

FreeBSD Handbook

Abstract

Welcome to FreeBSD! This handbook covers the installation and day to day use of *FreeBSD 14.2-RELEASE* and *13.4-RELEASE*. This book is the result of ongoing work by many individuals. Some sections might be outdated. Those interested in helping to update and expand this document should send email to the [FreeBSD documentation project mailing list](#).

The latest version of this book is available from the [FreeBSD web site](#). Previous versions can be obtained from <https://docs.FreeBSD.org/doc/>. The book can be downloaded in a variety of formats and compression options from the [FreeBSD download server](#) or one of the numerous [mirror sites](#). Searches can be performed on the handbook and other documents on the [search page](#).

Table of Contents

Preface	10
Intended Audience	10
Fourth Edition	10
Third Edition	11
Second Edition (2004)	11
First Edition (2001)	12
Organization of This Book	13
Conventions used in this book	16
Acknowledgments	17
I: Getting Started	19
1. Introduction	20
1.1. Synopsis	20
1.2. Welcome to FreeBSD!	20
1.3. About the FreeBSD Project	22
2. Installing FreeBSD	27
2.1. Synopsis	27
2.2. Minimum Hardware Requirements	27
2.3. Pre-Installation Tasks	27
2.4. Starting the Installation	31
2.5. Using bsdininstall	35
2.6. Allocating Disk Space	40
2.7. Fetching Distribution Files	63
2.8. Network Interfaces, Accounts, Time Zone, Services and Hardening	66
2.9. Troubleshooting	95
2.10. Using the Live CD	95
3. FreeBSD Basics	96
3.1. Synopsis	96
3.2. Virtual Consoles and Terminals	96
3.3. Users and Basic Account Management	99
3.4. Permissions	108
3.5. Directory Structure	113
3.6. Disk Organization	115
3.7. Mounting and Unmounting File Systems	121
3.8. Processes and Daemons	123
3.9. Shells	126
3.10. Text Editors	129
3.11. Devices and Device Nodes	129
3.12. Manual Pages	129

4. Installing Applications: Packages and Ports	132
4.1. Synopsis	132
4.2. Overview of Software Installation	132
4.3. Finding Software	133
4.4. Using pkg for Binary Package Management	134
4.5. Using the Ports Collection	143
4.6. Building Packages with poudriere	151
4.7. Post-Installation Considerations	154
4.8. Dealing with Broken Ports	155
5. The X Window System	156
5.1. Synopsis	156
5.2. Installing Xorg	156
5.3. Graphic card drivers	157
5.4. Xorg Configuration	159
5.5. Using Fonts in Xorg	164
6. Wayland on FreeBSD	169
6.1. Synopsis	169
6.2. Wayland Overview	169
6.3. The Wayfire Compositor	171
6.4. The Hikari Compositor	173
6.5. The Sway Compositor	175
6.6. Using Xwayland	176
6.7. Remote Desktop Using VNC	179
6.8. Wayland Login Manager	179
6.9. Useful Utilities	180
7. Network	182
7.1. Synopsis	182
7.2. Setting up the Network	182
7.3. Wired Networks	183
7.4. Wireless Networks	191
7.5. Hostname	195
7.6. DNS	195
7.7. Troubleshooting	197
II: Common Tasks	199
8. Desktop Environments	200
8.1. Synopsis	200
8.2. Desktop Environments	200
8.3. Browsers	207
8.4. Development tools	210
8.5. Desktop office productivity	212
8.6. Document Viewers	214

8.7. Finance	215
9. Multimedia	217
9.1. Synopsis	217
9.2. Setting Up the Sound Card	217
9.3. Audio players	220
9.4. Video players	221
9.5. Conferencing and Meetings	223
9.6. Image Scanners	225
10. Configuring the FreeBSD Kernel	229
10.1. Synopsis	229
10.2. Why Build a Custom Kernel?	229
10.3. Finding the System Hardware	230
10.4. The Configuration File	231
10.5. Building and Installing a Custom Kernel	233
10.6. If Something Goes Wrong	234
11. Printing	236
11.1. Quick Start	236
11.2. Printer Connections	237
11.3. Common Page Description Languages	238
11.4. Direct Printing	240
11.5. LPD (Line Printer Daemon)	240
11.6. Other Printing Systems	249
12. Linux Binary Compatibility	250
12.1. Synopsis	250
12.2. Configuring Linux Binary Compatibility	250
12.3. Linux userlands	251
12.4. Advanced Topics	255
13. WINE	259
13.1. Synopsis	259
13.2. WINE Overview & Concepts	260
13.3. Installing WINE on FreeBSD	262
13.4. Running a First WINE Program on FreeBSD	264
13.5. Configuring WINE Installation	265
13.6. WINE Management GUIs	273
13.7. WINE in Multi-User FreeBSD Installations	287
13.8. WINE on FreeBSD FAQ	289
III: System Administration	293
14. Configuration, Services, Logging and Power Management	294
14.1. Synopsis	294
14.2. Configuration Files	294
14.3. Managing Services in FreeBSD	299

14.4. Cron and Periodic	301
14.5. Configuring System Logging	305
14.6. Power and Resource Management	314
14.7. Adding Swap Space	320
15. The FreeBSD Booting Process	322
15.1. Synopsis	322
15.2. FreeBSD Boot Process	322
15.3. Device Hints	328
15.4. Shutdown Sequence	329
16. Security	330
16.1. Synopsis	330
16.2. Introduction	330
16.3. Securing Accounts	331
16.4. Intrusion Detection System (IDS)	337
16.5. Secure levels	339
16.6. File flags	341
16.7. OpenSSH	342
16.8. OpenSSL	347
16.9. Kerberos	351
16.10. TCP Wrappers	359
16.11. Access Control Lists	360
16.12. Capsicum	362
16.13. Process Accounting	362
16.14. Resource Limits	363
16.15. Monitoring Third Party Security Issues	366
16.16. FreeBSD Security Advisories	367
17. Jails and Containers	373
17.1. Synopsis	373
17.2. Jail Types	373
17.3. Host Configuration	376
17.4. Classic Jail (Thick Jail)	379
17.5. Thin Jails	381
17.6. Jail Management	390
17.7. Jail Upgrading	392
17.8. Jail Resource Limits	394
17.9. Jail Managers and Containers	394
18. Mandatory Access Control	396
18.1. Synopsis	396
18.2. Key Terms	397
18.3. Understanding MAC Labels	398
18.4. Planning the Security Configuration	402

18.5. Available MAC Policies	403
18.6. User Lock Down	410
18.7. Nagios in a MAC Jail	411
18.8. Troubleshooting the MAC Framework	415
19. Security Event Auditing	417
19.1. Synopsis	417
19.2. Key Terms	417
19.3. Audit Configuration	418
19.4. Working with Audit Trails	422
20. Storage	426
20.1. Synopsis	426
20.2. Adding Disks	426
20.3. Resizing and Growing Disks	427
20.4. USB Storage Devices	430
20.5. Creating and Using CD Media	433
20.6. Creating and Using DVD Media	439
20.7. Creating and Using Floppy Disks	444
20.8. Backup Basics	445
20.9. Memory Disks	449
20.10. File System Snapshots	451
20.11. Disk Quotas	453
20.12. Encrypting Disk Partitions	456
20.13. Encrypting Swap	462
20.14. Highly Available Storage (HAST)	463
21. GEOM: Modular Disk Transformation Framework	472
21.1. Synopsis	472
21.2. RAID0 - Striping	472
21.3. RAID1 - Mirroring	474
21.4. RAID3 - Byte-level Striping with Dedicated Parity	484
21.5. Software RAID Devices	485
21.6. GEOM Gate Network	490
21.7. Labeling Disk Devices	490
21.8. UFS Journaling Through GEOM	493
22. The Z File System (ZFS)	495
22.1. What Makes ZFS Different	495
22.2. Quick Start Guide	495
22.3. zpool Administration	502
22.4. zfs Administration	520
22.5. Delegated Administration	541
22.6. Advanced Topics	542
22.7. Further Resources	545

22.8. ZFS Features and Terminology	545
23. Other File Systems	553
23.1. Synopsis	553
23.2. Linux® File Systems	553
23.3. Windows® File Systems	554
23.4. MacOS® File Systems	555
24. Virtualization	557
24.1. Synopsis	557
24.2. FreeBSD as a Guest on Parallels Desktop for macOS®	557
24.3. FreeBSD as a Guest on VMware Fusion for macOS®	564
24.4. FreeBSD as a Guest on VirtualBox™	576
24.5. FreeBSD as a Host with VirtualBox™	577
24.6. Virtualization with QEMU on FreeBSD	580
24.7. FreeBSD as a Host with bhyve	605
24.8. FreeBSD as a Xen™-Host	621
25. Localization - i18n/L10n Usage and Setup	628
25.1. Synopsis	628
25.2. Using Localization	628
25.3. Finding i18n Applications	635
25.4. Locale Configuration for Specific Languages	635
26. Updating and Upgrading FreeBSD	638
26.1. Synopsis	638
26.2. FreeBSD Update	638
26.3. Updating Bootcode	646
26.4. Updating the Documentation Set	646
26.5. Tracking a Development Branch	647
26.6. Updating FreeBSD from Source	650
26.7. Tracking for Multiple Machines	657
26.8. Building on non-FreeBSD Hosts	658
27. DTrace	659
27.1. Synopsis	659
27.2. Implementation Differences	659
27.3. Enabling DTrace Support	660
27.4. Enabling DTrace in Out-of-Kernel Modules	661
27.5. Using DTrace	661
28. USB Device Mode / USB OTG	664
28.1. Synopsis	664
28.2. USB Virtual Serial Ports	664
28.3. USB Device Mode Network Interfaces	666
28.4. USB Virtual Storage Device	667
IV: Network Communication	669

29. Serial Communications	670
29.1. Synopsis	670
29.2. Serial Terminology and Hardware	670
29.3. Terminals	674
29.4. Dial-in Service	678
29.5. Dial-out Service	681
29.6. Setting Up the Serial Console	685
30. PPP	690
30.1. Synopsis	690
30.2. Configuring PPP	690
30.3. Troubleshooting PPP Connections	698
30.4. Using PPP over Ethernet (PPPoE)	701
30.5. Using PPP over ATM (PPPoA)	703
31. Electronic Mail	706
31.1. Synopsis	706
31.2. Mail Components	706
31.3. DragonFly Mail Agent (DMA)	707
31.4. Sendmail	709
31.5. Changing the Mail Transfer Agent	711
31.6. Mail User Agents	714
31.7. Advanced Topics	721
32. Network Servers	727
32.1. Synopsis	727
32.2. The inetd Super-Server	727
32.3. Network File System (NFS)	731
32.4. Network Information System (NIS)	735
32.5. Lightweight Directory Access Protocol (LDAP)	749
32.6. Dynamic Host Configuration Protocol (DHCP)	757
32.7. Domain Name System (DNS)	761
32.8. Zero-configuration networking (mDNS/DNS-SD)	764
32.9. Apache HTTP Server	764
32.10. File Transfer Protocol (FTP)	771
32.11. File and Print Services for Microsoft® Windows® Clients (Samba)	772
32.12. Clock Synchronization with NTP	775
32.13. iSCSI Initiator and Target Configuration	778
33. Firewalls	784
33.1. Synopsis	784
33.2. Firewall Concepts	785
33.3. PF	786
33.4. IPFW	802
33.5. IPFILTER (IPF)	818

33.6. Blacklistd	830
34. Advanced Networking	835
34.1. Synopsis	835
34.2. Gateways and Routes	835
34.3. Virtual Hosts	841
34.4. Wireless Advanced Authentication	842
34.5. Wireless Ad-hoc Mode	846
34.6. USB Tethering	850
34.7. Bluetooth	851
34.8. Bridging	860
34.9. Link Aggregation and Failover	866
34.10. Diskless Operation with PXE	872
34.11. Common Address Redundancy Protocol (CARP)	877
34.12. VLANs	879
V: Appendices	881
Appendix A: Obtaining FreeBSD	882
A.1. Mirrors	882
A.2. Using Git	885
A.3. Disc Copies	887
Appendix B: Bibliography	889
B.1. FreeBSD Bibliography	889
B.2. Security Reference	889
B.3. UNIX® History	889
B.4. Periodicals, Journals, and Magazines	890
Appendix C: Resources on the Internet	891
C.1. Websites	891
C.2. Mailing Lists	891
C.3. Usenet Newsgroups	894
Appendix D: OpenPGP Keys	895
D.1. Officers	895
FreeBSD Glossary	902
Colophon	922

Preface

Intended Audience

The FreeBSD newcomer will find that the first section of this book guides the user through the FreeBSD installation process and gently introduces the concepts and conventions that underpin UNIX®. Working through this section requires little more than the desire to explore, and the ability to take on board new concepts as they are introduced.

Once you have traveled this far, the second, far larger, section of the Handbook is a comprehensive reference to all manner of topics of interest to FreeBSD system administrators. Some of these chapters may recommend that you do some prior reading, and this is noted in the synopsis at the beginning of each chapter.

For a list of additional sources of information, please see [Bibliography](#).

Fourth Edition

The current version of the Handbook represents the cumulative effort of a working group that has been reviewing and updating all Handbook content. These are the major updates since the fourth edition of the Handbook.

- The Handbook has been converted from [Docbook](#) to [Hugo](#) and [AsciiDoctor](#)
- The [FreeBSD Documentation Portal](#) has been created.
- The [Introduction](#) chapter has been updated to improve the FreeBSD history and fix minor typos.
- The [Install](#) chapter has been updated with an upgraded synopsis, the latest changes to the installer, refreshed images, added alt text to images, and the removal of references to specific versions.
- The [Basics](#) chapter has updated tables, command outputs, and directory structure per man:hier.
- The [Ports](#) chapter has been updated and now simplifies package searches, updates software examples (Nginx replaces Apache), improves the [pkg\(8\)](#) bootstrap process, and adds new instructions for configuring and managing packages, including blocking and unblocking.
- The [X11](#) chapter has been updated to reflect the current state of graphics in FreeBSD, remove outdated references to old Intel drivers, configurations, and compiz, and relocate desktop environment instructions (such as KDE Plasma and GNOME) to the Desktop Environments chapter, as these environments now also support Wayland in addition to X11.
- The [Wayland](#) chapter has been added with information about installing and configuring Wayland under FreeBSD.
- The [Network](#) chapter has been created to cover basic wired and wireless network configuration, including hostname, DNS, and troubleshooting. Sections on wired networks, wireless, and IPv6 have been moved and updated with improved command outputs, use of sysrc, and better AsciiDoc syntax.
- The [Desktop](#) chapter has been updated with upgraded installation instructions for KDE Plasma,

GNOME, XFCE, MATE, Cinnamon, and LXQT, expanded browser options, a new development tools section, and updates to office productivity, document viewers, and finance sections.

- The [Multimedia](#) chapter has been reworked with updates to the sound section, new tables for sound mixers, audio players, and video players, guidance on automatic headphone switching, a new conferencing and meetings section, and a revised image scanners section.
- The [Linuxemu](#) chapter has been improved with updated instructions for setting up a Debian/Ubuntu base system using debootstrap.
- The [Config](#) chapter has been renamed for accuracy, with updates to service management, cron and periodic, syslog, power management, and swap sections. A new entry on config files was added, and the outdated tuning section was removed.
- The [Security](#) chapter has been updated with enhancements to VPN over IPSec, securing accounts, password hashes, sudo/doas, and OpenSSH/OpenSSL. New sections have been added covering IDS, secure levels, file flags, Capsicum, NFSv4 ACLs, and resource limits.
- The [Jails](#) chapter has been updated to include details on jail types (Thick, Thin, VNET, and Linux Jails), host system configuration, networking options, the jail configuration file, setup procedures, upgrade methods, resource limits, and different jail managers and container solutions.
- The [Mail](#) chapter has been updated to include information on DMA, upgrades to Sendmail, instructions for changing DMA and Sendmail to use different MTAs, and the removal of Dialup and Fetchmail sections, along with a reorganization of the chapter.
- The [Bibliography](#) has been extensively updated.

Third Edition

The current online version of the Handbook represents the cumulative effort of many hundreds of contributors over the past 10 years. The following are some of the significant changes since the two volume third edition was published in 2004:

- [WINE](#) has been added with information about how to run Windows® applications on FreeBSD.
- [DTrace](#) has been added with information about the powerful DTrace performance analysis tool.
- [Other File Systems](#) have been added with information about non-native file systems in FreeBSD, such as ZFS from Sun™.
- [Security Event Auditing](#) has been added to cover the new auditing capabilities in FreeBSD and explain its use.
- [Virtualization](#) has been added with information about installing FreeBSD on virtualization software.
- [Installing FreeBSD](#) has been added to cover installation of FreeBSD using the new installation utility, bsdinstall.

Second Edition (2004)

The third edition was the culmination of over two years of work by the dedicated members of the FreeBSD Documentation Project. The printed edition grew to such a size that it was necessary to

publish as two separate volumes. The following are the major changes in this new edition:

- [Configuration and Tuning](#) has been expanded with new information about the ACPI power and resource management, the `cron` system utility, and more kernel tuning options.
- [Security](#) has been expanded with new information about virtual private networks (VPNs), file system access control lists (ACLs), and security advisories.
- [Mandatory Access Control](#) is a new chapter with this edition. It explains what MAC is and how this mechanism can be used to secure a FreeBSD system.
- [Storage](#) has been expanded with new information about USB storage devices, file system snapshots, file system quotas, file and network backed filesystems, and encrypted disk partitions.
- A troubleshooting section has been added to [PPP](#).
- [Electronic Mail](#) has been expanded with new information about using alternative transport agents, SMTP authentication, UUCP, fetchmail, procmail, and other advanced topics.
- [Network Servers](#) is all new with this edition. This chapter includes information about setting up the Apache HTTP Server, ftpd, and setting up a server for Microsoft® Windows® clients with Samba. Some sections from [Advanced Networking](#) were moved here to improve the presentation.
- [Advanced Networking](#) has been expanded with new information about using Bluetooth® devices with FreeBSD, setting up wireless networks, and Asynchronous Transfer Mode (ATM) networking.
- A glossary has been added to provide a central location for the definitions of technical terms used throughout the book.
- A number of aesthetic improvements have been made to the tables and figures throughout the book.

First Edition (2001)

The second edition was the culmination of over two years of work by the dedicated members of the FreeBSD Documentation Project. The following were the major changes in this edition:

- A complete Index has been added.
- All ASCII figures have been replaced by graphical diagrams.
- A standard synopsis has been added to each chapter to give a quick summary of what information the chapter contains, and what the reader is expected to know.
- The content has been logically reorganized into three parts: "Getting Started", "System Administration", and "Appendices".
- [FreeBSD Basics](#) has been expanded to contain additional information about processes, daemons, and signals.
- [Installing Applications: Packages and Ports](#) has been expanded to contain additional information about binary package management.
- [The X Window System](#) has been completely rewritten with an emphasis on using modern

desktop technologies such as KDE and GNOME on XFree86™ 4.X.

- [The FreeBSD Booting Process](#) has been expanded.
- [Storage](#) has been written from what used to be two separate chapters on "Disks" and "Backups". We feel that the topics are easier to comprehend when presented as a single chapter. A section on RAID (both hardware and software) has also been added.
- [Serial Communications](#) has been completely reorganized and updated for FreeBSD 4.X/5.X.
- [PPP](#) has been substantially updated.
- Many new sections have been added to [Advanced Networking](#).
- [Electronic Mail](#) has been expanded to include more information about configuring sendmail.
- [Linux® Binary Compatibility](#) has been expanded to include information about installing Oracle® and SAP® R/3®.
- The following new topics are covered in this second edition:
 - [Configuration and Tuning](#).
 - [Multimedia](#).

Organization of This Book

This book is split into five logically distinct sections. The first section, *Getting Started*, covers the installation and basic usage of FreeBSD. It is expected that the reader will follow these chapters in sequence, possibly skipping chapters covering familiar topics. The second section, *Common Tasks*, covers some frequently used features of FreeBSD. This section, and all subsequent sections, can be read out of order. Each chapter begins with a succinct synopsis that describes what the chapter covers and what the reader is expected to already know. This is meant to allow the casual reader to skip around to find chapters of interest. The third section, *System Administration*, covers administration topics. The fourth section, *Network Communication*, covers networking and server topics. The fifth section contains appendices of reference information.

[Introduction](#)

Introduces FreeBSD to a new user. It describes the history of the FreeBSD Project, its goals and development model.

[Installing FreeBSD](#)

Walks a user through the entire installation process of FreeBSD 9.x and later using `bsdinstall`.

[FreeBSD Basics](#)

Covers the basic commands and functionality of the FreeBSD operating system. If you are familiar with Linux® or another flavor of UNIX® then you can probably skip this chapter.

[Installing Applications: Packages and Ports](#)

Covers the installation of third-party software with both FreeBSD's innovative "Ports Collection" and standard binary packages.

[The X Window System](#)

Describes the X Window System in general and using X11 on FreeBSD in particular. Also

describes common desktop environments such as KDE and GNOME.

Wayland

Describes the Wayland display server in general and using Wayland on FreeBSD in particular. Also describes common compositors such as Wayfire, Hikari and Sway.

Desktop Applications

Lists some common desktop applications, such as web browsers and productivity suites, and describes how to install them on FreeBSD.

Multimedia

Shows how to set up sound and video playback support for your system. Also describes some sample audio and video applications.

Configuring the FreeBSD Kernel

Explains why you might need to configure a new kernel and provides detailed instructions for configuring, building, and installing a custom kernel.

Printing

Describes managing printers on FreeBSD, including information about banner pages, printer accounting, and initial setup.

Linux® Binary Compatibility

Describes the Linux® compatibility features of FreeBSD. Also provides detailed installation instructions for many popular Linux® applications such as Oracle® and Mathematica®.

WINE

Describes WINE and provides detailed installation instructions. Also describes how WINE operates, how to install a GUI helper, how to run Windows® applications on FreeBSD, and offers other tips and solutions.

Configuration and Tuning

Describes the parameters available for system administrators to tune a FreeBSD system for optimum performance. Also describes the various configuration files used in FreeBSD and where to find them.

The FreeBSD Booting Process

Describes the FreeBSD boot process and explains how to control this process with configuration options.

Security

Describes many different tools available to help keep your FreeBSD system secure, including Kerberos, IPsec and OpenSSH.

Jails

Describes the jails framework, and the improvements of jails over the traditional chroot support of FreeBSD.

Mandatory Access Control

Explains what Mandatory Access Control (MAC) is and how this mechanism can be used to secure a FreeBSD system.

Security Event Auditing

Describes what FreeBSD Event Auditing is, how it can be installed, configured, and how audit trails can be inspected or monitored.

Storage

Describes how to manage storage media and filesystems with FreeBSD. This includes physical disks, RAID arrays, optical and tape media, memory-backed disks, and network filesystems.

GEOM: Modular Disk Transformation Framework

Describes what the GEOM framework in FreeBSD is and how to configure various supported RAID levels.

The OpenZFS storage platform

Describes the OpenZFS storage platform and provides a quick-start guide and information about advanced topics running OpenZFS under FreeBSD.

Other File Systems

Examines support for non-native file systems under FreeBSD like ext2, ext3 and ext4.

Virtualization

Describes what virtualization systems offer, and how they can be used with FreeBSD.

Localization - i18n/L10n Usage and Setup

Describes how to use FreeBSD in languages other than English. Covers both system and application level localization.

Updating and Upgrading FreeBSD

Explains the differences between FreeBSD-STABLE, FreeBSD-CURRENT, and FreeBSD releases. Describes which users would benefit from tracking a development system and outlines that process. Covers the methods users may take to update their system to the latest security release.

DTrace

Describes how to configure and use the DTrace tool from Sun™ on FreeBSD. Dynamic tracing can help locate performance issues, by performing real time system analysis.

USB Device Mode / USB OTG

Explains the use of USB Device Mode and USB On The Go (USB OTG) on FreeBSD.

PPP

Describes how to use PPP to connect to remote systems in FreeBSD.

Electronic Mail

Explains the different components of an email server and dives into simple configuration topics for the most popular mail server software: sendmail.

Network Servers

Provides detailed instructions and example configuration files to set up your FreeBSD machine as a network filesystem server, domain name server, network information system server, or time synchronization server.

Firewalls

Explains the philosophy behind software-based firewalls and provides detailed information about the configuration of the different firewalls available for FreeBSD.

Advanced Networking

Describes many networking topics, including sharing an Internet connection with other computers on your LAN, advanced routing topics, wireless networking, Bluetooth®, ATM, IPv6, and much more.

Obtaining FreeBSD

Lists different sources for obtaining FreeBSD media on CDROM or DVD as well as different sites on the Internet that allow you to download and install FreeBSD.

Bibliography

This book touches on many different subjects that may leave you hungry for a more detailed explanation. The bibliography lists many excellent books that are referenced in the text.

Resources on the Internet

Describes the many forums available for FreeBSD users to post questions and engage in technical conversations about FreeBSD.

OpenPGP Keys

Lists the PGP fingerprints of several FreeBSD Developers.

Conventions used in this book

To provide a consistent and easy to read text, several conventions are followed throughout the book.

Typographic Conventions

Italic

An *italic* font is used for filenames, URLs, emphasized text, and the first usage of technical terms.

Monospace

A **monospaced** font is used for error messages, commands, environment variables, names of ports, hostnames, user names, group names, device names, variables, and code fragments.

Bold

A **bold** font is used for applications, commands, and keys.

User Input

Keys are shown in **bold** to stand out from other text. Key combinations that are meant to be typed simultaneously are shown with + between the keys, such as:

Ctrl + **Alt** + **Del**

Meaning the user should type the **Ctrl**, **Alt**, and **Del** keys at the same time.

Keys that are meant to be typed in sequence will be separated with commas, for example:

Ctrl + **X**, **Ctrl** + **S**

Would mean that the user is expected to type the **Ctrl** and **X** keys simultaneously and then to type the **Ctrl** and **S** keys simultaneously.

Examples

Examples starting with C:\> indicate a MS-DOS® command. Unless otherwise noted, these commands may be executed from a "Command Prompt" window in a modern Microsoft® Windows® environment.

```
C:\> tools\fdimage floppies\kern.flp A:
```

Examples starting with # indicate a command that must be invoked as the superuser in FreeBSD. You can login as **root** to type the command, or login as your normal account and use **su(1)** to gain superuser privileges.

```
# dd if=kern.flp of=/dev/fd0
```

Examples starting with % indicate a command that should be invoked from a normal user account. Unless otherwise noted, C-shell syntax is used for setting environment variables and other shell commands.

```
% top
```

Acknowledgments

The book you are holding represents the efforts of many hundreds of people around the world. Whether they sent in fixes for typos, or submitted complete chapters, all the contributions have been useful.

Several companies have supported the development of this document by paying authors to work on it full-time, paying for publication, etc. In particular, BSDi (subsequently acquired by [Wind River Systems](#)) paid members of the FreeBSD Documentation Project to work on improving this book full time leading up to the publication of the first printed edition in March 2000 (ISBN 1-57176-241-8).

Wind River Systems then paid several additional authors to make a number of improvements to the print-output infrastructure and to add additional chapters to the text. This work culminated in the publication of the second printed edition in November 2001 (ISBN 1-57176-303-1). In 2003-2004, [FreeBSD Mall, Inc](#), paid several contributors to improve the Handbook in preparation for the third printed edition. The third printed edition has been split into two volumes. Both volumes have been published as The FreeBSD Handbook 3rd Edition Volume 1: User Guide (ISBN 1-57176-327-9) and The FreeBSD Handbook 3rd Edition Volume 2: Administrators Guide (ISBN 1-57176-328-7).

Part I: Getting Started

This part of the handbook is for users and administrators who are new to FreeBSD. These chapters:

- Introduce FreeBSD.
- Guide readers through the installation process.
- Teach UNIX® basics and fundamentals.
- Show how to install the wealth of third party applications available for FreeBSD.
- Introduce X, the UNIX® windowing system, and detail how to configure a desktop environment that makes users more productive.
- Introduce Wayland, a new display server for UNIX®.

The number of forward references in the text have been kept to a minimum so that this section can be read from front to back with minimal page flipping.

Chapter 10. Configuring the FreeBSD Kernel

10.1. Synopsis

The kernel is the core of the FreeBSD operating system. It is responsible for managing memory, enforcing security controls, networking, disk access, and much more. While much of FreeBSD is dynamically configurable, some users may wish to configure and compile a custom kernel.

After reading this chapter, you will know:

- When to build a custom kernel.
- How to take a hardware inventory.
- How to customize a kernel configuration file.
- How to use the kernel configuration file to create and build a new kernel.
- How to install the new kernel.
- How to troubleshoot if things go wrong.

All of the commands listed in the examples in this chapter should be executed as **root**.

10.2. Why Build a Custom Kernel?

Traditionally, FreeBSD used a monolithic kernel. The kernel was one large program, supported a fixed list of devices, and in order to change the kernel's behavior, one had to compile and then reboot into a new kernel.

Today, most of the functionality in the FreeBSD kernel is contained in modules which can be dynamically loaded and unloaded from the kernel as necessary. This allows the running kernel to adapt immediately to new hardware and for new functionality to be brought into the kernel. This is known as a modular kernel.

Occasionally, it is still necessary to perform static kernel configuration. Sometimes the needed functionality is so tied to the kernel that it can not be made dynamically loadable. Some security environments prevent the loading and unloading of kernel modules and require that only needed functionality is statically compiled into the kernel.

Building a custom kernel is often a rite of passage for advanced BSD users. This process, while time consuming, can provide benefits to the FreeBSD system. Unlike the GENERIC kernel, which must support a wide range of hardware, a custom kernel can be stripped down to only provide support for that computer's hardware. This has a number of benefits, such as:

- Faster boot time. Since the kernel will only probe the hardware on the system, the time it takes the system to boot can decrease.
- Lower memory usage. A custom kernel often uses less memory than the GENERIC kernel by omitting unused features and device drivers. This is important because the kernel code remains resident in physical memory at all times, preventing that memory from being used by applications. For this reason, a custom kernel is useful on a system with a small amount of RAM.

- Additional hardware support. A custom kernel can add support for devices which are not present in the GENERIC kernel.



When building a custom kernel, it is important to note that non-default configurations are less thoroughly tested than the GENERIC configuration. While customizing the kernel can provide specific benefits it also increases the risk of encountering build or runtime issues. Custom kernel configurations are recommended only for advanced users who have a specific reason for making changes and are willing to engage in the debugging process if necessary.

Before building a custom kernel, consider the reason for doing so. If there is a need for specific hardware support, it may already exist as a module.

Kernel modules exist in `/boot/kernel` and may be dynamically loaded into the running kernel using `kldload(8)`. Most kernel drivers have a loadable module and manual page. For example, the `ath(4)` wireless network driver has the following information in its manual page:

Alternatively, to load the driver as a module at boot time, place the following line in `loader.conf(5)`:

```
if_ath_load="YES"
```

Adding `if_ath_load="YES"` to `/boot/loader.conf` will load this module dynamically at boot time.

In some cases, there is no associated module in `/boot/kernel`. This is mostly true for certain subsystems.

10.3. Finding the System Hardware

Before editing the kernel configuration file, it is recommended to perform an inventory of the machine's hardware. On a dual-boot system, the inventory can be created from the other operating system. For example, Microsoft®'s Device Manager contains information about installed devices.



Some versions of Microsoft® Windows® have a System icon which can be used to access Device Manager.

If FreeBSD is the only installed operating system, use `dmesg(8)` to determine the hardware that was found and listed during the boot probe. Most device drivers on FreeBSD have a manual page which lists the hardware supported by that driver. For example, the following lines indicate that the `psm(4)` driver found a mouse:

```
psm0: <PS/2 Mouse> irq 12 on atkbd0  
psm0: [GIANT-LOCKED]  
psm0: [ITHREAD]  
psm0: model Generic PS/2 mouse, device ID 0
```

Since this hardware exists, this driver should not be removed from a custom kernel configuration file.

If the output of `dmesg` does not display the results of the boot probe output, instead read the contents of `/var/run/dmesg.boot`.

Another tool for finding hardware is `pciconf(8)`, which provides more verbose output. For example:

```
% pciconf -lv
ath0@pci0:3:0:0:      class=0x020000 card=0x058a1014 chip=0x1014168c rev=0x01 hdr
=0x00
    vendor      = 'Atheros Communications Inc.'
    device      = 'AR5212 Atheros AR5212 802.11abg wireless'
    class       = network
    subclass    = ethernet
```

This output shows that the ath driver located a wireless Ethernet device.

The `-k` flag of `man(1)` can be used to provide useful information. For example, it can be used to display a list of manual pages which contain a particular device brand or name:

```
# man -k Atheros
ath(4)          - Atheros IEEE 802.11 wireless network driver
ath_hal(4)      - Atheros Hardware Access Layer (HAL)
```

Once the hardware inventory list is created, refer to it to ensure that drivers for installed hardware are not removed as the custom kernel configuration is edited.

10.4. The Configuration File

In order to create a custom kernel configuration file and build a custom kernel, the full FreeBSD source tree must first be installed.

If `/usr/src/` does not exist or it is empty, source has not been installed. Source can be installed with Git using the instructions in [“Using Git”](#).

Once source is installed, review the contents of `/usr/src/sys`. This directory contains a number of subdirectories, including those which represent the following supported architectures: amd64, i386, powerpc, and sparc64. Everything inside a particular architecture’s directory deals with that architecture only and the rest of the code is machine independent code common to all platforms. Each supported architecture has a `conf` subdirectory which contains the GENERIC kernel configuration file for that architecture.

Do not make edits to GENERIC. Instead, copy the file to a different name and make edits to the copy. The convention is to use a name with all capital letters. When maintaining multiple FreeBSD machines with different hardware, it is a good idea to name it after the machine’s hostname. This example creates a copy, named MYKERNEL, of the GENERIC configuration file for the `amd64` architecture:

```
# cd /usr/src/sys/amd64/conf
# cp GENERIC MYKERNEL
```

MYKERNEL can now be customized with any **ASCII** text editor. The default editor is `vi`, though an easier editor for beginners, called `ee`, is also installed with FreeBSD.

The format of the kernel configuration file is simple. Each line contains a keyword that represents a device or subsystem, an argument, and a brief description. Any text after a `#` is considered a comment and ignored. To remove kernel support for a device or subsystem, put a `#` at the beginning of the line representing that device or subsystem. Do not add or remove a `#` for any line that you do not understand.



It is easy to remove support for a device or option and end up with a broken kernel. For example, if the `ata(4)` driver is removed from the kernel configuration file, a system using **ATA** disk drivers may not boot. When in doubt, just leave support in the kernel.

In addition to the brief descriptions provided in this file, additional descriptions are contained in **NOTES**, which can be found in the same directory as **GENERIC** for that architecture. For architecture independent options, refer to `/usr/src/sys/conf/NOTES`.



When finished customizing the kernel configuration file, save a backup copy to a location outside of `/usr/src`.

Alternately, keep the kernel configuration file elsewhere and create a symbolic link to the file:

```
# cd /usr/src/sys/amd64/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL
```

An **include** directive is available for use in configuration files. This allows another configuration file to be included in the current one, making it easy to maintain small changes relative to an existing file. If only a small number of additional options or drivers are required, this allows a delta to be maintained with respect to **GENERIC**, as seen in this example:

```
include GENERIC
ident MYKERNEL

options      IPFIREWALL
options      DUMMYNET
options      IPFIREWALL_DEFAULT_TO_ACCEPT
options      IPDIVERT
```

Using this method, the local configuration file expresses local differences from a **GENERIC** kernel.

As upgrades are performed, new features added to GENERIC will also be added to the local kernel unless they are specifically prevented using `nooptions` or `nodevice`. A comprehensive list of configuration directives and their descriptions may be found in [config\(5\)](#).



To build a file which contains all available options, run the following command as `root`:

```
# cd /usr/src/sys/arch/conf && make LINT
```

10.5. Building and Installing a Custom Kernel

Once the edits to the custom configuration file have been saved, the source code for the kernel can be compiled using the following steps:

Procedure: Building a Kernel

1. Change to this directory:

```
# cd /usr/src
```

2. Compile the new kernel by specifying the name of the custom kernel configuration file:

```
# make buildkernel KERNCONF=MYKERNEL
```

3. Install the new kernel associated with the specified kernel configuration file. This command will copy the new kernel to `/boot/kernel/kernel` and save the old kernel to `/boot/kernel.old/kernel`:

```
# make installkernel KERNCONF=MYKERNEL
```

4. Shutdown the system and reboot into the new kernel. If something goes wrong, refer to [The kernel does not boot](#).

By default, when a custom kernel is compiled, all kernel modules are rebuilt. To update a kernel faster or to build only custom modules, edit `/etc/make.conf` before starting to build the kernel.

For example, this variable specifies the list of modules to build instead of using the default of building all modules:

```
MODULES_OVERRIDE = linux acpi
```

Alternately, this variable lists which modules to exclude from the build process:

```
WITHOUT_MODULES = linux acpi sound
```

Additional variables are available. Refer to [make.conf\(5\)](#) for details.

10.6. If Something Goes Wrong

There are four categories of trouble that can occur when building a custom kernel:

config fails

If **config** fails, it will print the line number that is incorrect. As an example, for the following message, make sure that line 17 is typed correctly by comparing it to GENERIC or NOTES:

```
config: line 17: syntax error
```

make fails

If **make** fails, it is usually due to an error in the kernel configuration file which is not severe enough for **config** to catch. Review the configuration, and if the problem is not apparent, send an email to the [FreeBSD general questions mailing list](#) which contains the kernel configuration file.

The kernel does not boot

If the new kernel does not boot or fails to recognize devices, do not panic! Fortunately, FreeBSD has an excellent mechanism for recovering from incompatible kernels. Simply choose the kernel to boot from at the FreeBSD boot loader. This can be accessed when the system boot menu appears by selecting the "Escape to a loader prompt" option. At the prompt, type **boot** `kernel.old`, or the name of any other kernel that is known to boot properly.

After booting with a good kernel, check over the configuration file and try to build it again. One helpful resource is `/var/log/messages` which records the kernel messages from every successful boot. Also, [dmesg\(8\)](#) will print the kernel messages from the current boot.



When troubleshooting a kernel make sure to keep a copy of a kernel that is known to work, such as GENERIC. This is important because every time a new kernel is installed, `kernel.old` is overwritten with the last installed kernel, which may or may not be bootable. As soon as possible, move the working kernel by renaming the directory containing the good kernel:

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

The kernel works, but **ps(1)** does not

If the kernel version differs from the one that the system utilities have been built with, for example, a kernel built from `-CURRENT` sources is installed on a `-RELEASE` system, many system status commands like **ps(1)** and [vmstat\(8\)](#) will not work. To fix this, [recompile and install a world](#) built with the same version of the source tree as the kernel. It is never a good idea to use a

different version of the kernel than the rest of the operating system.