begins with "exec," Tines runs the rest of the line as a shell command.

5. Commandline Reference

Tines includes an extensive set of CLI commands.

The commands listed are sorted by function and the file name where they are defined.

To set a variable, use the command <variable> <value>.

5.1. Actions (actions.c)

5.1.1. system <command_string>

Opens a shell and runs the specified command.

5.1.2. action

Executes an external shell according to node content. It first checks the direct children, then the current node.

If it contains an http:// URL, the command described in browser_command is executed.

If it contains an '@' character, the characters surrounding it are interpreted as an email address and the mail_command is executed.

If the data starts with 'exec' the rest of the data is executed verbatim.

5.1.3. web_command <command_string>

The command executed when opening URLs. The detected URL replaces the * character.

5.1.4. mail_command <command_string>

The command executed when sending email. The address replaces the * character.

5.1.5. address (string)

The email address used as the parameter for mail_command.

5.2. Autosave (autosave.c)

5.2.1. tree_changed [increment]

An internal command, used to drive the autosave functionality. For large changes, pass a high number (around 1000) to make sure the changes are saved immediately. Otherwise, omit the increment.

5.2.2. autosave_check_timeout

Checks the timeout counter and autosaves if necessary.

5.2.3. autosave_timeout (integer)

The number of ticks before autosaving after change.

5.2.4. autosave_sync (integer)

A flag that determines whether the filesystem should be synced after autosave.

5.2.5. autosave_timer (integer)

The number of ticks since an unsaved change.

5.2.6. autosave_threshold (integer)

Autosave for every autosave_threshold nodes changed.

5.2.7. autosave_threshold_nodes_changed (integer)

The number of nodes changed since the last save.

5.3. Calendar (cal.c)

5.3.1. insert_cal <month> <year>

Inserts a monthly calendar below the current selection.

- <month>: 1-12
- <year>: the full (4-digit) year.

5.4. Macros (cli_macro.c)

5.4.1. macro [define] <cmd>

Defines or executes a macro.

When defining a macro, the commands following are part of the macro until reaching an 'end' command.

5.5. Clipboard (clipboard.c)

5.5.1. copy

Stores the current node and its sub-entries in the clipboard.

5.5.2. cut

Moves the current node and its subtree to the clipboard.

5.5.3. paste

Inserts the contents of the clipboard at the current position in the tree.

5.6. Event control (evilloop.c)

5.6.1. quit, q

Quits tines without prompting to save.

5.6.2. forced_up (integer)

If true, attempting to move beyond the first sibling moves to the parent entry.

5.6.3. forced_down (integer)

If true, attempting to move beyond the last sibling moves to the entry below the parent entry.

5.6.4. add_at_top (integer)

If true, items inserted automatically are added at the top (default 0, at the bottom).

5.7. Expand/collapse (expanded.c)

5.7.1. expand [--all|-a|--here|-h|--subtree]

Expands the current entry to show its sub-entries.

- Use the -a or --all option to expand all nodes in the tree.
- Use the -h or --here option to expand all descendants of the current node.
- Use the --subtree option to recursively expand all descendants of the current entry.

5.7.2. collapse [-a]

Collapse the current entry's sub-entries.

- Use the -a option to collapse all entries in the tree.
- Use the --subtree option to recursively collapse all descendants of the current entry.

5.8. File commands (file.c)

5.8.1. save

Saves the data.

5.8.2. revert

Reverts to the last saved version.

5.9. Import/export text (file_ascii.c)

5.9.1. export_ascii <filename>

Exports the tree as a tab-indented text file.

5.9.2. import_ascii <filename>

Imports a tab-indented text file below the current node.

5.9.3. `ascii_margin` (integer)

(present but not implemented) The margin that text export wraps at. Set to -1 to disable wrapping.

5.10. Import/export binary (file_binary.c)

These are internal commands used for autosave snapshots.

5.10.1. `export_binary` <filename>

Exports the tree as a binary file.

5.10.2. `import_binary` <filename>

Imports a binary file below the current node.

5.11. Import/export help (file_help.c)

These commands are used to generate embeddable text files for including into source. These functions will soon be deprecated as Tines will pull files from /usr/local/share in the near future.

5.11.1. `export_help` <filename>

Exports to help source format.

5.11.2. `import_help` <filename>

Imports help source format into the tree.

5.12. Import/export hnb (file_hnb.c)

5.12.1. `export_hnb` <filename>

Exports the tree in hnb format.

5.12.2. `import_hnb` <filename>

Imports an hnb file below the current node.

5.13. Import/export OPML (file_opml.c)

5.13.1. `export_opml` <filename>

Exports the tree as OPML.

5.13.2. import_opml <filename>

Imports an OPML file below the current node.

5.14. Import/export XML (file_xml.c)

5.14.1. import_xml <filename>

Imports an XML file below the current node.

5.14.2. export_xml <filename>

Exports the current node, its siblings, and all sublevels to 'filename' as XML markup.

Load an XML file with import_xml or tines -x file.xml to see how it looks inside tines.

5.14.3. xml_cuddle (integer)

If true, joins the data with nodes if there are no tags within the tag.

5.15. Export HTML (file_html.c)

5.15.1. export_html <filename>

Exports the tree as basic HTML.

5.15.2. export_htmlcss <filename>

Exports the tree as HTML with embedded CSS.

5.16. Export PostScript (file_ps.c)

5.16.1. export_ps <filename>

Exports the current entry, its siblings, and all sub-entries to a PostScript file suitable for printing.

5.17. Marks (mark.c)

5.17.1. set_mark <mark number>

Defines a location for use with 'go_mark'. The <mark_number> can be 0 through 9.

5.17.2. go_mark <mark number>

Moves to a location saved with 'set_mark'.

5.18. Attributes (node.c)

5.18.1. att_set <attribute> <value>

Sets an attribute for the current entry.

5.18.2. att_get <attribute>

Returns the specified attribute value for the current entry.

5.18.3. att_clear

Removes the specified attribute.

5.18.4. att_list

Lists the text and attributes for the current entry.

5.19. Prefs (prefs.c)

5.19.1. format (string)

The format of the current file.

5.19.2. rc_file (string)

The path and name of the rc file in use (the default is ~/.tinesrc).

5.19.3. db_file (string)

The path and name of the file tines is using (the default is ~/.tines).

5.19.4. default_db_file (string)

The path and file name of the default file tines loads if no other file is available on the command line.

5.19.5. showpercent (integer)

If true, to-do entries with child to-do entries show a percentage of completed sub-entries.

5.19.6. fixedfocus (integer)

If true, moves the tree up and down around the selection bar, which remains about 1/3 of the way down the screen.

5.19.7. savepos (integer)

If true, saves the current position along with the data. Supported in both hnb and OPML formats.

On save, Tines inserts a `<?pos="x"?>` processing instruction, where x is the current position. When reading the file, Tines sets the current position to the value found in the processing instruction.

5.19.8. readonly (integer)

If true, tines does not save any changes.

5.19.9. escdelay (integer)

Time, in milliseconds, Tines waits before assuming an ESC keypress is meant as an escape rather than part of a command sequence. The default is the value of the curses ESCDELAY variable.

5.20. Queries (query.c)

5.20.1. getquery <prompt>

Inputs a string from the user and puts it into the variable query.

The search function, and some of the export/import functions can use queries.

5.20.2. query (string)

The last query. May also be set as a variable.

5.21. Search (search.c)

5.21.1. search [-b|-f] <string>

Searches -b(ackward) or -f(oward) for the next entry with a substring matching the specified string.

5.22. Spellcheck (spell.c)

5.22.1. spell [-r]

Spellchecks the current node, or all children and following siblings recursively (if -r specified).

This function uses the command defined in 'spell_command'.

5.22.2. spell_command (string)

The command executed when spell checking a node.

Tines writes the entries to a temporary file.

5.23. Statistics (statcmd.c)

5.23.1. mem

Reports the memory usage of the application.

5.23.2. stats

Reports the number of items, leaf items and total number of words in the tree.

5.24. Current state (state.c)

5.24.1. save_state

Saves a temporary copy of the current tree and selected node in memory.

5.24.2. restore_state

Restores the copy of the tree saved with save_state.

5.25. Tree (tree_misc.c)

5.25.1. movenode <up|left|right|down>

Reposition the current node.

5.25.2. go <up|down|left|right|recurse|backrecurse|root|top|bottom>

Changes the current position in the tree:

- up/down: moves up or down in the current level.
- left: moves to the parent node.
- right: moves to the first child node. If no child node exists, tines creates one.
- root: moves to the top of the tree.
- top/bottom: moves to the first or last node.
- recurse: moves to the next node in the current level.
- backrecurse: moves to the first node in the current level.

5.25.3. outdent

Moves the active item and the following siblings one level to the left.

5.25.4. indent

Moves the active item and the following siblings one level to the right.

5.25.5. remove

Removes the active node. If the node has children, you have a chance to confirm the deletion.

5.25.6. commandline

Invokes the interactive command line in curses mode.

5.25.7. insert_below

Adds a new entry immediately below the active entry.

5.25.8. sort [-r|-a|-d]

Sorts the siblings of the currently selected node:

- -a=ascending
- -d=descending
- -r=random (shuffle)

5.25.9. toggle_todo

Toggles the visibility and usage of the checkbox for the current entry.

5.25.10. toggle_done

Toggles the checkbox for the currently active entry, if the entry has a checkbox.

5.25.11. toggle_text

Toggles the text marker for the current entry. (Introduced in 1.9.20.)

5.25.12. bind <key> <action> [parameter]

Binds the action to the specified key.

Use the virtual key '.' to bind subsequent actions; this allows binding multiple actions to a single keystroke.

The virtual key 'any' indicates any unbound keys. This can be used to produce an error message using 'bind any unbound'.

5.25.13. unbound

Outputs a message, indicating that the pressed key is not bound in the current context.

5.25.14. context <context>

Changes the current context.

Predefined contexts include 'main', 'lineedit', 'nodeedit', and 'confirm'. You can define other contexts as needed.

5.25.15. backspace

Erases the preceding character.

5.25.16. delete

Erases the following character.

5.25.17. left

Moves the cursor or selection left.

5.25.18. right

Moves the cursor or selection right.

5.25.19. up

Moves the cursor or selection up.

5.25.20. down

Moves the cursor or selection down.

5.25.21. pagedown

Moves down one page.

5.25.22. pageup

Moves up one page.

5.25.23. bkill_word

Erases the preceding word.

5.25.24. kill_word

Erases the following word.

5.25.25. bol

Moves the cursor to the beginning of line.

5.25.26. eol

Moves to the end of the line.

5.25.27. top

Moves to the topmost sibling of the current entry.

5.25.28. bottom

Moves to the bottommost sibling of the current entry.

5.25.29. skipword

Moves to the next word.

5.25.30. bskipword

Moves to the previous word.

5.25.31. cancel

Cancels editing of the node.

5.25.32. complete

Outputs the possible completions of the entered command or variable.

5.25.33. confirm

Confirms the question posed, (used in dialogs asking for the users' consent).

5.25.34. ignore

Ignores a pressed key without giving an error message.

5.25.35. join

Joins the currently edited node with the following sibling.

5.25.36. split

Splits the currently edited node at the cursor location.

5.25.37. kill_line

Erases the currently edited line.

5.25.38. kill_to_bol

Erases from the cursor to the beginning of the line.

5.25.39. kill_to_eol

Erases from the cursor to the end of the line.

5.26. Bindings (ui_binding.c)

Editing bindings and contexts.

5.26.1. Configuration commands

5.26.1.1. **bind** <key> <action> [parameter]

Creates a keybinding in the current context. The virtual key `..` adds actions to the last-defined key, allowing a single key to perform multiple actions. Each context may have up to 255 key bindings.

The virtual key `'any'` indicates any unbound keys. This can be used to produce an error message using `'bind any unbound'`.

5.26.1.2. **unbound**

Displays an error message for keys not bound to another action.

5.26.1.3. **ignore**

Ignores the current input. Can be used to bind a key you want silently ignored.

5.26.1.4. **confirm** <text>

Confirms the action.

5.26.1.5. **context** <name>

Switches to the named context. The following contexts are predefined:

- `main`: the primary context for navigating through the tree.
- `confirm`: used to confirm a user action (such as deleting).
- `nodeedit`: used for editing a node.
- `lineedit`: not currently used.

You can define up to 12 custom contexts. The default `.tinesrc` configuration defines several contexts.

5.26.1.6. **command** <param>

(executed in `evilloop.c`) Executes <param> as a CLI command.

5.26.1.7. **quit**

(executed in `evilloop.c`) Quits without saving.

5.26.2. Movement commands (entry-level)

5.26.2.1. **up**

Moves the cursor or selection up.

5.26.2.2. **down**

Moves the cursor or selection down.

5.26.2.3. left

Moves the cursor or selection left.

5.26.2.4. right

Moves the cursor or selection right.

5.26.2.5. top

Moves to the topmost sibling of the current entry.

5.26.2.6. bottom

Moves to the last sibling of the current entry.

5.26.2.7. pageup

Moves up a page of entries.

5.26.2.8. pagedown

Moves down a page of entries.

5.26.3. Movement commands (edit-level)

5.26.3.1. bol

Moves to the beginning of the entry.

5.26.3.2. eol

Moves to the end of the entry.

5.26.3.3. skipword

Moves to the beginning of the next word.

5.26.3.4. bskipword

Moves the cursor to the beginning of the first word to the left of the cursor.

5.26.4. Editing commands

5.26.4.1. backspace

Deletes the character to the left of the cursor.

5.26.4.2. delete

Deletes the character under the cursor.

5.26.4.3. kill_word

Deletes the word under the cursor.

5.26.4.4. bkill_word

Deletes the word to the left of the cursor.

5.26.4.5. kill_to_bol

Deletes all the text from the cursor to the beginning of the entry.

5.26.4.6. kill_to_eol

Deletes all the text from the cursor to the end of the entry.

5.26.4.7. kill_line

Deletes the line under the cursor.

5.26.4.8. cancel

Cancels edit mode without saving changes.

5.26.4.9. complete

Outputs possible completions of the partially-entered command or variable name.

5.26.4.10. join

Combines the subsequent entry with the following entry.

5.26.4.11. split

Splits the entry at the cursor position.

5.27. CLI Commands (ui_cli.c, libcli/cli.c)

5.27.1. add <string>

Inserts a new entry at the end of the current level, containing the specified string.

Note: The position does not move. To operate on the new entry, use 'go bottom' to move to the entry.

5.27.2. ls

Lists the nodes at the current level.

5.27.3. addc <parent> <string>

Inserts a new entry under the node named <parent>, with the text specified in <string>.

5.27.4. cd <path>

Moves to a different part of the tree.

5.27.5. pwd

Echoes the current path.

5.27.6. show_vars

Lists all defined variables and their values.

5.27.7. temp_str (string)

A temporary string variable for use in macros.

5.28. Node formatting (ui_draw.c)

5.28.1. display_format <format string>

Defines how each node is displayed. The format string syntax is:

- spaces: literal spaces
- i<width>: indents the specified width in characters
- -: inserts a bullet
- d: inserts the actual node data
- x<width>: reserved

A typical format string is 'i4- d'.

5.28.2. bullet_leaf <string>

The format for entries with no sub-entries.

Example: ' *'

5.28.3. bullet_parent <string>

The format for entries with unexpanded sub-entries.

Example: ' +'

5.28.4. bullet_parent_expanded <string>

The format for entries with expanded sub-entries.

Example: ' -'

5.28.5. bullet_todo_parent_undone <string>

The format for unchecked to-do entries with sub-entries.

Example: '[]'

5.28.6. bullet_todo_parent_done <string>

The format for checked to-do entries with sub-entries.

Example: '[X]'

5.28.7. bullet_todo_leaf_undone <string>

The format for unchecked to-do entries with no sub-entries.

Example: '[]'

5.28.8. bullet_todo_leaf_done <string>

The format for checked to-do entries with no sub-entries.

Example: '[X]'

5.28.9. bullet_textleaf <string>

The format for text entries with no sub-entries.

Example: 't *'

5.28.10. bullet_textparent <string>

The format for text entries with collapsed sub-entries.

Example: 't +'

5.28.11. bullet_textparent_expanded <string>

The format for text entries with expanded sub-entries.

Example: 't -'

5.29. Node editor (ui_edit.c)

5.29.1. edit

Invokes the node editor for the current node.

5.30. Menu definition (ui_menu.c)

5.30.1. menu <menu> <text> <cmd>

Defines a menu entry.

- <menu> specifies the menu name.
- <text> specifies the entry text.

- `<cmd>` specifies the associated command, macro, or context.

5.31. Messages (ui_overlay.c)

5.31.1. helptext <help for context>

Defines the help text for the current context.

The character `|` alternates between the menuitem and the menutext styles.

`||` is the escape sequence for a single pipe.

5.31.2. status [-c|--clear] <message>

Adds 'message' as the newest status line.

Use `-c` or `--clear` to clear all pending status messages off the screen.

BUG: Currently, specifying `-c` or `--clear` omits the message.

5.31.3. echo [-c|--clear] <message>

Alias for status.

5.32. Style (ui_style.c)

5.32.1. style <element> <fgcolor/bgcolor> <attributes>

Changes the drawing style of `<element>`. Available elements are: menuitem, menutext, parentnode, node, bullet, selected, parentsselected and background.