Opensource.com

7 open source desktop tools

A productivity guide for becoming more efficient on your desktop

By Kevin Sonney

ABOUT OPENSOURCE.COM

What is Opensource.com?

OPENSOURCE.COM publishes stories about creating, adopting, and sharing open source solutions. Visit Opensource.com to learn more about how the open source way is improving technologies, education, business, government, health, law, entertainment, humanitarian efforts, and more.

Submit a story idea: opensource.com/story

Email us: open@opensource.com

open
source
.com

Supported by
Red Hat

KEVIN SONNEY

KEVIN SONNEY IS A TECHNOLOGY PROFESSIONAL, MEDIA PRODUCER, AND PODCASTER. A Linux Sysadmin and Open Source advocate, Kevin has over 25 years in the IT industry, with over 15 years in Open Source.

Kevin hosts the weekly Productivity Alchemy Podcast. He and his wife, author and illustrator Ursula Vernon, co-host the weekly podcast Kevin and Ursula Eat Cheap (NSFW) and routinely attend sci-fi and comic conventions. Kevin also voices Rev. Mord on The Hidden Almanac.

FOLLOW KEVIN SONNEY

Twitter: @ksonney

INTRODUCTION

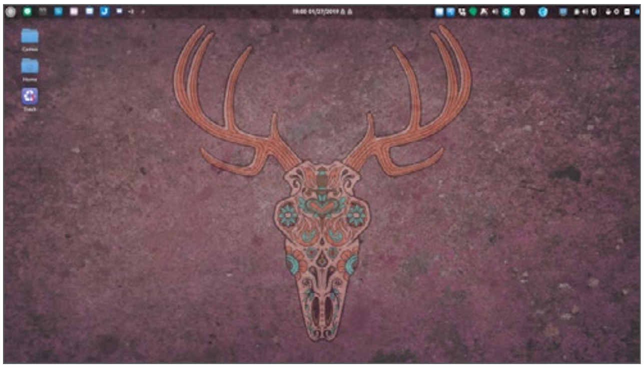
CHAPTERS

Introduction

IN OUR BUSY WORLD, people seem to constantly be searching for ways to be more productive. The usual round of recommendations is heavily biased towards closed source and proprietary software. it doesn’t have to be that way. Here are my picks for open source desktop tools to help increase efficiency so you have time be more productive.

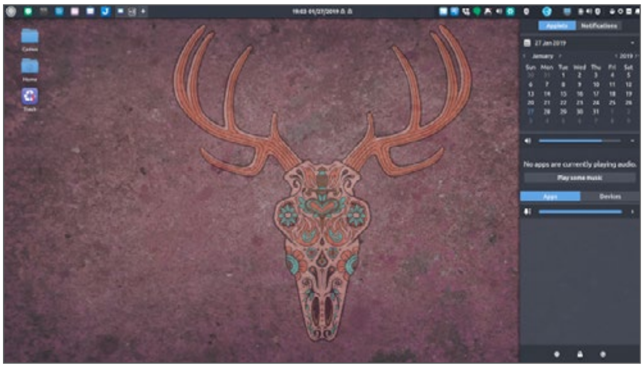
Budgie Desktop, a Linux environment

THERE ARE MANY, many desktop environments for Linux. From the easy to use and graphically stunning GNOME desktop [1] (default on most major Linux distributions) and KDE [2], to the minimalist Openbox [3], to the highly configurable tiling i3, there are a lot of options. What I look for in a good desktop environment is speed, unobtrusiveness, and a clean user experience. It is hard to be productive when a desktop works against you, not with or for you.

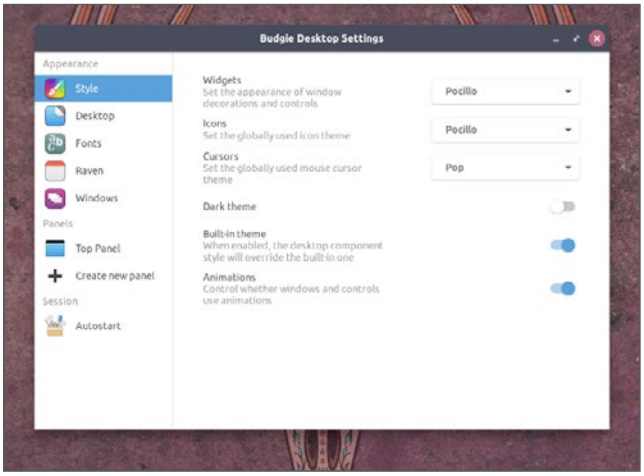


Budgie Desktop is the default desktop on the Solus [4] Linux distribution and is available as an add-on package for most of the major Linux distributions. It is based on GNOME and uses many of the same tools and libraries you likely already have on your computer.

The default desktop is exceptionally minimalistic, with just the panel and a blank desktop. Budgie includes an integrated sidebar (called Raven) that gives quick access to the calendar, audio controls, and settings menu. Raven also contains an integrated notification area with a unified display of system messages similar to MacOS's.



Clicking on the gear icon in Raven brings up Budgie's control panel with its configuration settings. Since Budgie is still in development, it is a little bare-bones compared to GNOME or KDE, and I hope it gets more options over time. The Top Panel option, which allows the user to configure the ordering, positioning, and contents of the top panel, is nice.



Budgie Desktop Settings

Widgets — Set the appearance of window decorations and controls: Pocillo

Icons — Set the globally used icon theme: Pocillo

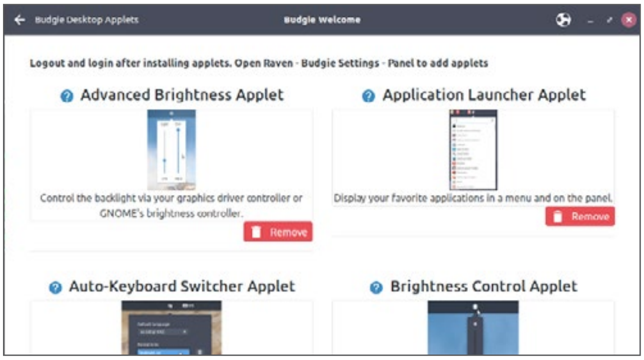
Cursors — Set the globally used mouse cursor theme: Pop

Dark theme: off

Built-in theme — When enabled, the desktop component style will override the built-in one: on

Animations — Control whether windows and controls use animations: on

The Budgie Welcome application (presented at first login) contains options to install additional software, panel applets, snaps, and Flatpack packages. There are applets to handle networking, screenshots, additional clocks and timers, and much, much more.



Logout and login after installing applets. Open Raven - Budgie Settings - Panel to add applets

Advanced Brightness Applet — Control the backlight via your graphics driver controller or GNOME's brightness controller.

Application Launcher Applet — Display your favorite applications in a menu and on the panel.

Auto-Keyboard Switcher Applet

Brightness Control Applet

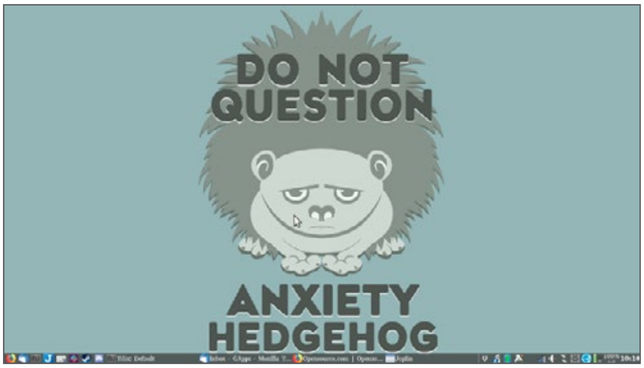
Budgie provides a desktop that is clean and stable. It responds quickly and has many options that allow you to customize it as you see fit..

Links

[1] https://www.gnome.org/
[2] https://www.kde.org/
[3] http://openbox.org/wiki/Main_Page
[4] https://getsol.us/solus/experiences/

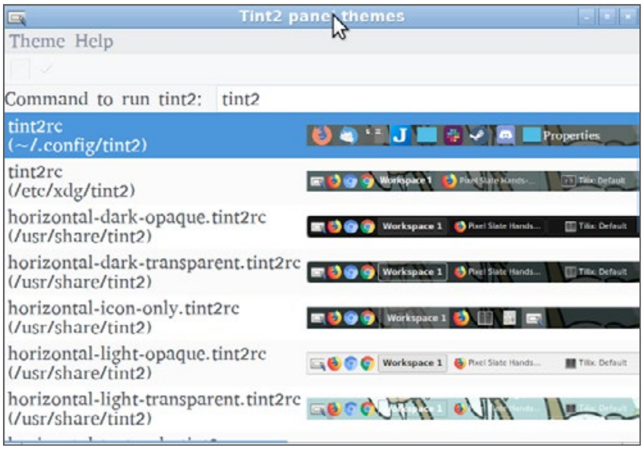
Tint2, an open source taskbar for Linux

ONE OF THE BEST WAYS for me to be more productive is to use a clean interface with as little distraction as possible. As a Linux user, this means using a minimal window manager like Openbox [1], i3 [2], or Awesome [3]. Each has customization options that make me more efficient. The one thing that slows me down is that none has a consistent configuration, so I have to tweak and re-tune my window manager constantly.



Tint2 [4] is a lightweight panel and taskbar that provides a consistent experience with any window manager. It is included with most distributions, so it is as easy to install as any other package.

It includes two programs, Tint2 and Tint2conf. At first launch, Tint2 starts with its default layout and theme. The default configuration includes multiple web browsers, the tint2conf program, a taskbar, and a system tray.



Tint2 panel themes

Theme Help

Command to run tint2: tint2

tint2rc
(~/.config/tint2)

tint2rc
(/etc/xdg/tint2)

horizontal-dark-opaque.tint2rc
(/usr/share/tint2)

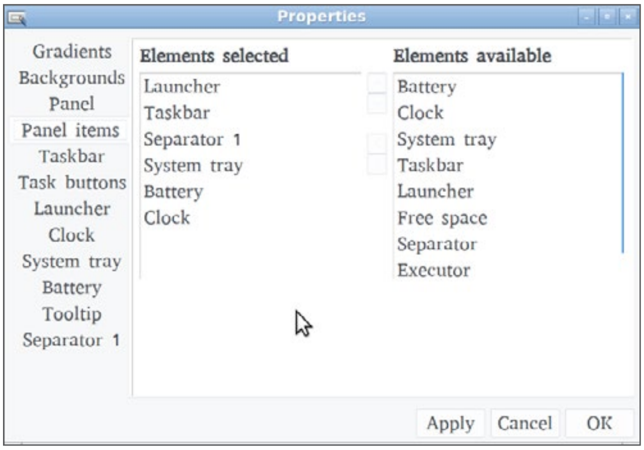
horizontal-dark-transparent.tint2rc
(/usr/share/tint2)

horizontal-icon-only.tint2rc
(/usr/share/tint2)

horizontal-light-opaque.tint2rc
(/usr/share/tint2)

horizontal-light-transparent.tint2rc
(/usr/share/tint2)

Launching the configuration tool allows you to select from the included themes and customize the top, bottom, and sides of the screen. I recommend starting with the theme that is closest to what you want and customizing from there.



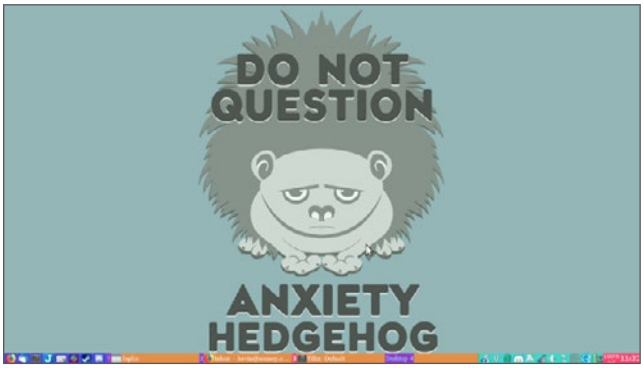
Properties

Gradients
Backgrounds
Panel
Panel items
Taskbar
Task buttons
Launcher
Clock
System tray
Battery
Tooltip
Separator 1

Elements selected: Launcher, Taskbar, Separator 1, System tray, Battery, Clock

Elements available: Battery, Clock, System tray, Taskbar, Launcher, Free space, Separator, Executor

Within the themes, you can customize where panel items are placed as well as background and font options for every item on the panel. You can also add and remove items from the launcher.



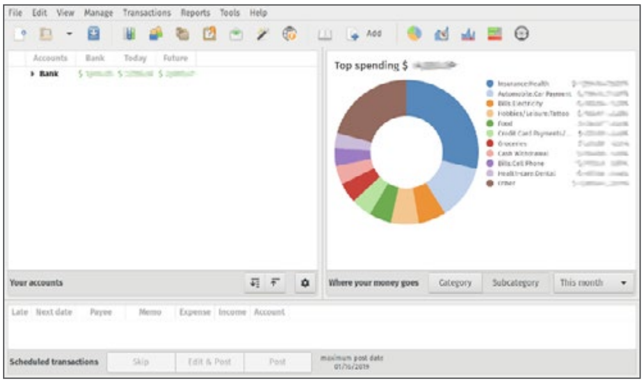
Tint2 is a lightweight taskbar that helps you get to the tools you need quickly and efficiently. It is highly customizable, unobtrusive (unless the user wants it not to be), and compatible with almost any window manager on a Linux desktop.

Links

[1] http://openbox.org/wiki/Main_Page
[2] https://i3wm.org/
[3] https://awesomewm.org/
[4] https://gitlab.com/o9000/tint2

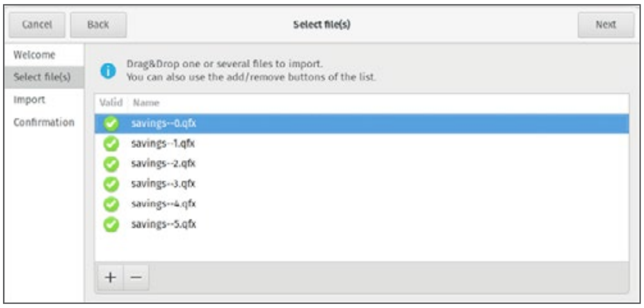
HomeBank, an open source personal finance app

MANAGING MY FINANCES can be really stressful. I don't look at my bank balance every day and sometimes have trouble keeping track of where my money is going. I often spend more time managing my finances than I need to, digging into accounts and payment histories to figure out where my money went. Knowing my finances are OK helps keep me calm and allows me to focus on other things.



HomeBank [1] is a personal finance desktop application that helps decrease this type of stress by making it fairly easy to keep track of your finances. It has some nice reports to help you figure out where you're spending your money, allows you to set up rules for importing transactions, and supports most modern formats.

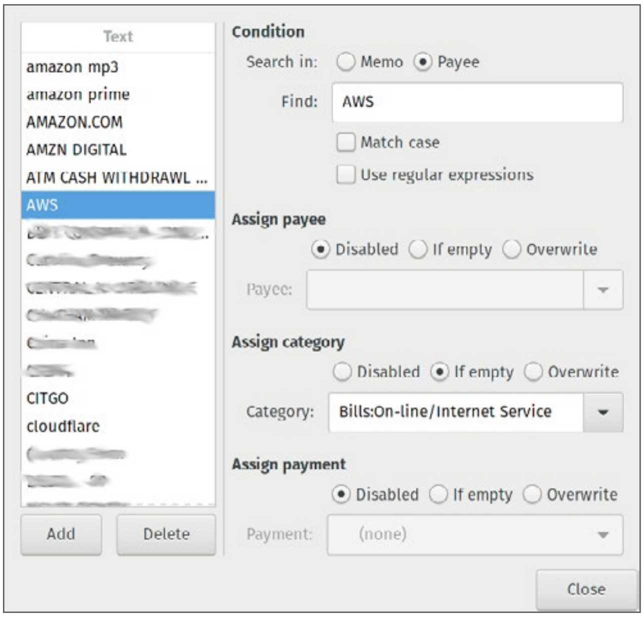
HomeBank is available on most distributions by default, so installation is very easy. When you start it up for the first time, it will walk you through setup and allow you to create an account. From there, you can either import one of the supported file formats or start entering transactions. The transaction register itself is just that—a list of transactions. Unlike some other apps [2], you don't have to learn double-entry bookkeeping [3] to use HomeBank.



Drag&Drop one or several files to import.
You can also use the add/remove buttons of the list.

- ✓ savings--0.qfx
- ✓ savings--1.qfx
- ✓ savings--2.qfx
- ✓ savings--3.qfx
- ✓ savings--4.qfx
- ✓ savings--5.qfx

Importing files from your bank is handled with another step-by-step wizard, with options to create a new account or populate an existing one. Importing into a new account saves a little time since you don't have to pre-create all the accounts before starting the import. You can also import multiple files into an account at once, so you don't need to repeat the same steps for every file in every account.



amazon mp3
amazon prime
AMAZON.COM
AMZN DIGITAL
ATM CASH WITHDRAWL ...
AWS
[redacted]
CITGO
cloudflare
[redacted]

Search in: ○ Memo ● Payee
Find: AWS
□ Match case
□ Use regular expressions

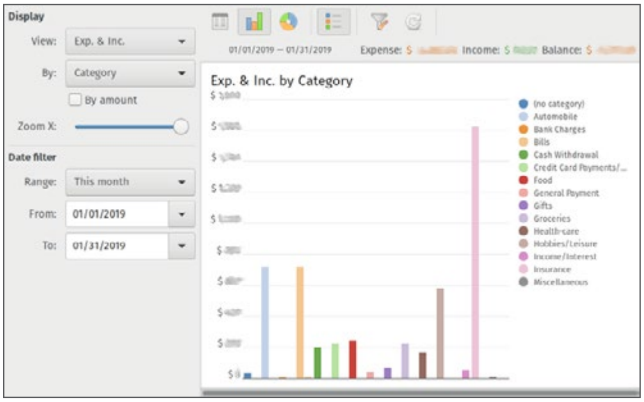
● Disabled ○ If empty ○ Overwrite
Payee:

○ Disabled ● If empty ○ Overwrite
Category: Bills:On-line/Internet Service

● Disabled ○ If empty ○ Overwrite
Payment: (none)

The one pain point I've had with importing and managing accounts is category assignment. Categories are what allow you to break down your spending and see what you are spending money on, in general terms. HomeBank, unlike commercial services (and some commercial programs), requires you to manually set up all the assignments. But this is generally a one-time thing, and then the categories can be auto-applied as transactions are added/imported. There is also a button to analyze the account and auto-apply things that already exist, which speeds up categorizing a large import (like I did the first time). HomeBank comes with a large number of categories you can start with, and you can add your own as well.

HomeBank also has budgeting features, allowing you to plan for the months ahead.



Display

View: Exp. & Inc.

By: Category

☐ By amount

Zoom X:

Date filter

Range: This month

From: 01/01/2019

To: 01/31/2019

01/01/2019 – 01/31/2019 Expense: $ [redacted] Income: $ [redacted] Balance: $ [redacted]

The big win, for me, is HomeBank's reports feature. Not only is there a chart on the main screen showing where you are spending your money, but there are a whole host of other reports you can look at. If you use the budget feature, there is a report that tracks your spending against your budget. You can also view those reports as pie and bar charts. There is also a trend report and a balance report, so you can look back and see changes or patterns over time.

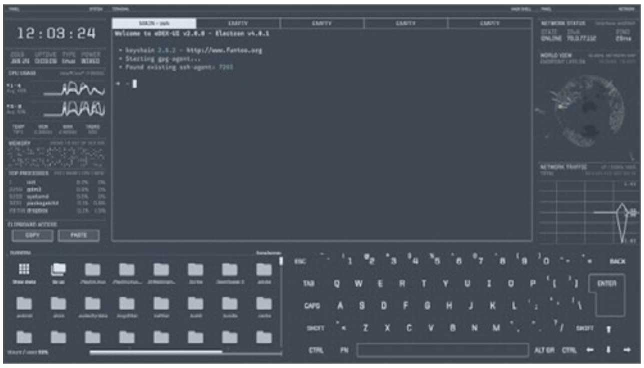
Overall, HomeBank is a very friendly, useful application to help you keep your finances in order. It is simple to use and really helpful if keeping track of your money is a major stress point in your life.

Links

[1] http://homebank.free.fr/en/index.php
[2] https://www.gnucash.org/
[3] https://en.wikipedia.org/wiki/Double-entry_bookkeeping_system

eDEX-UI, a Tron-influenced terminal program for tablets and desktops

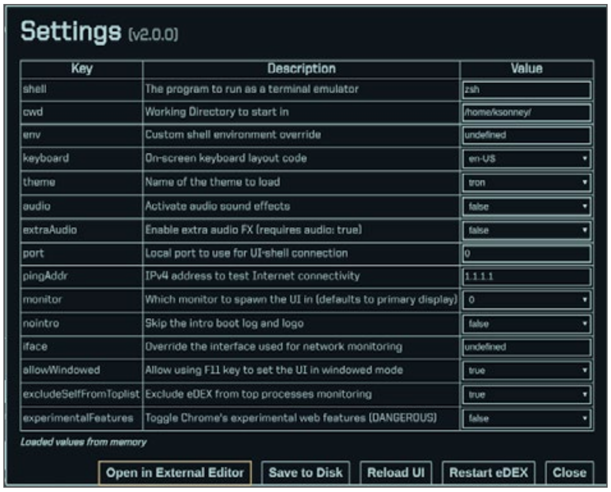
I WAS 11 YEARS old when *Tron* [1] was in movie theaters. I cannot deny that, despite the fantastical nature of the film, it had an impact on my career choice later in life.



```
Welcome to eDEX-UI v2.0.0 - Electron v4.0.1

 * keychain 2.8.2 ~ http://www.funtoo.org
 * Starting gpg-agent...
 * Found existing ssh-agent: 7203

→ ~ ▌
```

eDEX-UI [2] is a cross-platform terminal program designed for tablets and desktops that was inspired by the user interface in *Tron*. It has five terminals in a tabbed interface, so it is easy to switch between tasks, as well as useful displays of system information.

At launch, eDEX-UI goes through a boot sequence with information about the ElectronJS system it is based on. After the boot, eDEX-UI shows system information, a file browser, a keyboard (for tablets), and the main terminal tab. The other four tabs (labeled EMPTY) don't have anything loaded and will start a shell when you click on one. The default shell in eDEX-UI is Bash (if you are on Windows, you will likely have to change it to either PowerShell or cmd.exe).

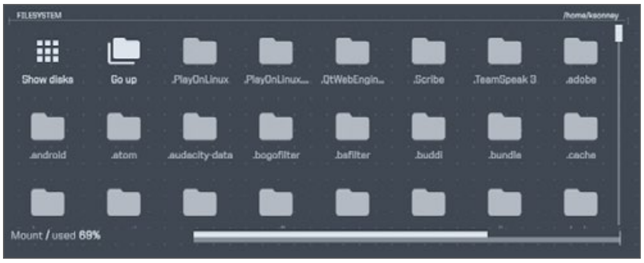


Settings (v2.0.0)

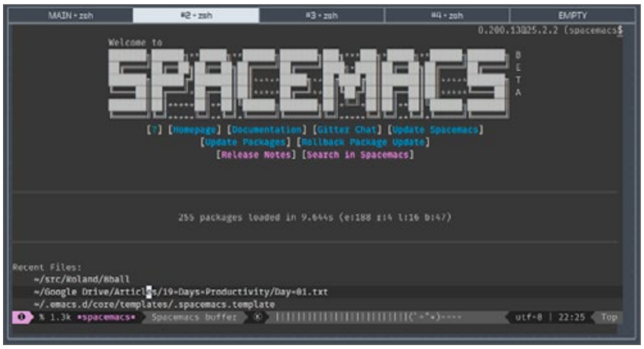
Key	Description	Value
shell	The program to run as a terminal emulator	zsh
cwd	Working Directory to start in	/home/ksonney/
env	Custom shell environment override	undefined
keyboard	On-screen keyboard layout code	en-US ▾
theme	Name of the theme to load	tron ▾
audio	Activate audio sound effects	false ▾
extraAudio	Enable extra audio FX (requires audio: true)	false ▾
port	Local port to use for UI-shell connection	0
pingAddr	IPv4 address to test Internet connectivity	1.1.1.1
monitor	Which monitor to spawn the UI in (defaults to primary display)	0 ▾
nointro	Skip the intro boot log and logo	false ▾
iface	Override the interface used for network monitoring	undefined
allowWindowed	Allow using F11 key to set the UI in windowed mode	true ▾
excludeSelfFromToplist	Exclude eDEX from top processes monitoring	true ▾
experimentalFeatures	Toggle Chrome's experimental web features (DANGEROUS)	false ▾

Loaded values from memory

Changing directories in the file browser will change directories in the active terminal and vice-versa. The file browser does everything you'd expect, including opening associated applications when you click on a file. The one exception is eDEX-UI's settings.json file (in .config/eDEX-UI by default), which opens the configuration editor instead. This allows you to set the shell command for the terminals, change the theme, and modify several other settings for the user interface. Themes are also stored in the configuration directory and, since they are also JSON files, creating a custom theme is pretty straightforward.



eDEX-UI allows you to run five terminals with full emulation. The default terminal type is xterm-color, meaning it has full-color support. One thing to be aware of is that the keys light up on the keyboard while you type, so if you're using eDEX-UI on a tablet, the keyboard could present a security risk in environments where people can see the screen. It is better to use a theme without the keyboard on those devices, although it does look pretty cool when you are typing.



```
                                                   0.200.13025.2.2 (spacemacs$
           Welcome to
           SPACEMACS  BETA

  [?] [Homepage] [Documentation] [Gitter Chat] [Update Spacemacs]
           [Update Packages] [Rollback Package Update]
              [Release Notes] [Search in Spacemacs]

         255 packages loaded in 9.644s (e:188 r:4 l:16 b:47)

Recent Files:
    ~/src/Roland/8ball
    ~/Google Drive/Articles/19-Days-Productivity/Day-01.txt
    ~/.emacs.d/core/templates/.spacemacs.template
 0  % 1.3k *spacemacs*  Spacemacs buffer  ⓚ  |||||||||||||||||||||||||(*^‿^*)----   utf-8 | 22:25  Top
```

While eDEX-UI supports only five terminal windows, that has been more than enough for me. On a tablet, eDEX-UI gives me that cyberspace feel without impacting my productivity. On a desktop, eDEX-UI allows all of that and lets me look cool in front of my co-workers.

Links

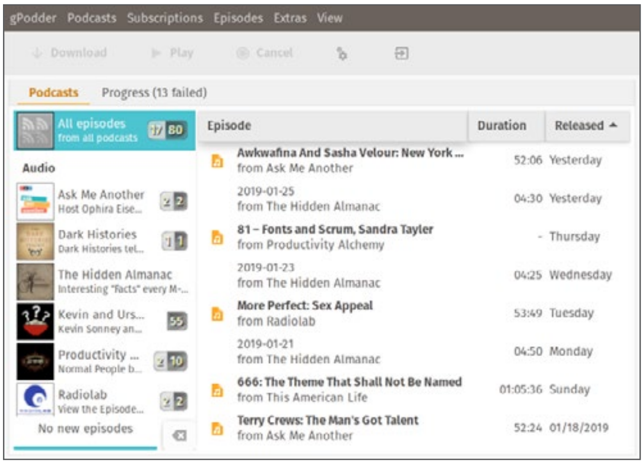
[1] https://en.wikipedia.org/wiki/Tron

[2] https://github.com/GitSquared/edex-ui

gPodder, an open source podcast client

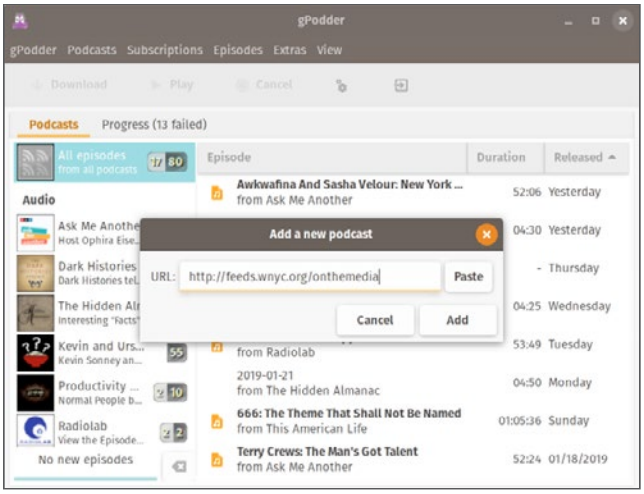
I LIKE PODCASTS. Heck, I like them so much I record three of them (you can find links to them in my profile [1]). I learn a lot from podcasts and play them in the background when I'm working. But keeping them in sync between multiple desktops and mobile devices can be a bit of a challenge.

gPodder [2] is a simple, cross-platform podcast downloader, player, and sync tool. It supports RSS feeds, FeedBurner [3], YouTube [4], and SoundCloud [5], and it also has an open source sync service that you can run if you want. gPodder doesn't do podcast playback; instead, it uses your audio or video player of choice.



Episode	Duration	Released ▲
Awkwafina And Sasha Velour: New York ... from Ask Me Another	52:06	Yesterday
2019-01-25 from The Hidden Almanac	04:30	Yesterday
81 – Fonts and Scrum, Sandra Tayler from Productivity Alchemy	-	Thursday
2019-01-23 from The Hidden Almanac	04:25	Wednesday
More Perfect: Sex Appeal from Radiolab	53:49	Tuesday
2019-01-21 from The Hidden Almanac	04:50	Monday
666: The Theme That Shall Not Be Named from This American Life	01:05:36	Sunday
Terry Crews: The Man's Got Talent from Ask Me Another	52:24	01/18/2019

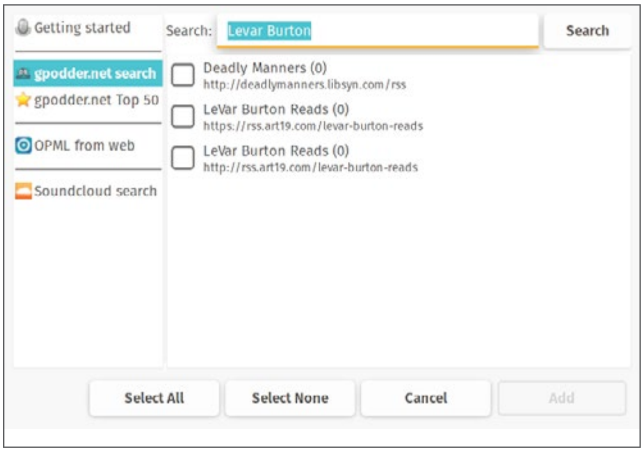
Installing gPodder is very straightforward. Installers are available for Windows and MacOS, and packages are available for major Linux distributions. If it isn't available in your distribution, you can run it directly from a Git checkout. With the "Add Podcasts via URL" menu option, you can enter a podcast's RSS feed URL or one of the "special" URLs for the other services. gPodder will fetch a list of episodes and present a dialog where you can select which episodes to download or mark old episodes on the list.



URL: http://feeds.wnyc.org/onthemedia

Episode	Duration	Released ▲
Awkwafina And Sasha Velour: New York ... from Ask Me Another	52:06	Yesterday
	04:30	Yesterday
	-	Thursday
	04:25	Wednesday
from Radiolab	53:49	Tuesday
2019-01-21 from The Hidden Almanac	04:50	Monday
666: The Theme That Shall Not Be Named from This American Life	01:05:36	Sunday
Terry Crews: The Man's Got Talent from Ask Me Another	52:24	01/18/2019

One of its nicer features is that if a URL is already in your clipboard, gPodder will automatically place it in its URL field, which makes it really easy to add a new podcast to your list. If you already have an OPML file of podcast feeds, you can upload and import it. There is also a discovery option that allows you to search for podcasts on gPodder.net [6], the free and open source podcast listing site by the people who write and maintain gPodder.



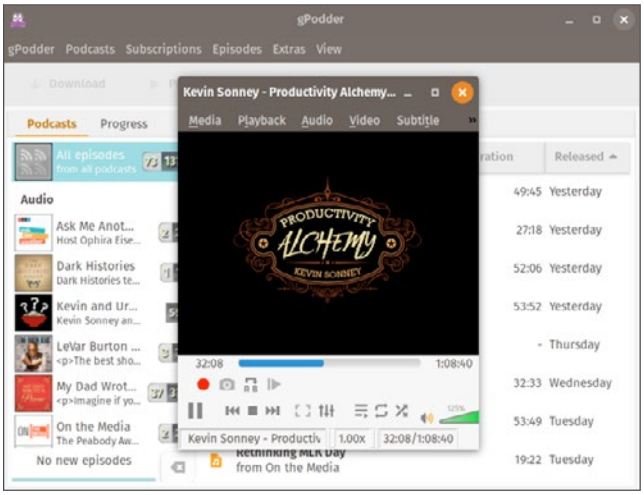
Search: Levar Burton

☐ Deadly Manners (0)
http://deadlymanners.libsyn.com/rss

☐ LeVar Burton Reads (0)
https://rss.art19.com/levar-burton-reads

☐ LeVar Burton Reads (0)
http://rss.art19.com/levar-burton-reads

A mygpo [7] server synchronizes podcasts between devices. By default, gPodder uses gPodder.net's [8] servers, but you can change this in the configuration files if want to run your own (be aware that you'll have to modify the configuration file directly). Syncing allows you to keep your lists consistent between desktops and mobile devices. This is very useful if you listen to podcasts on multiple devices (for example, I listen on my work computer, home computer, and mobile phone), as it means no matter where you are, you have the most recent lists of podcasts and episodes without having to set things up again and again.



Clicking on a podcast episode will bring up the text post associated with it, and clicking "Play" will launch your device's default audio or video player. If you want to use something other than the default, you can change this in gPodder's configuration settings.

gPodder makes it simple to find, download, and listen to podcasts, synchronize them across devices, and access a lot of other features in an easy-to-use interface.

Links

[1] https://opensource.com/users/ksonney
[2] https://gpodder.github.io/
[3] https://feedburner.google.com/
[4] https://youtube.com/
[5] https://soundcloud.com/
[6] http://gpodder.net/
[7] https://github.com/gpodder/mygpo
[8] http://gpodder.net/

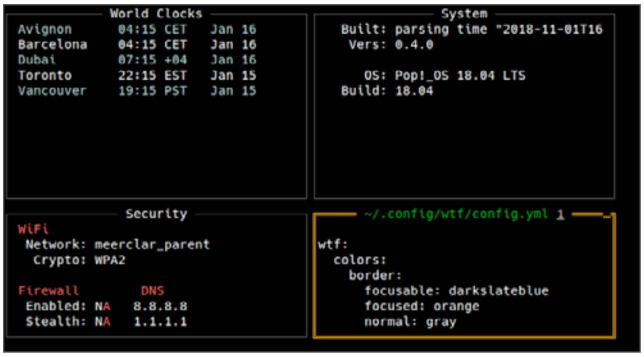
WTF, a dashboard for the terminal

ONCE UPON A TIME, I was doing some consulting at a firm that used Bloomberg Terminals [1]. My reaction was, "Wow, that's WAY too much information on one screen." These days, however, it seems like I can't get enough information on a screen when I'm working and have multiple web pages, dashboards, and console apps open to try to keep track of things.

While tmux [2] and Screen [3] can do split screens and multiple windows, they are a pain to set up, and the key-bindings can take a while to learn (and often conflict with other applications).

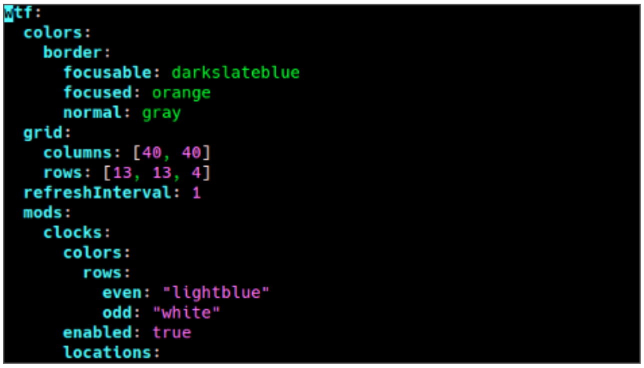
WTF [4] is a simple, easily configured information dashboard for the terminal. It is written in Go [5], uses a YAML configuration file, and can pull data from several different sources. All the data sources are contained in modules [6] and include things like weather, issue trackers, date and time, Google Sheets, and a whole lot more. Some panes are interactive, and some just update with the most recent information available.

Setup is as easy as downloading the latest release for your operating system and running the command. Since it is written in Go, it is very portable and should run anywhere you can compile it (although the developer only builds for Linux and MacOS at this time).



```
                 World Clocks                                     System
 Avignon       04:15 CET    Jan 16          Built: parsing time "2018-11-01T16
 Barcelona     04:15 CET    Jan 16           Vers: 0.4.0
 Dubai         07:15 +04    Jan 16
 Toronto       22:15 EST    Jan 15              OS: Pop!_OS 18.04 LTS
 Vancouver     19:15 PST    Jan 15          Build: 18.04

                   Security                          ~/.config/wtf/config.yml 1
 WiFi                                       wtf:
  Network: meerclar_parent                    colors:
   Crypto: WPA2                                 border:
                                                  focusable: darkslateblue
 Firewall       DNS                               focused: orange
  Enabled: NA   8.8.8.8                           normal: gray
  Stealth: NA   1.1.1.1
```

When you run WTF for the first time, you'll get the default screen, identical to the image above.

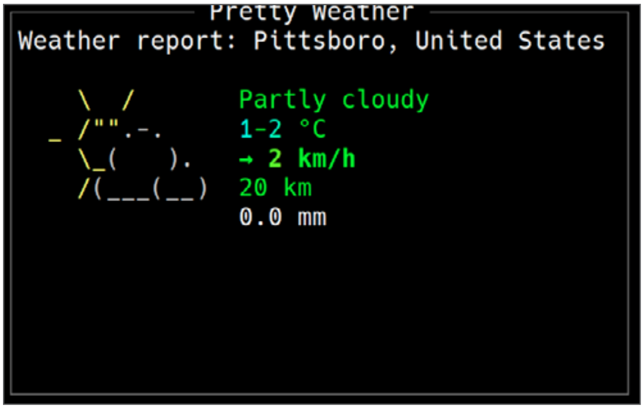


```
wtf:
  colors:
    border:
      focusable: darkslateblue
      focused: orange
      normal: gray
  grid:
    columns: [40, 40]
    rows: [13, 13, 4]
  refreshInterval: 1
  mods:
    clocks:
      colors:
        rows:
          even: "lightblue"
          odd: "white"
      enabled: true
      locations:
```

You also get the default configuration file in **~/.wtf/config.yml**, and you can edit the file to suit your needs. The grid layout is configured in the top part of the file.

```
grid:
  columns: [45, 45]
  rows: [7, 7, 7, 4]
```

The numbers in the grid settings represent the character dimensions of each block. The default configuration is two columns of 40 characters, two rows 13 characters tall, and one row 4 characters tall. In the code above, I made the columns wider (45, 45), the rows smaller, and added a fourth row so I can have more widgets.



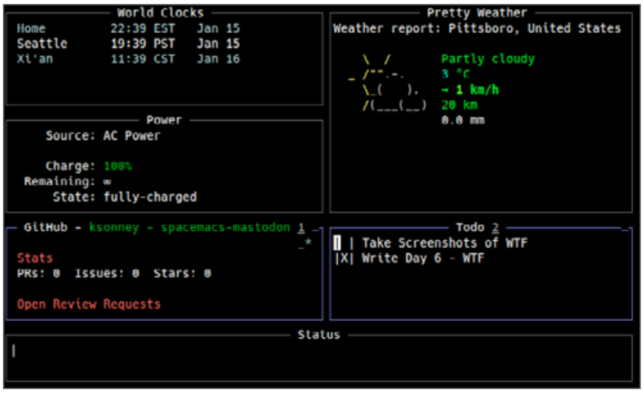
```
                Pretty Weather
Weather report: Pittsboro, United States

     \  /        Partly cloudy
   _ /"".-.      1-2 °C
     \_(   ).    → 2 km/h
     /(___(__)   20 km
                 0.0 mm
```

I like to see the day's weather [7] on my dashboard. There are two weather modules to chose from: Weather, which shows just the text information, and Pretty Weather [8], which is colorful and uses text-based graphics in the display.

```
prettyweather:
  enabled: true
  position:
    top: 0
    left: 1
    height: 2
    width: 1
```

This code creates a pane two blocks tall (height: 2) and one block wide (height: 1), positioned on the second column (left: 1) on the top row (top: 0) containing the Pretty Weather module.

Some modules, like Jira, GitHub, and Todo, are interactive, and you can scroll, update, and save information in them. You can move between the interactive panes using the Tab key. The \ key brings up a help screen for the active pane so you can see what you can do and how. The Todo module lets you add, edit, and delete to-do items, as well as check them off as you complete them.



```
World Clocks
Home          22:39 EST   Jan 15
Seattle       19:39 PST   Jan 15
Xi'an         11:39 CST   Jan 16

Power
    Source: AC Power

    Charge: 100%
 Remaining: ∞
     State: fully-charged

GitHub - ksonney - spacemacs-mastodon 1
                                       _*
Stats
PRs: 0  Issues: 0  Stars: 0

Open Review Requests
```

```
Pretty Weather
Weather report: Pittsboro, United States

    \  /       Partly cloudy
  _ /"".-.     3 °C
    \_(   ).   → 1 km/h
    /(___(__)  20 km
               0.0 mm

Todo 2
[ ] | Take Screenshots of WTF
|X| Write Day 6 - WTF
```

```
Status
|
```

There are also modules to execute commands and present the output, watch a text file, and monitor build and integration server output. All the documentation is very well done.

WTF is a valuable tool for anyone who needs to see a lot of data on one screen from different sources.

Links

[1] https://en.wikipedia.org/wiki/Bloomberg_Terminal

[2] https://github.com/tmux/tmux

[3] https://www.gnu.org/software/screen/

[4] https://wtfutil.com/

[5] https://golang.org/

[6] https://wtfutil.com/modules/

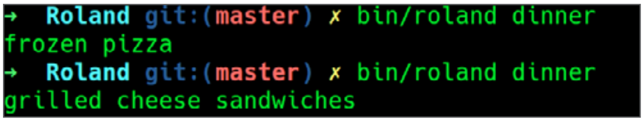
[7] https://wtfutil.com/modules/weather_services/weather/

[8] https://wtfutil.com/modules/weather_services/prettyweather/

Roland, a random selection tool for the command line

BY THE TIME THE WORKDAY has ended, often the only thing I want to think about is hitting the couch and playing the video game of the week. But even though my professional obligations stop at the end of the workday, I still have to manage my household. Laundry, pet care, making sure my teenager has what he needs, and most important: deciding what to make for dinner.

Like many people, I often suffer from decision fatigue [1], and I make less-than-healthy choices for dinner based on speed, ease of preparation, and (quite frankly) whatever causes me the least stress.



```
→  Roland git:(master) ✗ bin/roland dinner
frozen pizza
→  Roland git:(master) ✗ bin/roland dinner
grilled cheese sandwiches
```

Roland [2] makes planning my meals much easier. Roland is a Perl application designed for tabletop role-playing games. It picks randomly from a list of items, such as monsters and hirelings. In essence, Roland does the same thing at the command line that a game master does when rolling physical dice to look up things in a table from the *Game Master's Big Book of Bad Things to Do to Players*.

With minor modifications, Roland can do so much more. For example, just by adding a table, I can enable Roland to help me choose what to cook for dinner.

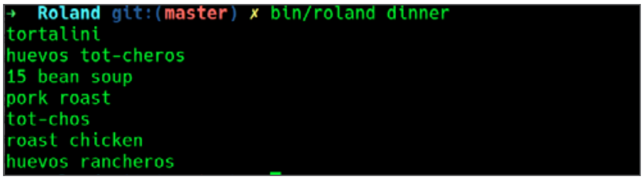
The first step is installing Roland and all its dependencies.

```
git clone git@github.com:rjbs/Roland.git
cpan install Getopt::Long::Descriptive Moose \
   namespace::autoclean List:AllUtils Games::Dice \
   Sort::ByExample Data::Bucketeer Text::Autoformat \
   YAML::XS
cd oland
```

Next, I create a YAML document named dinner and enter all our meal options.

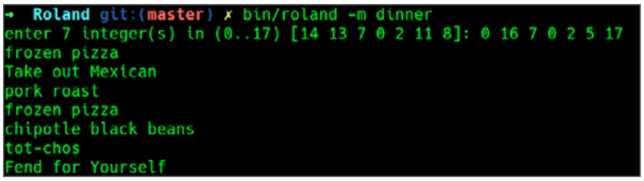
```
type: list
pick: 1
items:
- "frozen pizza"
 - "chipotle black beans"
 - "huevos rancheros"
 - "nachos"
 - "pork roast"
 - "15 bean soup"
 - "roast chicken"
 - "pot roast"
 - "grilled cheese sandwiches"
```

Running the command **bin/roland** dinner will read the file and pick one of the options.



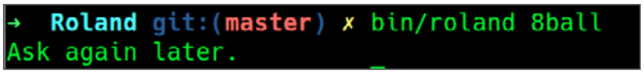
```
→  Roland git:(master) ✗ bin/roland dinner
tortalini
huevos tot-cheros
15 bean soup
pork roast
tot-chos
roast chicken
huevos rancheros
```

I like to plan for the week ahead so I can shop for all my ingredients in advance. The **pick** command determines how many items from the list to chose, and right now, the **pick** option is set to 1. If I want to plan a full week's dinner menu, I can just change **pick: 1** to **pick: 7** and it will give me a week's worth of dinners. You can also use the **-m** command line option to manually enter the choices.



```
→  Roland git:(master) ✗ bin/roland -m dinner
enter 7 integer(s) in (0..17) [14 13 7 0 2 11 8]: 0 16 7 0 2 5 17
frozen pizza
Take out Mexican
pork roast
frozen pizza
chipotle black beans
tot-chos
Fend for Yourself
```

You can also do fun things with Roland, like adding a file named **8ball** with some classic phrases.



```
→  Roland git:(master) ✗ bin/roland 8ball
Ask again later.
```

You can create all kinds of files to help with common decisions that seem so stressful after a long day of work. And even if you don't use it for that, you can still use it to decide which devious trap to set up for tonight's game.

Links

[1] https://en.wikipedia.org/wiki/Decision_fatigue
[2] https://github.com/rjbs/Roland