PRACTICAL PORTS

Want Some Toppings on Your Desk?

BY BENEDICT REUSCHLING

This column covers ports and packages for FreeBSD that are useful in some way, peculiar, or otherwise good to know about. Ports extend the base OS functionality and make sure you get something done or, simply, put a smile on your face. Come along for the ride, maybe you'll find something new.

Years ago, when I began my Unix journey as a university freshman, installers were a lot simpler. Getting to a desktop at the end was not the default. In fact, I struggled a lot at the beginning to get one going, and when I did, the computer I had back then did not have enough power to run it. So, I stayed in the terminal for a long time. This did not turn me away, as I could already do a lot more than on DOS. At least I had misc/mc to get me a little bit more than white text on a black background.

Later, I did get a working desktop running on X11R6. Nowadays, the installers are much better and often default to a graphical point-and-click interface. But when that happened, I was switching to FreeBSD where a curses-like installer is still the default. My hard-learned tricks in the non-GUI world proved useful. The FreeBSD handbook had me starting up a working desktop much faster than my stunts on other distributions. I still use a minimal desktop environment because, most of the time, I need to open a terminal to be productive. I used x11/icewm initially, then switched to x11/fluxbox. Then there was a period during my university studies where all the other students had flashy desktop effects like desktop-switching on a rotating cube, transparency, and such. It got boring after a while, because everyone had them, so they ended up being nothing special. Since I couldn't get them to work back then, I wanted to save face and ran x11-wm/enlightenment. That was interesting, because who does not compile config files into binaries to get some speed benefits, right? I guess some concepts don't catch on with other software systems—as revolutionary they might be.

Coming from Windows before I started my Unix experiences, I had certain expectations of a desktop. How do you launch applications if you don't have any icons on the desktop? And while we're at it: how do I put a flashy background image on fluxbox (which I switched back to)? I can't work this way! I mean, transparent terminal programs at least show you parts of it when you are not browsing the web. OK, there are plenty of terminal programs available that can do that, but most terminal emulators come with a big shopping list of dependencies. No thanks, I'll pass. Speaking of emulation, how about a trip to the days of cathode ray tubes that ruined our eyes? With x11/cool-retro-term nothing stops you from reliving those days. And don't blame me for your next eye appointment!

There was also the problem of launching applications without icons (and less than favorable quick launch menus). Having seen what the Mac could do with Alfred (alfredapp.com), it was only a matter of time until someone sat down and built something similar for the Unix desktop. Since the name of Bruce Wayne's butler was already taken, the next best name was x11/albert, of course. For those who want a good file manager, I hear good reports from x11-fm/dolphin in combination with devel/dolphin-plugins. Even x11-fm/xfe gets you running wild in your filesystem with a small runtime dependency list. For those who don't want to run a desktop to have terminals side by side, there is always sysutils/screen or sysutils/tmux at your side. Detaching and re-attaching screens and sessions on a server (which does not run a desktop unless you're on Windows) is common these days. When it starts to get complex, sysutils/tmuxinator can help you manage those sessions with relative ease. Or look at sysutils/byobu for a bit more elegance and system status notifications.

How do you launch applications if you don't have any icons on the desktop?

A while ago, I started getting into tiling window managers (perhaps because of liking tmux a little too much). Not only did that kill the "icons on the desktop" idea, it also impacted my habit of typing rather than pushing around a mouse all day. While getting an x11/i3 desktop running alongside x11/i3status, x11/py-i3-quickterm, I also needed to learn some new keystrokes to control it. The experience has been good so far and x11/i3-gaps looks like my next upgrade from there. But now that I have discovered that whole new world of mouseless desktopping, I have also learned of x11/rofi. Rofi is an application launcher like Albert. Give it a try, run rofi and it shows you what it can do. For example, ssh into one of those boxes with ease. Or browse your files or executables like on Apple's Spotlight application.

There is always a debate about work you can do better on a desktop versus in the terminal. "Surely you can't do graphics editing in the console?!" I hear skeptics say. Well, that depends on what needs doing. You can go a long way with graphics/ImageMagick7. Starting from cropping, rotating, and putting watermarks on images, or putting your scanned signature on a contract. You don't need to run graphics/gimp for that. Does it replace the Paintshop Pro experience of old? Probably not. But hey, most changes are small and easy to batch up in the terminal. Who wants to manually rotate all those holiday images from years ago taken by a shabby flip-phone camera?

At work, I discovered that some PDF meeting notes were not searchable. It turned out they were printed, signed by participants, and scanned into a PDF. That killed the full-text search and copy-and-paste as the PDF was an image now. I had discovered textproc/ocrmypdf which basically does the reverse. It applies OCR software and machine learning to restore the text portions of a PDF and converts it back to text. Since we had many such documents, batching them up in the terminal was quick work. I have not dared to think how long it would take to manually fix that, desktop program available or not. Use the right tool for the job, even if it's a desktop-based one. And while my own journey in desktop-land continues, I'll record my software findings for you along the way.

BENEDICT REUSCHLING is a documentation committer in the FreeBSD project and member of the documentation engineering team. He serves on the board of directors of the FreeBSD Foundation as vice president. In the past, he served on the FreeBSD core team for two terms. He administers a big data cluster at the University of Applied Sciences, Darmstadt, Germany. He's also teaching a course "Unix for Developers" for undergraduates. Together with Allan Jude, he is host of the weekly bsdnow.tv podcast.