lang Fr | Gb

ratpoison

ratpoison is a simple Window Manager with no fat library dependencies, no fancy graphics, no window decorations, and no rodent dependence. It is largely modeled after GNU Screen which has done wonders in the virtual terminal market. All interaction with the window manager is done through keystrokes. ratpoison has a prefix map to minimize the key clobbering that cripples Emacs and other quality pieces of software. ratpoison was written by Shawn Betts (sabetts@gmail.com).

Concepts

ratpoison uses the concept of panes to place and size windows. Instead of allowing windows to have arbitary shapes at arbitary locations on the screen, the display is divided into panes, the same way a physical window might contain several pieces of glass seperated by wood. In ratpoison, the panes are called frames, and windows are placed in them, maximised. ratpoison starts with one frame, which can be split into an arbitary number of smaller ones. Each frame can be split in half either horizontally or vertically. You can move among them, making different ones the current. For more information, see Splitting Frames.

Each frame has at most one window associated with it, which is visible in that frame. If you select a window that is associated with a frame, the focus will move to its associated frame, rather than moving the window to the current frame. If you select a window that is not associated with a frame, that window will be opened in the current frame and resized to fit that frame.

If the window associated with a frame does not fill the frame completely, the various gravity commands control how it is placed.

If no window was open in that frame before the current window was opened, the X root will be visible behind it.

General Use

When ratpoison starts you should see an empty X server. To open an x terminal hit C-t c. You can now run shell commands as you would on any terminal. Notice the terminal maximized full screen. C-t ! will run a single shell command and saves you the effort of opening a terminal.

Once you have a couple X programs running, you'll want to navigate between windows. To see what windows are being managed hit C-t w. Each window has a number. You can jump to a window by hitting C-t followed by the window's number. This assumes the the window's number is one digit. You can also switch to a window by typing in part of its name. To do this hit C-t '.

ratpoison allows you to cycle through the windows with C-t n and C-t p.

And That concludes a brief introduction on how to use ratpoison. Notice how we didn't have to drag a single window, or click a single maximize button? Beautiful wasn't it? Felt fast? Cool? It's modern computing at its best.

Windows

Manipulating Windows

The following are commands used to manipulate windows:

Command	Action & Keys
select n	This jumps you to window n where n is the window number as shown in the Program Bar. You can do the same trick with C-n too. To select no window, blanking the current frame, type `select -'.
select window-name	Go to a window by name. A shortcut is C-t '.
windows fmt	This displays the Program Bar which displays the windows you currently have running. The number before each window name is used to jump to that window. You can do this by typing C-t n where n is the number of the window. Note that only windows with numbers from 0 to 9 can be referenced using this keystroke. To reach windows with numbers greater than 9, use C-t ' and type the number at the prompt. After 5 seconds the Program Bar disappears. This command is bound to C-t w by default. When invoked from the command-line like this,'$ ratpoison -c windows' . Instead of a message bar, you will get a list of the windows printed to stdout. This allows you to write more advanced scripts than simple keyboard macros. This is where fmt comes into play. If windows is given an argument it treats it as the format string as described in set winfmt.
title title	Rename the currently active window. This name will remain for the duration of the window's life, unless you change it again. By default, the C-t A keystroke is bound to this command.
other	This toggles between the current window and the last window. By default, this is bound to C-t C-t.
prev	This jumps you to the previous window in the window list. By default, this is bound to C-t p.
next	This jumps you to the next window in the window list. This one is bound to three keystrokes, namely C-t n, C-t space, and C-t enter.
kill	This destroys the current window. Normally you should only need to use delete, but just in case you need to rip the heart out of a misbehaving window this command should do the trick. Also available as C-t K.
info fmt	Display information about the current window. At optional fmt argument can be passed to override the default format. info accepts the same format options as windows.

gravity g	Change the gravity of the current window. A normal window will default to the top-left corner of the screen, but it can also be placed at the bottom-right corner of the screen. Valid values for g are the 8 directions `northwest', `north', `northeast', `east', `southeast', `south', `southwest' and `west', clockwise from the top left corner. `center' will center the window in the frame. g and can be abbreviated to the standard compass 1 and 2 letter abbreviations (i.e. `nw', `s', etc). When called with no arguments, the current setting is returned.
delete	This deletes the current window. You can access it with the C-t k keystroke.
set infofmt fmt	Set the default window format for the info command. See set winfmt for accepted format characters.
set winname name	There are three resources ratpoison can get a window's name from: the WMNAME hint, the res_name from the WMCLASS hint, or the res_class from the WMCLASS hint. name can be `title' which is what most window managers put in the title bar, `name' which is the res_name, or `class' which is the res_class. When called with no arguments, the current setting is returned.
set wingravity g	Set the default gravity for normal windows. See the gravity command. When called with no arguments, the current setting is returned.
set winliststyle setting	The window list can be displayed in a row or a column. setting can be `row' or `column'.
number n target	Set a window's number to n. If another window occupies the requested number already, then the windows' numbers are swapped. The second argument, target, is optional. It should be the number of the window whose number will be changed. If target is omitted ratpoison defaults to the current window.
set transgravity g	Set the default alignment for transient windows. See the gravity command. When called with no arguments, the current setting is returned.
set maxsizegravity g	Set the default alignment for windows with maxsize hints. See the gravity command. When called with no arguments, the current setting is returned.
set border n	Set the border width for all windows. When called with no arguments, the current setting is returned.
set winfmt fmt	Set the default window format for the windows command. By default it is '%n%s%t'. See below for a list of valid format characters.

char	**window format**
'%a'	Application Name
'%c'	Resource Class
'%f'	The frame number the window is displayed in or a space if it is not in a frame
'%g'	The window's gravity setting
'%h'	The window's height
'%H'	The window's height increment hint
'%i'	X11 Window ID
'%l'	A unique number based on when the window was last accessed. The higher the number, the more recently it was accessed
'%n'	The window number
'%p'	Process ID ('?' if _NET_WM_PID isn't set)
'%s'	Window status (current window, last window, etc)
'%S'	The window's screen
'%t'	Window Name
'%T'	Whether the window is a transient or not

'%M'	Whether the window is a maxsize window or not
'%w'	The window's width
'%W'	The window's width increment hint
'%x'	the window's xinerama screen

Additionally there can be a number between the percent sign and the format character, denoting a maximum length this value is to truncate to, e.g. `%n%s%20t'.
When called with no arguments, the current setting is returned.

Window Classes

Window classes are a way of grouping windows together. Windows that are part of the same program generally have the same class. Ratpoison takes advantage of this to help you navigate between windows of the same class. This is useful if you only want to cycle through Emacs frames or XTerms.

Command	Action
inext	Go to the next window in the window list that is in the same class as the current window.
iprev	Go to the previous window in the window list that is in the same class as the current window.
iother	Go to the last accessed window that is in the same class as the current window.
cnext	Go to the next window in the window list that is in a different class from the current window.
cprev	Go to the previous window in the window list that is in a different class from the current window.
cother	Go to the last accessed window that is in a different class from the current window.

Unmanaged Windows

ratpoison can intentionally not manage windows. ratpoison keeps a list of strings and if any new window's name matches a string in the list, then it will not be picked up and managed by ratpoison. The following are commands to manipulate this list:

Command	Actions
clrunmanaged	Clear the unmanaged window list.
unmanage text	Add text to the unmanaged window list. Any window whose name matches any of the strings in the unmanaged window list will not be handled in any way by ratpoison. This only applies to new windows (not windows already managed by ratpoison). When called with no arguments, the list is returned.

Rudeness

Some programs will attempt to steal the focus without the users permission. Not only is this a sign of a lame programmers attempt to fix a window manager problem in the wrong place, it's just plain rude. By default ratpoison will honour these rudeness requests, but it doesn't have to. Use the rudeness command to deal with such programs.

Command	Action
rudeness n	The rudeness command lets you decide what windows pop-up automatically and when. This is often useful for those deep hack sessions when you absolutely can't be disturbed. n is a number from 0 to 15. Each of the four bits determine which requests ratpoison grants. When called with no arguments, the current setting is returned.

There are two kinds of windows: normal windows (like an xterm) and transient windows (generally pop-up dialog boxes). When a client program wants to display a new window it makes a requests to ratpoison. ratpoison then decides whether to grant the request and display the window or ignore it. A client program can also request that one of its windows be raised. You can customize ratpoison to either honour these requests (the default operation) or ignore them.

- Bit 0 : Tells ratpoison to grant raise requests on transient windows
- Bit 1 : Tells ratpoison to grant raise requests on normal windows
- Bit 2 : Tells ratpoison to grant display requests on new transient windows
- Bit 3 : Tells ratpoison to grant display requests on new normal windows

For example, if you wanted only wanted to grant transient windows raise requests and display requests you would type `rudeness 5'. If a request is not granted ratpoison will tell you about the request with a message like `Raise request from window 1 (emacs)'.

Groups

ratpoison provides functionality to group windows together. This coupled with saving and restoring frames configurations is what most people would call virtual desktops or workspaces.

While ratpoison doesn't explicitly provide support for such things, it does allow you to write scripts to this end. Such a script exists in contrib/ called rpws. Consult that file for details on setting up workspaces inside ratpoison.

Groups are more general purpose than workspaces. windows from one group can be visible along with windows from another group. If you switch to a different group nothing changes except the list of windows you can cycle through. ratpoison allows the user to move a window from one group to another, merge two groups, create new groups, and delete existing ones.

The following is a list of of commands used for manipulating groups.

Command	Action
gnew name	Create a new group with the name name. name is optional. The new group becomes the current group.
gnewbg name	This is the same as gnew except that the current group does not change.
groups	Display a list of groups with a similar format to windows.
gmove group	Move the current window to group.
gnext	Go to the next group in the list.
gother	Go to the last accessed group.
gprev	Go to the previous group in the list.
grename	Rename current group.

gselect group	Select a particular group by name or number. If group is not provided, ratpoison will interactively prompt for the group.
gmerge group	Merge group with the current group. All windows in group will be moved to the current group. group is not deleted.
gdelete group	Delete a group. group is optional. If it is not specified ratpoison will attempt to delete the current group. Only empty groups can be deleted. To empty a group see gmerge.

Frames

Splitting Frames

To split the current frame horizontally use C-t s. To split the current frame vertically use C-t S. If you have enough windows, you'll notice that the new frame will find a window for itself. You can now use the normal navigation commands to switch windows in the frame. Note, however, that if you switch by name or number to a window that is already in another frame, you'll switch to that frame.

Before too long, you'll probably want to switch to another frame. Use C-t tab to cycle through the frames. If you want to remove a frame use C-t R. ratpoison automatically adjusts the size of the other frames to take up the free space. Unfortunately ratpoison may not always fill it in the way you might like it to.

Finally, when you've had enough of the splitting and you just want good ol' full screen ratpoison press C-t Q to remove all splits and leave you with the current window full screen.

Command	Action
remove	Kill the current frame. This is a no-op if there is only one frame.
only	Kill all frames but the current one.
(v)split n	Split the current frame vertically in two. The last accessed window not occupying a frame will be the second window.
hsplit n	Split the current frame horizontally in two. The last accessed window not occupying a frame will be the second window.

n is either a fraction of the form x/y or a number. If it is a fraction then the current frame is resized to that fraction of its original size and the new frame takes up the remaining space. For instance, split 1/4 will split the current frame to a quarter of its original size and the new frame will then be 3/4 of the size of the original frame.
If it is a pixel, the original frame is resized to that many pixels. If n has a minus sign before it, then the new frame will shrink by that many pixels.

Resizing Frames

ratpoison provides a command, resize, that resizes the current frame. It is bound to the key C-t r by default. resize can be used non-interactively by providing two arguments: the number of pixels to grow horizontally and the number to grow vertically. For example, if you wanted to grow the current window by 10 pixels horizontally and shrink it vertically by 50 you could enter the command: resize

10 -50

When resizing interactively, the following keys are used:

Keys	Action on Frames
C-p	Grow the frame vertically.
C-n	Shrink the frame vertically.
C-f	Grow the frame horizontally.
C-b	Shrink the frame horizontally.
return	Accept the new frame size.
C-g	Abort and restore the frame to its original size.

The increment size used to resize the frame interactively is customized with the command set resizeunit.

Command	Action
set resizeunit pixels	Set the number of pixels a frame will grow or shrink by when being dynamically resized. When called with no arguments, the current setting is returned.
resize horizontal vertical	Resize the current frame by horizontal pixels horizontally, and vertical pixels vertically. If no arguments are given and the command is called interactively, ratpoison will let the user dynamically resize the frame using C-p to shrink vertically, C-n to grow vertically, C-b to shrink horizontally, C-f to grow horizontally, and s to shrink the frame to the size of the window (See the shrink command). When you have resized the frame to your liking, press Return to finish.
shrink	If a window has resize increment hints, such as xterms, the window may not be able to take up the whole frame. In this case, use this command to suck the frame up to the to window, reclaiming any wasted space.

Frame Navigation Commands

Here are the commands for Navigating frames.

Command	Action
fselect n	Select a frame by number. If an argument is passed to it then attempt to select the frame whose number is n. If not, ratpoison will print a number at the top left corner of each frame and wait for the user to type the number they wish to select. Currently there is no way to select a frame whose number is greater than 9 unless the number is passed as an argument.
curframe	Indicate which frame is the current frame.
focus	cycle through ratpoison's frames.
focusprev	cycle through ratpoison's frames backwards.
focusdown	Move to the frame below the current frame.
focuslast	Switch to the last focused frame.
focusleft	Move to the frame left of the current frame.
focusright	Move to the frame right of the current frame.
focusup	Move to the frame above the current frame.

Saving and Restoring Frame Sets

ratpoison provides two commands, fdump and frestore, that allow the user to save and restore frame configurations. Let's say, for example, you have split your desktop into several frames with some windows in these frames and now you want to quickly bring Emacs forward and browse some code (full-screen of course) then return to your funky frame configuration. You could use fdump to dump the frames, hit C-t Q to remove all frames, and then select your emacs window. When you've finished with emacs you could use frestore to restore the windows and frames.

If a frame contained a window when you dumped the frame layout but that window is not present when you restore the layout, the frame holding that window will be blank.

Calling fdump and frestore and copying and pasting the layout by hand each time is a bit cumbersome. There are some simple bindings in doc/sample.ratpoisonrc that allow you to save and restore frame layouts with the press of a key.

Command	Action
fdump screen-num	Dump the current frame layout as text. Without an argument the current screen's frames are dumped. With an argument the screen-numth screen is dumped. See Multiple Monitors.
frestore frames	Restore the frame layout based on the list of frames frames. frames should be the text that was printed after calling fdump.
undo	Undo the last change of frame layout. This is especially helpful after a only command. One can step at most maxundos steps back in frame layout history.
redo	redo the last change that was undone.

Frame Numbering

Frames are normally numbered starting from 0. But this can be changed with set framesels to, for instance, include letters as well.

```
set framesels abcdefghijklmnopqrstuvwxyz
```

The above code will bind letters to frames instead of numbers.

Command	Action
set framesels order	Tell ratpoison what alphanumeric character to give each frame and in what order. When called with no arguments, the current setting is returned.

Dedicated Frames

A dedicated frame is a frame that will not allow new windows to appear in it. Only the user may switch windows in this frame.

Command	Action
dedicate	Toggle whether the current frame is dedicated or not.

Multiple Monitors

When you've finally accumulated enough computer junk, you'll find yourself attaching a second monitor to your computer. ratpoison has functionality to help you get around your new and improved desktop space.

The X Windowing System assigns each monitor a screen number. To switch to another screen use the commands nextscreen and prevscreen. Or, sselect to jump to a specified screen. ratpoison will tell you which frame has focus by drawing the current frame indicator in it.

Many commands operate only on the current screen. This becomes apparent when you have 2 screens each with 1 frame. In each frame you have an xterm. If you try to switch to the other xterm with the command other, for instance, you'll get a message “No other window.” ratpoison means there's no other window to switch to in the current screen. If you want to switch to the other xterm you can switch to it by name (use select or C-t '), by number, or you can use nextscreen, prevscreen, and sselect.

Command	Action
nextscreen	This jumps you to the next X11 screen. nextscreen is used for dual-head displays and multiple monitor setups.
prevscreen	This jumps you to the previous X11 screen. prevscreen is used for dual-head displays and multiple monitor setups.
sselect n	This jumps you to the nth X11 screen. Screen numbers start at 0.
sdump	Like fdump, but dump information about each screen instead of each frame.
sfdump	Dump all the screen number and the frames on all screens.
sfrestore	restore a frame configuration created using sfdump.

Keystrokes

Interactive control of ratpoison is done entirely through keystrokes. This chapter explains how keystrokes are stored and manipulated.

ratpoison uses the Emacs style key notation. A combination of modifiers and one non-modifier key combine to invoke an action. The syntax is one or more modifiers seperated with dashes followed by a dash and the non-modifier key name. For instance, holding down control, shift, and super then pressing the spacebar would be described as: S-C-s-space

The following is a list of modifiers ratpoison accepts:

- S : Shift modifier
- C : Control modifier
- M : Meta modifier
- A : Alt modifier
- H : Hyper modifier
- s : Super modifier

ratpoison uses the X11 keysym names for keys. Alphanumeric key names are exactly what you see on your keyboard. Punctuation and other keys have longer names which vary from X server to X server.

To find the name of a key, see the describekey command. Or to find the name of a key not yet bound to an action, type C-t and then the key. ratpoison will tell you it isn't bound and give you the name of the key.

Key Maps

All keystrokes exist inside a keymap. When you press the prefix key you are accessing the 'root' keymap. By default all commands reside in the `root' key map and are accessed by pressing C-t.

There is also a top level key map, 'top'. Any keystroke in this key map can be accessed simply by pressing the key. This is where the prefix key resides.

The following example adds a C-x b key binding to switch windows, much like C-x b in Emacs. See the functions below for full descriptions.

```
# Create the key map
newkmap ctrl-x
# Bind b to 'select' on our new key map
definekey ctrl-x b select
# Attach our keymap to the top level key map via C-x.
definekey top C-x readkey ctrl-x
```

The following functions control creating, editing, and deleting key maps.

Command	Action on keymap
newkmap kmap	Create a new keymap named kmap.
delkmap kmap	Delete the keymap, kmap.
bind Key command	Bind a key to a ratpoison command on the `root' keymap. This command takes two arguments: the key to bind and the command to run. For example, to bind C-t R to restart ratpoison: bind R restart
unbind key	Unbind a keystroke on the 'root' keymap.
definekey kmap key command	definekey works exactly like bind except that it can bind keys on any key map (not just `root').
undefinekey kmap key	Like unbind except that you pass it a key map in kmap.
readkey kmap	Read a key from the keyboard and execute the command associated with it in the keymap, kmap.
link key	Call the command that key is bound to. For instance link C-t would call the command other and switch to the last window.
describekey keymap	An interactive way to find the command bound to a given key on the specified keymap. This command will wait for the user to type a key. When the user does, the command will display the command bound to this key.
set topkmap kmap	Set the top level keymap to kmap. You might use this to swap between several common keymappings or to implement modes.

Default Key Bindings

The default keystrokes are listed in this chapter. Not all commands are accessible by default by keys.

Keys	Action
C-t C-t	Switch to the last window.
C-t t	Sometimes you need to send a C-t to the current window. This keystroke does just that.
C-t 0-9	Switch to the numbered window.
C-t -	Select no window, essentially hiding all windows in the current frame.
C-t A / C-t C-A	Rename the current window. The window's new name will prevail for the rest of its lifetime.
C-t K / C-t C-K	Send a DestroyClient event to the current window. This will terminate the application without question.
C-t n / C-t C-n / C-t Return C-t C-Return / C-t Space / C-t C-Space	Go to next window.
C-t p / C-t C-p	Go to previous window.
C-t ' / C-t C-'	Go to a window by name. You will usually only need to type the first few characters of the window name.
C-t a / C-t C-a	Display the current time of day.
C-t c / C-t C-c	Open a new X terminal.
C-t :	This allows you to execute a single ratpoison command.
C-t !	Run a shell command.
C-t C-!	Run a shell command through an X terminal.
C-t i / C-t C-i	Display information about the current window.
C-t k / C-t C-k	Close the current window.
C-t l / C-t C-l	Redisplay the current window. Sometimes windows don't respond correctly to the initial maximize event and need some coaxing. This is a fancy way of saying there are still bugs in ratpoison. C-t l will force the current window to maximize.
C-t m / C-t C-m	Display the last message.
C-t v / C-t C-v	Display the version of ratpoison.
C-t V / C-t C-V	Display ratpoison's license.
C-t w / C-t C-w	Display the list of managed windows. The current window is highlighted.
C-t s / C-t C-s	Split the current window horizontally in two. The last accessed window not occupying a frame will be the second window.
C-t S / C-t C-S	Split the current window vertically in two. The last accessed window not occupying a frame will be the second window.
C-t tab	Cycle through ratpoison's frames.
C-t M-tab	Switch to the last focused frame.
C-t Q	Kill all frames but the current one.
C-t R	Kill the current frame. This is a no-op if there is only one frame.
C-t r / C-t C-r	Resize the current frame.
C-t b / C-t C-b	Banish the mouse to the lower right corner of the screen.
C-t ?	Display a help screen.
C-t f / C-t C-f	select a frame by number.
C-t F	Indicate which frame is the current frame.
C-t Down	Move to the frame below the current frame.
C-t Left	Move to the frame left of the current frame.
C-t Right	Move to the frame right of the current frame.

C-t Up	Move to the frame above the current frame.
C-t C-Down	Exchange the window in the current frame with the window in the frame below it.
C-t C-Left	Exchange the window in the current frame with the window in the frame to the left of it.
C-t C-Right	Exchange the window in the current frame with the window in the frame to the rigth of it.
C-t C-Up	Exchange the window in the current frame with the window in the frame above it.
C-t x / C-t C-x	Choose a frame and exchange the window in the current frame with the window in the chosen frame.

Hooks

One of the goals of ratpoison is to allow users to create exciting customization to fit their specific needs. Hooks allow a user to latch scripts onto certain events.

Each hook contains a list of commands to be executed when the appropriate event occurs in ratpoison. For example, if you want to warp the rat to corner of the screen every time you press a top level bound key, you could add this to you .ratpoisonrc file: addhook key banish

That should keep the rat out of your way.

- Command: addhook hook command : Add a command to hook. When the hook is run, command will be executed.
  The following hooks are available:

Command	Action
key	Run when a top level key is pressed (by default the only top level key is the prefix key).
switchwin	Run when the user switches to a different window in the current frame.
switchframe	Run when the user switches to another frame. This is also run when the user switches to a different screen, since a frame switch also occurs.
switchgroup	Run when the user switches to a different group.
deletewindow	Run when a window is deleted.
newwindow	Run after a new window is mapped.
titlechanged	Run when the current window's title changes.
quit	Run when ratpoison exits.
restart	Run when ratpoison restarts.

- Command: remhook hook command : Remove command from the hook. See addhook for a list of available hooks.
- Command: listhook hook : List the commands that will be run when hook is fired.

The Status Bar

ratpoison presents status and output through the status bar. By default it is located in the top right corner of the screen.

This chapter presents commands for manipulating the status bar.

Since it is the only visible evidence that ratpoison is running (as opposed to the invisible evidence including the lack of title bars and your favorite desktop background) there are also copious visual customizations available for those rainy days.

Command	Action
msgwait n	Set the bar's timeout in seconds. When called with no arguments, the current setting is returned.
lastmsg	Display the last message.
echo text	Display text as a message.
set inputwidth n	Set the width of the input window. When called with no arguments, the current setting is returned.
set font font	Set the font. font is a font string like `9x15bold'. When called with no arguments, the current setting is returned.
set framefmt fmt	Set the text that appears when the curframe command is called. fmt is a format string that accepts the same format characters as set winfmt.
set fgcolor color	Set the foreground color for all text ratpoison displays. color is any valid X11 color. When called with no arguments, the current setting is returned.
set bgcolor color	Set the background color for all text ratpoison displays. color is any valid X11 color. When called with no arguments, the current setting is returned.
set fwcolor color	Set the border color for the focused window. is any valid X11 color. When called with no arguments, the current setting is returned.
set bwcolor color	Set the border color for unfocused windows. is any valid X11 color. When called with no arguments, the current setting is returned.
set barpadding x y	Set the horizontal and vertical padding inside the bar. When called with no arguments, the current setting is returned.
set bargravity g	Set the default alignment for the message bar. See the gravity command. When called with no arguments, the current setting is returned.
set barborder n	Set the border width for the bar window. When called with no arguments, the current setting is returned.
set barinpadding n	Set whether the bar window appears at the edge of the screen when there is padding – that is, within the “padding” area – or whether it appears at the edge of the window area. “1” represents the former, “0” the latter. See the set padding and set bargravity commands. When called with no arguments, the current setting is returned.

Using Other Window Managers

There are times when a program has been so badly written that it is virtually impossible to use under ratpoison. Some authors have tailored their programs to certain window management paradigms so aggressively that very little can be done. Ratpoison has two commands to help you through these difficult times: tmpwm and newwm.

These commands should be used sparingly. They were created to allow users to understand how a

poorly designed program is intended to function so they can build a replacement or patch an existing alternative's missing functionality.

According to independant studies, tmpwm has been used almost exclusively to verify its correct operation – like a vintage sports car: always kept in prime condition and never used.

tmpwm and newwm are provided for boasting and completeness.

- Command: tmpwm WM : Gives control over to another window manager and regains control once it has terminated. WM is the path to the new window manager. This command is useful when you want to temporarily take a look at another window manager, or program under a different window manager, but you want to come back to ratpoison when you've finished your investigation.
- Command: newwm window-manager : This is a bad-bad command. It kills ratpoison and revives that ugly rodent! Yuck! Avoid!

Other Commands

The following is a list of commands that don't fit in any existing chapters.

Command	Action
abort	This is a pretty useless command. By default, it is bound to C-t g and its purpose is to abort the current chain of keystrokes (just like C-g in `Emacs').
alias name command	Allows you to name a ratpoison command something else. For instance, if you frequently open emacs you may want to make an alias called `emacs' that loads emacs. You would do it like this: alias emacs exec emacs An alias is treated exactly like a colon command in that you can call it from the colon prompt, bind it to a key, and call it non-interactively with ratpoison -c.
banish	Banish the mouse to the lower right corner of the screen.
banishrel	Banish the rat cursor to the lower right corner of the curren window. If there isn't a window in the current frame, it banishes the rat cursor to the lower right corner of the frame.
chdir	Change the current directory for ratpoison.
colon command	Run a ratpoison command.
compat	Install the now obsolete `def*' commands as aliases to the corresponding `set *' command.
set padding left top right bottom	Set the padding around the edge of the screen. When called with no arguments, the current setting is returned.
set waitcursor n	Set whether the rat cursor should change into a square when waiting for a key. A non-zero number means change the cursor. Zero means don't change the cursor. When called with no arguments, the current setting is returned.
set historysize n	Set how many lines of history should be recorded. When called with no arguments, the current setting is returned.
set historcompaction bool	Set whether to remove multiple equal lines from history, even if not adjacent. When called with no arguments, the current setting is returned.

set historexpansion bool	Set whether to expand ! using readline's libhistory in input. When called with no arguments, the current setting is returned.
escape key	Set the prefix to key. For example `escape C-b' sets the prefix key to <C-b>.
exchangedown	Exchange the current frame with the one below it.
exchangeleft	Exchange the current frame with the one to the left of it.
exchangeright	Exchange the current frame with the one to the right of it.
exchangeup	Exchange the current frame with the one above it.
exec command	Execute a shell command. By default, C-t ! does this.
execa command	Execute a shell command but don't record which frame it was executed from. The client's windows will pop up in whatever frame is current.
execf frame command	Execute a shell command and choose which frame the client's first window will open in. The client must be netwm compliant for this to work.
getenv env	Display the value of the environment variable, env.
getsel	Return the contents of the X11 selection.
help	Display a help screen that lists all bound keystrokes.
license	Display ratpoison's license. By default, this is bound to C-t V.
meta key	key is an optional argument. When key is omitted, send a C-t to the current window. Otherwise, send the key described by key to the current window. Note that some applications by default ignore the synthetic key that is sent using this command as it is considered a security hole. xterm is one such application. For example, if your `Emacs' window is focused, 'meta M-x' Would cause emacs to prompt for an extended command.
prompt prompt	This command is only useful when called non-interactively. prompt prompts the user for input using prompt and returns the input.
putsel text	Make text the X11 selection.
quit	Quit ratpoison.
ratinfo	Display the x y coordinates of the rat cursor relative to the screen.
ratrelinfo	Display the x y coordinates of the rat cursor relative to the current window or current frame if no window is focused.
ratrelwarp x y	Warp the rat to the specified location relative to the current rat position.
ratwarp x y	Warp the rat to the specified absolute location.
ratclick button	click the rat. button is either 1, 2, or 3. button defaults to button 1.
rathold state button	click the rat button down if state is `down' or release the button if state is `up'.
redisplay	Extend the current window to the whole size of its current frame and redisplay it. This can be used to: - redisplay normal windows or bring transient windows to the full size of the frame as only normal windows are maximized by ratpoison. - fix xterms that didn't catch ratpoison's initial maximize event.
restart	Restart ratpoison.
set var value	Set the value of a ratpoison variable. This command replaces the older `def*' variable get/set style. Here is a list of variables that can be set: framesels, winliststyle, barpadding, bgcolor, fgcolor, winname, winfmt, waitcursor, inputwidth, barborder, border, padding, font, bargravity, maxsizegravity, transgravity, wingravity, maxundos, resizeunit, historysize, historycompaction, historyexpansion.
setenv env value	Set the environment variable env to value

source file	Read a text file containing ratpoison commands.
startup_message state	Turn on or off the startup_message. This is most useful in your .ratpoisonrc file. state can be on or off. When called with no arguments, the current setting is returned.
swap destination-frame source-frame	When called interactively prompt for a frame and swap its window with the window in the current frame. An optional second argument allows swapping of windows between arbitrary frames.
time	Show current time in the status bar.
unalias name	Remove name from the list of defined aliases.
unsetenv env	Clear the value of the environment variable, env.
verbexec command	Verbosely exec the shell command command. Raptoison displays a message saying command was executed.
version	Print ratpoison version. By default, this is bound to C-t v.
warp state	Toggle rat warping. By default ratpoison saves the position of the rat when leaving a window and when the user returns to the window the rat's position is restored. This can be counter-intuitive, so you can toggle it with this command. state can be on or off.

Input

At various times ratpoison will prompt you for input. Ratpoison sports a fully featured line editor. The following table lists the keystrokes and actions:

Keys	Action
<C-g> / <escape>	abort the command requesting input.
<C-f> / <right arrow>	move forward a character.
<C-b> / <left arrow>	move backward a character.
<M-f>	move forward a word.
<M-b>	move backward a word.
<C-a> / <home>	move to the beginning of the line.
<C-e> / <end>	move to the end of the line.
<C-d> / <delete>	delete the character at point.
<M-d>	delete the word at point.
<backspace>	delete the character before the point.
<M-backspace>	delete the word before the point.
<C-k>	delete from the point to the end of the line.
<C-u>	delete from the point to the beginning of the line.
<C-y>	Yank the text from the X11 cut buffer.
<C-p> / <up arrow>	Cycle backwards through the history (This command does nothing if ratpoison was configured with the –disable-history configure option).
<C-n> / <down arrow>	Cycle forwards through the history (This command does nothing if ratpoison was configured with the –disable-history configure option).
<return>	submit the line of text.

<tab>	complete the text up to the point or if there are several possible completions, cycle through them. This only works in certain contexts. Tab completion will complete a shell command, a window name, a group name, and colon commands in their appropriate context (i.e. when being asked for a window name).
<S-iso-lefttab>	This is shift + tab by the way. This does the same as tab, but cycles backwards through the completions.

All input is stored in the same history list. By default ratpoison has a history length of 100 entries. This history is saved to the file ~/.ratpoison_history and is loaded when you start ratpoison. This means your history sticks between sessions. This assumes history has not been disabled on compilation.

Command Line Arguments

ratpoison supports command line arguments to request various actions when invoking ratpoison.

Options	Action
-h, --help	Display this help screen
-v, --version	Display the version
-d, --display	Specify the X display to connect to.
-s, --screen	Specify the screen to use. By default ratpoison runs on all screens. You can tell it to use just one with this option.
-c, --command	Send ratpoison a colon-command. This allows you to control ratpoison from the command-line. with the -c option you can script ratpoison using any programming language that can spawn a process. Some commands behave differently when invoked this way. Currently the only commands that behaves differently are the windows command and some def* commands. Instead of displaying the window list in a message window, it is printed to stdout. The output can then be captured and used in the ratpoison script. For instance, this could be used to check whether a program is running and if it is switch to its window otherwise launch it. It should also be noted that multiple -c options can be used. to facilitate writing scripts, the RATPOISON environment variable is set to the full path of the ratpoison binary. $ ratpoison -c split -c split Here ratpoison would split the current frame twice.
-i, --interactive	Force ratpoison to execute commands in interactive mode. This is used in conjunction with the -c option.
-f, --file	Specify an alternate configuration file. See Startup file.

Startup file

Now you've probably read the web page, and you've no doubt dug up some old file I forgot about. You're probably wondering, “say, didn't he say there was no configuration file to customize?”. Okay, ya got me. But let's be honest here: ratpoison is so pure and fast-acting, customization is barely worth the extra effort. In the off chance that you need to make ratpoison your own, we now support it.

On startup ratpoison looks for ~/.ratpoisonrc and runs it through the command parser. If

~/.ratpoisonrc does not exist, ratpoison tries /etc/ratpoisonrc. This means any command you can bind a key to or run at the command prompt (C-t :) you can execute in this rc file.

You can also use the -f option to specify another startup file, allowing you to switch between different configurations (see Command Line Arguments).

---

sources : wiki ratpoison