

GhostBSD: From Usability to Struggle and Renewal

BY ERIC TURGEON

This article isn't meant to be technical. Instead, it offers a high-level view of what happened through the years with GhostBSD, where the project stands today, and where we want to take it next. As you may know, GhostBSD is a user-friendly desktop BSD operating system built with FreeBSD. Its mission is to deliver a simple, stable, and accessible desktop experience for users who want FreeBSD's power without the complexity of manual setup. I started this journey as a non-technical user. I dreamed of a BSD that anyone could use.

The Beginning of GhostBSD

In 2007, I read Eric S. Raymond's *How To Become A Hacker*. It pointed to BSD Unix as a place for aspiring contributors to learn and grow. It sparked my curiosity about BSD, which led me to explore FreeBSD. At that time, I was using Ubuntu, and the question emerged: could FreeBSD become a desktop OS for non-technical users like Ubuntu? At that time, I was a non-technical user myself. I just liked Ubuntu's ease and wondered why FreeBSD couldn't be the same. In 2008, that question ignited my quest to create GhostBSD, and I started to learn everything I needed to create the project to make my vision of FreeBSD as approachable as Ubuntu.

It took me almost two years to figure everything out, experimenting with tools like FreeSBIE to craft a live CD and finding some code to build with GNOME. FreeSBIE was tricky for a beginner to wrangle. As a Canadian French speaker, the *FreeBSD Handbook* was helpful, but my English was limited, and I forced myself to learn it. I dug through forums and docs, piecing together GNOME builds, often breaking thighs more than making things work. Fun fact: the name "GhostBSD" comes from my wife. Back then, she was my girlfriend. She suggested it, and I went with it. We got married in September 2009, just as the project took shape. In November 2009, GhostBSD 1.0 Beta launched as a live CD running GNOME on FreeBSD 8.0. It was rough and full of issues, but it laid the foundation for what GhostBSD has become. Feedback from the FreeBSD community fueled the early progress. Along the way, some people joined to help and teach me about source control versioning and other things. Folks like Ovidiu Anghelescu nudged me toward SVN with his shell scripting know-how. I did learn a lot. We use Git now, but back then, it was all SVN.

I started this journey as a non-technical user. I dreamed of a BSD that anyone could use.

The Early Years

The first release was GhostBSD 1.0 Beta, which gave users a taste of FreeBSD with GNOME. It was a shaky start but a proof of concept. By 2010, version 1.5 added a text-based installer using `pc-sysinstall` from PC-BSD, making setup easier. Those early years involved learning FreeBSD in and out, shell scripting, programming, and from people like Ovidiu Angelescu. I leaned on Ovidiu for shell scripting tips. In 2011, version 2.5 introduced the graphical GBI installer, built on the same `pc-sysinstall` backend. That backend remains a core component today. Its reliability meant I didn't have to reinvent the wheel. It was a lesson in sticking with what works.

The groundwork for NetworkMgr started around 2012 as a GUI tool to manage Ethernet and Wi-Fi connections. It was a step toward usability, inspired by Linux's NetworkManager.

By 2015, I stripped it from `ghostbsd-build` to refine it separately as its tool. A significant shift hit in 2013 with version 3.5. GNOME 3's release was clunky and unstable on FreeBSD. It was laggy and resource-hogging. This situation clashed with GhostBSD's goals. We tried other desktop environments. We created multiple ISOs with different DEs like LXDE, XFCE, and Openbox, and the meaning of GhostBSD, "GNOME hosted on BSD" faced an identity crisis. When MATE, a fork of GNOME 2, emerged, it saved the day. We did switch to MATE for its simplicity and familiarity. It felt like GNOME 2's cozy and at home, running lighter on FreeBSD. At that time, I was unsure what to do with the other desktops when some people just disappeared from the project. I started dropping all the other desktops, but XFCE remained an option. MATE became the flagship, reshaping GhostBSD's focus on usability over Flash.

MATE became the flagship, reshaping GhostBSD's focus on usability over Flash.

System Base Shifts

I started working on Update Station in 2014 to bring GUI updates to GhostBSD users. I thought it was later, but after digging through GitHub, I found that it began taking shape around that time. At first, GhostBSD relied on FreeBSD's release source, tarballs, and official packages. At one point, we needed to start delivering updates for tools like NetworkMgr, which pushed us to build our package repositories. Our custom packages often clashed with FreeBSD version upgrades, creating friction and demanding a better solution. Also, `freebsd-update` was hard to automate for Update Station. `Freebsd-update` didn't fit our GUI-first goal. We started to look at what TrueOS was doing. I noticed the use of PkgBase, and it was intriguing. Their pkg-driven OS updates promised GUI freedom.

In 2018, GhostBSD 18.10 switched to TrueOS as its base. TrueOS offered PkgBase, letting us update the OS with the `pkg` tool, and we ditched `freebsd-update`. It also brought OpenRC, a modern service manager perfect for building a GUI to manage services, but that never happened. The shift to TrueOS allowed Update Station to upgrade software and OS from a graphical interface. It was a game-changer for us and allowed our users to upgrade the OS with Update Station. Later, TrueOS introduced OS ports, allowing us to build OS packages from the ports tree with `poudriere` and giving us finer control over updates.

Bumps in the Road

TrueOS shut down in 2020, putting pressure on us to maintain everything we'd gained. I initially wanted to keep OpenRC, but maintaining all services across ghostbsd-ports and ghostbsd-src became a solo struggle. Around that time, I was alone, primarily maintaining it, which was draining my time to improve and manage GhostBSD. In 2022, I dropped OpenRC for FreeBSD's simpler but reliable RC system. It meant less to juggle and more focus forward. By 2023, maintaining OS ports grew too taxing. In 2024, I decided to shift building our OS packages from FreeBSD-maintained PkgBase within SRC, streamlining the maintenance load and refocusing on user experience.

In January this year, I realized that building GhostBSD from STABLE was too much for our small team. After discussing it with the other contributors, I decided to switch back to FreeBSD RELEASE. Yes, we are losing early driver access, but we save time debugging STABLE changes, giving us the stability to build on. I am not saying that STABLE is always broken, but sometimes, some changes create problems.

Over the years, I've tried to manage GhostBSD to the best of my knowledge through critical choices that added difficulties but also brought gains for what was missing in GhostBSD. PkgBase and OS ports gave OS updates from a graphical user interface, but OpenRC piled on work. However, it also meant adding more to maintain than the project could handle. All the latest changes mark the return to GhostBSD's roots and a renewal of focusing on usability instead of over-complicating the maintenance of GhostBSD. It was a hard lesson to keep it simple.

I feel we're in a good place to focus on improving our tools.

Where GhostBSD Stands Presently

As I write this, we've just released GhostBSD 25.01-R14.2p1. It marks a shift from FreeBSD STABLE to FreeBSD RELEASE, using 14.2-RELEASE-p1 for more excellent stability. The new versioning of GhostBSD breaks down as follows: 25 for 2025, 01 for the GhostBSD patch, R for RELEASE, 14.2 for the FreeBSD version, and p1 for the FreeBSD patch. It aims to clarify releases for users. No more guessing what's what. With all the recent changes, I feel we're in a good place to focus on improving our tools.

Some of those tools are:

- **NetworkMgr:** A GUI software for Ethernet and Wi-Fi, modeled after Linux's Network-Manager. Simple clicks over CLI chaos.
- **Update Station:** A GUI software for software and OS upgrades that creates a Boot Environment backup before upgrading. Safety first!
- **Software Station:** A GUI software to install software that leverages pkg to install software. The point, click, done.
- **ghostbsd-build:** Used to build GhostBSD, including Joe Maloney's ZFS reroof hack for a read-write ZFS pool live session in RAM. Blazing fast for demos and installations.
- **Backup Station:** Added in September 2022 by Mike Jurbala, a GUI software that uses pybectl, an in-house Python module for interfacing with bectl, to manage Boot Environments. System snapshots made easy.

- **GBI and pc-sysinstall:** GhostBSD's GUI installer recently dropped UFS in the UI to leverage ZFS's strengths. ZFS's power shines over older ways.

What's Next for GhostBSD Future

For 2025, I plan to document some SOPs to make contributing to GhostBSD easier for everyone, hoping that new contributors won't need too much mentorship. I'll deploy a faster build server at my house for quicker package builds and as I write this, I am waiting for the PDU to arrive. I also want to finish an OEM-friendly installer to widen our reach, rework Update Station to install updates at boot, and improve NetworkMgr's integration with devd for stability. If possible, add support to create a home directory dataset with the option to encrypt with GBI. FreeBSD's upcoming AC and AX Wi-Fi support in 2025 or 2026 promises better connectivity speed, and I'm excited about that. Our laptops will love it.

I can't speak for other contributors, but we have a long list of tasks on GitHub. We do have a roadmap that can be found under the Development tab on GhostBSD.org. Check it out if you're curious.

In the long term, we are waiting for more donations to come in so we can buy an ARM(Ampere) server to start building GhostBSD arm64. In the meantime, GhostBSD still aims to be a fully GUI-driven OS that leverages ZFS, which is perfect for non-technical users who can benefit from what FreeBSD offers. We have conversations on creating components for a desktop to slowly replace MATE that align better with FreeBSD/GhostBSD, like a file manager that leverages ZFS. However, it's still just discussions. Nothing solid for the moment.

Conclusion

GhostBSD has been a journey with a chain of choices, from a 2009 live CD to what it has become today. It was started by a non-technical user dream and evolved into a community project. Each step, like custom packages, TrueOS, ZFS reroor live session, and PkgBase tackled a challenge. The past taught me resilience, the present offers stability, and the future invites you to help shape it. If you are interested in getting involved, please visit GhostBSD.org. You'll find a spot.

ERIC TURGEON is the founder and leader of GhostBSD. He's also a FreeBSD ports committer, focusing on maintaining MATE ports and the NetworkMgr port. Based in Canada, Eric has been passionate about BSD since the late 2000s. He balances his time between GhostBSD, FreeBSD contributions, work, and personal life. His drive comes from a desire to make BSD accessible, and he welcomes anyone to join the GhostBSD community.

.....

GhostBSD has been a journey with a chain of choices, from a 2009 live CD to what it has become today.