

# *The FreeBSD Appliance*

*Leveraging FreeBSD and Strategic Scripting to Deliver Storage  
and Virtualization Services*

**Michael Dexter**

**editor@callfortesting.org**

**BSDCan 2023**

# Please Read The Paper!

- It's Short and Sweet
- Highlights each of these building blocks
- What I wish I was handed 20 years ago

*My 2007 – 2023 paper trail leads to this point*

# The Best Use of 45 Minutes?

I am just one person.

The most valuable thing I can do:

Inspire You

# Some High-Level Themes

The use of OpenZFS is non-negotiable

Own the Stack

Spot the Patterns

Do not wait for the Fundamentals to Change

# “The FreeBSD Appliance”: FreeBSD

- Feature Rich
- Permissively-Licensed
- Unix Operating System

# FreeBSD: Feature Rich

- Excellent Networking – The Stack and packet filters
- Excellent File Systems – UFS, OpenZFS, GEOM Suite
- Excellent Observability – DTrace, DDB, libxo
- 20+ Years of Containers – Jail
- Two Hypervisors – bhyve and Xen
- CTRL-T!

# FreeBSD: Permissively-Licensed

- Two Clause BSD: Give credit where credit is due
- No obligation to give back but you're an idiot not to
- Ideal for appliances, but the CDDL components are file-level copyleft

*Please "steal" from a BSD before you  
implement from scratch*

# FreeBSD: Unix Operating System

- NOT to be confused with UNIX®
- But really, really similar and overlapping
- Standards-based/influenced: POSIX, SUS, RFC's, NIH

The thing on your phone, your home and enterprise appliances, hiding in macOS, hiding in Windows WSL, BSD, GNU/Linux, AIX, HP-UX, and on and on...



# “The FreeBSD Appliance”: **Appliance**

- Single Purpose
- Has a Ecosystem
- Has an Exit Strategy

# Appliance: Single Purpose

- As opposed to General Purpose computing
- FreeBSD is uniquely capable here
- Ultimately not capable of *not* doing its job

# Appliance: Has an Ecosystem

- External: Organic
  - Users!
  - User Support Community
    - Blogs
    - Forums
    - Messaging/Social Media Platforms
  - Contribution Vector
    - Right to Fork
    - No Hardware Lock-in

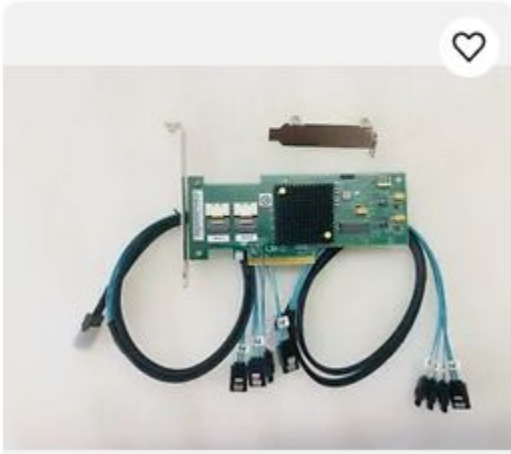
# Appliance: Has an Ecosystem

## Ecosystem Metrics of Success

873 results for **freenas**

[Save this search](#)

Shipping to: **K1N6N5** [v](#)



LSI 6Gbps SAS HBA 9200-8I IT Mode P20 ZFS FreeNAS unRAID +  
2\*SFF-8087 SATA US

Brand New

**\$39.99**

or Best Offer

+\$10.99 shipping  
from United States

**160 sold**

**Extra 5% off with coupon**

# Appliance: Has an Ecosystem

- External: Commercial
  - Business Users
  - Professional Support Community
  - Partners
  - Resellers/Integrators
  - Second-hand resellers - eBay!
  - Second-hand recertifiers

Looking at you Open Networking 😐

# Appliance: Has an Ecosystem

- Internal: Commercial
  - Marketing
  - Sales
  - Sales Engineering
  - Finance
  - Support
  - Warranties
  - Professional Services

Appliance: Has an Ecosystem

A Cathedral *and* a Bazaar

# Appliance: An Exit Strategy

- Freedom to Fork, Open Source (verb), Liberate!
  - What WE want
- Corporate Acquisition
  - What Shareholders want

Whose favorite appliance has been  
acquired and shelved?



# Implicit Buyer's Guide, Taxonomy, & Guide

Remember...

**Spot the Patterns**

**Do not wait for the Fundamentals to Change**

That said... Choose your own adventure!

# Choose Your Own Appliance Adventure!

## Choose an Exit Strategy

- Fork, Open Source (v.), Liberate!
- Acquisition! Buy that Tesla!

# Choose Your Own Appliance Adventure!

## Choose a Hardware Target

- Proprietary – A Black Box
- Supported – Exact Hardware
- Compatible – Most Hardware
- Virtual – Pretend Hardware

# Choose Your Own Appliance Adventure!

## Choose a License

- Proprietary
- Viral
- File-level Viral
- Permissive

# Choose Your Own Appliance Adventure!

## Choose a Kernel/OS

- Home Grown
- Windows
- GNU/Linux
- illumos
- BSD (Net, Free, Open, DragonFly...)

# Choose Your Own Appliance Adventure!

## Choose a File System

- OpenZFS, illumos ZFS
- Maybe something distributed on top

# Choose Your Own Appliance Adventure!

Choose your Secret Stuff

- Proprietary Stuff
- Hardware Drivers/Interfaces

# Choose Your Own Appliance Adventure!

## Choose Management Tools

- Console User Interface
- Command Line Interface
- Web User Interface
- Application Programming Interface

Human Interfaces vs. Machine Interfaces



# Congratulations!

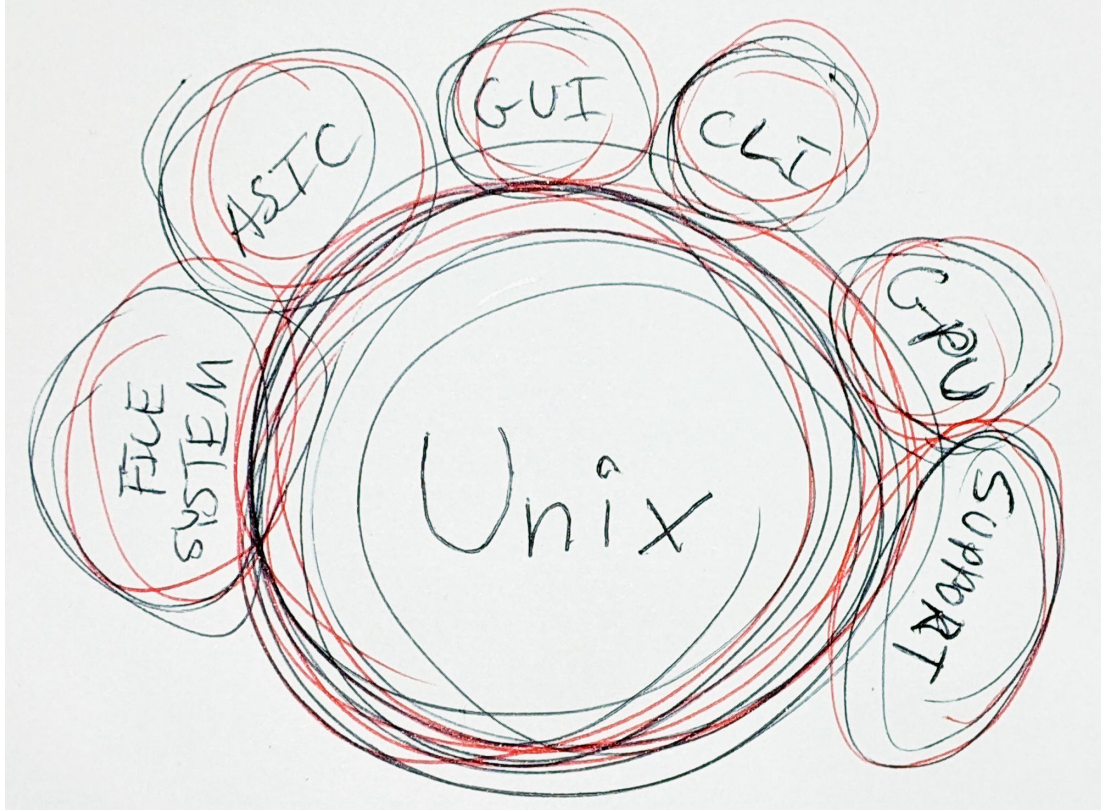
The majority of appliances  
follow this model and

**Unix Won!**



# WAIT! TIME OUT!

The  
Venn  
Diagram  
From  
Hell



Proprietary  
GPL Violating  
Abandoned  
Unpatched  
Black Boxes  
and E-Waste

# We Won! Yet Lost.

That's not what we asked for.

That's not where we want to be.

"I stopped playing the game." – James Holden

# Choosing Different Fundamentals

In the long term, maybe vendors aren't the answer

Public-benefit foundations are rarely perfect, but  
they don't get VC-funded and acquired

# Getting to where we want to be

Make it Easier

*That's* easier when you own the stack

# Make it Easier

Start here: Human and Machine Read and Writability

# Make it Easier: Formatted Output

```
smartctl -a /dev/...
```

Epiphany: NEITHER HUMAN NOR MACHINE READABLE

Added JSON... Changed the schema...

BUT NOT THE SCHEMA VERSION

# Make it Easier: Formatted Output

Bringing the Unix  
Philosophy to the 21st  
Century

`txt | jc | {}`

[blog.kellybrazil.com/2019/11/26/bringing-the-unix-philosophy-to-the-21st-century/](https://blog.kellybrazil.com/2019/11/26/bringing-the-unix-philosophy-to-the-21st-century/)



# Make it Easier: Formatted Output

`github.com/kellyjonbrazil/jc`

## About

CLI tool and python library that converts the output of popular command-line tools, file-types, and common strings to JSON, YAML, or Dictionaries. This allows piping of output to tools like jq and simplifying automation scripts.

~~Parses~~ Chases 100 Utilities!

# Machine Readability: Own the Stack

`wiki.freebsd.org/LibXo`

*Thank You Juniper!*

## In Base

```
arp df efitable iscsictl jls last lastlogin ls mount ndp  
netstat nfsstat procstat ps savecore sesutil vmstat w wc
```

## In Ports

```
sysutils/nsysctl sysutils/checkrestart sysutils/smart
```

# Machine Readability: Own the Stack

*Thank You Klara!*

Coming Soon

## OpenZFS JSON Output

# Machine Readability: Own the Stack

`wiki.freebsd.org/UniversalConfigurationLanguage`

In Base: `ctld`, `iovctl`

In Ports: `devel/uclcmd`

`bhyve_config(5)`: The plumbing for your format of choice!

# Make it Easier: UCL + flua + Jail = Jailua!

`gist.github.com/antranigv`

[antranigv](#) / [jailua.lua](#)

Created 2 months ago

FreeBSD jailua for Jail management

 2 files  0 forks  0 comments  0 stars

```
1  #!/usr/libexec/flua
2  local ucl = require('ucl')
3  local jail = require('jail')
4
5  function usage()
6      print(arg[0] .. " [ list | create jail.ucl | remove jail.ucl [jname] ]")
7  end
8
9  if #arg < 1 then
10     usage()
```

# Make it Easier: Open the Door!

May 1000 Flowers Bloom!

What if, like clockwork every September, CS 301 students built user and machine user interfaces in their languages of choice?

# Make it Easier: Understanding Before Solving

Meet weekly

Document the Problem

Understand the Problem

Help Each Other

Hack

Weekly Jail Call: `jail.freebsd.am`

# Make it Easier: “Embedded” FreeBSD

My journey started here

RPM Hell guided me to FreeBSD Jail in 2003

My first question: How do I build a minimum system?

Answer: Fix the FreeBSD Build Options



# Single-Purpose Hosts, Containers and VMs

*The bare-minimum binaries and dependencies to perform one or more related functions*

less (1) is more (1) (more or less)

# Single-Purpose Hosts, Containers and VMs

- Meaningful containment is assumed
- Attack surface is significantly reduced
- Deterministic inventory = better understanding
- Potentially higher performance
- Small size, potentially all nullfs-mounts
- Selectively read-only thanks to ZFS

# Single-Purpose Hosts, Containers and VMs

- FreeBSD build options configure the userland
- The kernel has modules, options, and devices
- Surprising coverage, with over 200 build options
- *Stunningly* broken for years
- All working come FreeBSD 13.0R, after MUCH work
- What do they look like?

# Anatomy of a Build Option

```
man src.conf ...
```

```
WITHOUT_BHYVE
```

```
Do not build or install bhyve(8),  
associated utilities, and examples.
```

```
This option only affects amd64/amd64.
```

```
Add WITHOUT_BHYVE=YES to /etc/src.conf
```

# Anatomy of a Build Option

```
/usr/src/usr.sbin/Makefile.amd64
```

```
.if ${MK BHYVE} != "no"
```

```
SUBDIR+=      bhyve
```

```
SUBDIR+=      bhyvectl
```

```
SUBDIR+=      bhyveload
```

```
.endif
```

# Structured & Automated with OccamBSD

- *“An application of Occam's razor to FreeBSD”*
- An OS reduced to its minimum components
  - Minimum components to build
  - Minimum components to boot on a VM
  - Minimum components to boot on hardware
- Add networking and other features as needed

# OccamBSD

- World and kernel build times in minutes
- Working OS in under 150 Megabytes
- Boot times in seconds
- Expected to be unrecognizable
- A flashback to 4.3BSD!
- *Very* educational...

# OccamBSD: Immediate Benefits

- Reveal abandoned components
- Reveal undocumented components
- Reveal cross-building issues
- Produce a “Rescue” ISO
- Perfect classroom OS (Confirmed by Antranig V)
- Instantly incorporated into LureOS by Antranig V



# OccamBSD: START HERE

- No, Seriously, Literally...
- The core OS used by all users at all times
- Where to begin documenting
- Where to begin auditing
- Where to begin fuzz testing
- Where to begin LEARNING

# OccamBSD: 20 Years in the Making

`github.com/michaeldexter/occambsd`

```
sh occambsd.sh -v -z -p profile-amd64-zfs.txt
```

-v VM image

-z OpenZFS thanks to `mkimage -t zfs!`

-p Profile

# OccamBSD: 20 Years in the Making

`github.com/michaeldexter/occambsd`

Note the prior art in the README.md

NanoBSD, picobsd, TinyBSD, Crochet,  
Poudriere image.sh, mkjail

# OccamBSD: 20 Years in the Making

`github.com/michaeldexter/occambsd`

Note `imagine.sh` in the same repo

Release Engineering “VM-IMAGES”  
are awesome

# OccamBSD: 20 Years in the Making

`github.com/michaeldexter/occambsd`

A great way fall in love in love  
with FreeBSD again

# 30 Years of Innovation, Big and Small

- Obviously Clang/LLVM, OpenZFS, DTrace, and bhyve
- Less obviously reproducible builds when reducing
- Less obviously `makefs(8) -t zfs`
- Less obviously `mount nullfs -f`
- Less obviously `sysrc(8)`

# 30 Years of Innovation, Big and Small

- Less obviously reproducible builds
  - “A build is **reproducible** if given the **same source code, build environment, and build instructions, any party** can recreate **bit-by-bit identical** copies of all specified artifacts.”
- `reproducible-builds.org`
- `wiki.freebsd.org/ReproducibleBuilds`
- Helpful when paring down the operating system

# 30 Years of Innovation, Big and Small

- Less obviously `makefs(8) -t zfs`
- `makefs -- create a file system image from a directory tree or amtree manifest`
- Thank you NetBSD!
- **NOW CREATES BOOTABLE ZFS-BASED “VM-IMAGES”**
- Thank you `markj@ ...`



# 30 Years of Innovation, Big and Small

- Less obviously `mount nullfs -f`
- The `mount_nullfs` utility supports mounting both *directories* and *single files*
- Traditionally only for directories but now supports files!
- A Jail can be created mostly, if not entirely out of components from the host, often read-only

# 30 Years of Innovation, Big and Small

- Less obviously `sysrc(8)`
- `sysrc -- safely edit system rc files`  
`sysrc hostname=current`  
`sysrc -c hostname=current ; echo $?`  
`0`
- However, it's not idempotent

# Make it Easier: Idempotence

Define a desired state, work to get there

This often involves doing nothing

`sysrc(8)` needs to learn to do nothing  
if nothing needs to be done

# Make it Easier: Idempotence

Define a desired state, work to get there

This often involves doing nothing

`sysrc(8)` needs to learn to do nothing  
if nothing needs to be done

`fetch(1) -i` needs to remember its job

# Make it Easier: Idempotence

```
hostname="occambsd"
if [ "$( sysrc -c hostname=$hostname )" ] ; then
    echo "Hostname $hostname is correct"
    logger "Hostname $hostname is correct"
else
    echo ; echo "Setting hostname $hostname"
    logger "Setting hostname $hostname"
    sysrc hostname="$hostname"
    service hostname restart
fi
```

(See the sample rc.local in the OccamBSD repo)

# Make it Easier: Idempotence Easy Button

Sound Familiar?

Feeling Impatient?

Search "Ansible NAS"

Admittedly the shortest-path easy button. Bye!

# Medium-Term Easy Buttons

Webmin is still going strong, Perl based, and  
permissively licensed

It's ZFS managers are making progress

Remember choosing your own adventure!

# Medium-Term Easy Buttons

Proxmox/bhyve/Xen?

More Perl, ZFS-aware, gaining in popularity



# Buttons...

That's what the easier buttons look like

I'm sticking with the hard button

Thank you!

*Questions?*

Michael Dexter

editor@callfortesting.org | @dexter@bsd.network

BSDCan 2023