

Wyrd v1.4 User Manual

Paul J. Pelzl

October 23, 2010

“Because you’re tired of waiting for your bloated calendar program to start up.”

Contents

1	Introduction	3
2	Installation	3
3	Quick Start	4
3.1	Overview	4
3.2	Navigation	4
3.3	Editing Reminders	5
3.4	Quick Reminders	5
3.5	Cutting and Pasting Reminders	6
3.6	Viewing Reminders	6
3.7	Searching for Reminders	7
3.8	Other Commands	7
3.9	Alarm Strategies	7
3.10	Miscellaneous	7
4	Advanced Configuration	8
4.1	wyrdrc Syntax	8
4.1.1	Including Other Rcfiles	8
4.1.2	Setting Configuration Variables	8
4.1.3	Creating Key Bindings	8
4.1.4	Removing Key Bindings	9
4.1.5	Setting the Color Scheme	9
4.2	Configuration Variables	9
4.3	Calendar Operations	12
4.4	Colorable Objects	15
5	Usage Tips	16
6	Licensing	16
7	Acknowledgments	16

8	Contact info	16
9	Miscellaneous	17

1 Introduction

Wyrd is a text-based front-end to Remind, a sophisticated calendar and alarm program available from Roaring Penguin Software, Inc.¹ Wyrd serves two purposes:

1. It displays reminders in a scrollable timetable view suitable for visualizing your calendar at a glance.
2. It makes creating and editing reminders fast and easy. However, Wyrd does not hide Remind's textfile programmability, for this is what makes Remind a truly powerful calendaring system.

Wyrd also requires only a fraction of the resources of most calendar programs available today.

2 Installation

This section describes how to install Wyrd by compiling from source. Wyrd has been packaged for a number of popular Linux/Unix variants, so you may be able to save yourself some time by installing from a package provided by your OS distribution.

Wyrd is designed to be portable to most Unix-like operating systems, including GNU/Linux, *BSD, and Mac OS X. Before installing Wyrd, your system must have the following software installed:

- OCaml² ≥ 3.08
- the ncurses library³ (and development headers)
- Remind⁴ $\geq 3.1.0$
- GNU make⁵
- the Unix pager application, `less`

Wyrd may be compiled by executing the following at the root of the source tree:

```
./configure  
make
```

After compiling, become root and execute

```
make install
```

to complete the installation process. The `make` command here should correspond to GNU make; on some systems (particularly *BSD), you may need to use `gmake`.

If your ncurses library was built with wide character support, Wyrd can be configured to render UTF-8 encoded reminders. To enable this option, use the command

```
./configure --enable-utf8
```

when configuring the sources.

¹http://www.roaringpenguin.com/penguin/open_source_remind.php

²<http://caml.inria.fr>

³<http://www.gnu.org/software/ncurses/ncurses.html>

⁴http://www.roaringpenguin.com/penguin/open_source_remind.php

⁵<http://www.gnu.org/software/make/>

3 Quick Start

This section describes how to use Wyrd in its default configuration. After familiarizing yourself with the basic operations as outlined in this section, you may wish to consult Section 4 to see how Wyrd can be configured to better fit your needs.

3.1 Overview

Before attempting to use Wyrd, learn how to use Remind. Wyrd makes no attempt to hide the details of Remind programming from the user. Aside from reading the Remind manpage, you may get some useful pointers by reading Mike Harris's article on 43 Folders⁶ or David Skoll's writeup on Linux Journal⁷. The 43 Folders Wiki also has a nice section on Remind⁸.

You can launch Wyrd using the default reminder file by executing `wyrd`. If desired, a different reminder file (or reminder directory) may be selected by executing `wyrd <filename>`.

At the top of the window is a short (incomplete) list of keybindings.

The left window displays a scrollable timetable view, with reminders highlighted in various colors. If the `DURATION` specifier is used for a reminder, the highlighted area is rendered with an appropriate size. Overlapping reminders are rendered using one of four different indentation levels so that all reminders are at least partially visible. If the current time is visible in this window, it is highlighted in red.

The upper right window displays a month calendar, with the color of each day representing the number of reminders it contains. The colors range across shades of white to blue to magenta as the number of reminders increases. The selected date is highlighted in cyan; if the current date is visible, it is highlighted in red.

The lower right window displays a list of the untimed reminders falling on the selected date.

The bottom window displays the full text of the `MSG` for the reminder or reminders that are currently selected.

3.2 Navigation

Action	Keypress
scroll up and down the schedule	<up>, <down> or k, j
jump back or forward by a day	<pageup>, <pagedown> or 4, 6 or <, > or H, L
jump back or forward by a week	8, 2 or [,] or K, J
jump back or forward by a month	{, }
jump to current date and time	<home>
jump to the next reminder	<tab>
switch between schedule and untimed reminders window	<left>, <right> or h, l
zoom in on the schedule	z
scroll the description window up and down	d, D

Notice that if you have a numeric keypad, the {4, 6, 8, 2} keys will let you move directionally in

⁶http://www.43folders.com/2005/02/guest_mike_harr.html

⁷<http://www.linuxjournal.com/article/3529>

⁸<http://wiki.43folders.com/index.php/Remind>

the month calendar view at the upper-right of the screen. Similarly, {H, J, K, L} will cause directional calendar movement using the standard mapping from `vi(1)`.

In addition to the hotkeys provided above, Wyrd lets you jump immediately to a desired date by pressing 'g', entering in a date specifier, and then pressing <return>. Any of the following date specifiers may be used:

- 8 digits representing year, month, and day: YYYYMMDD
- 4 digits representing month and day (of current year): MMDD
- 2 digits representing day (of current month and year): DD

(The date specifier format may be changed to DDMMYYYY; consult Section 4.2.)

3.3 Editing Reminders

Note: By default, Wyrd is configured to modify your reminder files using the text editor specified by the `$EDITOR` environment variable. (This configuration has been tested successfully with a number of common settings for `$EDITOR`, including 'vim', 'emacs', and 'nano'.) If you wish to use a different editor, see Section 4.

If you select a timeslot in the schedule view, then hit 't', you will begin creating a new timed reminder. Wyrd will open up your reminder file in your favorite editor and move the cursor to the end of the file, where a new reminder template has been created. The template has the selected date and time filled in, so in many cases you will only need to fill in a MSG value.

Similarly, hitting 'u' will begin creating an untimed reminder. 'w' will create a weekly timed reminder, and 'W' will create a weekly untimed reminder; 'm' will create a monthly timed reminder, and 'M' will create a monthly untimed reminder.

'T' and 'U' also create timed and untimed reminders (respectively), but first will provide a selection dialog for you to choose which reminder file you want to add this reminder to. The set of reminder files is determined by scanning the `INCLUDE` lines in your default reminder file. (If you use a reminder directory, then all *.rem files in that directory will be available along with all `INCLUDED` files.)

If you select a reminder (either timed or untimed) and hit <return>, you will begin editing that reminder. Wyrd will open up the appropriate reminders file in your editor and move the cursor to the corresponding REM line.

If you select a timeslot that contains multiple overlapping reminders, Wyrd will provide a dialog that allows you to select the desired reminder.

If you hit <enter> on a blank timeslot, Wyrd will begin creating a new timed or untimed reminder (depending on whether the timed or the untimed window is selected).

Finally, pressing 'e' will open the reminder file in your editor without attempting to select any particular reminder.

3.4 Quick Reminders

Wyrd offers an additional mode for entering simple reminders quickly. Press 'q', and you will be prompted for an event description. Simply enter a description for the event using natural language, then press <return>. Examples:

- meeting with Bob tomorrow at 11

- drop off package at 3pm
- wednesday 10am-11:30 go grocery shopping
- Board game night 20:15 next Fri
- 7/4 independence day
- 7/4/2007 independence day (next year)
- independence day (next year) on 2007-07-04

If your event description can be understood, Wyrd will immediately create the reminder and scroll the display to its location.

Currently the quick reminder mode tends to favor USA English conventions, as generalizing the natural language parser would require some work.

Wyrd also allows you to use the "quick reminder" syntax to create new reminders from the command line, using the `-a` or `--add` options. For example,

```
wyrd --add "dinner with neighbors tomorrow at 7pm"
```

would create a new reminder for tomorrow evening. When used in this mode, Wyrd exits silently with error code 0 if the reminder was successfully created. If the reminder could not be created (e.g. if the reminder syntax could not be parsed), Wyrd prints an error message and exits with a nonzero error code.

3.5 Cutting and Pasting Reminders

Reminders can be easily duplicated or rescheduled through the use of Wyrd's cutting and pasting features.

Selecting a reminder and pressing `'X'` will cut that reminder: the corresponding `REM` line is deleted from your reminders file, and the reminder is copied to Wyrd's clipboard. To copy a reminder without deleting it, use `'Y'` instead.

To paste a reminder from the clipboard back into your schedule, just move the cursor to the desired date/time and press `'P'`. Wyrd will append a new `REM` line to the end of your reminders file, and open the file with your editor. The `REM` line will be configured to trigger on the selected date. If the copied reminder was timed, then the pasted reminder will be set to trigger at the selected time using the original `DURATION` setting. (Additional Remind settings such as `delta` and `tdelta` are not preserved by copy-and-paste.)

If you wish to paste a reminder into a non-default reminders file, use `'P'`. This will spawn a selection dialog where you can choose the file that will hold the new reminder.

WARNING: Cutting a reminder will delete only the single `REM` command responsible for triggering it. If you are using more complicated Remind scripting techniques to generate a particular reminder, then the `cut` operation may not do what you want.

3.6 Viewing Reminders

Aside from viewing reminders as they fall in the schedule, you can press `'r'` to view all reminders triggered on the selected date in a `less(1)` window. Similarly, `'R'` will view all reminders triggered on or after the selected date (all non-expired reminders are triggered).

If you want to get a more global view of your schedule, Wyrd will also let you view Remind's formatted calendar output in a `less(1)` window. Pressing `'c'` will view a one-week calendar that contains the selected date, while pressing `'C'` will view a one-month calendar containing the selected date.

3.7 Searching for Reminders

Wyrd allows you to search for reminders with MSG values that match a search string. Press `'/'` to start entering a (case insensitive) regular expression. After the expression has been entered, press `<return>` and Wyrd will locate the next reminder that matches the regexp. Press `'n'` to repeat the same search. Entry of a search string may be cancelled with `<esc>`.

The regular expression syntax is Emacs-compatible.

Note: Sorry, there is no "search backward" function. The search function requires the use of `"remind -n"`, which operates only forward in time. For the same reason, there is a command to jump forward to the next reminder, but no command to jump backward to the previous reminder.

3.8 Other Commands

A list of all keybindings may be viewed by pressing `'?'`. You can exit Wyrd by pressing `'Q'`. If the screen is corrupted for some reason, hit `'Ctrl-L'` to refresh the display.

3.9 Alarm Strategies

You may wish to generate some sort of alarm when a reminder is triggered. Wyrd does not offer any special alarm functionality, because Remind can handle the job already. Check the Remind manpage and consider how the `-k` option could be used to generate alarms with the aid of external programs. For example, the following command will generate a popup window using `gxmessage(1)` whenever a timed reminder is triggered:

```
remind -z -k'gxmessage -title "reminder" %s &' ~/.reminders &
```

(A sensible way to start this alarm command is to place it in `~.xinitrc` so that it launches when the X server is started.) If you want some advance warning (say, 15 minutes), you can cause Remind to trigger early by setting a `tdelta` in the AT clause:

```
REM Nov 27 2005 AT 14:30 +15 MSG Do something
```

Alternatively, if you want to generate alarms only for specific reminders, consider using Remind's `RUN` command. This process could be easily automated by using the `templateN` configuration variables described in Section 4.

3.10 Miscellaneous

Remind's TAG specifier may be used to cause Wyrd to give special treatment to certain reminders. If a reminder line includes the clause `"TAG noweight"`, then Wyrd will not give that reminder any weight when determining the "busy level" colorations applied to the month calendar. If a reminder line includes the clause `"TAG nodisplay"`, then Wyrd will neither display that reminder nor give it any weight when determining the month calendar colorations. The tag parameters are case insensitive.

WARNING: These tag parameters are not guaranteed to interact well with other Remind front-ends such as `tkremind`.

4 Advanced Configuration

Wyrd reads a run-configuration textfile (generally `/etc/wyrdrc` or `/usr/local/etc/wyrdrc`) to determine key bindings, color schemes, and many other settings. You can create a personalized configuration file in `$HOME/.wyrdrc`, and select settings that match your usage patterns. The recommended procedure is to “include” the `wyrdrc` file provided with Wyrd (see Section 4.1.1), and add or remove settings as desired.

4.1 `wyrdrc` Syntax

You may notice that the `wyrdrc` syntax is similar to the syntax used in the configuration file for the Mutt email client (`muttrc`).

Within the `wyrdrc` file, strings should be enclosed in double quotes (`"`). A double quote character inside a string may be represented by `\"`. The backslash character must be represented by doubling it (`\\`).

4.1.1 Including Other Rfiles

Syntax: `include filename_string`

This syntax can be used to include one run-configuration file within another. This command could be used to load the default `wyrdrc` file (probably found in `/etc/wyrdrc` or `/usr/local/etc/wyrdrc`) within your personalized rfile, `~/.wyrdrc`. The filename string should be enclosed in quotes.

4.1.2 Setting Configuration Variables

Syntax: `set variable=value_string`

A number of configuration variables can be set using this syntax; check Section 4.2 to see a list. The variables are unquoted, but the values should be quoted strings.

4.1.3 Creating Key Bindings

Syntax: `bind key_identifier operation`

This command will bind a keypress to execute a calendar operation. The various operations, which should not be enclosed in quotes, may be found in Section 4.3. Key identifiers may be specified by strings that represent a single keypress, for example `"m"` (quotes included). The key may be prefixed with `"\C"` or `"\M"` to represent Control or Meta (Alt) modifiers, respectively; note that the backslash must be doubled. A number of special keys lack single-character representations, so the following strings may be used to represent them:

- `"<esc>"`
- `"<tab>"`
- `"<enter>"`
- `"<return>"`

- "<insert>"
- "<home>"
- "<end>"
- "<pageup>"
- "<pagedown>"
- "<space>"
- "<left>"
- "<right>"
- "<up>"
- "<down>"
- "<f1>" to "<f12>"

Due to differences between various terminal emulators, this key identifier syntax may not be adequate to describe every keypress. As a workaround, Wyrd will also accept key identifiers in octal notation. As an example, you could use `\024` (do *not* enclose it in quotes) to represent Ctrl-T.

Multiple keys may be bound to the same operation, if desired.

4.1.4 Removing Key Bindings

Syntax: `unbind key_identifier`

This command will remove all bindings associated with the key identifier. The key identifiers should be defined using the syntax described in the previous section.

4.1.5 Setting the Color Scheme

Syntax: `color object foreground background`

This command will apply the specified foreground and background colors to the appropriate object. A list of colorable objects is provided in Section 4.4. Wyrd will recognize the following color keywords: `black`, `red`, `green`, `yellow`, `blue`, `magenta`, `cyan`, `white`, `default`. The `default` keyword allows you to choose the default foreground or background colors. If you use `default` for your background color, this will access the transparent background on terminal emulators which support it.

4.2 Configuration Variables

The following configuration variables may be set as described in Section 4.1.2:

- `remind_command`
Determines the command used to execute Remind.

- `reminders_file`
Controls which Remind file (or Remind directory) Wyrd will operate on. The default is `~/reminders`.
- `edit_old_command`
Controls the command used to edit a pre-existing reminder. The special strings `'%file%'` and `'%line%'` will be replaced with a filename to edit and a line number to navigate to within that file.
- `edit_new_command`
Controls the command used to edit a new reminder. The special character `'%file%'` will be replaced with a filename to edit. Ideally, this command should move the cursor to the last line of the file, where the new reminder template is created.
- `edit_any_command`
Controls the command used for editing a reminder file without selecting any particular reminder. The special character `'%file%'` will be replaced with a filename to edit.
- `timed_template`
Controls the format of the REM line created when editing a new timed reminder. The following string substitutions will be made: `'%monname%'` - month name, `'%mon%'` - month number, `'%mday%'` - day of the month, `'%year%'` - year, `'%hour%'` - hour, `'%min%'` - minute, `'%weekdayname%'` - weekday name, `'%yday%'` - weekday number.
- `untimed_template`
Controls the format of the REM line created when editing a new untimed reminder. The substitution syntax is the same as for `timed_template`.
- `templateN`
Controls the format of a generic user-defined REM line template; *N* may range from 0 to 9. The substitution syntax is the same as for `timed_template`.
- `busy_algorithm`
An integer value specifying which algorithm to use for measuring how busy the user is on a particular day. If `busy_algorithm="1"`, then Wyrd will simply count the total number of reminders triggered on that day. If `busy_algorithm="2"`, then Wyrd will count the number of hours of reminders that fall on that day. (Untimed reminders are assumed to occupy `untimed_duration` minutes.)
- `untimed_duration`
An integer value that specifies the assumed duration of an untimed reminder, in minutes. This is used only when computing the busy level with `busy_algorithm="2"`.
- `busy_level1`
An integer value specifying the maximum number of reminders in a day (with `busy_algorithm="1"`) or maximum hours of reminders in a day (with `busy_algorithm="2"`) which will be colored using the color scheme for `calendar_level1`.
- `busy_level2`
Same as above, using the `calendar_level2` color scheme.

- `busy_level3`
Same as above, using the `calendar_level2` color scheme rendered in bold.
- `busy_level4`
Same as above, using the `calendar_level3` color scheme. Any day with more reminders than this will be rendered using the `calendar_level3` color scheme rendered in bold.
- `week_starts_monday`
A boolean value (`"true"` or `"false"`) that determines the first day of the week.
- `schedule_12_hour`
A boolean value that determines whether the timed reminders window is drawn using 12- or 24-hour time.
- `selection_12_hour`
A boolean value that determines whether the selection information is drawn with 12- or 24-hour time.
- `status_12_hour`
A boolean value that determines whether the current time is drawn using a 12- or 24-hour clock.
- `description_12_hour`
A boolean value that determines whether reminder start and end times are drawn using 12- or 24-hour time in the description window. This value also controls the format of timestamps in the formatted calendars produced by `view_week` and `view_month`.
- `center_cursor`
A boolean value that determines how the screen and cursor move during scrolling operations. When set to `"true"`, the cursor is fixed in the center of the timed reminders window, and the schedule scrolls around it. When set to `"false"` (the default), the cursor will move up and down the schedule during scrolling operations.
- `goto_big_endian`
A boolean value that determines how the `goto` operation will parse dates. When set to `"true"`, date specifiers should be in ISO 8601 (YYYYMMDD) format. When set to `"false"`, date specifiers should be in European style DDMMYYYY format.
- `quick_date_US`
A boolean value that determines how the `quick_add` operation will parse numeric dates with slashes, e.g. 6/1 (or 6/1/2006). When set to `"true"`, the first number is a month and the second is the day of the month (June 1). When set to `"false"`, these meanings of these two fields are switched (January 6).
- `number_weeks`
A boolean value that determines whether or not weeks should be numbered within the month calendar window. Weeks are numbered according to the ISO 8601 standard. The ISO standard week begins on Monday, so to avoid confusion it is recommended that `week_starts_monday` be set to `"true"` when week numbering is enabled.

- `home_sticky`
A boolean value that determines whether or not the cursor should "stick" to the "home" position. When this option is set to `"true"`, then after pressing the `<home>` key the cursor will automatically follow the current date and time. The effect is cancelled by pressing any of the navigation keys.
- `untimed_window_width`
An integer value that determines the target width of the month-calendar window and the untimed reminders window. The allowable range is 34 to $(\$COLUMNS - 40)$ characters, and Wyrd will silently disregard any setting outside this range.
- `advance_warning`
A boolean value that determines whether or not Wyrd should display advance warning of reminders. When set to `"true"`, Wyrd will invoke Remind in a mode that generates advance warning of reminders as specified in the reminder file.
- `untimed_bold`
A boolean value that determines whether or not Wyrd should render untimed reminders using a bold font.

For maximum usefulness, `busy_level1 < busy_level2 < busy_level3 < busy_level4`.

4.3 Calendar Operations

Every Wyrd operation can be made available to the interface using the syntax described in Section 4.1.3. The following is a list of every available operation.

- `scroll_up`
move the cursor up one element
- `scroll_down`
move the cursor down one element
- `next_day`
jump ahead one day
- `previous_day`
jump backward one day
- `next_week`
jump ahead one week
- `previous_week`
jump backward one week
- `next_month`
jump ahead one month
- `previous_month`
jump backward one month

- `home`
jump to the current date and time
- `goto`
begin entering a date specifier to jump to
- `zoom`
zoom in on the day schedule view (this operation is cyclic)
- `edit`
edit the selected reminder
- `edit_any`
edit a reminder file, without selecting any particular reminder
- `scroll_description_up`
scroll the description window contents up (when possible)
- `scroll_description_down`
scroll the description window contents down (when possible)
- `quick_add`
add a “quick reminder”
- `new_timed`
create a new timed reminder
- `new_timed_dialog`
same as previous, with a reminder file selection dialog
- `new_untimed`
create a new untimed reminder
- `new_untimed_dialog`
same as previous, with a reminder file selection dialog
- `new_templateN`
create a new user-defined reminder using `templateN`, where N may range from 0 to 9
- `new_templateN_dialog`
same as previous, with a reminder file selection dialog
- `copy`
copy a reminder to Wyrd’s clipboard
- `cut`
delete a reminder and copy it to Wyrd’s clipboard
- `paste`
paste a reminder from Wyrd’s clipboard into the schedule

- `paste_dialog`
same as previous, with a reminder file selection dialog
- `switch_window`
switch between the day schedule window on the left, and the untimed reminder window on the right
- `begin_search`
begin entering a search string
- `search_next`
search for the next occurrence of the search string
- `next_reminder`
jump to the next reminder⁹
- `view_remind`
view the output of `remind` for the selected date
- `view_remind_all`
view the output of `remind` for the selected date, triggering all non-expired reminders
- `view_week`
view Remind's formatted calendar for the week that contains the selected date (the in-calendar times-tamp formats are determined by the value of `description_12_hour`)
- `view_month`
view Remind's formatted calendar for the month that contains the selected date (the in-calendar times-tamp formats are determined by the value of `description_12_hour`)
- `refresh`
refresh the display
- `quit`
exit Wyrd
- `entry_complete`
signal completion of search string entry or date specifier
- `entry_backspace`
delete the last character of the search string or date specifier
- `entry_cancel`
cancel entry of a search string or date specifier

⁹The `next_reminder` operation locates reminders at the point they occur; if `advance_warning` is enabled, `next_reminder` will skip over any displayed warnings of an event.

4.4 Colorable Objects

Each of Wyrd's on-screen elements may be colored by the color scheme of your choice, using the syntax defined in Section 4.1.5. The following is a list of all colorable objects.

- `help`
the help bar at the top of the screen
- `timed_default`
an empty timeslot in the day-schedule window
- `timed_current`
the current time in the day-schedule window (if it is visible)
- `timed_reminder1`
a nonempty timeslot in the day-schedule window, indented to level 1
- `timed_reminder2`
a nonempty timeslot in the day-schedule window, indented to level 2
- `timed_reminder3`
a nonempty timeslot in the day-schedule window, indented to level 3
- `timed_reminder4`
a nonempty timeslot in the day-schedule window, indented to level 4
- `untimed_reminder`
an entry in the untimed reminders window
- `timed_date`
the vertical date strip at the left side of the screen
- `selection_info`
the line providing date/time for the current selection
- `description`
the reminder description window
- `status`
the bottom bar providing current date and time
- `calendar_labels`
the month and weekday labels in the calendar window
- `calendar_level1`
calendar days with low activity
- `calendar_level2`
calendar days with medium activity
- `calendar_level3`
calendar days with high activity

- `calendar_today`
the current day in the calendar window (if it is visible)
- `left_divider`
the vertical line to the left of the timed reminders window
- `right_divider`
the vertical and horizontal lines to the right of the timed reminders window

5 Usage Tips

- Wyrd fills in sensible defaults for the fields of a REM statement, but you will inevitably need to make some small edits to achieve the behavior you want. If you use Vim, you can make your life easier by installing the Vim-Latex Suite and then modifying your `~/.wyrdr` to use REM templates like this:

```
set timedtemplate="REM %monname% %mday% %year% <+>AT %hour%:%min%<+>
DURATION 1:00<+> MSG %\"<+>%\" %b"
set untimedtemplate="REM %monname% %mday% %year% <+>MSG %\"<+>%\" %b"
```

With this change, hitting Ctrl-J inside Vim (in insert mode) will cause your cursor to jump directly to the `<+>` markers, enabling you to quickly add any desired Remind delta and message parameters.
- The 43 Folders Wiki has a page on Wyrd¹⁰. This is a good place to look for other usage tips.

6 Licensing

Wyrd is Free Software; you can redistribute it and/or modify it under the terms of the GNU General Public License (GPL), Version 2, as published by the Free Software Foundation. You should have received a copy of the GPL along with this program, in the file `'COPYING'`.

7 Acknowledgments

Thanks, of course, to David Skoll for writing such a powerful reminder system. Thanks also to Nicolas George, who wrote the OCaml curses bindings used within Wyrd.

8 Contact info

Wyrd author: Paul Pelzl <pelzlpj@gmail.com>

Wyrd website: <http://pessimization.com/software/wyrd>

Wyrd project page (bug reports, code repository, etc.): <http://launchpad.net/wyrd>

¹⁰<http://wiki.43folders.com/index.php/Wyrd>

9 Miscellaneous

“Wyrd is a concept in ancient Anglo-saxon and Nordic cultures roughly corresponding to fate or personal destiny.” – *Wikipedia*