FreeBSD

使用手冊
摘要
歡迎使用 FreeBSD！本使用手冊涵蓋範圍包括了 FreeBSD 14.0-RELEASE, 13.2-RELEASE 與 FreeBSD 12.4-RELEASE 的安裝與平日操作的說明。
這份使用手冊是很多人的集體創作,而且仍然『持續不斷』的進行中,因此部份章節可能尚未仍未完成,如您有興趣協助本計畫的話,請寄電子郵件至 FreeBSD documentation project 郵遞論壇。
在 FreeBSD 網站可以找到本手冊的最新版本,舊版文件可從 https://docs.FreeBSD.org/doc/ 取得。本文件也提供各種格式與不同壓縮方式的版本可自 FreeBSD 下載伺服器或是其中一個鏡像網站下載。 此外,您可在搜尋頁面中搜尋本文件或其他文件的資料。
序

目

2. 系統

5. X Window

4. 應用程式

安裝：套件與

3. FreeBSD

基礎


術語

5.2.

述

5.1.

Port

4.7.

事項

編譯套件

4.6.

使用

Poudriere 軟件

4.5.

套使用

套管理 pkg

4.3.

搜尋安裝體的概要

3.12.

3.11.

點

裝

編輯

 Daemon

3.8.

案系統載與
d 檔

3.6.

結权限

3.4.

管理

3.3.

帳號

使用者與基礎

Console

3.2.

3.1.

述

概

Live CD

2.10.

使用

2.9.

疑難

2.8.

安裝後注意

認

間

2.6.

bsdinstall

2.5.

安裝

作

2.3.

準備工

最

體需

2.2.

硬

述

2.1.

劃

FreeBSD

1.3.

FreeBSD

1.2.

1.1.

.
23.6. 追蹤
24.2. DTrace
24.3. DTrace 启作差異
24.4. 使用開支援
25. USB Device Mode / USB OTG
26. 端機序出撥
26.5. 撥終端機列術語序概述
26.6. 撥序概述
27. PPP (PPPoA) 網使用太PPP (PPPoE)路乙
27.2. PPP 述
27.1. 設定
28.3. Sendmail 設定檔郵件成組述
28.6. 排解更改郵件代理程式
28.7. 寄件
28.8. 使用郵件撥號
28.9. SMTP 28.10. 使用郵件撥號
28.10. 29.1. 10. Microsoft™Windows™ 協伺服器網
29.7. 系統域名稱
29.6. 定協態 (DHCP)
29.5. 量定輕路資 (NIS)
29.4. 網訊路檔
29.2. inetd 伺服器述
29.1. 超級
給讀者的話
若您是第一次接觸 FreeBSD 的新手,可以在本書第一部分找到 FreeBSD 的安装程序,同时会逐步介绍 UNIX™ 的基础概念与一些常用、共通的东西。而阅读这部分并不难,只需要您有探索的精神和接受新概念。

读完这些之后,手册中的第二部分花很长篇幅介绍的各種广泛主题,相当值得系统管理者去注意。在阅读这些章节的内容时所需要的背景知识,都注明在该章的大纲里面,若不熟的话,可在阅读前先预习一番。

延伸阅读方面,可参阅 参考书目。

自第三版後的主要修订
您目前看到的这本手册代表著上百位貢献者歷時10年所累积的心血之作。以下為自2014年發佈的兩冊第三版後所做的主要修订:

• DTrace 增加说明有关强大的DTrace效能分析工具的资讯。

• 其他 檔案系统增加有关 FreeBSD非原生档案系统的资讯,如来自Sun™的ZSF。

• 安全事件稽查 增加的内容涵盖 FreeBSD的新稽查功能及其使用说明。

• 虚拟化 增加有关在虚拟化软體安装 FreeBSD的资讯。

• 安装 FreeBSD增加的内容涵盖使用新安装工具 bsdinstall 来安装 FreeBSD。

自第二版後的主要修订 (2004)
您目前看到的这本手册第三版是 FreeBSD文件計划的成員歷時兩年完成的心血之作。因文件內容成長到一定大小,印刷版需要分成兩冊發佈。新版的主要修订部分如下:

• 設定與調校 已针对新内容作更新,如:ACPI電源管理、cron 以及其他更多的核心調校選項說明內容。

• 安全性 增加了虛擬私人網路(VPN)、檔案系统的存取控制(Access Control List),以及安全報告。

• 強制存取控制(Mandatory Access Control)是此版本新増的章節。该章介绍:什么是MAC机制?以及如何运用它来使您的 FreeBSD系统更安全。

• 儲存設備 新増了像是:USB随身碟、檔案系統快照(Snapshot)、檔案系統配額(Quota)、檔案与網路为基是的檔案系统、以及如何对硬碟分割区作加密等详解。

• PPP 增加了疑難排解的章節。

• 電子郵件 新增有关如何使用其它的传送代理程式、SMTP认证、UUCP、fetchmail、procmail的运用以及其它進階主題。

• 網路伺服器 是該版中全新增的一章。这一章介绍如何架設Apache HTTP伺服器、ftpd 以及用於支援 Microsoft™Windows™客户端的Samba。其中有些段落來自原先的進階網路设定。

• 进階網路设定 新増有关在 FreeBSD中使用藍牙™装置、设定無線網路以及使用非同步传输模式(Asynchronous Transfer Mode, ATM)网络的介绍。

• 增加詞彙表,用以說明全書中出現的術語。

• 重新美編 书中所列的图表。
每個章節均加入概述，以便快速瀏覽該章節內容摘要，讓讀者所欲了解的部分。

內容架構重新組織成三大部分："入門"、"系統管理"以及"附錄"。

FreeBSD基礎新増了程序、Daemon以及信號(Signal)的介紹。

安裝應用程式：套件與Port新増了介紹如如何管理Binary套件的資訊。

X Window系統經過全面改寫，著重於在XFree86™ 4.X上的現代桌面技術，如KDE和GNOME。

FreeBSD開機程序更新相關內容。

儲存設備分別以兩個章節"磁碟"與"備份"來撰寫。我們認為這樣子會比單一章節來得容易瞭解。還有關於RAID (包含硬體、軟體RAID)的段落亦新増上去了。

序列通訊架構重新改寫，并更新至FreeBSD 4.X/5.X的內容。

PPP有相當程度的更新。

進階網路設定加入許多新內容。

電子郵件大量新增了設定sendmail的介紹。

Linux® Binary相容性新增了有關安裝Oracle™及SAP™ R/3™的介紹。

此外，第二版還新增章節，以介紹下列新主題：
◦ 設定與調校
◦ 多媒體

本書構架本書主要分為五大部分，第一部分入門：介紹FreeBSD的安裝、基本操作。讀者可根據自己的程度，循序或者跳過一些熟悉的主題來閱讀；第二部分一般作業：介紹FreeBSD常用功能，這部分可以不按順序來讀。每章前面都会有概述，概述會描述本章節涵蓋的內容和讀者應該已知的，這主要是讓讀者可以挑喜好的章節閱讀；第三部分系統管理：介紹FreeBSD老手所感興趣的各種主題部分；第四部分網路通訊：則包括網路和各式伺服器主題；而第五部分則為附錄包含各種有關FreeBSD的資源。
列印介紹如何在 FreeBSD 管理印表機，包含橫幅頁面、列印帳務以及初始設定等資訊。

Linux® Binary 相容性介绍 FreeBSD 的 Linux™ 相容性功能，同时提供许多热门的 Linux™ 应用程式详细的安装操作说明，例如 Oracle™ 及 Mathematica™。

设定与调校介绍可供系统管理者用来调校 FreeBSD 系统的可用参数来最佳化效率，同时也介绍 FreeBSD 用到的各种设定档以及到何处寻找这些设定档。

FreeBSD 开機程序介绍 FreeBSD 开機流程并说明如何使用设定选项控制開機流程。

安全性介绍许多可让您的 FreeBSD 系统更安全的各种工具，包含 Kerberos, IPsec 及 OpenSSH。

Jail 介绍 Jail Framework，以及 Jail 改进那些 FreeBSD 传统 chroot 不足的地方。

強制存取控制(MAC)说明什么是強制存取控制(Mandatory Access Control, MAC)及这个机制如何用来确保 FreeBSD 系统的安全。

安全事件稽查介绍什么是 FreeBSD 事件稽查，如何安装与设定，以及如何检查与监控稽查线索。

储存设备介绍如何在 FreeBSD 管理储存媒体及檔案系统，这包含实体磁碟、RAID 陣列、光碟与磁带媒体、記憶體为基础的磁碟以及網路檔案系统。

GEOM. 模組化磁碟转换框架介绍在 FreeBSD 中的 GEOM Framework 是什么，以及如何设定各种支援的 RAID 階層。

其他档案系统查询看 FreeBSD 还支援哪些非原生档案系统，如 Sun™ 的 Z 檔案系统。

虛拟化介绍虛擬化系统提供了哪些功能，以及如何在 FreeBSD 上使用。

在地化-i18n/L10n 使用与安装介绍如何在 FreeBSD 使用非英文的言语，这涵盖了系统及应用层的在地化。

更 新 与 升 級 FreeBSD 说明 FreeBSD-STABLE、FreeBSD-CURRENT 以 及 FreeBSD 離版之间的差异，并介绍那些使用者适何追踪开发系统以及程序的概述，这涵盖了使用者更新系统到最新安全性发布版本的方法。

DTrace 介绍如何在 FreeBSD 設定及使用 Sun™ 的 DTrace 工具，动态追蹤可以通过执行真实时间系统分析来协助定位效能问题。

序列通讯介绍如何使用拨入及拨出连线到您的 FreeBSD 系统的終端機与數據機。

PPP介绍如何在 FreeBSD 使用 PPP 来连线至遠端的系统。
電子郵件

說明組成電子郵件伺服器的各種元件，並深入說明如何設定最熱門的郵件伺服器軟體：sendmail。

網路伺服器提供詳細的操作說明與範例設定檔，讓您可安裝您的FreeBSD機器為網路檔案伺服器、網域名稱伺服器、網路資訊系統伺服器或時鐘伺服器。

防火牆說明軟體為基礎的防火牆背後的理念，並提供可用於FreeBSD中不同的防火牆設定的詳細資訊。

進階網路設定介紹許多網路主題，包含在您的區域網路(LAN)分享網路際網路連線給其他電腦、進階路由主題、無線網路、Bluetooth™、ATM、IPv6以及更多相關主題。

取得FreeBSD列出取得FreeBSD CDROM或DVD媒體的各種來源，以及在網路上的各種網站，讓您可以下載並安裝FreeBSD。

参考書目本書觸及許多不同主題，可能會讓您想更深入的了解，參考書目列出在文中引用的許多優秀書籍。

網路資源介紹了可讓FreeBSD使用者提出問題以及參與有關FreeBSD技術會談的多論壇。

OpenPGP金鑰列出數個FreeBSD開發人員的PGP指紋。

本書的編排體裁為了提供有一致性且易於閱讀的內容，以下是一些本書所遵循的編排體裁。

文字編排體裁斜體字用於：檔名、目錄、網址(URL)、強調語氣、以及第一次提及及的技術詞彙。

等寬字用於：錯誤訊息、指令、環境變數、Port名稱、主機名稱、帳號、群組、裝置名稱、變數、程式碼等。

粗體字以粗體字表示：應用程式、指令、按鍵。

使用者輸入鍵盤輸入以粗體字表示，以便與一般文字做區隔。組合鍵是同時按下一些按鍵，我們以+來表示連接，像是：Ctrl+Alt+Del是一起按下Ctrl、Alt以及Del鍵。

若要逐一按鍵，那會以逗號(,)來表示，像是：Ctrl+X, Ctrl+S是說：先同時按下Ctrl與X鍵，然後放開後再同時按下Ctrl與S鍵。
範例

以`C:`為開頭代表MS-DOS™的指令。若沒有特殊情況的話，這些指令應該是在Microsoft™ Windows™環境的"指令提示字元"視窗內執行。

範例

以`#`為開頭代表在FreeBSD中以超級使用者權限來執行的指令。你可先以root登入系統並下指令，或是以你自己的帳號登入再使用`su(1)`來取得超級使用者權限。

範例

以 `%` 為開頭代表在FreeBSD中以一般使用者帳號執行的指令。

範例

以`%`為開頭代表在FreeBSD中以一般使用者帳號執行的指令。除非有提到其他用法，否則都是預設為C-shell語法，用來設定環境變數及下其他指令的意

銘謝

您所看到的這本書是經過數百個分散在世界各地的人來努力而來的結果。無論他們只是糾正一些錯誤或提交完整的章節，所有的點滴貢獻都是非常寶貴有用的。


• 介紹FreeBSD給您。
• 在安裝過程中給您指引。
• 教您UNIX™的基礎及原理。
• 展示給您看如何安裝豐富的FreeBSD的應用軟體。
• 向您介紹X, UNIX™的視窗系統及詳細的桌面環境設定，讓您更有生產力。
Part I: \n入門

這部份是提供給初次使用FreeBSD的使用者和系統管理者。這些章節包括：

• 介紹FreeBSD給您。
• 在安裝過程給您指引。
• 教您UNIX®的基礎及原理。
• 展示給您看如何安裝豐富的FreeBSD的應用軟體。
• 向您介紹X，UNIX®的視窗系統及詳細的桌面環境設定，讓您更有生產力。

我們試著儘可能的讓這段文字的參考連結數目降到最低，讓您在讀使用手冊的這部份時可以不太需要常常前後翻頁。
多世界已經商許及上開源應用使用建產品多訊來做為置用、裝被立 FreeBSD 讓先進的及自全性能、成熟的安條款 FreeBSD 的，功測 NetEase FreeBSD 聞上運 - 名 FreeBSD 能而服以其網頁務功在用 1.2.2. FreeBSD 录由情燒你 FreeBSD DVD 請參閱抓回或是從站 CD-ROM。詳由程式碼個人都可以取得於每的 FreeBSD。 以下提供一些人 FreeBSD 办使用改廠商地被系統通常：

- FreeNAS
- iXsystems 商和
- iXsystems 為己基礎。的自產品業也管理著公司
dell EMC Isilon 的、可擴展的可靠社
ewith Quest KACE,並專注智慧財它們系統。整了己產打造整自的到 Dell EMC Isilon 活動大的 FreeBSD Shell 網端取,及運用了行雲皆路存能。

citrix 元含素中有效。過擁140上運都是在超支援語與件管理系統也可言需多其他 套軟體開發：體中央透工，開機 更便無磁碟作站變個人至可以更易於中伺服器變

- GNOME 多面選可擇化 KDE 型使用者介面。了，包許 FreeBSD 開源桌基礎。置成置建及入嵌其他裝為佳來式防火牆的路再由及平的自，的網全能以路環境健台端：

- 486 PC 甚變身出的伺服器，網路: 題，而能自在的必當流置兩開放性系統，不作計地版及問的交心由的具

- 有 FreeBSD 特研究些需要在電完成形對於設學圖那計件腦學系統、計了。其中結構路到：

- 教育

- 路務：

- 服它：

- 路

- 再由及平的自，的網全能以路環境健台端：

- 核心來運 FreeBSD - IronPort 作。後改路郵件反垃圾安 NetScaler 提供的第全設備層的、安應負層防火牆均載快取、內容路大

- Apple iOS 元虛擬檔空案系統以 - OS X Stack、量路大

- 由防火牆或安可以，器、將你沒易的有用到的全名稱伺服器如果需要輕 FreeBSD 的，業作有了完是研究佳環境碼及。FreeBSD 的

- 再使用嗎沒？

- 脳科學關領 FreeBSD 學域的比您是電也生

- 會中大部分面軟 - Apache 對設大庫及許的的網 Apple iOS 元虛擬檔空案系統以 - OS X Stack、量路大

- 中為設備 FreeBSD 系統管理 FreeBSD - KACE

- 硬件數量的網 Sina

- Sony Japan

- Rambler

- Hacker News

- Netcraft 購得BSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD

- 取得 FreeBSD
 FreeBSD 關了計數的開源催
16•
現韌上路取代 ZRouter
tpSence 能給所有使用者,專的 OS X 通過解來計功強
TrueOS 防火牆功能集,同時在商有的才和安數昂的能。開放它帶來有大多。和置由防火牆、建使用基礎的 NAS4Free 有以,
FreeBSD Jail 這基礎的 SMB/CIFS FTP 基礎的網頁介面、來簡化、支援與檔 UFS ZFS 良及最的安基礎,包括
ZFS, GELI, Capsicum, HAST 承業是以全性設備讓與管理以能控皆安功。這些業商伺服器擇萬的平了模線連每大過規它們。台台。它們選接著 FreeBSD 250 也的功 ZFS。TRIM 了 ZFS 支援能之外利且便擁型因是更的員環境詳人開發除的文件、活躍的原
Voxer 確保各種不同的網基礎的 FreeBSD 業設作運仰賴作。這些基礎系統包括
設備便電視節目的有報當地天氣端預到有安裝在各電視線 IntelliStar
The Weather Channel 將授權展產。並同時發群讓他們可大回饋與系統合
Spectra Logic 資,同時也可描入站郵件中的監感軟掃基礎,可訊意惡郵件和控出站郵件中的及體
Sophos Sony 台策略性高。產品處網作它們路立時網即的基礎的建為它路時網 32% 上路流量。以美做持與主 FreeBSD,並致力量貢獻修的的線正 OpenConnect
Netflix 了所有的網 32% 上路流量。以 Netflix 的
NetApp 基礎。改給回整 FreeBSD 能從回饋性開發的進內容會的
間基礎。FreeBSD 之為的共生關產品 Juniper 商是以劃展了計與供由便厂商之中,多兩開發開源個計的
的機
- Voxer - Voxer, FreeBSD 使用
- OpenConnect - FreeBSD 設備是以串電到用
- JunOS - FreeBSD 設備是以串電到用
- SecurOS - FreeBSD 設備是以串電到用
- Sophos - FreeBSD 設備是以串電到用
- Stormshield - FreeBSD 以串電到用
- nTier - FreeBSD 以串電到用
- FUDO - FreeBSD 以串電到用
- VeriSign.net 及隨附業務網以 DNS 經的與註冊主要
- NetApp - FreeBSD 基礎。改給回整 FreeBSD 能從回饋性開發的進內容會的
間基礎。FreeBSD 之為的共生關產品 Juniper 商是以劃展了計與供由便厂商之中,多兩開發開源個計的
的機
- Voxer - Voxer, FreeBSD 使用
- OpenConnect - FreeBSD 設備是以串電到用
- JunOS - FreeBSD 設備是以串電到用
- SecurOS - FreeBSD 設備是以串電到用
- Sophos - FreeBSD 設備是以串電到用
- Stormshield - FreeBSD 以串電到用
- nTier - FreeBSD 以串電到用
- FUDO - FreeBSD 以串電到用
- VeriSign.net 及隨附業務網以 DNS 經的與註冊主要
- NetApp - FreeBSD 基礎。改給回整 FreeBSD 能從回饋性開發的進內容會的
間基礎。FreeBSD 之為的共生關產品 Juniper 商是以劃展了計與供由便厂商之中,多兩開發開源個計的
的機
- Voxer - Voxer, FreeBSD 使用
- OpenConnect - FreeBSD 設備是以串電到用
- JunOS - FreeBSD 設備是以串電到用
- SecurOS - FreeBSD 設備是以串電到用
- Sophos - FreeBSD 設備是以串電到用
- Stormshield - FreeBSD 以串電到用
- nTier - FreeBSD 以串電到用
- FUDO - FreeBSD 以串電到用
- VeriSign.net 及隨附業務網以 DNS 經的與註冊主要
- NetApp - FreeBSD 基礎。改給回整 FreeBSD 能從回饋性開發的進內容會的
間基礎。FreeBSD 之為的共生關產品 Juniper 商是以劃展了計與供由便厂商之中,多兩開發開源個計的
的機
- Voxer - Voxer, FreeBSD 使用
- OpenConnect - FreeBSD 設備是以串電到用
- JunOS - FreeBSD 設備是以串電到用
- SecurOS - FreeBSD 設備是以串電到用
- Sophos - FreeBSD 設備是以串電到用
- Stormshield - FreeBSD 以串電到用
- nTier - FreeBSD 以串電到用
- FUDO - FreeBSD 以串電到用
- VeriSign.net 及隨附業務網以 DNS 經的與註冊主要
- NetApp - FreeBSD 基礎。改給回整 FreeBSD 能從回饋性開發的進內容會的
間基礎。FreeBSD 之為的共生關產品 Juniper 商是以劃展了計與供由便厂商之中,多兩開發開源個計的
的機
- Voxer - Voxer, FreeBSD 使用
- OpenConnect - FreeBSD 設備是以串電到用
- JunOS - FreeBSD 設備是以串電到用
- SecurOS - FreeBSD 設備是以串電到用
- Sophos - FreeBSD 設備是以串電到用
- Stormshield - FreeBSD 以串電到用
- nTier - FreeBSD 以串電到用
- FUDO - FreeBSD 以串電到用
- VeriSign.net 及隨附業務網以 DNS 經的與註冊主要
- NetApp - FreeBSD 基礎。改給回整 FreeBSD 能從回饋性開發的進內容會的
FreeBSD

FreeBSD has been widely used, thereby increasing its value. The cost of using these programs made it possible for people to provide programs for others to use, whether they were free or not. We believe that we should provide some resources that can be used. We are not indifferent to support. Sometimes, it is necessary to provide resources for others.
核心交檔案件計單獨開發作，多無論式的 FreeBSD 者或者錯和團隊對進希望些的人只需要在進一步參與計的計畫感計的作。架構工的透以者貢獻由來全發 FreeBSD 的世界像從的彈就具非就發佈軟版若寬途的選較限限的要的。少的有碼使用強制性通用我們條款公授權干（GPL）或者程式我認為倡導體最基本的，同時也是所要的，絕重，最不重要的，這個人之中交是在 2018 新的 7 年招提也是交之一，者的行職及修及錯誤就提供。與最大的的使用者自為我們最後一絕要的，重修及錯誤便進行互便修及錯誤就提供。與最大的的使用者自為我們最後一絕要的，重修及錯誤便進行互便修及錯誤就提供。與最大的的使用者自為我們最後一絕要的，重修及錯誤便進行互便修及錯誤就提供。
總而言之，我們的開發模式像是由鬆散的同心圓所組織。這個集中的模式設計的是讓 FreeBSD 的使用者更便利，可以很容易地追蹤同一個中央的程式庫，避免把潛在的貢獻者排除在外！而我們的目標是提供一個穩定的作業系統，並有大量相關的應用程式，讓使用者能夠輕鬆的安裝與使用——而這個開發模式對我們要完成這個目標來說運作的非常好。

我們對於那些想加入 FreeBSD 開發者的期待是：請保持如同前人一樣的投入，以確保繼續成功！

1.3.4. 第三方程式
除了基礎發行版之外，FreeBSD 提供了擁有上千個常用的程式的移植軟體的套件集，在撰寫本文的同時，已有超過 24,000 個 Port！Port 的範圍從 HTTP 伺服器到遊戲、語系、編輯器，幾乎所有東西都在裡面。完整的 Port 套件集需要將近 500 MB。要編譯一個 Port 您只需要切換目錄到您想安裝的程式目錄，然後輸入 make install，接著系統便會處理剩下的動作。您編譯的每個 Port 完整原始發行版內容是動態下載的，所以您只需要有足夠的磁碟空間來編譯您想要的 Port。幾乎所有 Port 都提供已經預先編譯好的"套件"，您可以透過簡單的指令來安裝 (pkg install)，提供那些不想要自行從原始碼編譯的人使用。更多有關套件與 Port 的資訊可於安裝應用程式：套件與 Port 取得。1.3.5. 其他文件
Chapter 2. Installing FreeBSD

2.1. Overview

This chapter, along with the next one, will explain how to install FreeBSD, and will also discuss some of the specialities of the operating system. It is important to understand that the installation system used on FreeBSD is a command line user interface, and as such, it is designed to work with the UNIX shell, which is the standard shell used in most UNIX systems. The installer will be used to create a new installation of FreeBSD, and will use the command line to perform most of the work. The installer will also be used to install FreeBSD on other systems, and will be used to configure the system. It is important to understand that the installer is not a graphical user interface (GUI), and that it is designed to work with the command line. The installer will be used to install FreeBSD on other systems, and will be used to configure the system. It is important to understand that the installer is not a graphical user interface (GUI), and that it is designed to work with the command line. The installer will be used to install FreeBSD on other systems, and will be used to configure the system. It is important to understand that the installer is not a graphical user interface (GUI), and that it is designed to work with the command line.
以下清項這些之前，先做目是好查檔符合媒的最低案並體需準備可以下載安裝確一準備工sparc64powerpcia64i386系統共用磁碟。需要專用的此時業法SMP處所有過列支援在2 GB位核心的支援支援器都有在多機系統都有SMP。多處E8870理器稱（Uniprocessor, UP）。Itanium™ 2包括和理器是組晶片目前的其他情請見體，並驅限制PAE但的的程式體能所PAE偵測超用該功能核心支援延伸能可在有功（Physical Address Extensions, PAE）體位實的階幾支援更相容乎所有處運支援更腦架構的型舊與記型桌處理器。Intel™ Xeon™包含：理器範核心例處與Intel64為商則。
兩種。隨身碟
USB 象
檔 2.3.1.1.
必須
不相
映
該要完
符
校
校
計
與
後的
計
的
來
像
之中的
下載完成之後, 下載同一個目
標
接用
檔。有
應
直
隨身碟
的系統可能會
強制
Mac OS™ 小寫
需要使用
能需要使用
隨身碟
該裝
否
割區
業
指向
。有些作
載, 以
失敗
行
執
非

• 使用 Procedure:
  1. 指令列工具
    BSD, Linux™ 皆
    可使用。要使用
    在
    含
    及
    的
    Port
    套
    件集, 也內
    含原始
    档
    案
    及
    档
    所需的
    件集。這個
    含原始
    套
    有所有安裝
    重
    有的資料。
    在
    除
    在
    要資料, 這個程序會
    既
    清
    繼之前, 請先
    其他作
    似
    類
    系統也會有
    的資料。
    小心
    例
    来
    置
    裝
    目
    确
    要使用的裝
    標
    位
    因
    指定
    置上已存
    例
    系
    系統
    的
    有
    所不同,
    的
    業
    例
    系
    的
    程序
    為
    位置
    必
    標
    的
    作
    程序
    使用方式
    來
    SUM(8)
    檔
    簡
    的安裝
    这是最精
    含
    档
    這是最精
    档
    FreeBSD
    案
    案, 包
    档
    FreeBSD
    案
    資料
    的安裝
    程序
    案, 包
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    件
    FreeBSD
    案
    資料
    的安裝
    案, 包
    档
    資料
    的安裝
    程序
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
    案
    案
    档
    FreeBSD
使用 Windows™ 製作磁碟映像

1. 取得 Image Writer Windows™
   Image Writer Windows™ 是一個免費應用程式，可以正確地將映像檔寫入隨身碟。可從 https://sourceforge.net/projects/win32diskimager/ 下載，並解壓縮到一個資料夾。

2. 用 Image Writer 寫入映像檔
   - 雙擊 Win32DiskImager 圖示啟動程式。
   - 確認 Device 显示的磁碟機代號是隨身碟的磁碟機代號。
   - 按下資料夾圖示選擇要寫入隨身碟的映像檔。
   - 按下【Save】按鈕確定映像檔名。
   - 確認所有東西都正確，隨身碟的資料夾並沒有在其他視窗開啟。所有東西準備好後，按下【Write】將映像檔寫入隨身碟。

您現在可以開始安裝 FreeBSD。

2.4. 開始安裝

- 預設安裝程序在下列訊息顯示之前不會對磁碟做任何更動：
  Your changes will now be written to disk. If you have chosen to overwrite existing data, it will be PERMANENTLY ERASED. Are you sure you want to commit your changes?

在這個警告訊息之前可以隨時中止安裝，若有任何設定錯誤的疑慮，只需在此時關閉電腦，將不會對系統磁碟做任何更改。

2.4.1. 本節將介紹如何使用根據準備安裝的媒體指示所準備的安裝媒體來開機。要使用可開機的 USB, 請在開啟電腦前插入 USB 隨身碟。要使用 CD 或 DVD, 則可在開啟電腦後第一時間插入媒體。如何設定系統使用插入的媒體開機依不同的系統架構會有所不同。

2.4.2. 在 PowerPC™ 開機

大部份機型，可於開機時按住鍵盤上的 C, 便可從 CD 開機。若在非 Apple™ 的鍵盤則可按住 Command + Option + O + F 或 Windows + Alt + O + F, 出現 0 > 提示時, 輸入 24。
在 SPARC64™ 開機大多數 SPARC64™ 系統會自動從磁碟開機，要從 CD 安裝 FreeBSD 需要進入 PROM。

要進入 PROM，需重新開機系統然後等候開機訊息出現。訊息會依機型而有所不同，但大致結果會如:

Sun Blade 100 (UltraSPARC-IIe), Keyboard Present

Copyright 1998-2001 Sun Microsystems, Inc. All rights reserved.

OpenBoot 4.2, 128 MB memory installed, Serial #51090132.

若系統繼續從磁碟開機，此時按上鍵盤上鍵盤上L1+A 或 Stop+A 或 透過序列 Console 送出 BREAK。當使用 tip 或 cu，~# 發出一個 BREAK 後，PROM 的提示會在單 CPU 的系統出現 ok，SMP 的系統出現 ok {0}，其中的數字代表啟動的 CPU 數。

此時，放入 CD 到磁碟機然後在 PROM 提示畫面輸入 boot cdrom。

2.4.4. FreeBSD 開機選單

從安裝媒體開機之後，會顯示如下的選單:

1. Boot Multi user [Enter]
2. Boot Single user
3. Escape to loader prompt
4. Reboot
5. Cons: Video

Options:
6. Kernel: default/kernel (1 of 1)
7. Boot Options

Autoboot in 7 seconds. [Space] to pause
在開機進入FreeBSD安裝程式前選單會等候使用者輸入10秒鐘，若已經安裝FreeBSD，則會在開機進入FreeBSD前等候。要暫停開機計時器來仔細查看選項，請按Space鍵。要選擇選項，按下明顯標示的數字、字元或鍵。選單有以下選項可選。

• 启動多使用者模式 (Boot Multi User)：這個選項會繼續FreeBSD開機程序，若開機計時器已經暫停，可按1、大寫或小寫B或Enter鍵。

• 启動單使用者模式 (Boot Single User)：這個模式用來修正已安裝的FreeBSD，如單使用者模式所述。可按2、大寫或小寫S進入這個模式。

• 離開到載入程式提示 (Escape to loader prompt)：這個選項會開機進入修復提示，這個模式含有限數量的低階指令，這個模式詳細說明於FreeBSD開機選項選單階段三。可按3或Esc進入這個提示。

• 重新開機 (Reboot)：重新開機系統。

• 設定開機選項 (Configure Boot Options)：開啟內部選單，詳細說明於FreeBSD開機選項選單。
• 详细资讯（Verbose）：切换这个选项为开（On）来查看开机程序中更详细的信息，这在诊断硬件问题时非常有用。

在做完所需的选项后，按下1或Backspace返回主开机选单，然后按下Enter继续开机进入FreeBSD。

FreeBSD执行装置侦测及载入安装程式时会显示一系列的开机资讯，开机完成之后会显示欢迎选单。

图3. 欢迎选单

按一下Enter选择预设的[Install]进入安装程式，接下来本章将介绍如何使用这个安装程式。

若要选择其他项目，可使用右或左方向键或颜色标示的字母选择想要的选单项目。

[Shell] 可用来自进入FreeBSD的Shell使用指令列工具在安装之前准备磁碟。

[Live CD] 选项可用来在安装之前试用FreeBSD，Live版本的详细说明于使用Live CD。

要重新检视开机资讯，包含硬体装置侦测，请按大写或小写S然后再按Enter进入Shell。在Shell提示之后输入more /var/run/dmesg.boot然后使用空白键来捲动资讯。当查看完毕后输入exit返回欢迎选单。

2.5. 使用bsdinstall
本节将告诉您在系统安装之前bsdinstall选单的顺序以及会询问的资讯类型，可使用方向键来选择选单的选项，然后按下Space选择或取消选单项目。当完成之后，按下Enter储存选项然后进入下一个画面。

2.5.1. 选择键键盘对应表
选单依据使用的系统Console，bsdinstall可能一开始显示的选单会如键盘对应选择。
To set a non-default key mapping for your keyboard, select [YES] and press Enter. This will display the keyboard menu.

If you wish to use the default key mapping, select [NO] and press Enter to skip this menu.

When setting the keyboard configuration, you can use the up and down arrows to select the closest keyboard mapping and press Enter to save your choices.

Press Esc to exit this menu and use the default keyboard mapping. If you are unsure which keyboard mapping to use, United States of America ISO-8859-1 is also an option.

In FreeBSD 10.0-RELEASE
The system console driver for FreeBSD defaults to standard "US" keyboard map. Other keymaps can be chosen below.

**Continue with default keymap**

- Armenian phonetic layout
- Belarusian
- Belgian
- Belgian (accent keys)
- Brazilian (accent keys)
- Brazilian (without accent keys)
- Bulgarian (BOS)
- Bulgarian (Phonetic)
- Canadian Bilingual
- Central European
- Central European (QWERTY)
- Colemak Mod-DH ergonomic alternative
- Colemak ergonomic alternative
- Croatian
- Czech (QWERTZ, accent keys)
- Danish
- Danish (MacBook)
2.5.3. 选择要安装的组件

接下来，bsdinstall会提示选择要安装的组件。
選擇要安裝的元件主要會根據系統的用途以及可用的磁碟空間。

- **FreeBSD核心（Kernel）及Userland統稱為基礎系統（Base system），是必須安裝的部份。**
  - 系統的架構，部份元件可能不會顯示。
- **doc** - 额外的說明文件，大部分是經年累月的產物，會安裝到 `/usr/shared/doc`。由FreeBSD文件計畫所提供的說明文件可在之後安裝，依照更新文件集中的指示操作。
- **games** - 一些傳統BSD遊戲，包含fortune, rot13以及其他。
- **lib32** - 在64-bit版本的FreeBSD供執行32-bit應用程式使用的相容性程式庫。
- **ports** - FreeBSD套件是一套可自動下載、編譯安裝第三方軟體套件的集合，安裝應用程式：套件與Port中會討論到如何使用Port套件集。
- **src** - 完整的FreeBSD原始碼，包含核心（Kernel）與Userland。雖然大多數的應用程式並不需要，但它可以編譯裝置驅動程式、核心模組或部份來自Port套件集的應用程式，它同時也用來做為開發FreeBSD本身所使用。整個原始碼樹需要1 GB的磁碟空間，重新編譯整個FreeBSD系統需要額外再5 GB的空間。

2.5.4. 透過網路安裝

於從網路安裝所示的選單只會在使用 `-bootonly.iso` CD安裝時顯示，因這個安裝媒體中並未含安裝檔的複本。由於安裝檔必須透過網路下載，此選單會告知要先設定網路介面。
Network Installation

Some installation files were not found on the boot volume. The next few screens will allow you to configure networking so that they can be downloaded from the Internet.
Please select the best suitable site for you or "other" if you want to specify a different choice. The "Main Site" directs users to the nearest project managed mirror via GeoDNS (they carry the full range of possible distributions and support both IPv4 and IPv6). All other sites are known as "Community Mirrors", not every site listed here carries more than the base distribution kits. Select a site!

<table>
<thead>
<tr>
<th>Mirror Selection</th>
<th>Main Site (GeoDNS)</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://download.freebsd.org">http://download.freebsd.org</a></td>
<td>Australia - IPv6</td>
</tr>
<tr>
<td>ftp://ftp.au.freebsd.org</td>
<td>Australia #3</td>
</tr>
<tr>
<td>ftp://ftp.at.freebsd.org</td>
<td>Austria - IPv6</td>
</tr>
<tr>
<td>ftp://ftp2.br.freebsd.org</td>
<td>Brazil #2</td>
</tr>
<tr>
<td>ftp://ftp3.br.freebsd.org</td>
<td>Brazil #3</td>
</tr>
<tr>
<td>ftp://ftp.dk.freebsd.org</td>
<td>Denmark - IPv6</td>
</tr>
<tr>
<td>ftp://ftp.fi.freebsd.org</td>
<td>Finland</td>
</tr>
<tr>
<td>ftp://ftp1.de.freebsd.org</td>
<td>Germany #1 - IPv6</td>
</tr>
<tr>
<td>ftp://ftp2.de.freebsd.org</td>
<td>Germany #2 - IPv6</td>
</tr>
</tbody>
</table>

[OK] [Other] [Cancel]
图11. FreeBSD 10.x 或更版本的磁碟分割选项

导引式（Guided）磁碟分割会自动设定磁碟的分割区（Partition），手自动（Manual）磁碟分割可让进阶的使用者使用选单项目建立自订的分割区，而Shell会开启Shell提示让进阶的使用者可以使用指示列工具如gpart(8), fdisk(8) 及bsdlabel(8)来建立自订的分割区。

ZFS磁碟分割只在FreeBSD 10及之后的版本可以使用, 可建立选择性加密的root-on-ZFS系统并支援开机环境（Boot environment）。

本节会介绍在配置磁碟分割时需要考量那些事情, 并且会示範各种磁碟分割的方式。

2.6.1. 规划分割区配置
配置档系统时要记得硬碟的资料传输的速度外轨较内轨快, 因此较小且大量存取的档案系统应要较接近磁碟的外轨, 而较大的分割区如/usr 应放置在磁碟较内部, 建议建立分割区的顺序如下:

/ , swap , /var 然后 /usr。

机器预期的用途会反映到/var分割区的大小, 这个分割区用来保存邮件（Mailbox）, 日志档（Log file）及印表机缓冲（Spool）。依照使用者数及保存的期间, 邮件及日志档可能成长为无法预期的大小, 一般来说大部分的使用很少会在/var需要超过1 GB的可用磁碟空间。

有时在/var/tmp会需要较多的空间, 当新软体安装, 套件工具会从套件中取出暂时的副本位置於/var/tmp。若在/var/tmp没有足够的空间, 要安装大型套件如Firefox, Apache OpenOffice或LibreOffice会很困难。

/usr 分割区保存了许多支援系统运作的档案, 包含FreeBSD Port套件集以及系统原始码, 这个分割区建议至少要有2 GB的空间。

在规划分割区大小时, 请牢记空间需求, 当因某个分割区空间不足时要改使用其他分割区时会很麻烦。
根據經驗，交換分割區應該為實體記憶體（RAM）的兩倍。使用最低需求的RAM來運作的系統會需要更多的交換空間來取得更好的表現。配置太小的交換空間可能會導致VM分頁掃描碼效率不佳，且往後增加更多的記憶體時可能會產生問題。

在有多個SCSI磁碟或IDE磁碟在不同控制器的大型系統建議在每個磁碟機上都設定交換空間，最多可至四個磁碟機。每個交換分割區的大小應接近相同。核心雖可以處以任意大小的交換空間，但內部資料結構擴充到4倍的最大交換分割區大小時，讓交換分割區擁有相同的大小可以讓核心可以最佳的方式串連各個磁碟的交換空間。規劃較大的交換空間是可以的，即使沒有使用到多少交換空間，這也會讓要從失控的程式恢復運作更容易，而不需強制重新啟動系統。

正確的磁碟分割，可以區隔頻繁寫入所產生的資料碎片與經常讀取的分割區，將寫入頻繁的分割區放在磁碟的邊緣可以增加I/O效率。雖然較大的分割區可能也需要增加I/O效率，但將這些分割區往磁碟邊緣移動所增加的效率並不會比將/var移動到磁碟邊緣所增加的效率來的顯著。

2.6.2. 引導式磁碟分割
當選擇這個方法時，選單上會顯示可用的磁碟，若電腦有安裝多個磁碟，則需選擇其中一個來安裝FreeBSD。選擇[Entire Disk]會自動建立通用的分割區配置來填滿整個磁碟。選擇[Partition]則會使用磁碟上未使用的空間來建立分割區配置。
Would you like to use this entire disk (ada0) for or partition it to share it with other operating systems? Using the entire disk will erase any data currently stored there.

- [Entire Disk]
- [Partition]

When confirmed that the disk partition is correct, select [Finish] to continue installing.
Please review the disk setup. When complete, press the Finish button.

<table>
<thead>
<tr>
<th>Device</th>
<th>Size</th>
<th>Type</th>
<th>Mount Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>ada0</td>
<td>50 GB</td>
<td>GPT</td>
<td></td>
</tr>
<tr>
<td>ada0p1</td>
<td>260 MB</td>
<td>efi</td>
<td>/boot/efi</td>
</tr>
<tr>
<td>ada0p2</td>
<td>57 GB</td>
<td>freebsd-ufs</td>
<td>/</td>
</tr>
<tr>
<td>ada0p3</td>
<td>3.0 GB</td>
<td>freebsd-swap</td>
<td>none</td>
</tr>
<tr>
<td>ada1</td>
<td>60 GB</td>
<td>(none)</td>
<td></td>
</tr>
</tbody>
</table>
Create partitions for. F1 for help.
No changes will be made until you select Finish.

 ada0 60 GB (none)
 ada1 60 GB
图16.
手動建立分割區
amd64電腦最適合的選擇通常是GPT，無法相容GPT的舊電腦則應使用MBR。而其他分割表格式一般會用在那些較罕見或較舊的電腦上。

表1. 磁碟分割表格式

<table>
<thead>
<tr>
<th>縮寫</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>APM</td>
<td>Apple Partition Map，用於PowerPC™。</td>
</tr>
<tr>
<td>BSD</td>
<td>無MBR的BSD標籤，因為非BSD的磁碟工具可能無法辨識該標籤，有時被稱做危險專用模式（Dangerously dedicated mode）。</td>
</tr>
<tr>
<td>GPT</td>
<td>GUID分割區表（<a href="http://en.wikipedia.org/wiki/GUID_Particint_Table%EF%BC%89%E3%80%82">http://en.wikipedia.org/wiki/GUID_Particint_Table）。</a></td>
</tr>
<tr>
<td>MBR</td>
<td>主開機記錄（<a href="http://en.wikipedia.org/wiki/Master_boot_record%EF%BC%89%E3%80%82">http://en.wikipedia.org/wiki/Master_boot_record）。</a></td>
</tr>
<tr>
<td>PC98</td>
<td>使用MBR改編，用於NEC PC-98電腦（<a href="http://en.wikipedia.org/wiki/Pc9801%EF%BC%89%E3%80%82">http://en.wikipedia.org/wiki/Pc9801）。</a></td>
</tr>
<tr>
<td>VTOC8</td>
<td>Volume Table Of Contents，用於Sun SPARC64及UltraSPARC電腦。</td>
</tr>
</tbody>
</table>

選擇完分割區表格式並建立之後，再選擇[Create]一次來建立分割區。Tab鍵可用來在欄位間移動游標。
FreeBSD GPT installation will use at least three types of partitions:

- **freebsd-boot**: stores FreeBSD boot code.
- **freebsd-ufs**: FreeBSD's UFS file system.
- **freebsd-swap**: FreeBSD swap space.

Another noteworthy partition type is **freebsd-zfs**, which is used to place FreeBSD ZFS file systems (ZFS). Please refer to the `gpart(8)` manual for available GPT partition types.

File systems can be divided into multiple partitions, and some people prefer to use the traditional method to separate `/`, `/var`, `/tmp`, and `/usr` on different partitions. Please refer to the example of creating traditional partitioned file systems.

In the Size column, you can use common abbreviations to input: `K` for KB, `M` for MB, `G` for GB.

An appropriate sector (Sector) alignment will provide the best performance, and having partition sizes as 4 KB even multiples can help ensure alignment on the 512-byte or 4K-byte sectors. Generally, using partition sizes of 1M or 1G as even multiples can be the simplest way to ensure each partition starts on a 4K even multiple. The only exception is: The **freebsd-boot** partition, due to the limitation of the current boot code, cannot be larger than 512K.

If a partition contains a file system, it will require a mount point (Mountpoint), so if you only need to create a UFS partition, then the mount point should be set to `/`.

Label is the partition name, and since the disk name or number may vary due to different controllers or ports, but the partition label will not change. Therefore, in files like `/etc/fstab`, using labels instead of disk names or partition numbers can make the system more resilient to changes.
標籤會於磁碟連結之後出現在/dev/gpt/。其他分割表格式的標籤格有不同功能，且標籤會在/dev中有各自的目錄。

每個分割區請使用獨一無二的標籤來避免相同名稱的衝突，標籤可以加入與電腦名稱、用途、地點有關的文字。例如，使用labroot或rootfslab來做為電腦名稱為lab的UFS根目錄分割區。

例1. 建立傳統分割的檔案系統分割區

傳統的分割區配置會將/ , /var, /tmp 以及/usr 分別使用不同的檔案系統與分割區。先建立GPT分割表格式，然後依照下表所示建立分割區。下表是針對20G目標磁碟的分割區大小，若在目標磁碟有更多可用的空間，則可增加交換空間(Swap)或/var 會比較有用。以下所示的標籤皆以ex為字首，代表"example"，讀者應照前面的說明使用其他獨一無二的標籤。

<table>
<thead>
<tr>
<th>分割區類型</th>
<th>大小</th>
<th>挂載點</th>
<th>標籤</th>
</tr>
</thead>
<tbody>
<tr>
<td>freebsd-boot</td>
<td>512K</td>
<td></td>
<td>freebsd-boot</td>
</tr>
<tr>
<td>freebsd-swap</td>
<td>4G</td>
<td></td>
<td>exswap</td>
</tr>
<tr>
<td>freebsd-ufs</td>
<td>2G</td>
<td>/var</td>
<td>exvarfs</td>
</tr>
<tr>
<td>freebsd-ufs</td>
<td>1G</td>
<td>/tmp</td>
<td>extmpfs</td>
</tr>
</tbody>
</table>

預設FreeBSD的gptboot會預期第一個UFS分割區為/ 分割區。

2.6.4. Root-on-ZFS

自動磁碟分割在FreeBSD 10.0-RELEASE之後支援了自動建立root-on-ZFS的安裝程序。這種磁碟分割模式只能使用整個磁碟，並會清除整個磁碟內的內容。安裝程式會自動建立對齊4k邊界的分割區然後強制ZFS使用4k扇區(Sector)。即使在512位元扇區的磁碟使用也很安全，並增加了確保在512位元的磁碟上建立儲存池(Pool)也可在未來加入4k扇區磁碟的好處，無論是作為額外的存儲空間或作為故障磁碟的替代品。安裝程式也可選擇性採用GELI磁碟加密，如使用geli做磁碟加密所介紹，若開啟磁碟加密，會建立一個內含/boot目錄的2 GB未加密的開機儲存池，這個儲存池中會儲存核心及其他開機必要的檔案。然後剩餘的空間會給ZFS儲存池使用。主要ZFS設定選單提供了數個設定選項來控制儲存池的建立。
选择T来设定储存池类型(Pool Type)以及要组成储存池的磁碟。自动ZFS安装程式目前仅支援建立单一顶层vdev，除了在串连(Stripe)模式。要建立更复杂的储存池，需使用Shell磁碟分割的操作来建立储存池。安装程式支援建立各种储存池类型，包含串连Stripe(不建议，没有备援功能)、镜像Mirror(效能较好，但可用空间较少)以及RAID-Z 1, 2,与3(分别有能力承受同时1,2与3个磁碟的损坏)。在选择储存池类型时会在萤幕的下方提示所需的磁碟数量，以及在使用RAID-Z时，每种配置最佳的磁碟数。
Select Virtual Device type:

- **stripe**: Stripe - No Redundancy
- **mirror**: Mirror - n-Way Mirroring
- **raid1**: RAID 1+0 - n x 2-Way Mirrors
- **raid2**: RAID-21 - Single Redundant RAID
- **raid3**: RAID-22 - Double Redundant RAID
- **raidz3**: RAID-23 - Triple Redundant RAID

[Press arrows, TAB or ENTER]
ZFS Configuration

raidzi: Not enough disks selected. (2 < 3 minimum)

CHANGE SELECTION  CANCEL

[Press arrows, TAB or ENTER]
图22. 分析磁碟主ZFS設定選單也允許使用者輸入儲存池名稱、關閉強制4k扇區對齊、開啟或關閉加密、切換GPT（建議）與MBR分割表類型以及選換交換空間容量。設定所有選項為想要的值之後，請選擇選單上方的安裝（Install）選項。

若開啟了GELI磁碟加密，安裝程式會提示輸入兩次用來加密磁碟的密碼。

gpart(8) show ada0:
40 125829040 ada0 GPT (60G)
  40 532480 1 efi (260M)
  532520 1024 2 freebsd-boot (512K)
  533544 984 - free - (492K)
  534528 4194304 3 freebsd-swap (2.0G)
  4728832 121098240 4 freebsd-zfs (58G)
  125627072 2008 - free - (1.0M)

camcontrol(8) inquiry ada0:

pass0: <VBOX HARDISK 1.0> ATA-6 device
pass0: 33.300MB/s transfers (UDMA2, PIO 65536bytes)

protocol ATA-6
device model VBOX HARDISK
firmware revision 1.0
serial number V00956971f-c387796c
additional product id
cylinders 16383

Enter a strong passphrase, used to protect your encryption keys. You will be required to enter this passphrase each time the system is booted.

[Enter passphrase]

[Use alpha-numeric, punctuation, TAB or ENTER]
最後修
改
然後安裝程序會
正
常
繼
續。

2.6.5. Shell 模式 磁碟分割
當要
做
進
階
的安裝時，bsdinstall
的
磁碟
分
割
選
單
可能
無
法
提
供
需
要
的
彈
性
。進
階
的
用
者
可
在
磁碟
分
割
選
單
選
擇
Shell
選項來
手
動
分
割
磁碟機
、
建
立
檔
案
系統
、
填
寫
/tmp/bsdinstall_etc/fstab
以
及
掛
載
檔
案
系統
到
/mnt
下。這些
動
作
完
成
之
後，
輸
入
exit
可
返
回
bsdinstall
繼
續
安
裝
程
序。

2.7.
確
認
安
裝
磁
碟
設
定
完
之
後，接
下
來
的
選
單
會
讓
您
在
格
式
化
所
選
的
硬
碟
之
前
有
最
後
次
機
會
做
變
更
，若
需
要
做
變
更
，可
選
Back
返
回
到
主
磁
碟
分
割
選
單
。
[Revert & Exit]
則
會
離
開
安
裝
程
式，不
會
對
硬
碟
做
任
何
變
更
。
To begin the actual installation, please select [Commit] and then press Enter.

Installation time will vary depending on the selected version, medium, and computer speed, and will provide information on the current progress.

First, the installation program will partition the selected disk, then initialize the partitions. If using boot-only media, it will begin downloading the selected files.
接着，会检验发行档的完整性来确保没有因下载过程中或安装媒体的读取过程中读取错误造成的损坏：
### Checksum Verification

- **base.txz**: Passed
- **kernel.txz**: Passed
- **lib32.txz**: Passed
- **ports.txz**: Passed
- **src.txz**: In Progress
- **tests.txz**: Pending

Verifying checksums of selected distributions.

**Overall Progress**: 64%
图28. 解开发行版档

所有选择的发行版档取出后，bsdinstall会显示第一次安装后设定画面，可用的安装后设定选项会在下一节说明。

2.8. 安装后注意

FreeBSD安装完之后，bsdinstall会在开机进入新安装的系统之前提示设定数个选项，本节将介绍这些设定选项。

2.8.1. 設定root密码

首先，必需设定root的密码，输入密码时，并不会直接在画面上显示输入的字元。输入完密码之后，必须再输入一次来确认没有输入错误。
FreeBSD Installer

Please select a password for the system management account (root):
Typed characters will not be visible.
Changing local password for root
New Password:
Retype New Password:
请选择一个网络接口来配置：

`en0 Intel(R) Legacy PRO/1000 MT 82540EM`

[ OK ]  [Cancel]
图31. 扫描无线网路存储点

网络网路会使用Service Set Identifier (SSID)来辨识，SSID是一段简短、独一无二的名称，用来命名每个网路。扫描时找到的SSID会列到清单，并会说明该网路可用的加密类型。若想连线的SSID并未出现在清单上，可选择[Rescan]再扫描一次，若想连线的网路仍然没有出现，请检查天线的连线是否有问题，或者尝试将电脑移动至更靠近存储点的位置，然后再次扫描。
然後，輸入加密資訊來連線到選擇的無線網路。

強烈建議使用WPA2加密，因為較舊的加密類型，如WEP僅提供微弱的安全性。若網路使用WPA2則需輸入密碼，也稱作Pre-Shared Key (PSK)。考量安全性，輸入到輸入框的字元會以星號顯示。
以下のウィンドウでSSIDを入力し、パスワードを設定してください。
图34.

IPv4 网路有两

种方式可以设

定 IPv4。

DHCP 会自

动设定网路

界面卡且该

网路上需有

DHCP 伺服

器才可使

用。否

则，必须

手动输入位

址的资讯

来做静态

设定。

请不要

随便输入

网路资

讯，因为

这不管用。如果

没有可用的

DHCP 伺服

器，可向

网路管理

者或网路

服务供应

商（Internet Service Provider, ISP）索取列

需要的网路

资讯的资

讯。

若有可用的

DHCP 伺服

器，请在接下

来的选

单中选

择 [Yes] 则会自

动设定网路

界面卡。当

找到 DHCP

服务器

并且取得系

统的位

址资讯时，安装程式会

出现一分

钟左右

的停

顿。
若沒有可用的DHCP伺服器，則選擇[No]然後在這個選單中輸入以下位址資訊:

```
Network Configuration
Would you like to use DHCP to configure this interface?
[ Yes ]  [ No ]
```
IPv4 靜態位置設定

- **IP 位址** (IP Address) - 要分配給這台電腦的IPv4位址。位址必須獨一無二且不可已被其他在區域網路上的設備使用。

- **子網路遮罩** (Subnet Mask) - 網路的子網路遮罩。

- **預設路由器** (Default Router) - IP位址所在網路段的預設通訊閘。

接下來的畫面會詢問是否要設定介面卡的IPv6位址，若可以且想使用IPv6，請選擇 `[Yes]`。

<table>
<thead>
<tr>
<th>IP Address</th>
<th>192.168.1.135</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnet Mask</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>Default Router</td>
<td>192.168.1.1</td>
</tr>
</tbody>
</table>
圖37. 選擇IPv6網路也有兩種方式可以設定IPv6。

StateLess Address AutoConfiguration (SLAAC)會自動向區域路由器請求取得正確的設定資訊，請參考http://tools.ietf.org/html/rfc4862取得進一步資訊。

靜態設定則需要手動輸入網路資訊。

若有可用的IPv6路由器，請在接下來的選單選擇[Yes]自動設定網路介面卡。當找到路由器並且取得系統的位址資訊時，安裝程式會出現一分鐘左右的停頓。
Would you like to try stateless address autoconfiguration (SLAAC)?

[ Yes ] [ No ]
IPv6 靜態位置設定

- IPv6 位址 (IPv6 Address) - 要分配給這台電腦的 IPv6 位址。位址必須獨一無二且不可已被其他在區域網路上的設備使用。

- 預設路由器 (Default Router) - IPv6 位址所在網段的預設通訊閘。

最後的網路設定選單是用來設定網域名稱系統 (Domain Name System, DNS) 的解析器，解析器會轉換主機名稱為網路位址。若已使用 DHCP 或 SLAAC 來自動設定網路介面卡，解析器設定 (Resolver Configuration) 的值可能會事先已填入，否則需輸入區域網路的網域名稱到搜尋 (Search) 欄位。

DNS #1 與 DNS #2 要填寫 DNS 伺服器的 IPv4 及 IPv6 位址，至少需填寫一個 DNS 伺服器。
图40. DNS 設定

2.8.3. 設定

接下來的選單會詢問系統時鐘要使用 UTC 或者當地時間。若有疑問時可選擇 [No] 使用更常用的當地時間。

图41. 選擇本地或 UTC 時鐘

---

Is this machine's CMOS clock set to UTC? If it is set to local time, or you don't know, please choose No here!

Yes [N]
接下来，一系列的选单会通过选择地理区域、城市及时区来判断正确的当地时间。设定时区可让系统自动更正区域时间的更改，如日光节约时间及正确执行其他时区相关功能。

图42. 选择区域使用方向键选择适当区域然后按下Enter。
Select a country or region

31 Moldova (Republic of)
32 Monaco
33 Montenegro
34 Netherlands
35 Norway
36 Poland
37 Portugal
38 Romania
39 Russian Federation
40 San Marino
41 Serbia
42 Slovakia
43 Slovenia
44 Spain
45 Sweden
46 Switzerland

[ OK ]  [ Cancel ]
Select a zone which observes the same time as your locality.

1. Spain (mainland)
2. Ceuta, Melilla
3. Canary Islands

[OK] [Cancel]
2.8.4. 启用服务
接下来的菜单用来设定哪些系统服务要在系统启动时执行。所有的服务为选中，只需开启系统运作真正需要的服务。

确认时区的缩写是否正确，若正确，按下 Enter 继续安装后设定。

Confirmation
Does the abbreviation 'CET' look reasonable?

[ Yes ]  [ No ]
<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sshd</td>
<td>Secure Shell daemon. Can be started from a remote server through an encrypted connection. Only enabled when a remote login is permitted.</td>
</tr>
<tr>
<td>moused</td>
<td>PS/2 mouse pointer on console.</td>
</tr>
<tr>
<td>ntpd</td>
<td>Synchronize system and network time.</td>
</tr>
<tr>
<td>ntpd_sync_on_start</td>
<td>Sync time on ntpd startup, even if offset is high.</td>
</tr>
<tr>
<td>powerd</td>
<td>Adjust CPU frequency dynamically if supported.</td>
</tr>
<tr>
<td>dumpdev</td>
<td>Enable kernel crash dumps to /var/crash.</td>
</tr>
</tbody>
</table>

2.8.5. Enable system crash dumps

Choose the services you would like to be started at boot:

- System Configuration
- Choose the services you would like to be started at boot:
  - sshd: Secure Shell daemon
  - moused: PS/2 mouse pointer on console
  - ntpd: Synchronize system and network time
  - ntpd_sync_on_start: Sync time on ntpd startup, even if offset is high
  - powerd: Adjust CPU frequency dynamically if supported
  - dumpdev: Enable kernel crash dumps to /var/crash

[OK]
Would you like to enable crash dumps? If you start having problems with the system it can help the FreeBSD developers debug the problem. But the crash dumps can take up a lot of disk space in /var.

< Yes >  < No >
图48．新增使用者帳號

请依照提示输入需要的使用者帳號资讯，输入使用者资讯的範例如 sample使用者帳號。
FreeBSD Installer
====================
Add Users

Username: imani
Full name: imani
Uid (Leave empty for default): 
Login group [iman]: 
Login group is imani. Invite imani into other groups? []: wheel
Login class [default]: 
Shell (sh csh ksh nologin) [sh]:
Home directory [/home/iman]: 
Home directory permissions (Leave empty for default): 
Use password-based authentication? [yes]: 
Use an empty password? (yes/no) [no]: 
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:


FreeBSD Installer
=================================
Add Users

Username: imani
Full name: imani
Uid (Leave empty for default):
Login group [imani]:
Login group is imani. Invite imani into other groups? [ ]: wheel
Login class [default]:
Shell (sh csh tsh nologin) [sh]:
Home directory [/home/imani]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username : imani
Password : ******
Full Name : imani
Uid : 1001
Class :
Groups : imani wheel
Home : /home/imani
Home Mode :
Shell : /bin/sh
Locked : no
Ok? (yes/no) [yes]:
Adduser: INFO: Successfully added imani to the user database.
Add another user? (yes/no) [no]:  

---

2.8.7. Finally, after the installation and configuration are complete, a final opportunity to make changes will be provided.
Setup of your FreeBSD system is nearly complete. You can now modify your configuration choices. After this screen, you will have an opportunity to make more complex changes using a shell.

- **Add User** - Add a user to the system
- **Root Password** - Change root password
- **Hostname** - Set system hostname
- **Network** - Networking configuration
- **Services** - Set daemons to run on startup
- **System Hardening** - Set security options
- **Time Zone** - Set system timezone
- **Handbook** - Install FreeBSD Handbook (requires network)

Complete the final setup by selecting [Exit].
The installation is now finished. Before exiting the installer, would you like to open a shell in the new system to make any final manual modifications?

[< Yes >   < No >]
完成安裝
若有需要做進一步或特殊的設定，選擇[Live CD]會開機進入安裝媒體的Live CD模式。若安裝已完成，選擇[Reboot]重新開機後啟動新的FreeBSD電腦。不要忘了移除FreeBSD安裝媒體，否則電腦會再次開機進入安裝程式。

FreeBSD開機的過程會顯示許多可以參考的訊息，系統開機完成後，會顯示登入提示，在login:提示時，輸入安裝時新增的使用者名稱。登入時避免直接使用root，請參考超級使用者帳號來取得當需要管理權時如何成為超級使用者的說明。

要查看開機過程顯示的訊息可按Scroll-Lock鍵來開啟卷軸暫存，然後可使用PgUp, PgDn以及方向鍵來捲動訊息。查看到完成之後再按Scroll-Lock鍵一次來解除畫面鎖定並返回Console。系統開機一段時間之後要查看這些訊息可在指令提示後輸入less/var/run/dmesg.boot，查看後按下q鍵便可回到指令列。

若在選擇要啟動的其他服務有啟動sshd，因系統會產生RSA及DSA金鑰第一次開機可能會有點慢，之後的開機便會恢復正常速度。接著會顯示金鑰的指紋(Fingerprint)，如這個範例:

Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
The key's randomart image is: +--[RSA1 1024]----+

76
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
The key's randomart image is:
+--[ DSA 1024]----+
|       ..          |
|      o oo         |
|     +  S          |
|    + + o          |
|o + ..+ .         |
|==o..o+E         |
|-----------------+

Starting sshd.
部份安裝問題可以透過更各種硬體元件的韌體來避免或緩解，特別是主機板。主機板的韌體通常稱為BIOS, 大部份主機板與電腦製造商會有網站可以取得升級程式與升級資訊。製造商通常會建議若沒有特殊原因盡量避免升級主機板BIOS，例如：重大更新，升級的程式可能會出錯，導致未更新完成的BIOS並讓電腦無法運作。

若系統在開機偵測硬體時卡住或安裝時運作異常，可能主因為ACPI, FreeBSD在i386, amd64及ia64平台廣泛的使用了系統ACPI服務來協助設定系統組態，若在開機時有偵測到該功能。不幸的是, ACPI驅動程式與系統主機板及BIOS韌體之間仍存在部份問題。可於開機載入程式的第三階段設定hint.acpi.0.disabled

set hint.acpi.0.disabled = "1"

每一次系統重開之後便會重設，因此需要在/boot/loader.conf檔案加入hint.acpi.0.disabled="1"。更多有關開機載入程式的資訊可於概述取得。

2.10. 使用Live CD

如歡迎選單所示bsdinstall的歡迎選單提供了[Live CD]選項，這對那些對FreeBSD是否為正確的作業系統尚存疑慮的人非常有幫助，這可讓這些人在安裝前測試一部份功能。

在使用[Live CD]之前必須注意以下幾點事項:

• 若要增加存取權限，必須透過認證。使用者名稱為root而密碼則是空白。
• 系統是直接從安裝媒介上執行，比起安裝到硬碟的系統，效能可能較差。
• 這個選項只提供指令提示，不會有圖型化介面。
Chapter 3. FreeBSD

3.1. 概述

接下来的这一章将涵盖 FreeBSD 作业系统的基本指令及功能。大部分的内容在 UNIX™-like 作业系统中都是相通的。如果您对这些内容熟悉的话，可以放心的跳过。如果您刚接触 FreeBSD，那您一定要仔细的读完这一章。

读完这一章，您将了解：
• 如何使用 FreeBSD 的虚拟 Console。
• 如何在 FreeBSD 建立与管理使用者与群组。
• UNIX™ 档案权限以及 FreeBSD 档案标记的运作方式。
• 预设的 FreeBSD 档案系统配置。
• FreeBSD 的磁碟组织。
• 如何挂载（Mount）、卸载（Umount）档案系统。
• 什么是程序、Daemon 以及信号（Signal）。
• 什么是 Shell，以及如何变更您预设的登入环境。
• 如何使用基本的文字编辑器。
• 什么是装置（Device）和装置节点（Device node）。
• 如何阅读操作手册以获得更多的资讯。

3.2. 虚拟 Console 与终端机

如果您没有将 FreeBSD 装置成开机时自动进入图形化模式，系统会进入指令登入提示像是这样的东西：

FreeBSD/amd64 (pc3.example.org) (ttyv0)

login:  

第一行包含了刚开机完系统的信息，amd64 代表此范例所使用的系统是执行 64-位元版本的 FreeBSD，这台主机的名称是 pc3.example.org，ttyv0 代表这是一个"系统 Console"。第二行则是登入人的提示讯息。

FreeBSD 是一个多使用者的系统，需要一套可以分辨不同使用者的方法。因此所有的使用者在执行程式之前必须"登入"系统以取得系统内程式的存取权限。每个使用者都有一组独一无二的使用者名称("username")及个人密码("password")。

要登入系统 Console 需要输入在系统安装时设定的使用者名称，请参考新增使用者，并按下 Enter。

接著输入該使用者名称的密码按一下 Enter。输入的密码为了安全起见不会显示在画面上。

如果您输入了正确的密码，您应该会看到今日讯息（Message of the day, MOTD），后面接著显示指令提示字元，依使用者建立时所选择的 Shell 会有不同的提示字元可能为 #, $ 或 %。看到指令提示代表使用者现在已经登入 FreeBSD 系统 Console 且已经准备好可以下指令。

3.2.1. 虚拟 Console

虽然系统 Console 已经可以用来与系统互动，但使用键盘来下指令使用 FreeBSD 系统的使用者通常会使用虚拟 Console 登入。因为系统讯息预设会显示在系统 Console，这些讯息会在使用者作業的过程中不断出现，让使用者难以专心作業。
 FreeBSD 預設提供多個虛擬 Console 可以輸入指令，每個虛擬 Console 都有自己的登入提示及 Shell，並且可以輕易地在虛擬 Console 間切換。這實際上讓指令輸入有了類似於圖型環境中可以同時開啟多個視窗的功能。

組合鍵 Alt+F1 至 Alt+F8 被 FreeBSD 保留用來切換虛擬 Console，使用 Alt+F1 可切換至系統 Console (ttyv0)，Alt+F2 可存取第一個虛擬 Console (ttyv1)，Alt+F3 可存取第二個虛擬 Console (ttyv2)，以此類推。當使用 Xorg 作為圖型 Console 時，組合鍵則改使用 Ctrl+Alt+F1 來切換回文字介面的虛擬 Console。

當您從一個 Console 切換到下一個的時候，FreeBSD 會切換顯示的內容，這就好像有很多虛擬的螢幕和鍵盤可以讓您輸入指令到 FreeBSD 執行。在某一個虛擬 Console 上執行的程式並不會因為使用者切到別的 Console 而停止執行。

請參考 kbdcontrol(1)，vidcontrol(1)，atkbd:(4)，syscons(4) 及 vt(4) 以取得更多有關 FreeBSD Console 及鍵盤驅動程式的技術說明。

FreeBSD 中虛擬 Console 的數量設定在 /etc/ttys 檔案中的下列章節:

<table>
<thead>
<tr>
<th>name</th>
<th>getty</th>
<th>type</th>
<th>status</th>
<th>comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ttyv0</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv1</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv2</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv3</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv4</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv5</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv6</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv7</td>
<td>/usr/libexec/getty Pc</td>
<td>xterm</td>
<td>on</td>
<td>secure</td>
</tr>
<tr>
<td>ttyv8</td>
<td>/usr/X11R6/bin/xdm -nodaemon</td>
<td>xterm</td>
<td>off</td>
<td>secure</td>
</tr>
</tbody>
</table>

要關閉虛擬 Console 只要在指定的虛擬 Console 該行設定的一開始加上註解號（#）。例如要將虛擬 Console 的數量由 8 個改為 4 個，則可將加在代表虛擬 Console 的 ttyv5 到 ttyv8 的最後四行一開始。

請勿將系統 Console ttyv0 加上註解號。注意，若有依照 X Window 系統安裝並設定 Xorg 時，會用到最後一個虛擬 Console (ttyv8)。

有關各欄位的設定及其他選項，請參閱 ttys(5) 說明。

3.2.2. 單使用者模式

FreeBSD 開機選單會提供一個選項為 “Boot Single User”，若選擇該項目，系統將會進入所謂 “單使用者模式” 的特殊模式。此模式通常用在修復系統無法開機或重設已忘掉的 root 密碼。在當使用者模式中無法使用網路及其他虛擬 Console，但有完整 root 實際系統的存取權限，而且預設是不須要輸入 root 密碼。也因此，要能透過實體鍵盤來操作才能進入此模式，在考量 FreeBSD 系統安全時須要限制可操作實體鍵盤的人員。
預設狀態為安全（secure），這代表誰能夠操作實體鍵盤不是不重要就是已受到實體安全規範管制。

若設定更改為不安全（insecure）則代表主機所在的環境不安全，因為任何人皆可接觸鍵盤。

當此行設定更改為不安全（insecure）時，當使用選擇單使用者模式時，FreeBSD 將會要求輸入 root 密碼。

請審慎考慮是否要更改為 insecure！因為萬一忘記 root 密碼的話，雖然還是有其他辦法可以登入單使用者模式，只是對不熟 FreeBSD 開機程序的人可就麻煩了。

3.2.3. 更改 Console 影像模式
FreeBSD Console 預設顯示大小可以調整為 1024x768、1280x1024 或其他顯示卡與螢幕有支援的解析度大小。要使用不同的影像模式需載入 VESA 模組:

```bash
# kldload vesa
```

要偵測硬體支援的影像模式，可使用 `vidcontrol(1)`。要取得支援的影像模式清單可輸入以下指令:

```bash
# vidcontrol -i mode
```

該指令會顯示硬體所支援的影像模式清單，要採用新的影像模式需以 root 使用者執行 `vidcontrol(1)` 指令:

```bash
# vidcontrol MODE_279
```

若可接受新的影像模式，可以在 `/etc/rc.conf` 加入設定，讓每次重開機後會自動生效:

```bash
allscreens_flags="MODE_279"
```

3.3. 使用者與基礎帳號管理
FreeBSD 允許多使用者同時使用電腦，在一次只能有一位使用者坐在我們前面使用鍵盤操作的同時，可讓任何數量的使用者透過網路登入到系統。每一位要使用該系統的使用者應有自己的帳號。

本章介紹:
• FreeBSD 系統中各種類型的使用者帳號。
• 如何加入、移除與修改使用者帳號。
• 如何設定用來控制使用者與群組允許存取的資源的限制。
• 如何建立群組與加入使用者作爲群組成員。
使用者的名
時到帳號期間
時碼更改密碼
Login class
登入類別
組群ID (GID)
ID (UID)
密碼
使用者名稱
都會有一些相關資訊:
系統的使用者每個在FreeBSD言及語境
使用者可以編輯環境來配己設器、合鍵使用系統的習Shell設定己預透合（Key自組慣、管理者以覆設定讓避免蓋其他使用者的,這可做什麼辨號在及使用者識
給配擁一的使用者
使用者
3.3.1.2.
nobody
nobody
帳號例系統子有
不會受到限作的將運務那使用者行,以安這些服及DNS執考全性,若所有的服超級為那麼來作是因,要這量務均、郵件,系統如用帳號例來務執行服系統
3.3.1.1.
帳號有三種主要帳號以的及類超級帳號:、使用者使用者系統帳號帳號型
超級帳號:
、使用者使用者
系統
帳號
型
在。
的長字個的長字個可以姓
密碼
過並不一但了使用者的寫字元訊、大,
定名有
類並可註真實。反映白似,這個資
無二的名稱
FreeBSD使用者辦一來獨用
到系統,管該日儘期入的目錄及檔帳號期帳號登期日法案仍存後,指定帳號帳號
不會有學建的需要有學例的。當壽命限限如,帳號生帳號時,可使用期立FreeBSD預情況設下機制。且可上權的資降低一個以讓著的顯小員的大些來控制使用者成
群組數
組員是一
群組數
組是一
群組識使用者所
一
題。數字相容
會先的將為65535UID使用
名稱的
二的來獨無用
屬是一個組
二的來獨無用
FreeBSD用識是一
字ID (User ID, UID)
組數
使用者一
使用
名稱的
二的來獨無用
FreeBSD用識是一
字ID (User ID, UID)

家目錄（Home directory）

家目錄是系統中某個目錄的完整路徑，這個目錄是使用者登入後的起點目錄。慣例上會將所有使用者目錄放置在 `/home/username` 或 `/usr/home/username`。每位使用者可以儲存他們的個人檔案及子目錄於他們自己的家目錄。

使用者 Shell (User shell)

Shell 提供了使用者預設的環境來與系統互動。有數種不同類型的 Shell，有經驗的使用者會有自己偏好的選擇，可儲存在自己的帳號設定。

3.3.1.3. 超級使用者帳號

超級使用者帳號，通常稱作 root，用來管理系統，沒有權限的限制，也因為這個原因，該帳號不應該用來做每日例行工作，如：寄信與收信、系統的一般探索或程式設計。

超級使用者並不像其他使用者帳號，可以沒有限制的操作，不正確的使用超級使用者帳號可能會造成可觀的災害。一般使用者帳號不會因為失誤而摧毀作業系統，所以建議登入一般使用者帳號，只有在指令需要額外權限時切換為超級使用者。

使用超級使用者下指令時永遠要再三檢查，由於一個多餘的空白或缺少的字元可能意味著無法挽回的資料遺失。

有數種方法可以提升為超級使用者權限，雖然可以直接登入為 root，但強烈不建議這樣做。

改使用 `su(1)` 切換為超級使用者。執行此指令時若指定 `-` 參數，該使用者會繼承 root 的使用者環境。執行此指令的使用者必須在 `wheel` 群組中，否則指令會失敗。使用者也必須要知道 root 使用者的密碼。

在此例當中，該使用者只在要執行 `make install` 時切換為超級使用者，因為這個步驟需要超級使用者權限。

指令完成之後，該使用者輸入 `exit` 離開超級使用者帳號並返回他的使用者帳號權限。

例 2. 以超級使用者的身分安裝程式

```
% configure
% make
% su -
Password:
# make install
# exit
%
```

內建的 `su(1)` 框架在單人系統或只有一位系統管理者的小型網路上可以運作得很好。另一種方式是安裝 `security/sudo` 套件或 `Port`。此軟體提供了活動記錄且允許管理者設定那個使用者可以用超級使用者執行那個指令。

3.3.2. 管理帳號

FreeBSD 提供了各種不同指令來管理使用者帳號，最常用的指令已摘要於管理使用者帳號的工具，接著有一些用法的範例。請參考每個工具的操作手冊來取得更多的詳細資訊與用法範例。

表 2. 管理使用者帳號的工具
adduser(8) 建議用來新增新使用者的指令列應使用程式。

rmuser(8) 建議用來移除使用者的指令列應使用程式。

chpass(1) 用來更改使用者資料庫資訊的工具。

passwd(1) 用來更改使用者密碼的指令列工具。

pw(8) 用來修改使用者帳號各方面資訊強大且靈活的工具。

3.3.2.1. adduser 建議用來新增新使用者的程式為 adduser(8)。當新使用者新增之後，此程式會自動更新 /etc/passwd 及 /etc/group，這同時也會建立新使用者的家目錄（複製 /usr/shared/skel 中的預設設定檔），並且可以選擇是要寄送歡迎訊息通知新使用者。這個工具必須使用超級使用者執行。

adduser(8) 工具採用互動的方式，只需幾個步驟便可建立新使用者帳號。如在 FreeBSD 新增使用者所示，可輸入必填的資訊或按 Return 鍵採用方括中的預設值。在此例當中，使用者被邀請加入 wheel 群組，這讓使用者可使用 su(1) 變成超級使用者。完成之後，此工具會詢問是否要建立其他的使用者或離開。

例 3. 在 FreeBSD 新增使用者

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? [ ]: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [ ]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   :
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
```

adduser: INFO: Successfully added (jru) to the user database.

Add another user? (yes/no): no

Goodbye!

3.3.2.2. rmuser

To remove one user, a superuser can use the command `rmuser(8)`. This command performs the following steps:

1. Remove the user's `crontab(1)` entry, if any.
2. Remove any jobs belonging to this user.
3. Stop all processes owned by this user.
4. Remove the user's local password file.
5. Selectively remove the user's home directory, if the user has one.
6. Remove the user's mail spool file.
7. Remove all files owned by the user that are stored in `/var/mail`.
8. Remove all files owned by the user that are stored in `/tmp`.
9. Finally, remove the user from `/etc/group`.

The `rmuser(8)` command cannot be used to remove superuser accounts, as this would effectively be a form of blanket destruction.

By default, it will use an interactive mode, as shown in the example:

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
```

```
Removing user (jru): mailspool home passwd.
```

3.3.2.3. chpass

Any user can use `chpass(1)` to change their own settings and personal data associated with their account. Superusers can change another user's account data.

When no arguments are given, `chpass(1)` will start an editor to make changes to the user's information. Once the user leaves the editor, the changes will be saved to the user database.
離開編輯器時，此工具會提示使用者輸入密碼，除非使用者執行此工具。

在以超級使用者的身份使用 `chpass` 中，超級使用者輸入了 `chpass jru`，並正在檢視這個使用者可以更改的欄位。若改以 `jru` 執行這個指令，只會顯示最後六個欄位供編輯，如同以一般使用者的身份使用 `chpass` 所示。

例 5.

```
#Changing user database information for jru.
Login: jru
Password: *
```

```
Uid [ #]: 1001
Gid [ # or name]: 1001
Change [ month day year]:
Expire [ month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

例 6.

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

指令 `chfn(1)` 以及 `chsh(1)` 皆連結至 `chpass(1)`，就如同 `ypchpass(1)`, `ypchfn(1)` 以及 `ypchsh(1)` 的關係。自從 NIS 支援自動化以後，便不再需要特別加上 `yp`，如何設定 NIS 在網路伺服器中有說明。

3.3.2.4. `passwd`

任何使用者皆可簡單使用 `passwd(1)` 更改自己的密碼。要避免意外或未授權的變更，這個指令在設定新密碼之前會提示使用者輸入原來的密碼：

```
86
```
更改您的密码
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done

超级使用者可以更改任何使用者的密码通过执行 passwd(1)时指定使用者名称。当此工具以超级使用者执行时，将不会提示输入使用者目前的密码，这可在使用者忘记原来的密码时更改密码。

例
8.
以超级使用者的身份更改其他使用者的密码
# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done

如同 chpass(1), yppasswd(1) 連結到 passwd(1), 因此 NIS 在兩個指令上皆可運作。
例 9. 使用 `pw(8)` 新增群組:

```
# pw groupadd teamtwo
# pw groupshow teamtwo
```

在本例中，1100 是 `teamtwo` 的 GID。目前 `teamtwo` 沒有任何成員，這個指令會加入 `jru` 作為 `teamtwo` 的成員。

例 10. 使用 `pw(8)` 加入使用者帳號到新的群組:

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
```

give `-M` 的參數是以逗號分隔的使用者清單，用來加入成員到新的（空的）群組或取代既有群組中的成員。對使用者來說這裡的群組成員與使用者列於密碼檔的主要群組不同，這代表在 `pw(8)` 使用 `groupshow` 時不會顯示做為使用者主要群組的成員，但會顯示在使用 `id(1)` 或同類工具查詢的資訊當中。當使用 `pw(8)` 加入使用者到某個群組，該指令只會處理 `/etc/group` 且不會嘗試自 `/etc/passwd` 讀取其他的資料。

例 11. 使用 `pw(8)` 加入新成員到群組:

```
# pw groupmod teamtwo -m db
# pw groupshow teamtwo
```

在本例當中，給 `-m` 的參數是以逗號分隔的使用者清單，用來加入使用者到群組。不像前面的例子，這些使用者會加入到群組，而非取代既有群組中的使用者。

例 12. 使用 `id(1)` 查看所屬群組:

```
% id jru
uid = 1001 (jru)
gid = 1001 (jru)
groups = 1001 (jru), 1100 (teamtwo)
```

在本例中，`jru` 是群組 `jru` 以及 `teamtwo` 的成員。

要取得更多有關此指令的資訊及 `/etc/group` 的格式，請參考 `pw(8)` 以及 `group(5)`。

3.4. 權限

在 FreeBSD 中，每個檔案與目都有相關的數個權限，且有許多年工具可以檢視與修改這些權限。了解權限如何運作是必需的，這可確保使用者能夠存取它們所需的檔案以及無法不正確的存取供作業系統或其他使用者擁有的檔案。
<table>
<thead>
<tr>
<th>等级</th>
<th>权限</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td>444</td>
<td>r--</td>
<td>所有者可读、可写，其他人可读但不可写。</td>
</tr>
<tr>
<td>644</td>
<td>rw-</td>
<td>所有者可读可写，组内成员可读可写，其他人只可读。</td>
</tr>
</tbody>
</table>
| 744  | rwx  | 所有者、组内成员、其他人权限都为可读可写可执行。

示例：
```bash
ls -l
```

输出：
```
total 530
1 root  wheel    7680 Sep  5 12:31 email.txt
1 root  wheel     512 Sep  5 12:31 otherfile
1 root  wheel     512 Sep  5 12:31 myfile
```

这些权限设置可以通过`chmod`命令来修改。例如，要将文件`myfile`的权限设置为所有者可读可写可执行，组内成员可读可写可执行，其他人只可读，可以使用命令：
```bash
chmod 744 myfile
```
3.4.1. 權限符

號

權限符

號

可

稱

做

符

號

表

示

使用

字元

的方式

來

取代使用

數

值

來設定檔

案或目

錄

的

權限

符

號

表

示

的

格式

依

序

為

(某

人

)(動

作

)(權限

), 可使用的

符

號

如下:

項

目

字

母

代表意

義

(u)

使用者

(g)

群組所有者

(o)

其他

全

部

("world")

(動

作

)+

增加

權限

(動

作

)-

移

除權限

(動

作

)=

指定

權限

(r)

讀取

(w)

寫入

(x)

執

行

(t)

Sticky位

元

(s)

設定UID

或

GID

如先前同

樣

使用

chmod(1)

指令來設定,

但

使用的參

數為

這些

字元

。例如, 您可以使用下

列指令

禁

止

其他使用者

存

取

檔

案

FILE:

```
% chmod go= FILE
```

若有

兩

個以

上

的

權限

更改

可以使用

逗

號

(,)

區

隔

。例如, 下

列指令

將

會

移

除群

組及全

部人

("world")

對檔

案

FILE的

寫入

權限,

並使

全

部人

對

該

檔

有

執

行

權限:

```
% chmod go-w,a+x FILE
```

3.4.2. FreeBSD

檔

案

旗

標

除

了前面提到的

檔

案

權限

外,

FreeBSD

支援

使用

"檔

案

旗

標"

。這些

旗

標增加

了

檔

案

的安

全性及

管理

性,

但

不包

含

目

錄。

有了

檔

案

旗

標

可

确保

在

某

些時

候

root

不會意外

将

檔

案修

改

或

移

除

。

修

改

的

檔

案

flag

僅

需要使用

拥

有

简易

的介面的

chflags(1)

工具。

例如,

標示

系统

禁止删

除

的

旗

標於檔

案

file1, 使用下

列指令:

```
# chflags sunlink file1
```

若要

移

除

系统

禁止删

除

的

旗

標,

只需要

简单

在

sunlink

前

加上

"no",

例如

```
# chflags nosunlink file1
```

使用

ls(1)

及

參

數

-lo

可

检

视檔

案目前的

旗

標:
多數的旗標僅能由 root 使用者來標示或移除，而部份旗標可由檔案所有者設定。

我們建議系統管理者可閱讀 chflags(1) 及 chflags(2) 说明以了解相關細節。

3.4.3. setuid、setgid 与 sticky 權限

除了已經探討過的權限外，這裡尚有另外三種特別的設定所有管理者都應該知道，這些設定為 setuid, setgid 以及 sticky 權限。

這些設定對某些一般不會授權給一般使用者的 UNIX™ 操作非常重要，它讓這些功能可運作。要了解這些權限，就必須說明真實使用者 ID (Real user ID) 與有效使用者 ID (Effective user ID) 的差異。

真實使用者 ID 即是擁有者或啟動程序者的 UID，而有效 UID 是執行程序所使用的使用者 ID。例如，passwd(1) 在使用者更改自己的密碼時會以真實使用者 ID 執行，然而，為了要更新密碼資料庫，該指令必須以 root 使用者做為有效 ID 執行，這讓使用者可以更改自己的密碼而不會遇到權限不足 (Permission Denied) 的錯誤。

setuid 權限可以透過在權限集前加上數字 (4) 來設定，如下範例所示:

```
# chmod 4755 suidexample.sh
```

現在 suidexample.sh 的權限會如下所示:

```
-rwsr-xr-x   1 trhodes  trhodes    63 Aug 29 06:36 suidexample.sh
```

注意，suid 現在取代了原來的執行位元成為指定檔案擁有者權限集的一部份，這會允許需要提升權限的工具，如 passwd(1) 可正常使用。

mount(8) 的 nosuid 選項會造成這類 Binary 執行失敗，但不會警告使用者。由於 nosuid Wrapper 可能可繞過該選項，因此該選項並非完全可靠。

實際來看這個範例，先開啟兩個終端機，其中一個用一般使用者輸入 passwd。在等待輸入新密碼的同時，檢查程序表並查看 passwd(1) 程序的使用者資訊:

於終端機 A:
```
Changing local password for trhodes
Old Password:
```

於終端機 B:
```
# ps aux | grep passwd
```

```
91
```
雖然使用一般使用者來執行 passwd(1)，但該程序使用了 root 的有效 UID。

setgid 權限的功能與 setuid 相似，當應用程式或工具使用此設定執行時，將會以擁有該檔案的群組來執行，而非執行該程序的使用者。

要在檔案設定 setgid 權限，需在 chmod(1) 的參數前加上：

```
# chmod 2755 sgidexample.sh
```

注意以下清單中，s 現在位於指定群組權限設定的欄位：

```
-rwxr-sr-x 1 trhodes  trhodes    44 Aug 31 01:49 sgidexample.sh
```

在以上這些範例中，雖然在例子中的 Shell script 是可執行的檔案，但並不會以其他的 EUID 或有效使用者 ID 执行，這是因為 Shell script 不會存取 setuid(2) 系統呼叫 (System call)。

setuid 及 setgid 權限位元可能會因允許提高權限而降低系統的安全性，因此有了第三個特殊的權限：sticky bit，可以加強系統的安全性。

當在目錄上設定 sticky bit，將只允許由檔案擁有者刪除檔案。這對避免公開目錄，如 /tmp 中的檔案被不擁有該檔案的人刪除非常有用。要使用這個權限，可在權限集前加上：

```
# chmod 1777 /tmp
```

sticky bit 權限會以 t 顯示於權限集的最後：

```
# ls -al / | grep tmp
```

```
drwxrwxrwt  10 root  wheel         512 Aug 31 01:49 tmp
```

目錄結構

認識 FreeBSD 的目錄架構，就可對系統有概略的基礎理解。最重要的莫过于整個目錄的根目錄，就是“/”目錄，該目錄會在開機時最先掛載 (mount)，裡面會有開機所會用到必備檔案。

此外，根目錄還紀錄其他檔案系統的掛載點相關設定。

「掛載點」就是讓新增的檔案系統，能接到上層的檔案系統 (通常就是「根目錄」檔案系統) 的目錄。在磁碟組織這邊對此有更詳細介紹。

標準的掛載點包括了 /usr/，/var/，/tmp/，/mnt/ 及 /cdrom/。這些目錄通常會記錄在 /etc/fstab 設定檔內。

/etc/fstab 是記錄各檔案系統及相關掛載點的表格。大部份在 /etc/fstab 有記錄的檔案系統，會在開機時由 rc(8) Script 來自動掛載，除非它們有設定 noauto 選項。

其中細節說明可參閱 fstab 檔。有關檔案系統架構的完整說明可參閱 hier(7)。

現在呢，讓我們大致先一窺常見的目錄有哪些吧。
目錄

說明

目錄

/bin/

單使用者(Single-user)、多使用者(Multi-user)兩種模式皆可使用的基本工具。

/boot/

作業系統開機過程會用到的程式、設定檔。

/boot/defaults/

預設的開機啟動設定檔，詳情請參閱loader.conf(5)。

/dev/

裝置節點(Device node)，詳情請參閱intro(4)。

/etc/

系統設定檔及一些Script檔。

/etc/defaults/

預設的系統設定檔，詳情請參閱rc(8)。

/etc/mail/

郵件傳輸代理程式，像是sendmail(8)的相關設定檔。

/etc/periodic/

每日、每週、每月透過cron(8)，執行的定期排程Script，詳情請參閱periodic(8)。

/etc/ppp/

ppp(8)設定檔。

/mnt/

系統管理者慣用當臨時掛載點的目錄。

/proc/

程序(Process)檔案系統，詳情請參閱procfs(5)及mount_procfs(8)。

/rescue/

緊急救援用途的一些靜態連結(Statically linked)的程式，詳情請參閱rescue(8)。

/root/

root帳號的家目錄。

/sbin/

供單使用者(Single-user)及多使用者(Multi-user)環境使用的系統程式及管理工具。

/tmp/

臨時檔案。一般而論，重開機之後/tmp內的東西會清除掉。而通常會將以記憶體為基礎(Memory-based)的檔案系統掛載在/tmp上。這些瑣事可透過tmpmfs相關的rc.conf(5)環境變數來自動完成。(或是在/etc/fstab內做設定，詳情請參閱mdmfs(8))。

/usr/

主要是使用者所安裝的工具程式、應用程式存放處。

/usr/bin/

常用工具、開發工具、應用軟體。

/usr/include/

標準C include檔案。

/usr/lib/

程式庫存放處。

/usr/libdata/

其他各式工具的資料檔。

/usr/libexec/

系統Daemon及系統工具程式(透過其他程式來執行)。

/usr/local/

存放一些自行安裝的執行檔、程式庫等等。同時，也是FreeBSD Port架構的預設安裝目錄。

/usr/local內的目錄架構大致與/usr相同，詳情請參閱hier(7)說明。但man目錄例外，它們是直接放在/usr/local底下，而非/usr/local/share，而Port所安裝的說明文件則在share/doc/port。

/usr/obj/

在編譯/usr/src目錄時所產生的相關架構目地檔。

/usr/ports/

FreeBSD Port套件集(選用)。

/usr/sbin/

由使用者執行的系統Daemon及系統工具。

/usr/shared/

各架構皆共通的檔案。

/usr/src/

BSD原始碼(或自行新増的)。
目錄

說明
/var/存放各種用途的日誌檔、臨時或暫時存放、列印或郵件的緩衝檔案。有時候,以記憶體為基礎的檔案系統也會掛載在/var。

/var/log/各項系統記錄的日誌檔。
/var/mail/各使用者的郵件檔案。
/var/spool/各種印表機、郵件系統的緩衝目錄。
/var/tmp/臨時檔案。這些檔案在重開機後通常仍會保留,除非/var是屬於以記憶體為基礎的檔案系統。
/var/yp/NIS對應表。

3.6.磁碟組織
FreeBSD用來尋找檔案的最小單位就是檔案的名稱了。檔案的名稱有大小寫之分,所以說readme.txt和README.TXT是兩個不同的檔案。

FreeBSD並不使用副檔名(.txt)來判別这是一个程式檔、文件檔或是其他類型的檔案。檔案或目錄的對應是藉由給定的檔案或目錄名稱,然後加上正斜線符號(/)；之後再視需要加上其他的目錄名稱。如果您有一個目錄foo,裡面有一個目錄叫作bar,這個目錄中又包含了一個叫作readme.txt的檔案,那麼這個檔案的全名,或者說檔案的路徑就是foo/bar/readme.txt。注意這與Windows™用來分隔檔案與目錄名稱所使用的不同,且FreeBSD在路徑上並不使用磁碟機代號或其他磁碟機名稱,意思是,在FreeBSD上不會有人輸入c:\foo\bar\readme.txt這種路徑。目錄及檔案儲存於檔案系統之中。每個檔案系統都有一個最上層的目錄,叫做根目錄。然後在這個根目錄下面才會有其他的目錄。其中一個檔案系統會被指定成為根檔案系統或/,其他的檔案系統均會掛載在該根檔案系統之下,不論在FreeBSD有多少個磁碟,所有目錄都會成為該磁碟的一部份。假設您有三個檔案系統,分別叫作A, B及C。

每個檔案系統都包含兩個目錄,叫作A1, A2(以此類推得B1, B2及C1, C2)。稱A為主要的檔案系統;如果您用ls(1)指令查看此目錄的內容,您會看到兩個子目錄:A1及A2,如下所示:

一個檔案系統必須以目錄形式掛載於另一個檔案系統上。因此,假設您將B掛載於A1之上,則B的根目錄就變成了A1,而在B之下的任何目錄的路徑也隨之改變:

如果 B 挂載在 A2 之上，則會變成：上面的路徑分別為 /A2/B1 及 /A2/B2。

檔案系統可以掛在其他檔案系統的目錄之上。延續之前的例子，C 檔案系統可以掛在檔案系統 B 的 B1 目錄之上，如圖所示：
您可以使用单一的一个大的根档案系统而不建立其他的档案系统。这有好处也有坏处。

使用多个档案系统的坏处
- 不同的档案系统在挂上的时候可以有不同的挂载参数 (Mount option)。
  
  举例来说，为求谨慎您可以将根档案系统设成唯读，以避免不小心删除或修改掉重要的档案。

将使用者可写入的档案系统 (例如 /home) 独立出来也可以让它们用 nosuid 的参数挂载，此选项可以让在这个档案系统中执行档的 suid/guid 位元失效，可让系统更安全。

FreeBSD 会自动根据您档案系统的方式来做最佳的档案配置方式。因此，一个有很多小档案、常常写入的档案系统跟只有几个较大档案的档案系统配置是不一样的。

如果您只有一个大的档案系统，这部分就没用了。

FreeBSD 的档案系统在停电的时候很稳固。然而，在某些重要的时候停电仍然会对档案系统结构造成损害。

分割成许多档案系统的话在系统在停电后比较能够正常启动，以便您在需要的时候将备份资料回存回来。
將這些建區決定最後在這些分間或裡。案系統要掛交換案系統。 FreeBSD必須立中區,接著置磁碟先切割區建的用的分要安裝 FreeBSD於建。
磁碟機類

型

磁碟機

裝

置

稱

SATA

及

IDE

硬碟

ada

或

ad

SCSI

硬碟

與

USB

儲存

裝

da

SATA

與

IDE

CD-ROM

光

碟機

cd

或

acd

SCSI

CD-ROM

光

碟機

cd

軟碟機

fd

各種

非標準

CD-ROM

光

碟機

mcd

代表

Mitsumi CD-ROM

以

及

scd

代表

Sony

CD-ROM

光

碟機

SCSI

磁帶機

sa

IDE

磁帶機

ast

RAID

磁碟機

範

例

包

含

aacd

代表

Adaptec™

AdvancedRAID,

mlxd

及

mlyd

代表

Mylex™,

amrd

代表

AMI MegaRAID™,

idad

代表

Compaq Smart RAID,

twed

代表

3ware™ RAID.

例

13.

磁碟

、

切

割區及

分

區

命

名

範

例

名稱

意

義

ada0s1a

第一個

SATA

硬碟

((ada0))

上

第一個

切

割區

(s1)

的第一個分

區

(a)

。

da1s2e

第二個

SCSI

硬碟

(da1)

上

第二個

切

割區

(s2)

的第

五

個分

區

(e)

。

例

14.

磁碟

的概念

模

型

此

圖

顯

示

FreeBSD

中

連

接到系統的第一個

SATA

磁碟機

內部

配置圖

。假

設

這個

磁碟

的容

量

是

250

GB

,並且包

含

了一個

80 GB

的

切

割區及

一個

170 GB

的

切

割區

(MS-DOS™

的分

割區)

。第一個

切

割區

是

Windows™NTFS

檔

案系統的

C:

磁碟機

,第二個

則

安裝了

FreeBSD

。本範

例

中安裝的

FreeBSD

有

四

個資料分

區及

一個

交換

分

區

。這

四

個分

區

中各有一個

檔

案系統。分

區

a

是

根檔

案系統、分

區

d

是

/var/

、分

區

e

是

/tmp/

,而分

區

f

是

/usr/

。分

區

字

母

c

用

來

代表

整

個

切

割區

,因此並不作

為

一

般

分

區

使用。
3.7. 挂载与卸载档案系统

就像一颗树。/dev, /usr 以及其他在根目录下的目录就像树根，而这些树根上又有分枝，像是 /usr/local 等。

因为某些原因，我们会将一些目录分别放在不同的档案系统上。如 /var 包含了可能会满出来的 log/，spool/ 等目录以及各式各样的暂存档。

把根档案系统塞到满出来显然不是个好主意，所以往往会比较倾向于把 /var 从 / 中拉出来。

另一个常见到把某些目录放在不同档案系统上的理由是：这些档案在不同的实体或虚拟磁碟机上。

像是网络档案系统（Network File System）详情可参考网络档案系统（NFS）或是光碟机。

3.7.1. fstab 档

在 /etc/fstab 里面有设定的档案系统会在开机（FreeBSD 开机程序）的过程中自动地被挂载（除非该档案系统有加上 noauto 参数）。档案内容的格式如下：

```
device       /mount-point fstype     options      dumpfreq     passno
```

device 已存放在的装置名称，详情请参阅磁碟装置名称。
mount-point

案系統要

掛

載到的目

錄

(該目

錄

必須

存

在

)

fstype

檔

案系統

類

型

,這是要

傳

給

mount(8)

的參

數

。FreeBSD

預

設

的

檔

案系統是

ufs

。

options

可讀可

寫

(Read-Write)

的

檔

案系統用

rw

,而

唯

讀

(Read-Only)

的

檔

案系統

則

是用

ro

,後面

視

需要

還

可以

加

其他

選項

。常

見

的

選項

如

noauto

是用在不要

於開機過

程中自

動

的

掛

載的

檔

案系統。其

他

選項

可參

閱

mount(8)

說明。

dumpfreq

由

此

項

目

決

定

那

些

檔

案系統需

要

傾

印

。如果這格

空

白

則

以

零

為

預

設

值。

passno

這個

項

目

決

定檔

案系統

檢

查

的

順

序。

對於

要

跳過

檢

查

的

檔

案系統,

它們

的

passno

值要

設

為

零

。

根檔

案系統的

passno

值

應設

為

一

(因

為

需要

比

所有其他的

還

要先

檢

查

)

,而其他的

檔

案系統的

passno

值

應

該要

設

得

比

一大。

若有多個

檔

案系統

具

有相同的

passno

值,

則

fsck(8)

會

試

著

並

行

地

(如果可能的

話)

檢

查

這些

檔

案系統。

更多

關於

/etc/fstab

d檔

案格式

及選

項

的資

訊

請參

閱

fstab(5)

說明文

件。

3.7.2.

使用

mount(8)

mount(8)

指令

是

拿

來

掛

載

檔

案系統用的。基本的操

作

指令

格式如下

:

# mount device mountpoint

在

mount(8)

裡面有提到大

一

堆

的

選項

,不過

最常用的

就

是這些

:

掛

載

選項

-a

把

/etc/fstab

裡面所有

還

沒

有

被

掛

載、

沒

有

被

標

記

成

serrat

而且

沒

有用

-t

排

除

的

檔

案系統

掛

載

起

來

。

選項

-d

執

行所有的

動

作,

但

是不

真

的去

呼

叫

掛

載的系統

呼

叫

(System call)

。這個

選項

和

-v

搭

配

拿

來

推測

mount(8)

將

要

做什麼動

作時很

好

用。

選項

-f

強

迫掛

載不

乾淨

的

檔

案系統

(危險)

,或是用

來強制

取

消

寫入

權限

(把

檔

案系統的

掛

載

狀

態

從可

存

取

變

成

唯

讀

)。

選項

-r

用

唯

讀的方式

掛

載

檔

案系統。這個

選項

和在

-o

選項

中

指定

ro

參

數

是一

樣

的。

選項

-t

fstype

用

指定

的

檔

案系統

型態

來

掛

載

指定

的

檔

案系統,或是在有

-a

選項

時只

掛

載

指定

型態

的

檔

案系統。預

設

的

檔

案系統

類

型

為

"ufs"

。

選項

-u

更

新

檔

案系統的

掛

載

選項

。

選項

-v

顯

示詳細

資

訊

。
在預設情況下，指令顯然做什麼是新一次。因此您可以式的執行動觀是新一次。數字正將其後之參數等等，用了多 PID 是指令少記憶名稱在來列出執行的是用行之程序，而且可以指令正的顯行中的程序，有要看系統指令執行個相當有用的程序檢視法有用規信 Daemon 這是習慣，並叫主要的寄送mail 但沒如做慣 httpd 名稱是在上實際的縮寫通常說。後面都會一個 "d" 做為 Deamon 加名字執行的程式字母 BIND 魔吉祥物就 是樂叉耙著快小隻穿為何著三的因。的也原, 是一惡帆布鞋拿 BSD 的裡他們詞祂屬不事惡陣來或臘一些有用的神話中的在背：邪祂做Daemon 把稱型到送傳用程式。從一個站信一個站的程式，也是這種另的應用另一類外，像是我們把這種程式例應何天如說，網頁伺服器求網頁方面的要任它東西。整都在回，而是在執端機有些程式並不是連開始收的終行的時使用者的候就直從中斷與線設計成一。一直都會是 PID 永遠是第一個程序，所以他的程序特做有一個別是一個程序，而您們的程序的。也是程序。父指令 Shell 程序都是跑行的就執身一個您用這種方式 Shell 這個程序的程序，大部份的程序都有 Shell 對父父輸入指令程序是如啟：您，權限過檔啟案 (有者是用及。前面有提到是這個程序可以組擁來群的置訊 PID) 一獨一個程序都有一個無做數每二的作的。維跑以運常而有些系統程序是一每個跑在花時間就正叫做一個的程式一個。過間系統，也 FreeBSD 就是說在同一時程序與Daemon 要注意的是 -a 檔損壞主意，載檔，或是造成電案系統內的資料。卸造好機並不是個強制加上及案系統，蠻這也是一個解析檔 setgid 不或標上旗。擁組群一。
新内容，可用显示动更会自隔应这个指上实际间量该程式的总体使用指是是记忆用程式需要的，而量则空用。记忆量用实际体目前一个是size)另实际用(Resident size)----，量掉在预还记忆会空体的告下多用栏的会分成(Total两少您程序用边。在这，一个是量间情况及执。花的时、使用者、指令行的PID CPU以正费在间名稱栏所来成，和构由好几位个的部份是取中取得。快少有多磁碟改由記憶資料從標示時态CPU间ZFS在不同的少。若有载模組上狀案系统別花了多入，會有一行檔分了，別及間被是系统头分就裡面的其他正占還在交換開是在數在行、有多体有有多程序執字少講空磁忙、系统的(等上評估)度(開機)方式時次間自及現在的時以新。重開機負近的資行前訊兩行或輸出頭PID均示出、系统平)六分成開顯載(ZFS五個部份。
In the following example, we show you how to send signals to a running program and how to interpret the error messages that are returned.

### Procedure

1. First, we need to find the process ID of the program we want to send a signal:

   ```bash
   PID=$(pgrep -f inetd)
   ```

2. Next, we can use the `kill` command to send a signal to the program:

   ```bash
   kill -s HUP $PID
   ```

   Here, `-s` specifies the signal number, and `$PID` is the process ID we obtained in step 1.

3. If the program is running in the background, it may not respond to the signal immediately. You can use the `su` command to switch to the user who owns the program:

   ```bash
   su -c 'kill -s HUP $PID'
   ```

   This command will issue the signal to the program as if you were the owner.

### Error Messages

When a signal is sent to a process, it may either accept or reject the signal. If the signal is accepted, the program will either ignore it or perform some action appropriate to the signal. If the signal is rejected, the program will either die or continue to run.

Here are some common error messages you might receive:

- **`kill: No such process`**
  - This message is returned if the process ID is invalid or the program is not running.

- **`Operation not permitted`**
  - This message is returned if you are not authorized to send the signal to the program.

- **`httpd: Can't send signal`**
  - This message is returned if the signal cannot be sent to the program for some reason.

If you receive any of these messages, you should check the program's documentation or seek help from the program's maintainer.
不同的环境变量设定，任何程序都可以读取此对应的关
是使用了 (Variable/Key) 变数特
档把您名字元
打几个想和显示只 Shell
案就别作余的部份您有。
把剩案分档是补常先首。自功下输入指令或能之一档重，于了。点符合就选，至 shell 手。 如果是那要用易上可能会到想会您也到师父程式。 如果您是一个要用 Shell 那哪个人的对于使用像是可以从前有还 Shell 包含 FreeBSD 也内案管理、集以。
相同

指令在bash下则是:

```
% export EDITOR="/usr/local/bin/emacs"
```

要展开以显示目前环境变数中的值，只要在指令列输入环境变数之前加上`

```bash
$ 
```

字元。举例来说，`echo $TERM`会显示出目前$TERM的设定值。

Shell中有特殊字元用来表示特殊资料，我们将其称作Meta-character。其中最常见的Meta-character是`*`字元，它代表了档名中的任意字元。Meta-character可以用在搜寻档名，例如输入`echo *`会和`ls`得到几乎相同的结果，因为shell会将所有符合`*`字元的档案由`echo`显示出来。

为了避免Shell翻译这些特殊字元，我们可以在这些特殊字元前放一个反斜线(`\`)字元使它们跳脱(S.escape)Shell的翻译。举例来说，`echo $TERM`会印出你目前终端机的设定，`echo \$TERM`则会直接印出$TERM这几个字。

3.9.1. 变更Shell

最简单的变更Shell方法就是透过`chsh`指令。执行`chsh`将会使用环境变数中EDITOR指定的字汇编辑器，如果没有设定，则预设是`vi(1)`。请更改Shell:

```
% chsh -s /usr/local/bin/bash
```

新的Shell必须已列於/etc/shells里。若是依照安装应用程序：套件与Port说明由Port套件集来装的Shell，那就自动列入该档案里。若仍缺少，请使用以下指令加入档案（请将路径替换成新的Shell的路径）：

```
# echo /usr/local/bin/bash >> /etc/shells
```

然后重新执行chsh(1)。

3.9.2. 进阶Shell技巧

UNIX™ Shell不只是指令的直译器，它是一个强大的工具可让使用者执行指令、重新导向指令的输出、重新导向指令的输入并将在指令串连在一起改进最终指令的输出结果。当这个功能与内建的指令混合使用时，可提供一个可以最佳化效率的环境给使用者。

Shell重新导向是将一个指令的输出或输入传送给另一个指令或档案。例如，要撷取`ls(1)`指令的输出到一个档案，可以重新导向输出:

```
% ls > directory_listing.txt
```

目录的内容现在会列到directory_listing.txt中，部份指令可以读取输入，例如`sort(1)`。要排序这个清单，可重新导向输入:

```
% sort < directory_listing.txt
```

输入的内容会被排序后呈现到画面上，要重新导向该输入到另一个档案，可以重新导向输出：

```
105
```
這些手冊可以使用簡短介紹該程式的基本操作。在許多程式中，都會有上數個詳細文件。FreeBSD 的操作手冊會放在 /etc/doc 目錄中，並且會列出開機過體裝的 Boot Message (開機訊) 存放的開機程式當中，開機訊也会偵測到硬碟、鍵盤等。開機訊是硬機表的運算結果。開機訊的編輯通常是透過 Emacs 或 vim 完成。這些程式的編輯文件如 Configuration File (安裝其他程式)。FS 也強烈推遲編輯器。這些程式在修補時會自省，您可以在 Emacs 開始為後期比較難是也。長件集提供。這些更強習學 FreeBSD Port 提案的程式。要了解程式有修補過時，程式名稱的是代表的是插件編入能與操作都螢幕所有功能。其中的程式是直接插入的編輯器。您也可以編輯種文提供的 FreeBSD Port 或器。指令，如 less(1)、sort(1) 等，例如：
filename | sort(1) > sorted.txt
將指令接合，管 "|" 的一組程式。基本到向子標準輸出行運交付指令的輸出導引輸出導到或自遞與通過描述符 (File descriptor) 指令執案個。UNIX™ 子，的管即算過描述符用途是使用這些案符案輸入可能提供。每或案都有來源任定符，輸入何可能提供。這三種機是以來張表印中的可能是皆息被滑鼠可能的檔輸出的用自一種定符，輸入何可能提供。當流含錯誤描述了 (stdin)、(stdout) 及標準符案標準輸出於上述例 UNIX™ 管所有的範符來向過檔中，導重會新 (File descriptor) 指令執案個。
其中

要

要瞭解指令的名稱。舉例來說，要知道

ls(1)

的詳細用法，就可以打：

% man ls

操作手冊被分成很多個章節，每個章節有不同的主題。在 FreeBSD 中操作手冊有以下章節:

1. 使用者指令。
2. 系統呼叫(System call)與錯誤編號。
3. C 程式庫函數。
4. 裝置驅動程式。
5. 檔案格式。
6. 遊戲及其他程式。
7. 其他資訊。
8. 系統維護與操作指令。
9. 系統核心介面。

有些情況會有同樣主題會同時出現於不同章節。舉個例子，系統內會有 chmod 使用者指令，但同時也有 chmod() 系統呼叫。在這種情況下，要告訴 man(1) 要查詢的章節編號:

% man 1 chmod

如此一來就會查詢使用者指令 chmod(1)。通常在寫文件時會把有參考到特定章節的號碼寫在括號內。所以 chmod(1) 就是指使用者指令，而 chmod(2) 則是指系統呼叫。

若不曉得操作手冊的名稱，可以使用 man -k 來以關鍵字查詢所有操作手冊的描述:

% man -k mail

這個指令會顯示所有描述中有使用到關鍵字 "mail" 的指令。這等同使用 apropos(1)。

想想要閱讀所有在/usr/bin 底下的指令說明則可輸入:

% cd /usr/bin
% man -f * | more

或

% cd /usr/bin
% whatis *
| more

3.12.1. GNU Info

FreeBSD 有多應用程式與工具來自自由軟體基金會(Free Software Foundation, FSF)。除了操作手冊之外，這些程式提供了另外一種更具彈性的超文字檔案，叫做 Info 檔。這些檔案可以使用 Info 軟體來閱讀。
要使用 `info(1)` 指令，只需输入:

```
% info
```

要查询简单说明请按 `h` 键，若要查询快速指令参考请按 `?` 键。
4.1. 概述
FreeBSD內建豐富的系統工具，此外FreeBSD提供了兩種安裝第三方軟體的套件管理技術：由原始碼安裝的FreeBSD Port套件集，以及由預先編譯好的Binary安裝的Binary套件集。兩種方法都可使用本地的媒體或網路來安裝軟體。

讀完這章，您將了解：
• Binary套件集與Port的差別。
• 如何找到已移植到FreeBSD的第三方軟體。
• 如何使用pkg管理Binary套件。
• 如何編譯來自Port套件集的第三方軟體原始碼。
• 如何找到應用程式已安裝的檔案來完成安裝後的設定。
• 若軟體安裝失敗要如何處理。

4.2. 安裝軟體的概要
通常要在UNIX™系統上安裝第三方軟體時，有幾個步驟要作：
1. 找到並且下載軟體，該軟體有可能以原始碼或Binary格式發佈。
2. 自發佈的格式解壓縮軟體。發佈的格式通常為tarball並以程式壓縮，如compress(1), gzip(1), bzip2(1)或xz(1)。
3. 找到位於INSTALL, README或者doc/子目錄下的檔案閱讀如何安裝該軟體。
4. 若軟體是以原始碼的格式發佈則需要編譯該軟體。這可能會需要修補Makefile或執行configureScript。
5. 測試並安裝該軟體。

FreeBSD Port是設計用來自動化從原始碼編譯應用程式整個程序的一系列檔案，組成Port的檔案包含了自動下載、解壓縮、修補、編譯與安裝應用程式的必要資訊。

若軟體尚未被FreeBSD採用並測試，可能會需要經過一些修正才能正常安裝並執行。雖然如此，目前已有超過24,000個第三方應用程式已經被移植到FreeBSD。當可行時，這些應用程式也會做成預先編譜好的套件(Package)供下載。

這些Binary套件可使用FreeBSD套件管理指令來管理，如pkg install。

雖然兩種技術非常相似，但Binary套件及Port有各自的優點。要視您要安裝的應用程式需求來選擇。

Binary套件優點
• 應用程式壓縮Binary套件的tarball會比壓縮原始碼的tarball還要小。
• 安裝Binary套件不需要編譜的時間，對於較慢的電腦要安裝大型的應用程式如Mozilla, KDE或GNOME這點顯得相當重要。
• Binary套件不需要了解在FreeBSD上編譜軟體的流程。
**Port Suite Advantages**

- Binary suites must be compiled as much as possible on most systems, usually using more general compile options. Port can change these compile options.

- Some application compile options may be relevant to the functionality being installed. For example, Apache has many different built-in options that can be set.

  - In some situations, the same application may exist in multiple Binary suites, such as Ghostscript with `ghostscript` and `ghostscript-nox11` variants to distinguish whether Xorg is installed.

  - If an application has any of the above compile options, this method cannot be used to distinguish Binary suites.

- Some software licenses prohibit the Binary format from being published. Such software must be distributed in source code form and compiled by the end user.

- Some people do not trust Binary published versions, preferring to read the source code to look for hidden problems.

- Source code can be used with custom patches.

To keep track of Port updates, subscribe to the [FreeBSD Port mailing list](https://vuxml.freebsd.org/) and the [FreeBSD Port problems mailing list](https://lists.freebsd.org/mailman/listinfo/freebsd-problems).

Please check https://vuxml.freebsd.org/ to see if there are any security issues or input `pkg audit -F` to check if all installed applications have known vulnerabilities.

---

**Search Software on FreeBSD**

FreeBSD package lists are continuously growing, and there are several ways to find the software you want to install:

- FreeBSD has a searchable list of the latest applications at https://www.FreeBSD.org/ports/.
  - You can search by application name or software category to find Port.

  - [Dan Langille](https://www.dl-name.com/) maintains FreshPorts.org, providing complete search tools and tracking changes in the Port suite. Registered users can set up personal notification lists for Port updates.

  - If you cannot find a specific application, you can first search on SourceForge.net or GitHub.com and then return to the FreeBSD website to check if the application has been ported.

To search for Binary suite files on the package repository:

```
# pkg search subversion
```

- `git-subversion-1.9.2`
- `java-subversion-1.8.8_2`
- `p5-subversion-1.8.8_2`
- `py27-hgsubversion-1.6`
- `py27-subversion-1.8.8_2`
- `ruby-subversion-1.8.8_2`
- `subversion-1.8.8_2`
- `subversion-book-4515`
- `subversion-static-1.8.8_2`
- `subversion16-1.6.23_4`
- `subversion17-1.7.16_2`

- `suite` package name, version number, and if a Port uses Python as a foundation, it will also include the Python version used to compile the suite.
Port 會有多個版本可使用,如 Subversion, 因編譯選項不同,有多個版本可用,這個例子中即指靜態連結版本的 Subversion。在指定要安裝的套件時,最好使用 Port 源來指定該應用程式,Port 源是指應用程式在 Port 樹中的路徑。

```
# pkg search -o subversion
```

```diff
+devel/git-subversion
+java/java-subversion
+devel/p5-subversion
+devel/py-hgsubversion
+devel/py-subversion
+devel/ruby-subversion
+devel/subversion16
+devel/subversion17
+devel/subversion
+devel/subversion-book
+devel/subversion-static
```

PKG 源支援使用 Shell 萬手字元 (globs)、正規表示法、描述或檔案庫中的其他其他內容。在安裝 ports-mgmt/pkg 或 ports-mgmt/pkg-devel 之後,可參考 pkg-search(8) 以取得更多詳細資訊。

• 若 Port 套件集已安裝,有數個方法可以查詢 Port 樹中的本地版本。要找到 Port 所在的分類,可輸入 whereis file,其中 file 是要安裝的程式:

```
# whereis lsof
```

lsof: /usr/ports/sysutils/lsof

或者,也可使用 echo(1):

```
# echo /usr/ports/*/*lsof*
```

/usr/ports/sysutils/lsof

請注意,這也會顯示已下載至 /usr/ports/distfiles 目錄中任何已符合條件的檔案。

• 另一個方法是使用 Port 套件集內建的搜尋機制來找軟體。要使用搜尋的功能需先 cd 到 /usr/ports 然後執行 make search name=program-name,其中 program-name 代表軟體的名稱。舉例搜尋 lsof:

```
# cd /usr/ports
# make search name=lsof
```

Port: lsof-4.88.d,8
Path: /usr/ports/sysutils/lsof
Info: Lists information about open files (similar to fstat(1))
Maint: ler@lerctr.org
Index: sysutils
B-deps:
內建的搜尋機制會使用索引檔內的資訊。若出現資訊指出需要INDEX,可執行make fetchindex來下載最新的索引檔。當INDEX檔存在時,make search方可執行請求的搜尋動作。

"Path:"此行代表Port的所在位置。若不要接受這那麼多資訊,可使用quicksearch功能:

```
# cd /usr/ports
# make quicksearch name=lsof
```

Port: lsof-4.88.d,8
Path: /usr/ports/sysutils/lsof
Info: Lists information about open files (similar to fstat(1))

若要進行更有深度的搜尋,使用make search key=string或make quicksearch key=string其中string是要搜尋的文字。該文字可以是一部份的註解、描述或相依套件,當不清楚程式的名稱時可以找到與特定主題相關的Port。

當使用search或quicksearch時,搜尋的字串不分大小寫。搜尋"LSOF"會與搜尋"lsof"產生相同的結果。
CD /usr/ports/ports-mgmt/pkg
make
make install clean

pkg_*: 当升级原使用的旧版工具的既有系统时，必须将资料库转换成新的格式，如此新的工具才会知道有那些已安装过的套件。

pkg:安装完后，必须执行以下指令将套件资料库从旧版格式转换到新版格式:

# pkg2ng

新安装的版本因尚未安装任何第三方软件因此不须做这个步骤。

这个步骤无法还原。一旦套件资料库转换成pkg的格式，旧版pkg_*工具就该继续使用。

套件资料库转换的过程可能会因内容转换为新版本产生错误。通常，这些错误皆可忽略，即使如此，仍然有在执行pkg2ng后无法成功转换的软件清单，这些应用程式则必须手动重新安装。

为确保FreeBSD Port套件集会将新软件的资讯注册到pkg而非旧版套件资料库，FreeBSD版本10.0之前需要在/etc/make.conf加入此行:

WITH_PKGNG= yes

预设pkg会使用FreeBSD套件镜像站(Repository)的Binary套件。若要取得有关编译自订套件档资讯，请参考使用Poudriere编译套件。

其他pkg设定选项目录请参考pkg.conf(5)。

pkg的用法资讯可在pkg(8)手册或不加任何参数执行pkg来取得。

每个pkg指令参数皆记录库在指令操作手册。要阅读pkg install的操作手册，可执行以下指令:

# pkg help install

# man pkg-install

本章节剩余的部份将会示范使用pkg执行常用的Binary套件管理工作。每个示例的指令皆会提供多个参数可使用，请参考指令的说明或操作手册以取得详细资讯或更多范例。

4.4.2. 有关已安装套件的资讯
有关已安装在系统中的套件资讯可透过执行pkg info来检视，若执行时未指定任何参数，将会列出所有已安装或指定的套件版本。

例如，要查看已安装的pkg版本可执行：

113
# pkg info pkg
pkg-1.1.4_1

4.4.3.
安裝與移除套件要安裝Binary套件可使用以下指令，其中packagename為要安裝的套件名稱:

# pkg install packagename

這個指令會使用檔案庫的資料來決定要安裝的軟體版本以及是否有任何未安裝的相依。例如，要安裝curl:

# pkg install curl

Updating repository catalogue
/usr/local/tmp/All/curl-7.31.0_1.txz          100% of 1181 kB 1380 kBps 00m01s
/usr/local/tmp/All/ca_root_nss-3.15.1_1.txz   100% of  288 kB 1700 kBps 00m00s
Updating repository catalogue

The following 2 packages will be installed:

Installing ca_root_nss: 3.15.1_1
Installing curl: 7.31.0_1

The installation will require 3 MB more space
0 B to be downloaded
Proceed with installing packages
[y/N]: y
Checking integrity...done
[1/2] Installing ca_root_nss-3.15.1_1...done
[2/2] Installing curl-7.31.0_1...done
Cleaning up cache files...Done

新的套件以及任何做為相依安裝的額外套件可在已安裝的套件清單中看到:

# pkg info
ca_root_nss-3.15.1_1    The root certificate bundle from the Mozilla Project
curl-7.31.0_1   Non-interactive tool to get files from FTP, GOPHER, HTTP (S servers)

不再需要的套件可以使用pkg delete來移除，例如:

114
pkg delete curl

The following packages will be deleted:
curl-7.31.0_1

The deletion will free 3 MB

Proceed with deleting packages

y/N]: y

[1/1] Deleting curl-7.31.0_1...
done

pkg upgrade

这个指令将会比对已安装的版本与在档案库分类中的版本，并从档案库升级这些套件。

pkg audit -F

pkg autoremove

Packages to be autoremoved:
ca_root_nss-3.15.1_1

The autoremoval will free 723 kB

Proceed with autoremoval of packages

y/N]: y

Deinstalling ca_root_nss-3.15.1_1...
```
sudo pkg prime-list

是一個別名指令，定義在/usr/local/etc/pkg.conf，尚有許多其他相關指令可以用來查詢系統的套件資料庫，例如，指令`pkg prime-origins`可用來取得上述清單的來源Port目錄:
```
```
# pkg prime-origins
www/nginx
security/openvpn
security/sudo
```

這份清單可以用來重新編譯所有安裝在系統中的套件，使用`ports-mgmt/poudriere`或`ports-mgmt/synth`這類的編譯工具。

要將一個安裝好套件註記成"自動"可以用:
```
# pkg set -A 1 devel/cmake
```

當套件為末端套件(Leaf Package)且被註記為"自動"，則會被`pkg autoremove`挑選出來。

要註記一個安裝好套件為"非自動"可以用:
```
# pkg set -A 0 devel/cmake
```

4.4.7.還原套件資料庫不如傳統的套件管理系統，`pkg`有自己的套件資料庫備份機制，此功能預設是開設的。

要停止週期的Script備份套件資料庫可在`periodic.conf(5)`設定`
```
```
daily_backup_pkgdb_enable="NO"
```
```
若要還原有週期Script所產生的備份必須在還原前先解壓縮。

要手動備份pkg資料庫，可執行以下指令，並替換`/path/to/pkg.sql`為適當的檔案名稱與位置:
```
# pkg backup -d /path/to/pkg.sql
```

4.4.8.移除過時的套件預設`pkg`會儲存Binary套件在快取目錄定義在`pkg.conf(5)`中的`PKG_CACHEDIR`，只會保留最後安裝的套件複本。較舊版的`pkg`會保留所有先前的套件，若要移除這些過時的Binary套件，可執行:
```
```
使用以下指令可清空全部的快取:

# pkg clean

4.4.9.

修改套件Metadata

在FreeBSD Port套件集中的軟體可能會經歷主要版本的修改,要解決這個問題可使用pkg內建的指令來更新套件來源。這非常有用,例如lang/php5重新命名為lang/php53,因此lang/php5从此之后代表版本5.4。

要更改上述例子中的套件来源,可执行:

# pkg set -o lang/php5:lang/php53

再一个例子,要更新lang/ruby18为lang/ruby19,可执行:

# pkg set -o lang/ruby18:lang/ruby19

最后的一个例子,要更改libglut共用程式库的来源从graphics/libglut改成graphics/freeglut,可执行:

# pkg set -o graphics/libglut:graphics/freeglut

在更改套件来源之后,非常重要的事是要重新安装套件,来让相依的套件也同时使用修改后的来源。要强制重新安装相依套件,可执行:

# pkg install -Rf graphics/freeglut

4.5.

使用Port套件集

Port套件集是指数个Makefiles、修补及描述档案,每一组这些档案可用以编译与安装在FreeBSD上的一个应用程式,即称为一个Port。

预设,Port套件集储存在/usr/ports的子目录下。在应用程式可以使用Port编译之前,必须先安装Port套件集。若在安装FreeBSD时没有安装,可以使用以下其中一种方式安装:

Procedure: Portsnap

方法FreeBSD的基础系统内含Portsnap,这是一个可用以取得Port套件集简单又快速的工具,较建议多数使用者使用这个方式。此工具会连线到FreeBSD的网站,验证金钥,然后下载Port套件集的新版本。该金钥是用来检验所有已下载档案的完整性。
1. 必須安裝Subversion才可用來取出（Check out）Port樹。若已存在Port樹的複本，可使用此方式安裝Subversion:

   # cd /usr/ports/devel/subversion
   # make install clean

2. 取出Port樹的複本:

   # svn checkout https://svn.FreeBSD.org/ports/head /usr/ports

3. 若需要，在第一次Subversion取出後可使用以下指令更新/usr/ports:

   # svn update /usr/ports

Port套件集中含有代表不同軟體分類的目錄，每個分類底下的子目錄代表每個應用程式，每個內含數個用來告訴FreeBSD如何編譯與安裝該程式檔的應用程式子目錄即稱作Port Skeleton，每個Port Skeleton會含有以下檔案及目錄:
Makefile: 内含用来说明应使用程式如何编译、要安装该程式到那的叙述句。

distinfo: 内含编译Port必须下载的档案名称及校验码（Checksum）。

files/：此目录含有编译与安装程式到FreeBSD时所需的修补档。此目录也可能含有其他用来编译Port的档案。

pkg-descr: 提供程式更详细说明。

pkg-plist: Port安装的所有档案清单，也同时会告诉Port系统解除安装时要移除哪些档案。

部份Port含有pkg-message或其他档案用来处理特殊情况。要取得有关这些档案的详细资讯，以及Port的概要可参考FreeBSD Porter's Handbook。

Port中并不含实际的原始码，即为distfile，在编译Port解压缩时会自动下载的原始码到/usr/ports/distfiles。

4.5.1. 安装Port
下面我们会介绍如何使用Port套件集来安装、移除软体的基本方法。

make可用的目标及环境变数详细说明可参阅ports(7)。

在编译任何Port套件前，请先确认已经如前章节所叙述之方法更新Port套件集。安装任何第三方软体皆可能会导致安全性漏洞，建请在安装前先行阅读https://vuxml.freebsd.org了解Port已知的安全性问题。或者在每次安装新Port前执行pkg audit -F。此指令可以设定在每日系统安全性检查时自动完成安全性稽查及更新漏洞资料库。

要取得更多信息，请参考pkg-audit(8)及periodic(8)。

使用Port套件集会假设有您拥有可正常连线的网络，同时也会需要超级使用者的权限。

要编译并安装Port，需切换目录到要安装的Port底下，然后输入make install，信息中会显示安装的进度：

```
# cd /usr/ports/sysutils/lsof
# make install
```

```>
lsof_4.88D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles.>
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===> Extracting for lsof-4.88...
... [extraction output snipped]
... [compilation output snipped]
```

```>
>> Checksum OK for lsof_4.88D.freebsd.tar.gz.
==> Patching for lsof-4.88.d,8
==> Applying FreeBSD patches for lsof-4.88.d,8
==> Configuring for lsof-4.88.d,8...
... [configure output snipped]
... [compilation output snipped]
```
Installing for lsof-4.88.d,8

[installation output snipped]

Generating temporary packing list
Compressing manual pages for lsof-4.88.d,8
Registering installation for lsof-4.88.d,8

SECURITY NOTE:
This port has installed the following binaries which execute with increased privileges.

/usr/local/sbin/lsof

lsof is a program that requires advanced permissions to execute. Therefore, when the program is installed, a security alert will be displayed. Once installed, it will demonstrate instructions.

Some Shell scripts may set the PATH environment variable to contain directories where commands are available for quick access, to increase the efficiency of executing these commands. For tcsh Shell users, rehash should be run to make new installed commands available without specifying the full path. If using sh Shell, use hash -r. Please refer to the Shell documentation for more information.

During the installation process, the system will create a subdirectory for use when storing temporary files created during compilation. You can remove this directory to save disk space and avoid issues when updating to a new version of the Port:

# make clean

若想在进行这些额外步骤时进行操作，可以在编译Port时使用make install clean。
變數。請參考 `fetch(3)` 取得完整的變數清單。

對於那些無法一直连线到网際網路的使用者，可在 `/usr/ports` 下执行 `make fetch` 来下载所有的 distfiles，或是可在某个分类的目录中，例如 `/usr/ports/net`，或指定的 Port Skeleton 中执行。要注意的是，若 Port 有任何的相依，在分类或 Port Skeleton 中执行此指令并不会下载相依在其他分类的 Port distfiles。可使用 `make fetch-recursive` 来下载所有相依 Port 的 distfiles。

在部份少数情况下，例如当公司或组织有自己本地的 distfiles 档案库，可使用 `MASTER_SITES` 变数来覆盖在 Makefile 中指定的下载位址。当要指定替代的位址时可:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE=ftp://ftp.organization.org/pub/FreeBSD/ports/distfiles fetch
```

也可使用 `WRKDIRPREFIX` 及 `PREFIX` 变数来覆盖预设的工作及目标目录。例如:

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

会编译在 `/usr/home/example/ports` 的 Port 并安装到 `/usr/local` 下。然后:

```
# make PREFIX=/usr/home/example/local install
```

会编译在 `/usr/ports` Port 并安装到 `/usr/home/example/local`。然后:

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

来同时设定工作及目标目录。

这些变数也可做为环境变数设定，请参考您使用的 Shell 操作手册来取得如何设定环境变数的说明。

4.5.2. 移除已安装的 Port

安装的 Port 可以使用 `pkg delete` 解除安装。使用这个指令的范例可以在 `pkg-delete(8)` 操作手册找到。

或者，可在 Port 的目录下执行 `make deinstall`:

```
# cd /usr/ports/sysutils/lsof
make deinstall
```

```none
Deinstalling for sysutils/lsof
Deinstallation has been requested for the following 1 packages:
lsof-4.88.d,8
The deinstallation will free 229 kB
[1/1] Deleting lsof-4.88.d,8...
done
```

建议阅读 Port 解除安装后的信息，若有任何相依该 Port。
4.5.3. 升级与管理 Port 的工具

Port 套件集含有数个工具可以进行升级，每一种工具都有其优点及缺点。以往大多 Port 安装会使用 Portmaster 或 Portupgrade，现在有较新的 Synth 可使用。

那一种工具对特定系统是最佳选择取决于系统管理员。建议在使用任何这些工具之前先备份数据。

4.5.3.1. 使用 Portmaster 升级 Port

ports-mgmt/portmaster 是可用来升级已安装 Port 的小巧工具，它只使用了随 FreeBSD 基础系统安装的工具，不需要相依其他 Port 或资料库便可在 FreeBSD 使用，要使用 Port 安装此工具可:

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

Portmaster 将 Port 定义成四种类型:

• 根 Port：没有相依且也不被任何其他 Port 相依。

• 主干 Port：没有相依，但被其他 Port 相依。

• 分支 Port：有相依，且其被其他 Port 相依。

• 枝 Port：有相依，但没有被其他 Port 相依。

要列出这四个类别并搜寻是否有新版:

```
# portmaster -L
```
Root ports

(No dependencies, not depended on)

ispell-3.2.06_18

screen-4.0.3

New version available: screen-4.0.3_1

tcpflow-0.21_1

7 root ports

Branch ports

(Have dependencies, are depended on)

apache22-2.2.3

New version available: apache22-2.2.8

137 total installed ports

83 have new versions available

此指令用来升级所有过时的Port:

# portmaster -a

Portmaster会删除已存在的Port前备份。若成功安装新版Portmaster会删除该备份。使用-b来让Portmaster不会自动删除备份。加入-i可启动Portmaster的互动模式,会在升级每个Port前提示信息。尚有多种可用的其他选项,请阅读portmaster(8)的操作手册来取得详细的用法。

若升级的过程发生错误,可加入-f来升级并重新编译所有Port:

# portmaster -af

Portmaster也可用来安装新的Port到系统,在编译及安装新Port前升级所有相关模组。要使用这个功能,要指定Port位于Port套件中的位置:

# portmaster shells/bash

更多有关ports-mgmt/portmaster的资讯可至其pkg-descr取得。
# cd /usr/ports/ports-mgmt/portupgrade
# make install clean

在执行升级之前使用此工具，建议使用 `pkgdb -F` 扫描已安装的 Port 并修正该指令回报的所有资讯不一致的套件。

要升级所有安装在系统上的 Port，可使用 `portupgrade -a`，或者加上 `-i` 会在每个套件升级时询问确认：

```
# portupgrade -ai
```

若想升级指定的应用程式而非所有可用的 Port 可使用 `portupgrade pkgname`，非常重要的，要加上 `-R` 来先升级指定应用程式所有相依的 Port：

```
# portupgrade -R firefox
```

若使用 `-P`，Portupgrade 会先在 `PKG_PATH` 清单中的本地目录中搜寻可用的套件。若本地没有可用的套件，则会从远端下载。若套件无法在本地或远端找到，Portupgrade 会使用 Port 来安装。要避免完全使用 Port 安装，可使用 `-PP`，这个选项会告诉 Portupgrade 若没有套件可用时放弃安装：

```
# portupgrade -PP gnome3
```

若只想下载 Port distfiles 或套件，使用 `-P` 参数。若不要编译或安装任何东西，使用 `-F`。请参考 `portupgrade` 的操作手册来取得所有可用选项的更多信息。

更多有关 `ports-mgmt/portupgrade` 的资讯可至其 `pkg-descr` 取得。

4.5.4. Port 与磁碟

使用 Port 套件集会随着时问消耗磁碟空间。在编译与安装 Port 完之后，在 Port Skeleton 中执行 `make clean` 可清除暂存的 work 目录。若使用 Portmaster 来安装 Port，则会自动移除该目录，除非使用 `-K`。若有安装 Portupgrade，此指令将会移除所有在 Port 套件集的本地复本中找到的 work 目录：

```
# portsclean -C
```

除此之外，许多过时的原始码发档会储存于 `/usr/ports/distfiles`。使用 Portupgrade 删除所有不再被任何 Port 所引用的 distfiles：

```
# portsclean -D
```

Portupgrade 可以移除所有未被任何安装在系统上的 Port 所引用的 distfiles：

```
# portsclean -DD
```

若有安装 Portmaster，则可使用：

```
```
portmaster --clean-distfiles

預設，若distfile應要被刪除，這個指令會以互動的方式向使用者確認。

除了以上指令外，ports-mgmt/pkg_cutleaves可自動移除不再需要使用的Port。

4.6. 使用Poudriere編譯套件

Poudriere是一個使用BSD授權條款用來建立與測試FreeBSD套件的工具。它使用FreeBSD Jail來建立獨立的編譯環境，這些Jail可以用來編譯與目前所在系統不同FreeBSD版本的套件，也同樣可以在主機為amd64的系統上編譯供i386使用的套件。套件編譯完成後的目錄配置會與官方鏡像站完全相同。這些套件可由pkg(8)及其他套件管理工具使用。

Poudriere可使用ports-mgmt/poudriere套件或Port安裝。安装完成後會有一個範例的設定檔/usr/local/etc/poudriere.conf.sample。複製此檔案到/usr/local/etc/poudriere.conf，編輯複製的檔案來配合本地的設定。

雖然在系統上執行poudriere並不一定要使用ZFS，但使用了是有幫助的。當使用了ZFS，則必須在/usr/local/etc/poudriere.conf指定ZPOOL及FREEBSD_HOST應設定到一個最近的鏡像站。定義CCACHE_DIR可開啟使用devel/ccache快取的功能來快取編譯結果並減少那些需時常編譯的程式碼的編譯次數。

將poudriere資料集放到一個獨立的目錄並掛載到/poudriere可能會比較方便，其他設定項目採用預設值便足夠。

偵測到的處理器數量可用來定義要同時執行多少個編譯。並給予足夠的虛擬記憶體，不論是RAM或交換空間，若虛擬記憶體不足，編譯Jail的動作將會停止並被清除，會造成奇怪的錯誤訊息。

4.6.1. 初始化Jail與Port樹

在設定之後，初始化poudriere來安裝Jail及其所需的FreeBSD樹與Port樹。使用-j來指定Jail的名稱以及-v來指定FreeBSD的版本。在執行FreeBSD/amd64的系統上可使用-a來設定要使用的架構為i386或amd64，預設會採用使用uname所顯示的架構。

```
# poudriere jail -c -j 10amd64 -v 10.0-RELEASE
====>> Creating 10amd64 fs... done
====>> Fetching base.txz for FreeBSD 10.0-RELEASE amd64
/poudriere/jails/10amd64/fromftp/base.txz 100% of   59 MB 1470 kBps 00m42s
====>> Extracting base.txz... done
====>> Fetching src.txz for FreeBSD 10.0-RELEASE amd64
/poudriere/jails/10amd64/fromftp/src.txz 100% of  107 MB 1476 kBps 01m14s
====>> Extracting src.txz... done
====>> Fetching games.txz for FreeBSD 10.0-RELEASE amd64
/poudriere/jails/10amd64/fromftp/games.txz 100% of  865 kB  734 kBps 00m01s
====>> Extracting games.txz... done
====>> Fetching lib32.txz for FreeBSD 10.0-RELEASE amd64
/poudriere/jails/10amd64/fromftp/lib32.txz 100% of   14 MB 1316 kBps 00m12s
====>> Extracting lib32.txz... done
====>> Cleaning up... done
====>> Jail 10amd64 10.0-RELEASE amd64 is ready to be used
```

poudriere ports -c -p local

Creating local fs... done

Extracting portstree "local"...

Looking up portsnap.FreeBSD.org mirrors... 7 mirrors found.

Fetching public key from ec2-eu-west-1.portsnap.freebsd.org... done.

Fetching snapshot tag from ec2-eu-west-1.portsnap.freebsd.org... done.

Fetching snapshot metadata... done.

Fetching snapshot generated at Tue Feb 11 01:07:15 CET 2014: 94a3431f0ce567f6452ffde4fd3d7d3c6e1da143efec76100% of 69 MB 1246 kBps 00m57s

Extracting snapshot... done.

Verifying snapshot integrity... done.

Fetching snapshot tag from ec2-eu-west-1.portsnap.freebsd.org... done.

Fetching snapshot metadata... done.

Updating from Tue Feb 11 01:07:15 CET 2014 to Tue Feb 11 16:05:20 CET 2014.

Fetching 4 metadata patches... done.

Applying metadata patches... done.

Fetching 0 metadata files... done.

Fetching 48 patches. (48/48) 100.00% done.

done.

Applying patches... done.

Fetching 1 new ports or files... done.

/tester/CHANGES

/tester/COPYRIGHT

Building new INDEX files... done.

在一台电脑，poudriere 可使用多组设定在多个 Jail 編譯来自不同 Port 樹的 Port。用來定義這些組合的自訂設定稱作 sets，可在安裝 ports-mgmt/poudriere 或 ports-mgmt/poudriere-devel 後參考 poudriere(8) 中的 CUSTOMIZATION 章節取得詳細的資訊。

在此處示範的基本設定放了單一個 jail，port 及 set 特定的 make.conf 在/usr/local/etc/poudriere.d。在此範例使用的檔案名稱由 Jail 名稱、Port 名稱及 set 名稱所組成: 10amd64-local-workstation-make.conf。系統 make.conf 與這個新的檔案在編譯時會被合并為編譯 Jail 要使用的 make.conf。

要編譯的套件會輸入到 10amd64-local-workstation-pkglist:

editors/emacs
devel/git

ports-mgmt/pkg

126
可使用以下方式設定選項及相依:

```bash
# poudriere options -j 10amd64 -p local -z workstation -f 10amd64-local-workstation-pkglist
```

最後，編譯套件並建立套件檔庫:

```bash
# poudriere bulk -j 10amd64 -p local -z workstation -f 10amd64-local-workstation-pkglist
```

在執行時，按下 `Ctrl+t` 可以顯示目前編譯的狀態，Poudriere 也會編譯在 `/poudriere/logs/bulk/jailname` 中的檔案，可用在網頁伺服器來顯示編譯資訊。

完成之後，新套件現在我可以從 poudriere 檔案庫來安裝。

要取得更多使用 poudriere 的資訊，請參考 `poudriere(8)` 及主網站 [https://github.com/freebsd/poudriere/wiki](https://github.com/freebsd/poudriere/wiki)。

### 4.6.2. 設定 pkg 客戶端

使用 Poudriere 檔案庫雖然可以同時使用自訂的檔案庫與官方檔案庫，但有時關閉官方檔案庫會有幫助。這可以透過建立一個設定檔覆蓋並關閉官方的設定檔來完成。

建立 `/usr/local/etc/pkg/repos/FreeBSD.conf` 包含以下內容:

```ini
FreeBSD: { enabled: no }
```

通常最簡單要提供 poudriere 檔案庫給客戶端的方式是透過 HTTP。安裝一個網頁伺服器來提供套件目錄，通常會像:

```
/usr/local/poudriere/data/packages/10amd64
```

其中 `10amd64` 是編譯的名稱。

若要連往套件檔案庫的 URL 是:

```
http://pkg.example.com/10amd64
```

則在 `/usr/local/etc/pkg/repos/custom.conf` 的檔案庫設定檔為:

```ini
custom: { url: "http://pkg.example.com/10amd64", enabled: yes, }
```

### 4.7. 安裝後的注意事項

不論軟體是從套件或 Port 安裝，大部份的第三方應用程式安裝完後需要做某種程度的設定，下列指令與位置可以用來協助找到應用程式安裝了什麼。

- 大部份應用程式安裝會在 `/usr/local/etc` 安裝至少一個預設的設定檔，若應用程式有大量設定檔的時則會建立一個子目錄來存放這些設定檔。範例的設定檔案名稱通常使用 `.sample`。
結尾，設定檔應要仔細查看並可能要做一些編輯讓設定檔符合系統的需求，要編輯設定檔範本前需先複製該檔案並去掉.sample副檔名。

• 應用程式提供的文件會安裝到/usr/local/shared/doc，且應用程式也同時會安裝操作手冊，在繼續使用應用程式前應先查看這些文件。

• 部分應用程式會以服務的方式執行，在啟動應用程式前前需要加入設定到/etc/rc.conf。這些應用程式通常會安裝啟動Script到/usr/local/etc/rc.d，請參考啟動服務來取得更資訊。

• 依設計，應用程式不會在安裝時執行其啟動Script，也不會在解除安裝或升級時執行其中止Script，這留給各系統的管理者去做決定。

• csh(1)的使用者應要執行rehash來更新已知Binary清單到Shell的PATH。

• 使用pkg info來了了解應用程式安裝了那些檔案、操作手冊以及Binary。

4.8. 处理損壞的Port

當發現某個Port無法順利編譯或安裝，可以嘗試以下幾種方法解決：

1. 搜尋問題回報資料庫看該Port有沒有待審核的修正，若有的話可以使用該修正來修正問題。

2. 寻求維護人員的協助，在Port Skeleton目錄中輸入make maintainer或閱讀Port的Makefile來取得維護人員的電子郵件位址。寄給維護人員的郵件內容請記得要包含Port的Makefile中的$FreeBSD:一整行及輸出的錯誤訊息。

有一些Port並非由個人維護，而是由郵遞論壇維護，有許多，但並非全部，只要郵件地址長的像freebsd-listname@FreeBSD.org都是，寄信時記得代入實際的論壇名稱。

尤其是由ports@FreeBSD.org所維護的Port都不是由特定個人維護，而該Port的修正與支援都是來自訂閱該郵递論壇的一般社群所提供，我們隨時歡迎志工參與!

若寄信後沒有取得任何回應，可以依照撰寫FreeBSD問題回報的說明使用Bugzilla提出問題回報。
要知有些基本概念及運細件的所有然元作方式，並不是道必須節對它們開始：

讀完這章，您解：體資支援影化環境，請參考更多有關 Xorg 型何圖化環境視視來提供環境管理程式。本章 FreeBSD 型圖安裝設定安裝並動將 5.1.

Chapter 5. X Window

• 型化工具來完成大部分的視窗修動 x11-wm 套是可換換，有些景主題以。Port 件集的觀："風面, 有些" 允許有些來同的使用介面支援虛擬桌視窗給所管理。可用的這部分把事該有標題及題列每窗長相，等等是上窗視要用切視能上行動幕移上並不的什麼樣內視。溝通進行。透通網與伺服器之系統執貴情比用程式。在這種較上 XX 強電上較伺服器，腦客戶端比執庭，通常型 X 伺服器跟室公

XTerm 是相行，"上執桌機上" X 的主是要在最相機上強悍認"X" 般認般輸入點板感 X 像自其他輸出如平(投。這困惑自的理電鍵盤工滑鼠鍵盤作包、腦顯架構最路為初設中"client-server" 用架構心計是以網
桌
面
環境 (Desktop Environment)

KDE 與 GNOME 會 被稱 作 桌 面 環境 是 因 為 包 含 了 完 整 常 用 桌 面 作 業 的 應 用 程 式 , 這 些 應 用 程 式 可 能 包 含 文 書 軟 體 、 網 頁 瀏 覽 器 及 遊 戲。

聚焦政 策 (Focus Policy)

視 窗 管 理 程 式 負 責 滑 鼠 指 標 的 聚 焦 政 策 。

聚 焦 政 策 指 的 是 如 何 決 定 使用 中 及 接 收 鍵 盤 輸 入 的 視 窗 。

常 見 人 熟 悉 的 聚 焦 政 策 叫 做 “click-to-focus” , 這 個 模 式 中 , 滑 鼠 點 選 到 的 視 窗 便 會 处 於 作 用 中 (Active) 的 狀 態 。

在 “focus-follows-mouse” 模 式 滑 鼠 指 標 所 在 的 視 窗 是 作 用 中 的 視 窗 , 只 要 把 滑 鼠 移 到 其 他 視 窗 就 可 以 改 變 作 用 中 的 視 窗 , 若 滑 鼠 移 到 根 視 窗 (Root Window) , 則 會 聚 焦 在 根 視 窗 。

在 “sloppy-focus” 模 式 , 既 使 滑 鼠 移 到 根 視 窗 , 仍 然 會 聚 焦 在 最 後 聚 焦 的 視 窗 上 , 此 模 式 只 有 當 滑 鼠 進 入 新 的 視 窗 時 才 會 聚 焦 於 該 視 窗 。

“click-to-focus” 模 式 用 滑 鼠 點 擊 來 決 定 作 用 中 的 視 窗 , 且 該 視 窗 會 被 置 頂 到 所 有 其 它 視 窗 之 前 , 即 使 滑 鼠 移 到 其 他 視 窗 , 所 有 的 鍵 盤 輸 入 仍 會 由 該 視 窗 所 接 收 。

不同的 視 窗 管 理 程 式 支 援 不 同 的 聚 焦 模 式 , 全 部 都 支 援 click-to-focus 且 其 中 大 部 分 支 援 其 他 模 式 , 請 查 看 視 窗 管 理 程 式 的 說 明 文 件 來 了 解 可 用 的 聚 焦 模 式 。

視 窗 元 件 (Widget)

視 窗 元 件 指 的 是 在 所 有 在 用 戶 介 面 上 可 被 點 選 或 操 作 的 項 目 , 這 包 括 按 鈕 、 核 選 方 塊 、 單 選 按 鈕 、 圖 象 及 清 單。

視 窗 元 件 工 具 包 (Widget toolkit) 是 指 用 來 建 立 型 化 應 用 程 式 的 一 系 列 視 窗 元 件 。 目 前 有 數 個 有 名 的 視 窗 元 件 工 具 包 , 包 含 KDE 所 使用 的 Qt 、 GNOME 所 使用 的 GTK+。

因 此 應 用 程 式 會 依 其 發 開 時 所 選 用 的 視 窗 元 件 工 具 包 而 有 不 同 的 外 觀 。

5.3. 安 裝 Xorg 在 FreeBSD , Xorg 可 透 過 套 件 或 Port 來 安 裝 。

使 用 Binary 套 件 的 安 裝 速 度 比 介 紹 但 可 用 的 自 訂 選 擇 比 少 ;

```
# pkg install xorg
```

要 從 Port 套 件 集 編 譯 與 安 裝 :

```
# cd /usr/ports/x11/xorg
# make install clean
```

兩 種 安 裝 方 式 皆 可 完 整 安 裝 Xorg 系 統 , 對 大 部 分 用 戶 應 該 選 用 Binary 套 件 安 裝 。

較 精 簡 版 本 的 X 系 統 仿 宜 規 有 經 驗 的 用 戶 使用 , 可 至 x11/xorg-minimal 取 得 。 這 個 版 本 就 不 會 安 裝 大 部 分 的 文件 、 函 數 庫 及 應 用 程 式 , 而 部 分 應 用 程 式 會 需 要 額 外 的 元 件 才 能 運 作 。

5.4. Xorg 設 定 5.4.1. 快 速 開 始 Xorg 支 援 大 部 分 常 見 的 顯 示 卡 、 鍵 盤 以 及 指 標 裝 置 。

顯 示 卡 、 顯 示 器 以 及 輸 入 裝 置 會 自 動 偵 査 , 無 須 任 何 手 動 設 置 。

除 非 自 動 設 置 失 敗 , 否 則 請 勿 建 立 xorg.conf 或 执行 -configure 步 驟 。
1. 若 Xorg 曾經在電腦使用過，可先將現有的設定檔重新命名或移除:

```
# mv /etc/X11/xorg.conf ~/xorg.conf.etc
# mv /usr/local/etc/X11/xorg.conf ~/xorg.conf.localetc
```

2. 加入要執行 Xorg 的使用者到 video 或 wheel 群組，以便在可用時能開啟 3D 加速。

   要加入使用者 jru 到任一個可用的群組:

```
# pw groupmod video -m jru || pw groupmod wheel -m jru
```

3. 預設內含 TWM 視窗管理程式，啟動 Xorg 時便會啟動該視窗管理程式:

```
% startx
```

4. 在部份較舊版的 FreeBSD，在切換回文字 Console 前系統 Console 必須設為 vt(4) 才可正常運作，請參考核心模塊設定 (Kernel Mode Setting, KMS)。

5.4.2. 可加速影像處理的使用者群組

要存取 /dev/dri 需要允許顯示卡的 3D 加速功能，這通常只需要將要執行 X 的使用者加入 video 群組。此處使用 pw(8) 將使用者 slurms 加入 video 群組，若沒有 video 則會加入 wheel 群組:

```
# pw groupmod video -m slurms || pw groupmod wheel -m slurms
```

5.4.3. 核心模塊設定 (Kernel Mode Setting, KMS)

當電腦顯示從 Console 切換到高螢幕解析度供 X 使用時，必須設定影像輸出模。最近版本的 Xorg 使用了核心內部的系統來讓切換模更有效率。較舊版的 FreeBSD 使用的 sc(4) 不知道到 KMS 系統的存，在會導致關閉 X 之後即始仍在運作但系統 Console 卻呈現空白。較新版的 vt(4) Console 可避免這個問題。

加入此行到 /boot/loader.conf 來開啓 vt(4):

```
kern.vty=vt
```

5.4.4. 設定檔

通常不需要做手動設置，除非自動設置無法運作，否則請不要手動建立設定檔。

5.4.4.1. 頭目錄

Xorg 會查看數個目錄來尋找設定檔，在 FreeBSD 建議使用 /usr/local/etc/X11/來存放這些設定檔，使用這個目錄可以幫助將應用程式檔與作業系統檔分離。

儲存設定檔在傳統的 /etc/X11/仍可運作，但並不建議將應用程式檔與基礎 FreeBSD 檔混合在一 起存放。
5.4.4.2. 使用多档，每档设定一个指定项目会较传统使用单一xorg.conf 设定来的简单。这些档案会存放于主设定档目录下的xorg.conf.d/子目录，完整路径通常为/usr/local/etc/X11/xorg.conf.d/。

於本节稍後会有这些档案的範例。

传统单一xorg.conf 的方式仍可运作，但比起在xorg.conf.d/子目录中的多档设定方式较不明瞭且没有弹性。

5.4.5. 显示卡

由於最近 FreeBSD 版本所做的变更，现在可以使用由 Port 或套件所提供的绘图驱动程式，所以使用者可以使用下列表中的驱动程式。

Intel KMS 驱动程式

大多 数使用 Intel KMS 驱动程式的Intel 显示卡支援 2D 与 3D 加速。

驱动程式名称：i915kms

大多 数使用 Radeon KMS 驱动程式的旧 AMD 显示卡支援 2D 与 3D 加速。

驱动程式名称：radeonkms

大多 数使用 AMD KMS 驱动程的新 AMD 显示卡支援 2D 与 3D 加速。

驱动程式名称：amdgpu

参考文献

Intel™ 3D 加速在大多 数 Intel™ 显示晶片都有支援，最新到 Ivy Bridge (HD Graphics 2500, 4000, 及 P4000) 包含 Iron Lake (HD Graphics) 与 Sandy Bridge (HD Graphics 2000)。

驱动程式名称：intel

参考文献
请至 https://en.wikipedia.org/wiki/List_of_Intel_graphics_processing_units。

AMD™ Radeon 显示卡支援 2D 及 3D 加速，最新到 HD6000 系列。

驱动程式名称：radeon

参考文献
请至 https://en.wikipedia.org/wiki/List_of_AMD_graphics_processing_units。

NVIDIA 有数个 NVIDIA 驱动程式可於Port 套件集中的x11 分类取得，请安装其中与显卡相符的驱动程式。

参考文献
请至 https://en.wikipedia.org/wiki/List_of_Nvidia_graphics_processing_units。

混合组合绘图晶片

部份笔记型电脑加入额外处理单元到那些内建晶片组合或处理。Optimus 结合了 Intel™ 及 NVIDIA 的硬体，Switchable Graphics 或 Hybrid Graphics 则是结合了 Intel™ 或 AMD™ 处理器与 AMD™ Radeon GPU。

这些混合绘图系统的工作方式均不同，FreeBSD 的 Xorg 尚无法驱动所有的混合绘图系统版本。

部份电脑提供了 BIOS 的选项可以关闭其中一个绘图介面卡或选择discrete
Modem, can be used with one of the standard cards to drive the system. For example, it is possible to turn off Optimus system's NVIDIA GPU and make Intel™ display chip available with Intel™ driver program to drive.

BIOS settings will vary by computer model, and in some cases, it is possible to enable two GPUs, with the Device section in the settings file only using the main GPU to run the system.

Other less common display cards may be found in the Port suite's x11-drivers directory. If no specific driver program is available to support the display card, x11-drivers/xf86-video-vesa driver program can be used to drive. This program can be installed with x11/xorg, or manually installed. When no specific driver program is specified, Xorg will try to use this driver program to drive the display card.

x11-drivers/xf86-video-scfb is a non-specific display card driver program, which can be used on many UEFI and ARM™ systems.

To set the display driver program in the configuration file:

Example 15. Select Intel™ display driver program:

```
Section "Device"
Identifier "Card0"
Driver     "intel"
EndSection
```

If there is more than one display card, the BusID identifier can be removed and set to the desired display card, and the Bus ID list can be used to obtain this information:

```
pciconf -lv | grep -B3 display
```

Example 16. Select Radeon display driver program:

```
Section "Device"
Identifier "Card0"
Driver     "radeon"
EndSection
```

Example 17. Select VESA display driver program:

```
Section "Device"
Identifier "Card0"
```

要設定UEFI或ARM™電腦使用scfb動程式:

例18. 在單檔中選擇scfb影像動程式:
/usr/local/etc/X11/xorg.conf.d/driver-scfb.conf

```ini
Section "Device"
Identifier "Card0"
Driver     "scfb"
EndSection
```

5.4.6. 電子顯示器幾乎所有顯示器都支援延伸顯示辨識資料(Extended Display Identification Data, EDID)，Xorg會使用EDID與顯示器通訊並偵測支援的解析度與更新頻率，然後選擇最適合的設定組合使用該顯示器。

其他顯示器支援的解析度可透過在設定檔中設定想要的解析度來選擇，或者在X伺服器啟動之後使用xrandr(1)。

使用xrandr(1)執�行xrandr(1)不加任何參數可檢查影像輸出及已偵測到的顯示器模式清單:

```
% xrandr
Screen 0: minimum 320 x 200, current 3000 x 1920, maximum 8192 x 8192
DVI-0 connected primary 1920x1200+1080+0
(495mm x 310mm)
1920x1200     59.95  
1600x1200     60.00
1280x1024     85.02    75.02    60.02
1280x960      60.00
1152x864      75.00
1024x768      85.00    75.08    70.07    60.00
832x624       74.55
800x600       75.00    60.32
640x480       75.00    60.00
720x400       70.08
DisplayPort-0 disconnected
HDMI-0 disconnected
```

這個結果顯示DVI-0輸出被用來顯示解析度為1920x1200像素於更新頻率約60 Hz的畫面，未有顯示器連接到DisplayPort-0與HDMI-0接頭。

可使用xrandr(1)來選擇其他的顯示模式。例如要切換為1280x1024於60 Hz:

```
```
筆記型電腦使用外部顯示輸出到投影機是常見的作業。

不同裝置間輸出接頭的類型與數量也不同，給每個輸出的名稱在不同驅動程式間也不同。在某些驅動程式稱為HDMI-1的輸出在其他驅動程式則可能稱為HDMI1。因此第一個步驟是執行xrandr(1)列出所有可用的輸出:

```
% xrandr
```

Screen 0: minimum 320 x 200, current 1366 x 768, maximum 8192 x 8192

LVDS1 connected 1366x768+0+0 (normal left inverted right x axis y axis)

344mm x 193mm

1366x768      60.04*

1024x768      60.00+

800x600       60.32    56.25

640x480       59.94

VGA1 connected (normal left inverted right x axis y axis)

1280x1024     60.02 +  75.02

1280x960      60.00

1152x864      75.00

1024x768      75.08    70.07    60.00

832x624       74.55

800x600       72.19    75.00    60.32    56.25

640x480       75.00    72.81    66.67    60.00

720x400       70.08

HDMI1 disconnected (normal left inverted right x axis y axis)

DP1 disconnected (normal left inverted right x axis y axis)

已找到四個輸出：內建面版的LVDS1，外接的VGA1，HDMI1以及DP1接頭。

投影機已連接至VGA1輸出，現在使用xrandr(1)來設定該輸出到投影機（原始解析度）並加入額外的空間到桌面上的右側:

```
% xrandr --output VGA1 --auto --right-of LVDS1 --auto
```

會選擇使用EDID偵測到的解析度與更新頻率。若未正確偵測解析度，可替換--auto為--mode然後給予固定值。例如大部份的投影機可使用1024x768解析度，則可設定--mode 1024x768。

xrandr(1)通常會在.xinitrc執行以在X啟動時設定適合的模式。

在檔案中設定螢幕解析度

在設定檔設定螢幕解析度為1024x768:

例19. 在單檔中設定螢幕解析度

```
/usr/local/etc/X11/xorg.conf.d/screen-resolution.conf
```
少數顯示器沒有EDID，可設定HorizSync及VertRefresh為顯示器支援的頻率範圍。

例20. 手動設定顯示器频率
/usr/local/etc/X11/xorg.conf.d/monitor0-freq.conf

5.4.7. 輸入裝置
5.4.7.1. 鍵盤
鍵盤配置
鍵盤上標準按鍵的位置稱做配置(Layout)。配置與其他可調整的參數列於xkeyboard-config(7)。預設為United States配置，要選擇其他的配置可在InputClass設定XkbLayout與XkbVariant選項。這會套用所有符合該類別的輸入裝置。

這個例子選擇French鍵盤配置使用oss變體。

例21. 設定鍵盤配置
/usr/local/etc/X11/xorg.conf.d/keyboard-fr-oss.conf
22. Setting multiple keyboard configurations in United States, Spanish, and Ukrainian, and switching between these configurations using the Alt+Shift combination. Configurations can be enhanced using x11/xxkb or x11/sbxkb.

```
Section "InputClass"
Identifier  "All Keyboards"
MatchIsKeyboard "yes"
Option      "XkbLayout" "us, es, ua"
EndSection
```

5.4.7.2. Mouse and pointer devices have many mouse parameters that can be adjusted using settings options. Please refer to mousedrv(4) for a complete list.

Mouse button settings can be configured in xorg.conf under the InputDevice section. For example, to set the number of mouse buttons to 7:

```
Section "InputDevice"
Identifier  "Mouse0"
Option      "Buttons" "7"
EndSection
```
5.4.8. 手動設定

在某些情况下，Xorg的自动设定无法在特定硬件上运作，或需要使用不同的设定。针对这些情况会建立自订的设定档。

非必要请勿手动建立设定档，非必要的手动设置会造成运作不正常。

设定档可由Xorg根据侦测到的硬件产生，这个档案对一開始自订设定很有帮助。

产生xorg.conf:

```bash
# Xorg -configure
```

设定档会储存至/root/xorg.conf.new，做任何需要的更改，然后使用以下指令测试该档案:

```bash
# Xorg -config /root/xorg.conf.new
```

在新设定档调整与测试过后，便可分割成较小的档案放置到正常的位置/usr/local/etc/X11/xorg.conf.d/。

5.5. 在Xorg使用字型

5.5.1. Type1字型

由於Xorg内的预设字型用在典型的桌面出版应用程式并不理想，大字型会呈现锯齿状边缘，看起来很不专业，小字型几乎完全看不清。不过，这里有几个免费高品质的Type1（PostScript™）字型可用，且容易在Xorg使用。例如，URW字型集（Times Roman™，Helvetica™，Palatino™及其它）。

Freefont字型集（x11-fonts/freefonts）包含更多字型，但其中大部分是给图形软体如GIMP所使用的字型，并不能完全作为萤幕字型使用。此外，Xorg可以简单的设定使用TrueType™字型。

更多有关本主题的详细资讯，请参考X(7)操作手册或TrueType™字型。

要由Binary套件安装上述的Type1字型集可执行以下指令:

```bash
# pkg install urwfonts
```

或由Port套件集编译，可执行以下指令:

```bash
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

同样的安装方式也适用Freefont或其他字型集。要让X伺服器侦测到这些新安装的字型，可加入适当的设定到X伺服器设定档（/etc/X11/xorg.conf），内容为:

```bash
FontPath "/usr/local/shared/fonts/urwfonts/"
```

或者在X session的指令列执行:

```bash
% xset fp+ /usr/local/shared/fonts/urwfonts
```
這樣便可，但在 X session 關閉時將會失效，除非將設定加入啟動檔（一般的 startx session 可在 ~/.xinitrc 設定，若透過圖型化登入管理程式如 XDM 登入時則在 ~/.xsession 設定）。第三種方式是使用新/usr/local/etc/fonts/local.conf，如反鋸齒字型的示範。

5.5.2. TrueType™ 字型 Xorg 內建支援繪製 TrueType™ 字型，目前有兩個模組可以支援這項功能。在本例中使用 freetype 模組，由於此模組與其他字型繪製後端較為一致。要開啓 freetype 模組只需要將下行加入到 /etc/X11/xorg.conf 中的 "Module" section。

```
Load  "freetype"
```

現在要建立一個儲存 TrueType™ 字型的目錄（例如，/usr/local/shared/fonts/TrueType）然後複製所有 TrueType™ 字型到這個目錄。要注意 TrueType™ 字型並無法直取自 Apple™ Mac™，Xorg 使用的字型必須為 UNIX™/MS-DOS™/Windows™ 檔案複製到讓目錄之後，使用 mkfontscale 來建立 fonts.dir 來讓 X 字型繪製程式知道安裝了新的檔案。

```
# pkg install mkfontscale
```

然後在目錄中建立 X 字型檔的索引:

```
# cd /usr/local/shared/fonts/TrueType
# mkfontscale
```

接著加入 TrueType™ 目錄到字型路徑。這個動作與 Type1 字型中所介紹的方式相同:

```
% xset fp+ /usr/local/shared/fonts/TrueType % xset fp rehash
```

或直接加入 FontPath 一行到 xorg.conf。

現在 Gimp, Apache OpenOffice 以及其他 X 應用程式應可以辨識到已安裝的 TrueType™ 字型。

極小的字型（以高解析度在網頁中顯示的字）與極大的字型（在 StarOffice™ 中）現在會看起來比較樣了。

5.5.3. 反鋸齒字型 所有可在 /usr/local/shared/fonts/ 及 ~/.fonts/ 找到的 Xorg 字型均可在 Xft-aware 的應用程式使用反鋸齒的效果。大多最接近的應用程式均為 Xft-aware 的，包括 KDE, GNOME 以及 Firefox。

要控制那一些字型要做反鋸齒或設定反鋸齒的屬性，需建立 /usr/local/etc/fonts/local.conf 檔案（若檔案存在則編輯）。在這個檔案中可以調整 Xft 字型系統的數項進階功能，本章節僅介紹部份簡單的項目，要取得進一步資訊，請參考 fonts-conf(5)。

這個檔案必須使用 XML 格式，小心文字大小寫，且要確定所有標籤均正常結尾。檔案的開頭使用常見的 XML 檔首，接著為 DOCTYPE 定義，然後是<fontconfig> 標籤:

```
<?xml version="1.0"?
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
```
如同前面所提到的，所有在 /usr/local/shared/fonts/ 与 ~/.fonts/ 的 字型 均可在 Xft-aware 的 应用程式 做 反锯齿 效果，若您 想 要 加入 除了 上两者 以外 的 目录，可 加入 如下行 路径 到 /usr/local/etc/fonts/local.conf:

```
<dir>/path/to/my/fonts</dir>
```

加入 新 字型 及额外 新 字型 目录 之后，需 重新 建立 字型 快取:

```
# fc-cache -f
```

反锯齿 效果 会让 文字 的 边缘 变模糊，这会让 非常 小的 文字 更 能 阅读 且 去除 大型 文字 的 "锯齿"，但 套用 在 一般 的 文字 会 费眼 疲劳。要 排除 小于 14 点 的 字型 大小 使用 反锯齿 效果，可 加入 这些 行:

```
<match target="font">
<test name="size" compare="less">
<double>14</double>
</test>
<edit name="antialias" mode="assign">
<bool>false</bool>
</edit>
</match>
<match target="font">
<test name="pixelsize" compare="less" qual="any">
<double>14</double>
</test>
<edit mode="assign" name="antialias">
<bool>false</bool>
</edit>
</match>
```

反锯齿 所 生产 的 间距 对 部份 等宽 字型 并不 合宜，尤其 在 使用 KDE 时 会 成 为 一个问题。可能 的 修正 方式 是 强制 这类 字型 的 间距 为 100，可 加入 以下行:

```
<match target="pattern" name="family">
<test qual="any" name="family">
<string>fixed</string>
</test>
<edit name="family" mode="assign">
<string>mono</string>
</edit>
</match>
```

```
140
```
這會設定等寬字型的其他常用名稱為"mono",然後加入:
部份字型,如Helvetica,在使用反鋸齒時可能會發生問題,通常會像垂直切成兩半的字型,最差還可能會導致應用程式當掉。要避免這個問題,可考慮加入以下設定到local.conf:

```
<match target="pattern" name="family">
<test qual="any" name="family">
<string>Helvetica</string>
</test>
<edit name="family" mode="assign">
<string>sans-serif</string>
</edit>
</match>
```

編輯local.conf完之後,請確認有使用 femme結尾,若沒有使用會讓所做的更改被忽略。

使用者可透過建立自己的~/.config/fontconfig/fonts.conf來加入個人化的設定,此檔案使用與上述說明相同的XML格式。

最後一點:若有使用LCD螢幕,可能會想使用子像素取樣(Sub-pixel sampling),這基本上會分开處理(水平分隔紅、綠、藍色彩組成)來提高垂直解析度,結果可能是無法預料的。要開啓這個功能,加入下行到local.conf的任一處:

```
<match target="font">
<test qual="rgba">
<const>unknown</const>
</test>
<edit name="font-family" mode="assign">
<string>helvetica</string>
</edit>
</match>
```
依據不同的顯示器類型可能會需要將rgb更改為bgr，vrgb或vbgr：可實驗看看然後看那一個效果最好。

5.6. X顯示管理程式

Xorg 提供了 X 顯示管理程式（X Display Manager, XDM），可用來做登入階段的管理。XDM 提供了一個圖型化的介面來選擇要連結的顯示伺服器以及輸入認證資訊（登入與密碼）。

本節將示範如何設定 FreeBSD 的 X 顯示管理程式。部份桌面環境會提供自己圖型化登入管理程式，请参考 GNOME 設定 GNOME 顯示管理程式（GNOME Display Manager）的操作方式及 KDE 設定 KDE 顯示管理程式（KDE Display Manager）的操作方式。

5.6.1. 設定 XDM

要安裝 XDM 可使用 x11/xdm 套件或 Port。安裝完成之後，可設定 XDM 在開機時執行，只需編輯 /etc/ttys 中的此項目:

```
ttyv8 /usr/local/bin/xdm -nodaemon
```
更改（off）為開（on）然後儲存編輯。在此項目中的 ttyv8 代表 XDM 會在第9個虛擬終端機執行。XDM 的設定目錄位於 /usr/local/etc/X11/xdm。此目錄中包含數個可用來更改 XDM 行為與外觀的檔案以及在 XDM 執行時用來設定桌面的一些 Script 及程式，XDM 設定檔摘要了每個檔案的功能。這些檔案正確的語法與用法在 xdm(1) 有說明。

### 表6. XDM 設定檔

<table>
<thead>
<tr>
<th>檔案</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xaccess</td>
<td>連線到 XDM 所需的通訊協定稱做 X 顯示管理程式連線通訊協定（X Display Manager Connection Protocol, XDMCP），此檔案為客戶端認證規則，用來控制來自遠端器的 XDMCP 連線。預設此檔案不允許任何遠端的客戶端連線。</td>
</tr>
<tr>
<td>Xresources</td>
<td>此檔案控制 XDM 顯示選擇器及登入畫面的外觀。預設的設定簡單的矩形登入視窗，上方用較大的字型顯示機器的主機名稱，並在下方顯示“Login:”與“Password:”提示。此檔案的格式與 Xorg 說明文件中說明的 app-defaults 檔案相同。</td>
</tr>
<tr>
<td>Xservers</td>
<td>登入選擇時在選擇器上要提供的本地及遠端顯示清單。</td>
</tr>
<tr>
<td>Xsession</td>
<td>預設的登入階段 Script，使用者登入之後由 XDM 執行。這會指向使用者自訂的登入階段 Script 於 ~/.xsession。</td>
</tr>
</tbody>
</table>
Xsetup_* 用來在顯示選擇器與登入介面之前自動執行應用程式。每個顯示器各有一個 Script，名稱為 Xsetup_*, 其中 * 為本地顯示編號。正常情況這些 Script 會在背景執行一兩個程式，例如 xconsole。xdm-config 用來設定所有在此機器上執行的顯示的全域設定檔。xdm-errors 內含由伺服器程式產生的錯誤訊息，若 XDM 尋求啟動的顯示沒有回應，可查看此檔案來取得錯誤訊息。以登入階段為基礎，這些訊息也會寫入至使用者的 ~/.xsession-errors。xdm-pid XDM 的執行程序 ID。

5.6.2. 設定遠端存取
預設只有同系統的使用者可以使用 XDM 登入。要啟用讓其他系統的使用者可連線到顯示伺服器，需編輯存取控制規則及啟用連線傾聽程式。

要設定 XDM 傾聽任何遠端的連線，在 /usr/local/etc/X11/xdm/xdm-config 中的 DisplayManager.requestPort 行前加上 ! 來解讀該行:

```
# SECURITY: do not listen for XDMCP or Chooser requests
```
Comment out this line if you want to manage X terminals with xdm
```
DisplayManager.requestPort:     0
```

儲存編輯並重新啟動 XDM，要限制遠端存取，請看 /usr/local/etc/X11/xdm/Xaccess 中的範例項目，並參考 xdm(1) 取得進一步資訊。

5.7. 桌面環境
本節將介紹如何在 FreeBSD 系統安裝三種熱門的桌面環境。一套桌面環境的範圍可從簡單的視窗管理程式到完整的桌面應用程式集。有上百套的桌面環境可在 Port 套件集的 x11-wm 分類取得。

5.7.1. GNOME
GNOME 是一個擁有友善使用者介面的桌面環境，它包括用於啟動應用程式和顯示狀態的面板、一系列工具與應用程序及一套可讓應用程式更容易進行合作、相互一致的協定。更多有關 FreeBSD GNOME 的訊息可在 https://www.FreeBSD.org/gnome 取得，該網站包含有關在 FreeBSD 安裝、設定和管理 GNOME 的額外文件。

這套桌面環境可以從套件安裝:
```
# pkg install gnome3
```
也可使用以下指令從 Port 編譯 GNOME，GNOME 是一套大型的應用程式，即使在速度較快的電腦上，也會需要花費一些時間編譯。
```
# cd /usr/ports/x11/gnome3
# make install clean
```

GNOME 需要掛載 /proc。加入下列到 /etc/fstab 讓系統啟動時會自動掛載這個檔案系統：
 GNOME 使用了 D-Bus 以及 HAL 的 Message bus 与 Hardware abstraction。这两个应用程式会随著 GNOME 的相依一併自动安装，但需要在 /etc/rc.conf 开启，这样在系统开机关时才会启动:

dbus_enable="YES"
hald_enable="YES"

安装完之后,需设定让 Xorg 启动 GNOME。最简单的办法是开启 GNOME Display Manager, GDM, 该程式已做为 GNOME 套件或 Port 的一部分安装了, 可加入下行到 /etc/rc.conf 来开启:

gdm_enable="YES"

通常也会需要启动所有的 GNOME 服务, 可加入下行到 /etc/rc.conf:

gnome_enable="YES"

GDM 则会在系统开机关时自动启动。

第二种启动 GNOME 的方法是在设定完 ~/.xinitrc 后在指令列输入 startx。若这个档 案已经存在, 代替启动目前视窗管理程式的那一行, 改为启动/usr/local/bin/gnome-

session。若档案不存在, 则使用以下指令建立一个:

```
% echo "exec /usr/local/bin/gnome-session" >> ~/.xinitrc
```

第三种方法是使用 XDM 做为显示管理程式, 在这个方法需要建立一个可执行的 ~/.xsession:

```
% echo "exec /usr/local/bin/gnome-session" >> ~/.xsession
```

5.7.2. KDE

KDE 是另一套易於使用的桌面环境。这个桌面环境提供了一致外观的应用程式、标准化的选单和工具列、组合键、配色方案、国际化与集中、对话语框导向的桌面设定。更多有关 KDE 可在 http://www.kde.org/ 取得。要取得 FreeBSD 特定的资讯, 则可参考 http://freebsd.kde.org。

要安装 KDE 套件, 请输入:

```
# pkg install x11/kde5
```

或者要使用 KDE Port 编译, 可使用以下指令, 采用 Port 方式安装会有选单可以选择要安装的元件。 KDE 是一个大型的应用程式, 即使在较快的电脑上仍需要花一段时间来编译。

```
# cd /usr/ports/x11/kde5
# make install clean
```
KDE 需要挂载 /proc。加入下行到 /etc/fstab 让系统启动时会自动挂载这个档案系统:

```
proc           /proc       procfs  rw  0   0
```

KDE 使用了 D-Bus 以及 HAL 的 Message bus 与 Hardware abstraction。这两个应用程序会随着 KDE 的依赖一併自动安装,但需要在 /etc/rc.conf 开启,这样在系统开机时才会启动:

```
dbus_enable="YES"
hald_enable="YES"
```

自 KDE Plasma 5 开始, KDE Display Manager, KDM 便停止开发,可能的替代方案为 SDDM, 要安装该套件可输入:

```
# pkg install x11/sddm
```

加入下行到 /etc/rc.conf:

```
sddm_enable="YES"
```

第二种执行 KDE 的方法是在在指令列输入 startx。要采用这个方式, 需要加入下行到 ~/.xinitrc:

```
exec ck-launch-session startkde
```

第三种启动 KDE 的方式是透过 XDM, 要使用这个方法需要建立一个可执行的 ~/.xsession 如下:

```
%
echo
"exec ck-launch-session startkde"
> ~/.xsession
```

启动 KDE 之后, 请参考内建的说明系统来取得更多有关如何使用各种选单及应用程序的资讯。

5.7.3. Xfce

Xfce 是以 GNOME 使用的 GTK+ 工具包做为开发的桌画面境, 但是它更轻巧且提供了一种简单、高效、易于使用的桌面。它可完全自订设定、附有选单、Applet 及应用程序启动器的主面板、提供档案管理程式和音效管理程式并且可设定主题。由于它是快速、轻巧、高效的桌画面境, 因此它非常适合有记忆体限制的较旧或较慢机器。更多有关 Xfce 的资讯可至 http://www.xfce.org 取得。

要安装 Xfce 套件:

```
# pkg install xfce
```

或者使用 Port 编译:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```
使用了 D-Bus 作為 Message bus，由於是 Xfce 的相依，因此會自動安装，但仍在 /etc/rc.conf 中開啟該程式才會在系統開機時啟動:
```
dbus_enable="YES"
```
不像 GNOME 或 KDE，Xfce 沒有自己登入管理程式，要能用 startx 指令列啟動 Xfce 之前需先加入其項目到 ~/.xinitrc:
```%
  echo ~/.usr/local/etc/xdg/xfce4/xinitrc
```
另—種方式是使用 XDM，要設定這個方式需建立一個可執行的 ~/.xsession:
```%
  echo ~/.usr/local/etc/xdg/xfce4/xinitrc
```
5.8. 安装 Compiz Fusion 要令使用桌面更令人愉快的方法是用炫麗的 3D 效果。安装 Compiz Fusion 套件非常简单，但設定該套件需要一些未在 Port 说明书中的步骤。
5.8.1. 設定 FreeBSD nVidia 驱动程式
桌面特效需要使用相當程度的顯示卡，對於以 nVidia 為基礎的顯示卡，需要使用專用的驅動程序來取得較佳的性能。其他顯示卡的使用可以跳過這一節，並繼續 xorg.conf 設定。
要知道需要那一種 nVidia 驱动程式是正確的之後，接下來的安装程序跟安装其他套件—样简单。例如，要安装最新的驱动程式:
```#
  pkg install x11/nvidia-driver
```
驱动程式會建立—个需要在系统啟動時載入的核心模组，加入下行到 /boot/loader.conf:
```nvidia_load="YES"
```
要立即載入核心模组到执行中的核心可以下 kldload nvidia 指令，但需要注意，若不是在開機時載入，某些 Xorg 版本會無法正常運作。因此編輯完 /boot/loader.conf 之後建議要重新開機。
核心模组載入之後，您只需要更改 xorg.conf 的其中—行來啟用專用的驅動程式:
```/etc/X11/xorg.conf

Driver "nv"
```
然後更改該行為:
```/etc/X11/xorg.conf

Driver "nvidia"
```
如往常一般啟動GUI，您應該會看到nVidia的啟動畫面，其他東西應如往常般運作。

5.8.2. 設定xorg.conf來啟動桌面特效

要開啟Compiz Fusion需要修改`/etc/X11/xorg.conf`:

加入以下`Section`來開啓合成特效:

```
Section "Extensions"
Option         "Composite" "Enable"
EndSection
```

找到“Screen” section，長的應該如下所示:

```
Section "Screen"
Identifier     "Screen0"
Device         "Card0"
Monitor        "Monitor0"
...
```

然後加入以下兩行（在“Monitor”之後）:

```
DefaultDepth    24
Option         "AddARGBGLXVisuals" "True"
```

找到您欲使用的螢幕解析度所在的“Subsection”，例如，您想使用1280x1024，則找到如下所示的Section。

若想使用的解析度不在任何Subsection之中，您可以手動加入對應的項目:

```
SubSection     "Display"
Viewport    0 0
Modes      "1280x1024"
EndSubSection
```

桌面合成需要24 bit的色彩深度，更改上述Subsection為:

```
SubSection     "Display"
Viewport    0 0
Depth       24
Modes      "1280x1024"
EndSubSection
```

最後確認為在“Module” section中已經載入“glx”與“extmod”模組:
前面所述的动作可以执行 `x11/nvidia-xconfig` 来自动完成（使用 root）：

```
# nvidia-xconfig --add-argb-glx-visuals
# nvidia-xconfig --composite
# nvidia-xconfig --depth=24
```

5.8.3. 安装与设定 Compiz Fusion

安装 Compiz Fusion 如同安装其他套件一样简单：

```
# pkg install x11-wm/compiz-fusion
```

安装完成之后，开启您的图形化桌面，然后在终端机的画面输入以下指令（使用一般使用者）：

```
% compiz --replace --sm-disable --ignore-desktop-hints ccp &
% emerald --replace &
```

由于您的视窗管理程式（例如：Metacity，若您使用 GNOME）会被替换为 Compiz Fusion，您的萤幕会闪坏几秒。而 Emerald 会处理视窗的装饰（例如：关闭、最小化、最大化按鈕、标题列及其他相关）。

您或许可以将这些指令改写成较小的 Script 然后在启动时自动执行（加到 GNOME 桌面的“Sessions”中）：

```
#!/bin/sh
compiz --replace --sm-disable --ignore-desktop-hints ccp &
emerald --replace &
```

储存这个 Script 到您的家目录所在位置，例如：start-compiz，然后让该档可以执行：

```
% chmod +x ~/start-compiz
```

接著使用 GUI 将该档案加入启动程式 Startup Programs（位于 GNOME 桌面的系统 System，偏好设定 Preferences，工作阶段 Sessions）。

要选择想使用的特效与相关设定，可执行（一样使用一般使用者） Compiz Config 设定管理程式 Compiz Config Settings Manager：

```
% ccsm
```

在 GNOME 中，也可在系统 System，偏好设定 Preferences 选单中找到。
若您在编译时选择了“gconf support”，您便可以使用gconf-editor在apps/compiz下查看设定。

5.9. 疑难排解
若滑鼠无法使用，您将需要做第一次设定方可继续。在最近的Xorg版本，使用自动侦测装置会忽略在xorg.conf中的InputDevice section。要采用旧的方式，需在此档加入下行到ServerLayout或ServerFlags section:

```
Option "AutoAddDevices" "false"
```

输入装置便可以如先前版本一样设定，连同其他所需的选项（如：切换键盘配置）。

如同前面有说明过，hal Daemon预设会自动侦测您的键盘，因此您的键盘配置或型号可能不正确，桌面环境如GNOME，KDE或Xfce会提供设定键盘的工具。即使用如此，还是有可能透过setxkbmap(1)工具或hal的设定规则的协助来直接设定键盘属性。

举例来说，若有人想要使用PC 102键的键盘，采用法语配置，我们需要建立一个给hal的键盘设定档，名称为x11-input.fdi，并储存在/usr/local/etc/hal/fdi/policy目录。这个档应该要有以下几行:

```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
<device>
<match key="info.capabilities" contains="input.keyboard">
<merge key="input.x11_options.XkbModel" type="string">pc102</merge>
<merge key="input.x11_options.XkbLayout" type="string">fr</merge>
</match>
</device>
</deviceinfo>
```

若这个档已经存在，只需要复制并贴上您的档中有关键盘设定的那几行。

您会需要重新启动您的机器来让hal读取这个档。

也是可以从X终端机或Script下指令来做同样的设定:

```
% setxkbmap -model pc102 -layout fr
```

/usr/local/shared/X11/xkb/rules/base.lst中列出各种可用的键盘、配置与设定。
Model Name: Monitor Model
HorizSync: 30-107
VertRefresh: 48-120

End Section

多數顯示器都支援自動偵測同步頻率，並不需要手動設定這些數值。對於那些不支援自動偵測的顯示器，請輸入由製造商提供的數值來避免損壞顯示器。

X允許在支援的顯示器使用DPMS (Energy Star)功能，xset(1)程式可以控制逾時並可強制待機 (Standby)、暫停 (Suspend) 或關閉 (Off) 模式。若您想為您的顯示器開啟DPMS功能，您需要加入下行到顯示器 (Monitor) 的Section:

Option "DPMS"

在編輯器還未關閉xorg.conf.new設定檔前，選擇想要使用的預設解析度及色彩深度。這些項目可在 "Screen" section定義:

Section "Screen"
Identifier "Screen0"
Device "Card0"
Monitor "Monitor0"
DefaultDepth 24
SubSection "Display"
Viewport 0 0
Depth 24
Modes "1024x768"
EndSubSection
EndSection

DefaultDepth 关键字代表預設執行要使用的色彩深度，這個設定可以被Xorg(1)的指令列參數-depth覆蓋。

Modes 关键字代表執行要使用的解析度，注意，只有VESA標準模式才支援目標系統的繪圖硬體來定義解析度。在上述的例子中，預設使用的色彩深度為每像素24 bit，這個色彩深度可用的解析度為1024 x 768像素。

最後，儲存設定檔並使用測試模式來測試上述的設定。

若一且運作正常，設定檔需要安裝到Xorg(1)會尋找的常用設定檔位置，通常是/etc/X11/xorg.conf或/usr/local/etc/X11/xorg.conf。

# cp xorg.conf.new /etc/X11/xorg.conf

現在已經完成了Xorg的設定程序。Xorg現在可以使用startx(1)工具啟動。Xorg伺服器也可以使用xdm(1)來啟動。
5.9.1. Setting

To set the Intel™ i810 chipset, you need to use the AGP API to control Xorg. Please refer to the agp(4) manual for more information.

This can also let you set any other graphics card on the motherboard. Note that, in systems not compiled with the agp(4) kernel module, it is not possible to use kldload(8) to load the module, so the driver must be compiled into the kernel and started at boot time. This necessitates either compiling or using /boot/loader.conf to load it.

5.9.2. Adding Widescreen Displays to the Configuration File

This section requires some advanced settings knowledge. If you still fail to produce usable settings using the standard setting tools, you may find enough information in the log files to manually build usable settings. In this case, you will need to use a text editor.

Currently, the Widescreen (WSXGA, WSXGA+, WUXGA, WXGA, WXGA+, etc.) format supports 16:10 and 10:9 formats; other formats may cause problems. For example, some common screen resolutions:

- 2560x1600
- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

In some cases, you may need to add these resolutions to Section "Screen":

```plaintext
Section "Screen"
Identifier "Screen0"
Device     "Card0"
Monitor    "Monitor0"
DefaultDepth 24
SubSection "Display"
Viewport  0 0
Depth     24
Modes     "1680x1050"
EndSubSection
EndSection
```

Xorg can obtain resolution information through I2C/DDC, so it knows the screen can handle frequencies and resolutions.

If the driver does not contain the modes that the screen can handle (ModeLines), Xorg needs some indication.

Through /var/log/Xorg.0.log, you can get enough information to manually build a ModeLine. You will need to find information similar to the following:

(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border: 0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border: 0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max 170 MHz

These are called EDID information, and using EDIT information to build a ModeLine requires putting the data in the correct order:

```plaintext
151
```
將資訊放入之後，本例中Section "Monitor"中的ModeLine會看起來像這樣:

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

便完成編輯步驟，接著需要在您的寬螢幕顯示器啟動X。

5.9.3. Compiz Fusion
疑難排解
5.9.3.1. 我已經安裝了Compiz Fusion，但在執行了您所提到的指令後，我的視窗的標題列與按鈕便消失。是那裡有問題？

您可能忘記在/etc/X11/xorg.conf中的設定。請重新檢查這個檔案，特別是DefaultDepth及AddARGBGLXVisuals指令項。

5.9.3.2. 當我執行指令來啟動Compiz Fusion，X伺服器便當掉了，然後我又返回Console。是那裡有問題？
若您檢查/var/log/Xorg.0.log，您可能可以找到當X啟動時所發生的錯誤訊息。最常發生的錯誤會是:

```
(EE) NVIDIA(0):     Failed to initialize the GLX module
(EE) NVIDIA(0):     please check in your X
(EE) NVIDIA(0):     log file that the GLX module has been loaded
in your X server, and that the module is the NVIDIA GLX module.  If
you continue to encounter problems, Please try reinstalling the NVIDIA driver.
```

會發生這個情形通常是因為您升級了Xorg，您需要重新安裝x11/nvidia-driver套件來重編譯glx。
Part II: 一般作業
既然基礎的部分已經提過了，接下來的這個部分將會討論一些常會用到的 FreeBSD 的特色，這些章節包括:

• 介紹給您常見且實用的桌面應用軟體：瀏覽器、辦工工具、文件閱覽程式等。
• 介紹給您眾多 FreeBSD 上可用的多媒體工具。
• 解釋如何編譯量身訂做的 FreeBSD 核心以增加額外系統功能的流程。
• 詳細描述列印系統，包含桌上型印表機及網路印表機的設定。
• 展示給您看如何在您的 FreeBSD 系統中執行 Linux 應用軟體。

這些章節中有些需要您預先閱讀相關文件，在各章節開頭的概要內會提及。
6.1. 概述
随着 FreeBSD 越来越受欢迎，它同时适合作为每日使用的桌面系统。FreeBSD 套件或 Port 有超过 24,000 个可用的应用程式，可以简单的建立一个自订的桌面环境来执行各种不同的桌面应用程式。本章将示范如何安装数个桌面应用程式，包含网页浏览器、办公软件、文件阅览程式及财务软件。

比起重头设定与编译，较偏好使用 FreeBSD 桌面环境已预先编译好版本的使用者可参考 trueos.org 网站。

在阅读这章之前，你必须了解如何:
• 使用套件或 Port 安装其他软体如安装应用程式：套件与 Port 所叙述。
• 安装 X 与视窗管理程式如 X Window 系统所叙述。

要取得有关如何设定多媒体环境的资讯，请参考 多媒体。

6.2. 浏览器
在 FreeBSD 中并未预先安装好网页浏览器。但在 Port 套件集中的 www 分类中有许多浏览器可以採 Binary 套件安装或自 Port 套件集编译的方式安装。

KDE 和 GNOME 桌面环境都有提供自有的 HTML 浏览器。请参考 桌面环境来了解更多媒体环境的资讯。

有一些轻量化浏览器可使用，包含 www/dillo2, www/links 及 www/w3m。

本章节将示范如何安装下列常见网页浏览器并说明该应用程式是否需要用到大量资源、花费大量时间自 Port 编译或何主要的相依套件。

<table>
<thead>
<tr>
<th>应用程式名称</th>
<th>所需资源</th>
<th>自 Port 安装时说明</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firefox</td>
<td>多</td>
<td>多</td>
</tr>
<tr>
<td>Opera</td>
<td>少</td>
<td>少</td>
</tr>
<tr>
<td>Konqueror</td>
<td>多</td>
<td>KDE 程式库</td>
</tr>
<tr>
<td>Chromium</td>
<td>多</td>
<td>Gtk+ 程式库</td>
</tr>
</tbody>
</table>

6.2.1. Firefox
Firefox 是一套开放源码的浏览器，它具备符合 HTML 标准的显示引擎、页签浏览、弹出视窗封锁、扩充套件、强化安全性及其他更多功能。Firefox 的基础使用了 Mozilla 的程式库。

要安装最新释出版本的 Firefox 套件可输入:
```
# pkg install firefox
```

要安装延长支援发布 (Extended Support Release, ESR) 版本的 Firefox，可使用：
```
154
```
pkg install firefox-esr

在本地版本可在www/firefox-i18n及www/firefox-esr-i18n取得。

使用Port套件可以用原始碼編譯成您想要的Firefox版本。此範例編譯www/firefox,其中firefox可替換為ESR或在本地化版本來安裝。

# cd /usr/ports/www/firefox
# make install clean

6.2.2. Opera

Opera是個具備完整功能、符合標準且輕量、執行速度快的瀏覽器。它同時也具備了內建的郵件、新聞閱讀器、IRC客戶端、RSS/Atom來源閱讀器等。

可用的版本有兩種原生的FreeBSD版本及Linux™模擬模式下執行的版本。

以下指令可安裝FreeBSD Binary套件版本的Opera,替換opera為linux-opera則可改安裝Linux™版本。

# pkg install opera

或者,可安裝Port套件集中的版本,以下範例會編譯原生的版本:

# cd /usr/ports/www/opera
# make install clean

要安裝Linux™則替換opera為linux-opera。

要安裝Adobe™Flash™附加元件,需先編譯www/linux-flashplayerPort,因受到授權條款限制無法事先編譯為Binary套件。然後再安裝www/opera-linuxplugins。以下範例示範如How編譯Port中的這兩個應用程式:

# cd /usr/ports/www/linux-flashplayer
# make install clean

# cd /usr/ports/www/opera-linuxplugins
# make install clean

安裝完成後,開啟瀏覽器檢查附加元件是否存在,在網址列輸入opera:plugins並按下Enter鍵,便會有清單顯示目前可用的附加元件。

若要安裝Java™附加元件請接著安裝java/icedtea-web。

6.2.3. Konqueror

Konqueror不只是個網頁瀏覽器,它同時也是檔案管理器和多媒體瀏覽器。它包含在x11/kde4-baseapps套件或Port中。

Konqueror使用支援WebKit以及它自有的KTHML。

WebKit是一套被許多現代瀏覽器所使用的繪圖引擎,包含Chromium。要在FreeBSD的Konqueror使用WebKit需安裝www/kwebkitpart套件或Port。此範例範使用Binary套件安裝:
pkg install kwebkitpart

從 Port 套件集安裝:

# cd /usr/ports/www/kwebkitpart
# make install clean

要啟動 Konqueror 中的 WebKit 點選 "Settings"、"Configure Konqueror"。在 "General" 設定頁面內點選 "Default web browser engine" 旁的下拉示選單並變更 "KHTML" 為 "WebKit"。

Konqueror 也支援 Flash™，"如何在 Konqueror 上安裝 Flash™ 的說明可參考 http://freebsd.kde.org/howtos/konqueror-flash.php。

6.2.4. Chromium

Chromium 是一個開放源始碼的瀏覽器計畫，該計畫的目標是要建立一個安全、快速且更穩定的網頁瀏覽體驗。Chromium 的功能有頁籤式瀏覽、彈出視窗鎖、擴充套件等等。

Chromium 可以使用套件來安裝，只要輸入:

# pkg install chromium

或者可從 Port 套件集的原始碼編譯 Chromium:

# cd /usr/ports/www/chromium
# make install clean

Chromium 的執行檔為 /usr/local/bin/chrome，並非 /usr/local/bin/chromium。

6.3. 办公工具

當開始進行辦公，使用者通常會找好用的辦公軟體或是好上手的文書處理程式。雖然有些桌面環境像是 KDE 已經提供了辦公軟體，但並沒有預設的辦公軟體，FreeBSD 提供多套辦公軟體以及圖型化文書處理程式，不論您用那種的視窗管理程式都能使用。本章節元範如如何安裝以下熱門的辦公軟體以及說明該應用程式所需的資源、自 Port 安裝時間或者是否有其他主要相依套件。

應用程式名稱| 所需資源 | 自 Port 安裝時間 | 主要相依套件 |
---|---|---|---|
Calligra| 少 | 多 | KDE |
AbiWord| 少 | 少 | Gtk+或 GNOME |
The Gimp| 少 | 多 | Gtk+ |
Apache OpenOffice| 多 | 非常多 | JDK™及 Mozilla |
LibreOffice| 有點多 | 非常多 | Gtk+或 KDE/ GNOME 或 JDK™ |
6.3.1. Calligra KDE 桌面環境中內含辦公軟體可以與 KDE 分開安裝。Calligra 中也有可在其他辦公軟體中找到的標準元件，如 Words 是文件處理程式、Sheets 是試算表程式、Stage 可管理投影片以及 Karbon 用來繪製圖型文件。

在 FreeBSD 中可以使用套件或 Port 的方式安裝，要使用套件安裝：
```
# pkg install calligra
```
若沒有可用的套件，可改使用 Port 套件集安裝：
```
# cd /usr/ports/editors/calligra
# make install clean
```

6.3.2. AbiWord

AbiWord 是一個免費的文件處理軟體，外觀和感覺都近似於 Microsoft™ Word。

它非常快速，包含許多功能而且非常容易上手。AbiWord 可以輸入或輸出許多檔案格式，包括一些有專用的格式，例如 Microsoft™ .rtf 格式。

要安裝 AbiWord Binary 套件，可使用下列指令：
```
# pkg install abiword
```
若沒有 Binary 套件版本，也可以從 Port 套件集中編譯安裝：
```
# cd /usr/ports/editors/abiword
# make install clean
```

6.3.3. The GIMP

對於影像的編輯及修改來說，The GIMP 是非常精緻的影像處理軟體。

它可以當作簡單的繪圖軟體或是高品質的相片處理軟體。它支援為數眾多的外掛程式及指令稿(script-fu)介面。
The GIMP 可以讀寫許多檔案格式。它也支援掃描器和手寫板。

要安裝套件可：
```
# pkg install gimp
```
或使用 Port 套件集安裝：
```
# cd /usr/ports/graphics/gimp
# make install clean
```
在 Port 套件集的 graphics 分類(freebsd.org/ports/)下也包涵了許多 GIMP 相關的附加元件，說明檔及使用手冊。
6.3.4. Apache OpenOffice

Apache OpenOffice is open-source office software, developed by Apache Software Foundation’s Incubator team. It contains all the complete office software组合: word processor, spreadsheet, presentation software, and drawing software. It also includes different language user interfaces, spell checker and dictionary.

Apache OpenOffice’s word processor uses native XML file formats to increase portability and flexibility. Spreadsheet program supports macro functions and uses the ability to use external data libraries.

Apache OpenOffice is very stable and can execute on Windows™, Solaris™, Linux™, FreeBSD and Mac OS™ X operating systems. For more information about Apache OpenOffice, you can visit openoffice.org webpage. For FreeBSD specific information, you can refer to porting.openoffice.org/freebsd/

To install Apache OpenOffice:

```
# pkg install apache-openoffice
```

Once the installation is completed, you can run Apache OpenOffice by:

```
% openoffice-X.Y.Z
```

Where X.Y.Z is the version number of the installed Apache OpenOffice. The first execution of Apache OpenOffice will ask some questions and will create a .openoffice.org directory in the user’s home directory.

If you cannot get the desired version of Apache OpenOffice from the package, you can choose to compile it from Port. Nevertheless, you must note that the compile process will require a large amount of disk space and time:

```
# cd /usr/ports/editors/openoffice-4
# make install clean
```

If you want to compile a localized version, you can replace the above instructions with:

```
# make LOCALIZED_LANG=your_language install clean
```

Replace your_language with the correct ISO code. The supported language codes list is in files/Makefile.localized, located in the Port directory.

6.3.5. LibreOffice

LibreOffice is a free office software developed by documentfoundation.org. It is compatible with other主流办公软体 and can be used on various platforms. It is a branch of Apache OpenOffice, containing the applications found in a complete office productivity suite: file processor, spreadsheet, presentation software, drawing software, data library management, and mathematical tools.

LibreOffice also supports many languages and internationalization has been extended to the interface, spell checker and dictionary.

LibreOffice’s file processor uses native XML file format to increase portability and flexibility, spreadsheet program supports macro functions that can connect to external data libraries.

LibreOffice is very stable and can execute on Windows™, Linux™, FreeBSD and Mac OS™ X. More information about LibreOffice can be found at libreoffice.org.

To install the English version of LibreOffice:

```
158
```
```bash
# pkg install libreoffice
```

Port套件集的編輯器類（[freebsd.org/ports/](http://freebsd.org/ports/))中含有多個LibreOffice的語系。

安裝在地化套件時，請替換libreoffice為在地化套件的名稱。

套件安裝之後，輸入以下指令來執行LibreOffice：

```bash
% libreoffice
```

第一次啟動的過程中會詢問一些問題並在使用者的家目錄建立.libreoffice資料夾。

若找不到想使用的LibreOffice套件，也可從Port編譯，但這會要大量的磁碟空間及漫長的時間編譯。以下例子示範編譯英文版本：

```bash
# cd /usr/ports/editors/libreoffice
# make install clean
```

要編譯在地化版本，則需cd進入想用的語言Port目錄。支援的語言可在Port套件集的編輯器類（[freebsd.org/ports/](http://freebsd.org/ports/)）中找到。

### 6.4. 文件瀏覽程式

UNIX™出現之後，有一些新的文件格式越來越熱門，這些文件所需的檢視程式可能並不在基礎系統中。本節將示範如何安裝以下文件檢視程式：

- 应用程式名稱
- 所需資源自Port安裝時主要相依套件
- Xpdf
- FreeType
- gv
- Xaw3d
- Geeqie
- Gtk+
- GNOME ePDFView
- Gtk+
- Okular
- KDE

#### 6.4.1. Xpdf

如果你要一個小型的FreeBSD PDF閱讀軟體，Xpdf是一個輕量而且有效的瀏覽器。

它只需要非常少的資源而且十分穩定。它只使用標準的X字型且不需要額外的工具包（Toolkit）。

安裝Xpdf套件：

```bash
# pkg install xpdf
```

若沒有可用的套件版本，可使用Port套件集安裝：

```bash
# cd /usr/ports/graphics/xpdf
# make install clean
```
完成安裝後，執行 `xpdf` 並使用滑鼠右鍵開啟選單。

6.4.2. `gv`  
`gv` 是 PostScript™ 和 PDF 的閱讀器。它構建於 `ghostview` 的基礎上，不過因為使用 `Xaw3d` 視窗元件工具包，所以外觀看起來比較漂亮。
`gv` 有許多可設定的功能，比方紙張方向、紙張大小、縮放比例、和反鋸齒（Anti-aliasing）等。

而且幾乎所有的使用都可以從鍵盤或滑鼠來完成。

安裝 `gv` 套件：
```
# pkg install gv
```
若沒有可用的套件版本，可使用 Port 套件集安裝：
```
# cd /usr/ports/print/gv
# make install clean
```

6.4.3. `Geeqie`  
`Geeqie` 是由已經停止維護的 `GQView` 專案所衍生出来的分支，并致力開發新功能並整合已有的修補。
`Geeqie` 是一套影像管理軟體，支援單鍵閱覽檔案、啟動外部編輯器、縮圖預覽等功能。
它也有幻燈片模式及一些基本的檔案操作的功能，能輕鬆的管理大量影像並找出重複的檔案。
`Geeqie` 也支援使用全螢幕閱覽及國際化。

安裝 `Geeqie` 套件：
```
# pkg install geeqie
```
若沒有可用的套件版本，可使用 Port 套件集安裝：
```
# cd /usr/ports/graphics/geeqie
# make install clean
```

6.4.4. `ePDFView`  
`ePDFView` 是一套小巧的 PDF 文件檢視程式，只使用了 Gtk+ 與 Poppler 程式庫。
它目前還在開發當中，但已經可以開啟大部份 PDF 檔案（甚至加密過的）、儲存文件複本及支援使用 CUPS 來列印。

要以套件安裝 `ePDFView`：
```
# pkg install epdfview
```
若沒有可用的套件版本，可使用 Port 套件集安裝：
```
# cd /usr/ports/graphics/epdfview
# make install clean
```
Okular

Okular是一套通用的文件检视程式，以KDE的KPDF为基础。它可以开启多种文件格式，包括PDF、PostScript™、DjVu、CHM、XPS以及ePub。

要以套件安装Okular:
```
# pkg install okular
```
若没有可用的套件版本，可使用Port套件集安装:
```
# cd /usr/ports/graphics/okular
# make install clean
```

### 6.5. 财务

如果有任何理由你想在你的FreeBSD桌面上管理你的个人财务，这里有一些功能强大、使用简单的应用程式可供安装。这些财务管理软体之中有些是相容于流行的Quicken或Excel文件。

这节涵盖了下面这些软体:

<table>
<thead>
<tr>
<th>应用程式名称</th>
<th>所需资源</th>
</tr>
</thead>
<tbody>
<tr>
<td>GnuCash</td>
<td>GNOME</td>
</tr>
<tr>
<td>Gnumeric</td>
<td>GNOME</td>
</tr>
<tr>
<td>KMyMoney</td>
<td>KDE</td>
</tr>
</tbody>
</table>

#### 6.5.1. GnuCash

GnuCash是GNOME团队努力成果中的一部分，GNOME团队主要提供亲切而强大的桌面应用程式给终端使用者。使用GnuCash可以持续追踪收入与花费、银行账户以及股票证券等。它的特性是介面直观但功能仍非常专业。

GnuCash提供了智慧的计算器、多层帐户系统以及快速键及自动完成功能。它也能分开单一的报告至数个详细的部份。GnuCash也能够汇入及合并Quicken QIF档文件。它也能处理大部分国际的日期及通用货币之格式。

安装GnuCash套件:
```
# pkg install gnucash
```
若没有可用的套件版本，可使用Port套件集安装:
```
# cd /usr/ports/finance/gnucash
# make install clean
```

#### 6.5.2. Gnumeric

Gnumeric是GNOME社群所开发的试算表程式。它的特点是能够根据储存格格式「猜出」使用者的输入来自动补齐的系统。它也能够汇入许多热门的档文件格式，像是Excel, Lotus 1-2-3以及Quattro Pro。
它有大量内部的函数而且能够使用常用的存储格格式，像是：数字、货币、日期、时间及其他格式等。

安装Gnumeric套件：
```
# pkg install gnumeric
```
若没有可用的套件版本，可使用Port套件集安装：
```
# cd /usr/ports/math/gnumeric
# make install clean
```

6.5.3. KMyMoney

KMyMoney是一套个人财务管理应用程式，由KDE社群所开发。KMyMoney的目标是提供可在商业个人财务管理应用程式中找到的重要功能，它也强调简单易用及其功能间采用合适的复杂记帐。

KMyMoney可从标准Quicken QIF档案汇入资料、追踪投资、处理多种货币并提供财务报表。

要以套件安装KMyMoney：
```
# pkg install kmymoney-kde4
```
若没有可用的套件版本，可使用Port套件集安装：
```
# cd /usr/ports/finance/kmymoney-kde4
# make install clean
```
Chapter 7. 多媒体

7.1. 概述

FreeBSD 广泛地支援各种音效卡，让使用者可以享受来自电脑上的高传真音质（Hi-Fi），此外还包括了录制和播放 MPEG Audio Layer 3 (MP3)、Waveform Audio File (WAV)、Ogg Vorbis 以及其他多种格式声音的力。同时 FreeBSD Port 套件也包含了许多可以让您可以录音、修复音效以及控制 MIDI 配置的应用程式。

FreeBSD 也能播放一般的视讯档和 DVD。FreeBSD Port 套件中包含可编码、转换以及播放各种影像媒体的应用程式。

本章会说明如何设定 FreeBSD 上的音效卡、影像播放器、电视卡及扫描器。同时会说明有哪些应用程式可以使用这些装置。

读完这章，您将了解:

• 设定 FreeBSD 上的音效卡。
• 音效设定疑难排解。
• 播放、录制 MP3 及其他音档格式。
• FreeBSD 系统播放影像的准备工具。
• 播放 DVD 的 .mpg 及 .avi 档。
• 撷取（Rip）CD 和 DVD 的内容至档。
• 设定电视卡。
• 在 FreeBSD 安装 MythTV。
• 设定影像扫描机。
• 设定蓝牙耳机。

在开始阅读这章之前，您需要:

• 知道如何安装应用程式如安装应用程式: 套件与 Port 所叙述。

7.2. 设定音效卡

开始设定之前，必须先知道你的音效卡型号。FreeBSD 支援多种音效卡，请检查支援的音效硬体表（Hardware Notes），以确认你的音效卡是否支援及如何在 FreeBSD 上驱动。

要使用音效装置，必须要载入正确的驱动程式才行。最简单方式就是以 kldload(8) 来载入核心模组。以下范例载入 Intel 规格内建的音效晶片驱动程式:

```
# kldload snd_hda
```

要开機時自動载入驱动程式，需将驱动程式加到 /boot/loader.conf 档，以此驱动程式为例:

```
snd_hda_load="YES"
```

其他可用的音效卡模组清单列于 /boot/defaults/loader.conf。当不确要使用何种驱动程式时，可载入 snd_driver 模组:
It is the metadriver that loads all the most generic audio drivers and uses them to accelerate the search for the correct audio driver. You can also add metadriver to the /boot/loader.conf file to load all audio drivers.

To find out which audio card driver is used after loading snd_driver metadriver, please enter cat /dev/sndstat.

7.2.1. Setting up a custom kernel to provide sound support

This section is for users who prefer to statically compile in support for the sound card in a custom FreeBSD kernel. For more information about recompiling a kernel, refer to Setting FreeBSD.

When using a custom kernel to provide sound support, make sure that the audio framework driver exists in the custom kernel configuration file:

```
device sound
```

Next, add support for the sound card. To continue the example of the built-in audio chipset based on the Intel specification from the previous section, use the following line in the custom kernel configuration file:

```
device snd_hda
```

Be sure to read the manual page of the driver for the device name to use for the driver.

Non-PnP ISA sound cards may require the IRQ and I/O port settings of the card to be added to /boot/device.hints. During the boot process, loader(8) reads this file and passes the settings to the kernel. For example, an old Creative SoundBlaster™ 16 ISA non-PnP card will use the snd_sbc(4) driver in conjunction with snd_sb16. For this card, the following lines must be added to the kernel configuration file:

```
device snd_sbc
device snd_sb16
```

If the card uses the 0x220 I/O port and IRQ 5, these lines must also be added to /boot/device.hints:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

The syntax used in /boot/device.hints is described in sound(4) and the manual page for the driver of the sound card.

The settings shown above are the defaults. In some cases, the IRQ or other settings may need to be changed to match the card. Refer to snd_sbc(4) for more information about this card.
After loading the required module or rebooting into the custom kernel, the sound card should be detected. To confirm, run `dmesg | grep pcm`. This example is from a system with a built-in Conexant CX20590 chipset:

```
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 5 on hdaa0
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 6 on hdaa0
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> at nid 31,25 and 35,27 on hdaa1
```

The status of the sound card may also be checked using this command:

```
# cat /dev/sndstat
```

```
FreeBSD Audio Driver
(newpcm: 64bit 2009061500/amd64)
```

```
Installed devices:
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> (play/rec) default
```

The output will vary depending upon the sound card. If no `pcm` devices are listed, double-check that the correct device driver was loaded or compiled into the kernel. The next section lists some common problems and their solutions.

If all goes well, the sound card should now work in FreeBSD. If the CD or DVD drive is properly connected to the sound card, one can insert an audio CD in the drive and play it with `cdcontrol(1)`:

```
% cdcontrol -f /dev/acd0 play 1
```

Audio CDs have specialized encodings which means that they should not be mounted using `mount(8)`.

Various applications, such as `audio/workman`, provide a friendlier interface. The `audio/mpg123` port can be installed to listen to MP3 audio files.

Another quick way to test the card is to send data to `/dev/dsp`:

```
% cat filename > /dev/dsp
```

where `filename` can be any type of file. This command should produce some noise, confirming that the sound card is working.

The `/dev/dsp*` device nodes will be created automatically as needed. When not in use, they do not exist and will not appear in the output of `ls(1)`.

7.2.3. 設定藍芽音效装置

Connecting to a Bluetooth device is out of scope for this chapter. Refer to Bluetooth for more information.

To get Bluetooth sound sink working with FreeBSD’s sound system, users have to install...
# pkg install virtual_oss

requires cuse to be loaded into the kernel:

# kldload cuse

To load cuse during system startup, run this command:

# sysrc -f /boot/loader.conf cuse_load=yes

To use headphones as a sound sink with audio/virtual_oss, users need to create a virtual device after connecting to a Bluetooth audio device:

# virtual_oss -C 2 -c 2 -r 48000 -b 16 -s 768 -R /dev/null -P /dev/bluetooth/headphones -d

headphones in this example is a hostname from /etc/bluetooth/hosts. BT_ADDR could be used instead.

請參考 virtual_oss(8) 取得更多資訊。

7.2.4. 疑難排解音效

常見錯誤 lists some common error messages and their solutions:

<table>
<thead>
<tr>
<th>錯誤</th>
<th>解決方式</th>
</tr>
</thead>
<tbody>
<tr>
<td>sb_dspwr(XX) timed out</td>
<td>The I/O port is not set correctly.</td>
</tr>
<tr>
<td>bad irq XX</td>
<td>The IRQ is set incorrectly. Make sure that the set IRQ and the sound IRQ are the same.</td>
</tr>
<tr>
<td>xxx: gus pcm not attached, out of memory</td>
<td>There is not enough available memory to use the device.</td>
</tr>
<tr>
<td>xxx: can't open /dev/dsp!</td>
<td>Type fstat</td>
</tr>
</tbody>
</table>

Modern graphics cards often come with their own sound driver for use with HDMI. This sound device is sometimes enumerated before the sound card meaning that the sound card will not be used as the default playback device. To check if this is the case, run dmesg and look for pcm. The output looks something like this:

... hdac0: HDA Driver Revision: 20100226_0142 hdac1: HDA Driver Revision: 20100226_0142 hdac0: HDA Codec #0: NVidia (Unknown) 166
In this example, the graphics card (NVidia) has been enumerated before the sound card (Realtek ALC889). To use the sound card as the default playback device, change `hw.snd.default_unit` to the unit that should be used for playback:

```
# sysctl hw.snd.default_unit=n
```

where `n` is the number of the sound device to use. In this example, it should be 4. Make this change permanent by adding the following line to `/etc/sysctl.conf`:

```
hw.snd.default_unit=4
```

It is often desirable to have multiple sources of sound that are able to play simultaneously. FreeBSD uses "Virtual Sound Channels" to multiplex the sound card's playback by mixing sound in the kernel.

Three `sysctl(8)` knobs are available for configuring virtual channels:

```
# sysctl dev.pcm.0.play.vchans=4
# sysctl dev.pcm.0.rec.vchans=4
# sysctl hw.snd.maxautovchans=4
```

This example allocates four virtual channels, which is a practical number for everyday use. Both `dev.pcm.0.play.vchans=4` and `dev.pcm.0.rec.vchans=4` are configurable after a device has been attached and represent the number of virtual channels `pcm0` has for playback and recording. Since the `pcm` module can be loaded independently of the hardware drivers, `hw.snd.maxautovchans` indicates how many virtual channels will be given to an audio device when it is attached. Refer to `pcm(4)` for more information.

The number of virtual channels for a device cannot be changed while it is in use. First, close any programs using the device, such as music players or sound daemons.
7.2.6. Setting mixer channel values

The default values for the different mixer channels are hardcoded in the source code of the pcm(4) driver. While sound card mixer levels can be changed using mixer(8) or third-party applications and daemons, this is not a permanent solution. To instead set default mixer values at the driver level, define the appropriate values in /boot/device.hints, as seen in this example:

```
hint.pcm.0.vol="50"
```

This will set the volume channel to a default value of 50 when the pcm(4) module is loaded.

7.3. MP3 Music

This section describes some MP3 players available for FreeBSD, how to rip audio CD tracks, and how to encode and decode MP3s.

7.3.1. MP3 Player

A popular graphical MP3 player is Audacious. It supports Winamp skins and additional plugins. The interface is intuitive, with a playlist, graphic equalizer, and more. Those familiar with Winamp will find Audacious simple to use. On FreeBSD, Audacious can be installed from the multimedia/audacious port or package. Audacious is a descendant of XMMS.

The audio/mpg123 package or port provides an alternative, command-line MP3 player. Once installed, specify the MP3 file to play on the command line. If the system has multiple audio devices, the sound device can also be specified:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
```

High Performance MPEG 1.0/2.0/2.5 Audio Player for Layers 1, 2 and 3
version 1.18.1; written and copyright by Michael Hipp and others
free software (LGPL) without any warranty but with best wishes

Playing MPEG stream from Foobar-GreatestHits.mp3 ...

MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo

Additional MP3 players are available in the FreeBSD Ports Collection.

7.3.2. Ripping CD Tracks

Before encoding a CD or CD track to MP3, the audio data on the CD must be ripped to the hard drive. This is done by copying the raw CD Digital Audio (CDDA) data to WAV files.

The cdda2wav tool, which is installed with the sysutils/cdrtools suite, can be used to rip audio information from CDs.

With the audio CD in the drive, the following command can be issued as root to rip an entire CD into individual, per track, WAV files:

```
# cdda2wav -D 0,1,0 -B 168
```
In this example, the `-D 0,1,0` indicates the SCSI device 0,1,0 containing the CD to rip. Use `cdrecord -scanbus` to determine the correct device parameters for the system.

To rip individual tracks, use `-t` to specify the track:

```
# cdda2wav -D 0,1,0 -t 7
```

To rip a range of tracks, such as track one to seven, specify a range:

```
# cdda2wav -D 0,1,0 -t 1+7
```

To rip from an ATAPI (IDE) CDROM drive, specify the device name in place of the SCSI unit numbers. For example, to rip track 7 from an IDE drive:

```
# cdda2wav -D /dev/acd0 -t 7
```

Alternately, `dd` can be used to extract audio tracks on ATAPI drives, as described in 複音樂CD.

### 7.3.3. MP3編碼與解碼

Lame is a popular MP3 encoder which can be installed from the `audio/lame` port. Due to patent issues, a package is not available.

The following command will convert the ripped WAV file `audio01.wav` to `audio01.mp3`:

```
# lame -h -b 128 --tt "Foo Song Title" --ta "FooBar Artist" --tl "FooBar Album" --ty 2014 --tc "Ripped and encoded by Foo" --tg Genre audio01.wav audio01.mp3
```

The specified 128 kbits is a standard MP3 bitrate while the 160 and 192 bitrates provide higher quality. The higher the bitrate, the larger the size of the resulting MP3. The `-h` turns on the "higher quality but a little slower" mode. The options beginning with `-t` indicate ID3 tags, which usually contain song information, to be embedded within the MP3 file. Additional encoding options can be found in the lame manual page.

In order to burn an audio CD from MP3s, they must first be converted to a non-compressed file format. XMMS can be used to convert to the WAV format, while mpg123 can be used to convert to the raw Pulse-Code Modulation (PCM) audio data format.

To convert `audio01.mp3` using mpg123, specify the name of the PCM file:

```
# mpg123 -s audio01.mp3 > audio01.pcm
```

To use XMMS to convert a MP3 to WAV format, use these steps:

**Procedure: Converting to WAV Format in XMMS**

1. Launch XMMS.
2. Right-click the window to bring up the XMMS menu.
3. Select Preferences under Options.
4. Change the Output Plugin to "Disk Writer Plugin".
5. Press Configure.
6. Enter or browse to a directory to write the uncompressed files to.
7. Load the MP3 file into XMMS as usual, with volume at 100% and EQ settings turned off.
8. Press Play. The XMMS will appear as if it is playing the MP3, but no music will be heard. It is actually playing the MP3 to a file.
9. When finished, be sure to set the default.
Output Plugin back to what it was before in order to listen to MP3s again. Both the WAV and PCM formats can be used with cdrecord. When using WAV files, there will be a small tick sound at the beginning of each track. This sound is the header of the WAV file. The audio/sox port or package can be used to remove the header:

```
% sox
  -t wav
  -r 44100
  -s
  -w
  -c 2 track.wav track.raw
```

Refer to 建立與使用 CD 媒體 for more information on using a CD burner in FreeBSD.

7.4. 影片播放

Before configuring video playback, determine the model and chipset of the video card. While Xorg supports a wide variety of video cards, not all provide good playback performance. To obtain a list of extensions supported by the Xorg server using the card, run `xdpyinfo` while Xorg is running.

It is a good idea to have a short MPEG test file for evaluating various players and options. Since some DVD applications look for DVD media in `/dev/dvd` by default, or have this device name hardcoded in them, it might be useful to make a symbolic link to the proper device:

```
# ln -sf /dev/cd0 /dev/dvd
```

Due to the nature of `devfs(5)`, manually created links will not persist after a system reboot. In order to recreate the symbolic link automatically when the system boots, add the following line to `/etc/devfs.conf`:

```
link cd0 dvd
```

DVD decryption invokes certain functions that require write permission to the DVD device. To enhance the shared memory Xorg interface, it is recommended to increase the values of these `sysctl(8)` variables:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

7.4.1. 動畫處理能力

There are several possible ways to display video under Xorg and what works is largely hardware dependent. Each method described below will have varying quality across different hardware. Common video interfaces include:

1. Xorg: normal output using shared memory.
2. XVideo: an extension to the Xorg interface which allows video to be directly displayed in drawable objects through a special acceleration. This extension provides good quality playback even on low-end machines. The next section describes how to determine if this extension is running.
3. SDL: the Simple Directmedia Layer is a porting layer for many operating systems, allowing cross-platform applications to be developed which make efficient use of sound and graphics. SDL provides a low-level abstraction to the hardware which can sometimes be more efficient.
than the Xorg interface. On FreeBSD, SDL can be installed using the `devel/sdl20` package or port.

4. DGA: the Direct Graphics Access is an Xorg extension which allows a program to bypass the Xorg server and directly alter the framebuffer. Because it relies on a low level memory mapping, programs using it must be run as `root`. The DGA extension can be tested and benchmarked using `dga(1)`. When `dga` is running, it changes the colors of the display whenever a key is pressed. To quit, press `q`.

5. SVGAlib: a low level console graphics layer.

7.4.1.1. XVideo

To check whether this extension is running, use `xvinfo`:

```
% xvinfo
XVideo is supported for the card if the result is similar to:
X-Video Extension version 2.2
screen #0
Adaptor #0: "Savage Streams Engine"
number of ports: 1
port base: 43
operations supported: PutImage
supported visuals:
  depth 16, visualID 0x22
  depth 16, visualID 0x23
number of attributes: 5
  "XV_COLORKEY" (range 0 to 16777215)
    client settable attribute
    client gettable attribute (current value is 2110)
  "XV_BRIGHTNESS" (range -128 to 127)
    client settable attribute
    client gettable attribute (current value is 0)
  "XV_CONTRAST" (range 0 to 255)
    client settable attribute
    client gettable attribute (current value is 128)
  "XV_SATURATION" (range 0 to 255)
    client settable attribute
    client gettable attribute (current value is 128)
  "XV_HUE" (range -180 to 180)
    client settable attribute
    client gettable attribute (current value is 0)
maximum XvImage size: 1024 x 1024
Number of image formats: 7
  id: 0x32595559 (YUY2)
  guid: 59555932-0000-0010-8000-00aa00389b71
```
The formats listed, such as YUV2 and YUV12, are not present with every implementation of XVideo and their absence may hinder some players.
This section introduces some of the software available from the FreeBSD Ports Collection which can be used for video playback.

### 7.4.2.1. MPlayer

MPlayer is a command-line video player with an optional graphical interface which aims to provide speed and flexibility. Other graphical front-ends to MPlayer are available from the FreeBSD Ports Collection.

MPlayer can be installed using the `multimedia/mplayer` package or port. Several compile options are available and a variety of hardware checks occur during the build process. For these reasons, some users prefer to build the port rather than install the package.

When compiling the port, the menu options should be reviewed to determine the type of support to compile into the port. If an option is not selected, MPlayer will not be able to display that type of video format. Use the arrow keys and spacebar to select the required formats. When finished, press Enter to continue the port compile and installation.

By default, the package or port will build the `mplayer` command line utility and the `gmplayer` graphical utility. To encode videos, compile the `multimedia/mencoder` port. Due to licensing restrictions, a package is not available for MEncoder.

The first time MPlayer is run, it will create `~/.mplayer` in the user's home directory. This subdirectory contains default versions of the user-specific configuration files.

This section describes only a few common uses. Refer to `mplayer(1)` for a complete description of its numerous options.

To play the file `testfile.avi`, specify the video interfaces with `-vo`, as seen in the following examples:

```
% mplayer -vo xv testfile.avi
% mplayer -vo sdl testfile.avi
% mplayer -vo x11 testfile.avi
# mplayer -vo dga testfile.avi
# mplayer -vo 'sdl:dga' testfile.avi
```
It is worth trying all of these options, as their relative performance depends on many factors and will vary significantly with hardware.

To play a DVD, replace `testfile.avi` with `dvd://N-dvd-device DEVICE`, where `N` is the title number to play and `DEVICE` is the device node for the DVD. For example, to play title 3 from `/dev/dvd`:

```bash
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```

The default DVD device can be defined during the build of the MPlayer port by including the `WITH_DVD_DEVICE=/path/to/desired/device` option. By default, the device is `/dev/cd0`. More details can be found in the port's `Makefile.options`.

To stop, pause, advance, and so on, use a keybinding. To see the list of keybindings, run `mplayer -h` or read `mplayer(1)`.

Additional playback options include `-fs -zoom`, which engages fullscreen mode, and `-framedrop`, which helps performance.

Each user can add commonly used options to their `~/.mplayer/config` like so:

```
vo=xv
fs=yes
zoom=yes
```

Mplayer can be used to rip a DVD title to a `.vob`. To dump the second title from a DVD:

```bash
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-device /dev/dvd
```

The output file, `out.vob`, will be in MPEG format.

Anyone wishing to obtain a high level of expertise with UNIX™ video should consult `mplayerhq.hu/DOCS` as it is technically informative. This documentation should be considered as required reading before submitting any bug reports.

Before using `mencoder`, it is a good idea to become familiar with the options described at `mplayerhq.hu/DOCS/HTML/en/mencoder.html`. There are innumerable ways to improve quality, lower bitrate, and change formats, and some of these options may make the difference between good or bad performance. Improper combinations of command line options can yield output files that are unplayable even by `mplayer`.

Here is an example of a simple copy:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

To rip to a file, use `-dumpfile` with `mplayer`.

To convert `input.avi` to the MPEG4 codec with MPEG3 audio encoding, first install the `audio/lame` port. Due to licensing restrictions, a package is not available. Once installed, type:

```
% mencoder input.avi -oac mp3lame -lameopts br=192 -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

174
This will produce output playable by applications such as mplayer and xine. input.avi can be replaced with dvd://1 -dvd-device /dev/dvd and run as root to re-encode a DVD title directly. Since it may take a few tries to get the desired result, it is recommended to instead dump the title to a file and to work on the file.

7.4.2.2. xine

xine is a video player with a reusable base library and a modular executable which can be extended with plugins. It can be installed using the multimedia/xine package or port. In practice, xine requires either a fast CPU with a fast video card, or support for the XVideo extension. The xine video player performs best on XVideo interfaces.

By default, the xine player starts a graphical user interface. The menus can then be used to open a specific file. Alternatively, xine may be invoked from the command line by specifying the name of the file to play:

% xine -g -p mymovie.avi

Refer to xine-project.org/faq for more information and troubleshooting tips.

7.4.2.3. Transcode

Transcode provides a suite of tools for re-encoding video and audio files. Transcode can be used to merge video files or repair broken files using command line tools with stdin/stdout stream interfaces.

In FreeBSD, Transcode can be installed using the multimedia/transcode package or port. Many users prefer to compile the port as it provides a menu of compile options for specifying the support and codecs to compile in. If an option is not selected, Transcode will not be able to encode that format. Use the arrow keys and spacebar to select the required formats. When finished, press Enter to continue the port compile and installation.

This example demonstrates how to convert a DivX file into a PAL MPEG-1 file (PAL VCD):

% transcode -i input.avi -V --export_prof vcd-pal -o output_vcd
%

% mplex -f 1 -o output_vcd.mpg output_vcd.m1v output_vcd.mpa

The resulting MPEG file, output_vcd.mpg, is ready to be played with MPlayer. The file can be burned on a CD media to create a video CD using a utility such as multimedia/vcdimager or sysutils/cdrdao.

In addition to the manual page for transcode, refer to transcoding.org/cgi-bin/transcode for further information and examples.

7.5. TV card

TV card can let you watch wireless, wired television programs. Many cards are through RCA or S-video inputs to receive video, and some cards also receive FM radio functionality.

FreeBSD can support TV cards through the bktr(4) driver program, as long as these cards use Brooktree Bt848/849/878/879 video capture chips. Furthermore, to verify if the card has the电视台功能 supported, you can refer to bktr(4) manual to see the supporting hardware list.
7.5.1. 載入動程式要用電視卡的話,就要載入bktr(4)動程式,這個可以透過在/boot/loader.conf 檔加上下面這一行就可以了:

```
bktr_load="YES"
```

或者可以將電視卡支援靜態編譯到自訂的核心當中,若要這麼做則可在自訂核心設定檔加入以下行:

```
device   bktr
device  iicbus
device  iicbb
device  smbus
```

之所以要加上這些額外的動程式,是因為卡的各組成部分都是透過I2C匯流排而相互連接的。接下來,請編譯、安裝新的核心。

要測試調諧器(Tuner)是否被正確的偵測,請先重新啟動系統。電視卡應該會出現在開機訊息中,如同此範例:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

該訊息會依硬體不同而有所不同。若必要,可以使用sysctl(8)系統偵測的參數或者自訂核心設定選項。例如要強制使用Philips SECAM調諧器則可加入下列行至自訂核心設定檔:

```
options OVERRIDE_TUNER=6
```

或使用sysctl(8):

```
# sysctl hw.bt848.tuner=6
```

請參考bktr(4)查看sysctl(8)可用的參數說明及核心選項。

7.5.2. 好用的應用程式

To use the TV card, install one of the following applications:

- multimedia/fxtv provides TV-in-a-window and image/audio/video capture capabilities.
- multimedia/xawtv is another TV application with similar features.
- audio/xmradio provides an application for using the FM radio tuner of a TV card.

More applications are available in the FreeBSD Ports Collection.
If any problems are encountered with the TV card, check that the video capture chip and the tuner are supported by `bktr(4)` and that the right configuration options were used. For more support or to ask questions about supported TV cards, refer to the `freebsd-multimedia` mailing list.

**MythTV**

MythTV is a popular, open source Personal Video Recorder (PVR) application. This section demonstrates how to install and setup MythTV on FreeBSD. Refer to `mythtv.org/wiki` for more information on how to use MythTV.

MythTV requires a frontend and a backend. These components can either be installed on the same system or on different machines.

The frontend can be installed on FreeBSD using the `multimedia/mythtv-frontend` package or port. Xorg must also be installed and configured as described in the X Window System. Ideally, this system has a video card that supports X-Video Motion Compensation (XvMC) and, optionally, a Linux Infrared Remote Control (LIRC)-compatible remote.

To install both the backend and the frontend on FreeBSD, use the `multimedia/mythtv` package or port. A MySQL™ database server is also required and should automatically be installed as a dependency. Optionally, this system should have a tuner card and sufficient storage to hold recorded data.

**7.6.1. Hardware**

MythTV uses Video for Linux (V4L) to access video input devices such as encoders and tuners. In FreeBSD, MythTV works best with USB DVB-S/C/T cards as they are well supported by the `multimedia/webcamd` package or port which provides a V4L userland application. Any Digital Video Broadcasting (DVB) card supported by webcamd should work with MythTV. A list of known working cards can be found at `wiki.freebsd.org/WebcamCompat`. Drivers are also available for Hauppauge cards in the `multimedia/pvr250` and `multimedia/pvrxxx` ports, but they provide a non-standard driver interface that does not work with versions of MythTV greater than 0.23. Due to licensing restrictions, no packages are available and these two ports must be compiled.

The `wiki.freebsd.org/HTPC` page contains a list of all available DVB drivers.

**7.6.2. Setting up MythTV**

MythTV backend can be installed in two ways: with the package:

```
# pkg install mythtv
```

or from the port set:

```
# cd /usr/ports/multimedia/mythtv
# make install
```

Once installed, set up the MythTV database:

```
# mysql -uroot -p < /usr/local/shared/mythtv/database/mc.sql
```

Then, configure the backend:

```
177
```
Finally, start the backend:
```
# sysrc mythbackend_enable=yes
# service mythbackend start
```

In FreeBSD, access to image scanners is provided by SANE (Scanner Access Now Easy), which is available in the FreeBSD Ports Collection. SANE will also use some FreeBSD device drivers to provide access to the scanner hardware.

FreeBSD supports both SCSI and USB scanners. Depending upon the scanner interface, different device drivers are required. Be sure the scanner is supported by SANE prior to performing any configuration. Refer to [http://www.sane-project.org/sane-supported-devices.html](http://www.sane-project.org/sane-supported-devices.html) for more information about supported scanners.

This chapter describes how to determine if the scanner has been detected by FreeBSD. It then provides an overview of how to configure and use SANE on a FreeBSD system.

### 7.7.1. 檢查掃描器

The GENERIC kernel includes the device drivers needed to support USB scanners. Users with a custom kernel should ensure that the following lines are present in the custom kernel configuration file:

```
device usb
device uhci
device ohci
device ehci
```

To determine if the USB scanner is detected, plug it in and use `dmesg` to determine whether the scanner appears in the system message buffer. If it does, it should display a message similar to this:

```
ugen0.2: <EPSON> at usbus0
```

In this example, an EPSON Perfection™ 1650 USB scanner was detected on `/dev/ugen0.2`.

If the scanner uses a SCSI interface, it is important to know which SCSI controller board it will use. Depending upon the SCSI chipset, a custom kernel configuration file may be needed. The GENERIC kernel supports the most common SCSI controllers. Refer to `/usr/src/sys/conf/NOTES` to determine the correct line to add to a custom kernel configuration file. In addition to the SCSI adapter driver, the following lines are needed in a custom kernel configuration file:

```
device scbus
device pass
```

Verify that the device is displayed in the system message buffer:
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers

If the scanner was not powered-on at system boot, it is still possible to manually force detection by performing a SCSI bus scan with `camcontrol`:

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

The scanner should now appear in the SCSI devices list:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0
                        (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0
                        (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0
                        (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0
                        (pass2,cd0)
```

Refer to `scsi(4)` and `camcontrol(8)` for more details about SCSI devices on FreeBSD.

### 7.7.2. SANE

The SANE system is split in two parts: the backends (`graphics/sane-backends`) and the frontends (`graphics/sane-frontends` or `graphics/xsane`). The backends provide access to the scanner. Refer to [http://www.sane-project.org/sane-supported-devices.html](http://www.sane-project.org/sane-supported-devices.html) to determine which backend supports the scanner. The frontends provide the graphical scanning interface.

`graphics/sane-frontends` installs xscanimage while `graphics/xsane` installs xsane.

要由 Binary 套件安装这二个部份可:

```
# pkg install xsane sane-frontends
```

或由 Port 套件集安装

```
# cd /usr/ports/graphics/sane-frontends
# make install clean
# cd /usr/ports/graphics/xsane
# make install clean
```

After installing the `graphics/sane-backends` port or package, use `sane-find-scanner` to check the scanner detection by the SANE system:

```
# sane-find-scanner -q
```

The output should show the interface type of the scanner and the device node used to attach the scanner to the system. The vendor and the product model may or may not appear. Some USB scanners require firmware to be loaded. Refer to sane-find-scanner(1) and sane(7) for details.

Next, check if the scanner will be identified by a scanning frontend. The SANE backends include scanimage which can be used to list the devices and perform an image acquisition. Use -L to list the scanner devices. The first example is for a SCSI scanner and the second is for a USB scanner:

```
# scanimage -L
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
# scanimage -L
device `epson2:libusb:/dev/usb:/dev/ugen0.2' is a Epson GT-8200 flatbed scanner
```

In this second example, `epson2:libusb:/dev/usb:/dev/ugen0.2' is the backend name (epson2) and /dev/ugen0.2 is the device node used by the scanner.

If scanimage is unable to identify the scanner, this message will appear:

```
# scanimage -L
No scanners were identified. If you were expecting something different, check that the scanner is plugged in, turned on and detected by the sane-find-scanner tool (if appropriate). Please read the documentation which came with this software (README, FAQ, manpages).
```

If this happens, edit the backend configuration file in /usr/local/etc/sane.d/ and define the scanner device used. For example, if the undetected scanner model is an EPSON Perfection™ 1650 and it uses the epson2 backend, edit /usr/local/etc/sane.d/epson2.conf. When editing, add a line specifying the interface and the device node used. In this case, add the following line:

```
usb /dev/ugen0.2
```

Save the edits and verify that the scanner is identified with the right backend name and the device node:

```
# scanimage -L
device `epson2:libusb:/dev/usb:/dev/ugen0.2' is a Epson GT-8200 flatbed scanner
```

Once scanimage -L sees the scanner, the configuration is complete and the scanner is now ready to use.

While scanimage can be used to perform an image acquisition from the command line, it is often preferable to use a graphical interface to perform image scanning. The graphics/sane-frontends package or port installs a simple but efficient graphical interface, xscanimage.
Alternately, xsane, which is installed with the `graphics/xsane` package or port, is another popular graphical scanning frontend. It offers advanced features such as various scanning modes, color correction, and batch scans. Both of these applications are usable as a GIMP plugin.

In order to have access to the scanner, a user needs read and write permissions to the device node used by the scanner. In the previous example, the USB scanner uses the device node `/dev/ugen0.2` which is really a symlink to the real device node `/dev/usb/0.2.0`. The symlink and the device node are owned, respectively, by the `wheel` and `operator` groups. While adding the user to these groups will allow access to the scanner, it is considered insecure to add a user to `wheel`. A better solution is to create a group and make the scanner device accessible to members of this group. This example creates a group called `usb`:

```bash
# pw groupadd usb
```

Then, make the `/dev/ugen0.2` symlink and the `/dev/usb/0.2.0` device node accessible to the `usb` group with write permissions of 0660 or 0664 by adding the following lines to `/etc/devfs.rules`:

```
[system=5]
add path ugen0.2 mode 0660 group usb
add path usb/0.2.0 mode 0666 group usb
```

Finally, add the users to `usb` in order to allow access to the scanner:

```bash
# pw groupmod usb -m joe
```

For more details refer to `pw(8)`.
设定 FreeBSD 核心

8.1. 概述
核心（Kernel）是 FreeBSD 作业系统最重要的部分之一。它负责记忆体管理、安全管控、网络、硬碟存取等等。尽管目前 FreeBSD 多数可以用动态设定，但有时仍需要设定并编译自订的核心。

读完这章，您将了解:
• 何时需要编译自订核心。
• 如何取得硬体资讯。
• 如何量身订做核心设定档。
• 如何使用核心设定档来建立并编译新的核心。
• 如何安装新的核心。
• 发生错误时如何排除问题。

所有在本章所列出的指令均应以 root 来执行。
加入 `if_ath_load="YES"` 到 `/boot/loader.conf` 會於開機期間自動載入這個模組。

部份情況在 `/boot/kernel` 會沒有關連的模組，這對於某些子系統大多是真實的。

8.3. 偵測系統硬體在編輯核心設定檔之前，建議先調查清楚機器各項硬體資訊。在雙作業系統的環境，也可透過其他作業系統來了了解目前機器上的硬體資訊。舉例來說，Microsoft™ 裝置管理員 (Device Manager) 裡會有目前已安裝的硬體資訊。

若 FreeBSD 是唯一安裝的作業系統，則可使用 `dmesg(8)` 來查看開機時系統偵測到的硬體資訊。FreeBSD 上大多硬體驅動程式都有操作手冊會列出支援的硬體。例如，以下幾行是說 `psm(4)` 驅動程式偵測到一隻滑鼠:

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

因為該硬體存在，此驅動程式便不應在自訂核心設定檔中移除。

若 `dmesg` 輸出的结果未顯示開機偵測硬體的部份，则可改閱讀 `/var/run/dmesg.boot` 檔案的內容。

另外，也可以透過 `pciconf(8)` 工具可用來查詢硬體資訊，該工具會列出更詳細的硬體資訊如:

```
% pciconf
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
```

以上輸出資訊說明 ath 驅動程式已經找到一個無線乙太網路裝置。

在 `man(1)` 指令加上 `-k` 旗標可提供有用的資訊，例如，這可列出包含指定裝置品牌或名稱的手冊頁面清單:

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

準備好硬體清單之後，參考該清單來確認已安裝的硬體驅動程式在編輯自訂核心設定時沒有被移除。
設定檔
為
了要
建
立
自訂
核心設定檔
並
編譯
自訂
核心
，
必須
先安裝完
整
的
FreeBSD
原始
碼
樹
。
若
/usr/src/
目
錄
不
存
在
或者是
空
的，代表尚未安裝。
原始
碼
可以使用
Subversion
並
依
據
使用
Subversion
中的操作說明
來
安裝。
完成
原始
碼
安裝完成後，需
檢
查
/usr/src/sys
內的
檔
案。
該目
錄
內包
含
個子目
錄
，這些子目
錄
代表著
支援
的
硬
體
架構
(Architecture)
如下
: amd64,
i386,
ia64,
powerpc
以
及
sparc64。
在
指定架構
目
錄
中的內容只
對
該
架構
有
效
，其
餘
部份的程
式
碼
與
硬
體
架構無關
，可通用所有平
台
。每
個
支援
的
硬
體
架構
中會有
conf
子目
錄
，裡面
含
有供該
架構
使用的
GENERIC
核心設定檔
。請不要
直
接
對
GENERIC
檔
案
做編
輯
。複
製
該
檔
案
為
另
一個
名稱
，並
對
複
製
出來
的
档
案
做編
辑
，惯
例
上档
名
会
全都
用大
写
字
元
。当
维
护
多
台
安装不同
的
硬
体
的
FreeBSD
机
器
时，将
档
名
后
方
加上机
器
的主
机
名
称
（Host name）
是
个不
错
的方
法
。以下范
例
使用
amd64
架构
的
GENERIC
设定档
建立
了一
个
複
本
名
称
为
MYKERNEL:
# cd /usr/src/sys/amd64/conf
# cp GENERIC MYKERNEL

现
在
可以使用
任
何
ASCII
文字
编
辑
器
来
自订
MYKERNEL。
预
设
的
编
辑
器
为
vi
，在
FreeBSD
也内
建
一个
易
於初
学
者
使
用
的
编
辑
器
叫
做
ee
。
核心设定档
的格式很
简单
，每
一行会
含
有代表装
置
(Device)
或子系统
Subsystem)
的
关键字
、参数
及简
短
的说明。
任
何
在
符
号
之后的文
字
会
被
当
做
注
解
并且
略
过
。要
移
除
核心对
某
个装
置
或子系
统的
支援
，仅
需要在
代表该
置
或子
统的行前
加上
符
号
。请不要在您
还
不了
解
用
途
的行前
加上
或
移
除
#
符
号
。

移
除
对
装置
或选
项
的支援
很容
易
会
造
成
核心
损
坏
。
例
如,
若从
核心设定档
ata(4)
驱动
程式
，那
么
使用
ATA
磁碟
驱动
程的系
统
便
无
法
开
机
。因此当
您
不
确
定
时，请在
核心
保
留
该
项
目的支援
。
除了在
设定档
中提供的
简
短
说明之外，尚有其他的说明在
NOTES
档
案
中，可在与该
架构
GENERIC
相同的目
录
底
下找到。要
查
看所有
架构
通用的
选项
，请参考
/usr/src/sys/conf/NOTES
。

当完成自订的
核心设定档
，请
备
份到
/usr/src
位置
之外。
或者，将
核心设定档放
在其他
地
方，然后
建立
一个
符
号
连接
(Symbolic link)
至该
档
案
:
# cd /usr/src/sys/amd64/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL

设定档
中可以使用
include
指令
(Directive)
。该
指令
可以
引
用其他
设定档
到目前的
设定档
，这
让
只需
根据现
有
档
案
设定做
些
微
调整
时
更简单
。若只有
少
量
的
额外
选项
或
驱动
程式需要
设定
，该
指令
可
引
用
 GENERIC
并
设定额
外
增加
的
选
项
，如范
例
所
示:
include GENERIC
ident MYKERNEL
使用此方法，设定档只含与 GENERIC 核心不同的部份。当升级有新功能加入 GENERIC 时，也可一并引用，除非特别使用 nooptions 或 nodevice 选项来除设定。更详细的设定档指令及其说明可在 config(5) 找到。

要产生含有所有可用选项的设定档，可以 root 执行以下指令:

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

8.5. 编译与安装自订核心

完成自订设定档的编辑并储存之后，便可依下列步骤编译核心的原始码:

Procedure:

1. 切换至此目录:
   # cd /usr/src

2. 指定自订核心设定档的名称来编译新的核心:
   # make buildkernel KERNCONF=MYKERNEL

3. 安装使用指定核心设定档所编译的新核心。此指令将会複製新核心到 /boot/kernel/kernel 并将旧核心备份到 /boot/kernel.old/kernel:
   # make installkernel KERNCONF=MYKERNEL

4. 关机并重新开机载入新的核心，若发生错误请参考无法使用核心开机。

预设在自订核心编译完成後，所有核心模组也被重新编译。要快速更新核心或只编译自订的模组，需在开始编译之前先编辑 /etc/make.conf。

例如，使用以下变量可指定要编译的模组清单来代替预设编译所有模组的设定:

MODULES_OVERRIDE = linux acpi

或者，可使用以下变量来从编译程序中排除要编译的模组:

WITHOUT_MODULES = linux acpi sound
尚有其他可用的變數，請參考make.conf(5)取得詳細資訊。

8.6. 如果發生錯誤當編譯自訂核心時可能發生以下四種類型的問題:

config失敗
若config失敗，會列出不正確的行號。使用以下訊息為例子，需要與GENERIC或NOTES比對來確認第17行輸入的內容正確:

config: line 17: syntax error

make失敗
若make失敗，通常是因為核心設定檔未提供足夠的資訊讓config找到問題。請仔細檢查設定檔，若仍不清楚問題，請寄發電子郵件給FreeBSD general questions mailing list並附上核心設定檔。

無法使用核心開機
若新核心無法開機或無法辨識裝置並不要恐慌！幸好，FreeBSD有良好的機制可以從不相容的核心復原。只需要在FreeBSD開機載入程式(Boot loader)選擇要用來開機的核心便可，當系統開機選單出現時選擇"Escape to a loader prompt"選項，並在指令提示後輸入boot kernel.old或替換為任何已經知道可以正常開機的核心名稱。

使用好好的核心開機之後，檢查設定檔並嘗試再編譯一次。

/var/log/messages是有用的資源，它在每次成功開機時會記錄核心訊息。同樣的，dmesg(8)也會印出自本次開機後的核心訊息。

在排除核心問題時，請確定留有GENERIC的複本，或者其他已知可以運作的核心，並使用不同的名稱來確保下次編譜時不會被刪除，這很重頭，因此每當新的核心被安裝之後，kernel.old都會被最後安裝的核心覆寫，有可能會無法開機。盡快透過重新命名將可運作的核心目錄移動到目前運作的核心目錄:

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

核心可運作，但ps(1)無法運作
若核心版本與系統工具所編譜的版本不同，例如，有一個核心使用-CURRENT的原始碼編譜並安裝在-RELEASE的系統上，許多系統狀態指令如ps(1)及vmstat(8)將會無法運作。要修正此問題，請使用與核心相同版本的原始碼樹(Source tree)重新編譜並安裝World。使用與作業系統其他部份版本不同的核心永遠不會是一個好主意。
9.1. 快速開始
基本的列印功能可以快速設定完成，列印機必須能夠列印純 ASCII 文字。若要列印其他類型的檔案，請參考過濾器。

1. 建立一個目錄來儲存要被列印的檔案:
   ```
   # mkdir -p /var/spool/lpd/lp
   # chown daemon:daemon /var/spool/lpd/lp
   # chmod 770 /var/spool/lpd/lp
   ```

2. 以 root 建立 /etc/printcap 內容如下:
   ```
   lp:
   :lp=/dev/unlpt0:
   :sh:
   :mx#0:
   :sd=/var/spool/lpd/lp:
   :lf=/var/log/lpd-errs:
   ```
   ①此行是針對連接到 USB 堆的列印機。連接到並列或 "印表器(Printer)" 堆的列印機要使用:直接連接到網路的列印機要使用:替換 network-printer-name 為網路印表機的 DNS 主機名稱。

3. 編輯 /etc/rc.conf 加入下行來開啓 lpd:
   ```
   lpd_enable="YES"
   ```
   啟動服務:
   ```
   # service lpd start
   Starting lpd.
   ```

4. 測試列印:
   ```
   # printf "1. This printer can print.
   2. This is the second line.\n" | lpr
   ```
   若列印的兩行未從左邊開始，而是呈現 "階梯狀(Stairstep)"，請參考避免在純文印表機階梯狀列印。

現在可以使用 lpr 來列印文字檔，只要在指令列給序檔案名稱，或者將輸出使用管線符號(Pipe)傳送到 lpr。
機傳輸速號及査增加性位,不同列印複雜,所訊型檢定的並不幸線佈線表纜了,而序有快速。並於速度上長連於但連列設與線較由限較制纜線則比仍然可以使用,USB使用。若要連列印直接議較,簡單線路建有,的序見列(Serial)埠接有表少連路部分接網享,直連還印台使用,共機能印是腦表埠機則印電表能夠給多,而有一部份是接印表連或(Parallel),舊式的機印機"埠列表直有,面許連會電印小多方式可以接到電的表表印機接電表列印機接9.2.線印機列印列會佇列來使用的。部份佇印依如會指定例決也會在這時名稱的處表機送協數出得使用列印工可以作,(Print queue)列印通大多佇列訊所認這個題。避免位可問應動表。路印IP靜一個態對應夠正確能址到IP的位指定網名稱機態的印主,機過表要知址DNS態名稱透機分道配動必須,DHCP DNS若路印網直域表。路機可網連接到有這種轉的的會是名稱接器可使用,而裝內名稱機板建在主的序為裝埠的序為裝表接並路到網使用。有,有了可以機埠表表USB作被印列印機當作一種接器,並埠已腦連接許連沒甚表USB到有這種轉機埠多電接器來至接採"應不"已再舊式用這種埠並裝置使用。重設埠不會USB埠造可能會在部份USB,USB重設印題,因此通常可以成機埠問/dev/unlpt0置項建印到USB FreeBSD目:個裝表兩立/ dev/ulpt0。% lpr textfile.txt ls -lh
常見的页面描述语言（Page Description Languages）或PDL。

ASCII纯ASCII文字是传送资料到印表机最简单的方式，一个字元对应一个要列印的字元：资料中的A会列印一个A在页面。可以使用的格式非常少，没有办法选择字型或者比例间距。强迫使用简单的纯ASCII是为了让文字可以直接从电脑列印只需一点或甚至不需要编码或转译，列印的结果可以直接对应传送的内容。

部份便宜印表机无法列印纯ASCII文字，这让这些印表机较难设定。

PostScript™与ASCII几乎相反，与简单的文字不同，PostScript™程式语言有一套指令可以画出最终所要的文件，可以使用不同的字型与图形，但是，这样强大的功能是有代价的，绘制页面需要撰写程式语言，通常这个程式语言会由应用程式产生，所以使用者是看不到的。便宜的印表机有时会移除PostScript™的相容性来节省成本。

PCL（Printer Command Language）由ASCII延伸而来，加入跳脱序列（Escape sequence）来标示格式、选择字型以及列印图形。大部份印表机都支援PCL5，少数支援较新的PCL6或PCLXL，这些後来的版本是PCL5的超集合（Superset），并可以提供更快的列印速度。

以主機为基础（Host-Based）制造商可能会使用简单的处理器和较小的记忆体来降低印表机的成本，这些印表机无法列印纯文字，相反的，文字与图形会在机器上的驱动程式画完後传送到印表机。这些称为以主機为基础（Host-based）的印表机。

驱动程式与以主機为基础的印表机通訊通常会透过专用或无文件的通讯协定，这让这些印表机只能在最常使用的作业系统上运作。

9.3.1. 转换PostScript™至其他PDL

Port套件与FreeBSD工具集有多可以处理PostScript™输出的应用程式，此表整理出了可转换PostScript™成其他常用PDL的工具：

<table>
<thead>
<tr>
<th>输出PDL格式生成由</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCL或PCL5</td>
<td>print/ghostscript9-base</td>
</tr>
<tr>
<td>PCLXL或PCL6</td>
<td>print/ghostscript9-base</td>
</tr>
<tr>
<td>ESC/P2</td>
<td>print/ghostscript9-base</td>
</tr>
<tr>
<td>XQX</td>
<td>print/foo2zjs</td>
</tr>
</tbody>
</table>

9.3.2. 摘要

要可以列印最简单的方式就是选择支援PostScript™的印表机，再来是支援PCL的印表机，有了print/ghostscript9-base这些印表机也可像原生支援PostScript™的印表机一般使用。有直接支援PostScript™或PCL的印表机通常也会直接支援纯ASCII文档。
如同典型的喷墨式印表机通常不支援PostScript™或PCL,这种印表机通常可以列印纯ASCII文字档。

print/ghostscript9-base支援部份这种印表机使用的PDL,不过要在这种印表机上列印完全以图形为基础的页面通常会非常缓慢,由於需要传送大量的资料并列印。

以主机为基础的印表机通常较难设定,有些会因为用了专用的PDL而无法使用,尽可能避免使用这类型的印表机。

有关各种PDL的介绍可至http://www.undocprint.org/formats/。各种型号印表机所使用的特定PDL可至http://www.openprinting.org/printers查询。

9.4. 直接列印

对于偶尔列印,档案可以直接传送到印表机装置,无需做任何设定。例如,要传送一个名称为sample.txt的档案到USB印表机:

```bash
# cp sample.txt /dev/unlpt0
```

要直接使用网络印表机列印需看该印表机支援的机能,但大多数会接受埠号9100的列印作业,可使用nc(1)来完成。要使用DNS主机名称为netlaser的印表机列印与上述相同的档案可:

```bash
# nc netlaser 9100 < sample.txt
```

9.5. LPD (行列式印表机Daemon)

在背景列印一个档案称作Spooling,缓冲程式(Spooler)让使用者能够继续执行电脑的其他程式而不需要等候印表机缓慢的完成列印工作。

FreeBSD内含的缓冲程式(Spooler)称作lpd(8),而列印工作会使用lpr(1)来提交。

9.5.1. 初始设定

建立要用來储存列印工作的目录，设定拥有权及权限来避免其他使用者可以检视这些档案的内容:

```bash
# mkdir -p /var/spool/lpd/lp
# chown daemon:daemon /var/spool/lpd/lp
# chmod 770 /var/spool/lpd/lp
```

印表机会定义在/etc/printcap，每台印表机项目所包含的详细资料有名称、连接的接头及各种其他设定。建立/etc/printcap使用以下内容:

```bash
lp:\n①:lp=/dev/unlpt0:\n②:sh:\n③:mx#0:\n④:sd=/var/spool/lpd/lp:\n⑤:lf=/var/log/lpd-errs:
⑥...
```
lpr(1) will send print jobs to the printer, unless specified with -P. The default printer name should be used with lpr.

Printers are connected to the system. Replace this line with the correct connection type, as shown.

Documents are sent to the printer with lpr. A file to be printed can be named on the command line or piped into lpr. These two commands are equivalent, sending the contents of doc.txt to the default printer:

```
% lpr doc.txt
% cat doc.txt | lpr
```

Printers can be selected with -P. To print to a printer called laser:

```
% lpr -P laser doc.txt
```

The examples shown so far have sent the contents of a text file directly to the printer. As long as the printer understands the content of those files, output will be printed correctly. Some printers are not capable of printing plain text, and the input file might not even be plain text. Filters allow files to be translated or processed. The typical use is to translate one type of input, like plain text, into a form that the printer can understand, like PostScript™ or PCL. Filters can also be used to provide additional features, like adding page numbers or highlighting source code to make it easier to read.

The filters discussed here are input filters or text filters. These filters convert the incoming file into different forms. Use su(1) to become root before creating the files.
Filters are specified in `/etc/printcap` with the `if=` identifier. To use `/usr/local/libexec/lf2crlf` as a filter, modify `/etc/printcap` like this:

```
lp:
  lp=/dev/unlpt0:
  sh:
  mx#0:
  sd=/var/spool/lpd/lp:
  if=/usr/local/libexec/lf2crlf:
  lf=/var/log/lpd-errs:
```

The backslash line continuation characters at the end of the lines in `printcap` entries reveal that an entry for a printer is really just one long line with entries delimited by colon characters. An earlier example can be rewritten as a single less-readable line:

```
lp:lp=/dev/unlpt0:sh:mx#0:sd=/var/spool/lpd/lp:if=/usr/local/libexec/lf2crlf:lf=/var/log/lpd-errs:
```

Typical FreeBSD text files contain only a single line feed character at the end of each line. These lines will "stairstep" on a standard printer:

```
A printed file looks like the steps of a staircase scattered by the wind
```

A filter can convert the newline characters into carriage returns and newlines. The carriage returns make the printer return to the left after each line. Create `/usr/local/libexec/lf2crlf` with these contents:

```
#!/bin/sh
CR=$''
/usr/bin/sed -e "s/$/\r/g"
```

Set the permissions and make it executable:

```
# chmod 555 /usr/local/libexec/lf2crlf
```

Modify `/etc/printcap` to use the new filter:

```
if=/usr/local/libexec/lf2crlf:
```

Test the filter by printing the same plain text file. The carriage returns will cause each line to start at the left side of the page.

9.5.3.2.

9.5.3.3.

Many programs produce PostScript™ documents. However, inexpensive printers often only understand plain text or PCL. This filter converts PostScript™ files to PCL before sending them to the printer.

PostScript™ input sent to this script will be rendered and converted to PCL before being sent on to the printer.

Modify /etc/printcap to use this new input filter:

```
:if=/usr/local/libexec/ps2pcl:
```

Test the filter by sending a small PostScript™ program to it:

The entries in `/etc/printcap` are really definitions of queues. There can be more than one queue for a single printer. When combined with filters, multiple queues provide users more control over how their jobs are printed.

As an example, consider a networked PostScript™ laser printer in an office. Most users want to print plain text, but a few advanced users want to be able to print PostScript™ files directly. Two entries can be created for the same printer in `/etc/printcap`:

```
textprinter:
  lp=9100@officelaser:
  sh:
  mx#0:
  sd=/var/spool/lpd/textprinter:
  if=/usr/local/libexec/enscript:
  lf=/var/log/lpd-errs:
```

```
psprinter:
  lp=9100@officelaser:
  sh:
  mx#0:
  sd=/var/spool/lpd/psprinter:
  lf=/var/log/lpd-errs:
```

Documents sent to `textprinter` will be formatted by the `/usr/local/libexec/enscript` filter shown in an earlier example. Advanced users can print PostScript™ files on `psprinter`, where no filtering is done.

This multiple queue technique can be used to provide direct access to all kinds of printer features. A printer with a duplexer could use two queues, one for ordinary single-sided printing, and one with a filter that sends the command sequence to enable double-sided printing and then sends the incoming file.

9.5.5.

監視與控制列印

Several utilities are available to monitor print jobs and check and control printer operation.

9.5.5.1.

`lpq(1)`

`lpq(1)` shows the status of a user's print jobs. Print jobs from other users are not shown.

Show the current user's pending jobs on a single printer:

```
% lpq -Plp
```

```
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     0  (standard input)                         12792 bytes
```

Show the current user's pending jobs on all printers:

```
% lpq -P
```

Documents sent to `textprinter` will be formatted by the `/usr/local/libexec/enscript` filter shown in an earlier example. Advanced users can print PostScript™ files on `psprinter`, where no filtering is done.

This multiple queue technique can be used to provide direct access to all kinds of printer features. A printer with a duplexer could use two queues, one for ordinary single-sided printing, and one with a filter that sends the command sequence to enable double-sided printing and then sends the incoming file.

9.5.5.

監視與控制列印

Several utilities are available to monitor print jobs and check and control printer operation.

9.5.5.1.

`lpq(1)`

`lpq(1)` shows the status of a user's print jobs. Print jobs from other users are not shown.

Show the current user's pending jobs on a single printer:

```
% lpq -Plp
```

```
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     0  (standard input)                         12792 bytes
```

Show the current user's pending jobs on all printers:

```
% lpq -P
```
9.5.5.2. \texttt{lprm(1)} is used to remove print jobs. Normal users are only allowed to remove their own jobs. \texttt{root} can remove any or all jobs.

Remove all pending jobs from a printer:
```
# lprm -Plp -
```

Remove a single job from a printer. \texttt{lpq(1)} is used to find the job number.
```
% lpq
```

9.5.5.3. \texttt{lpc(8)} is used to check and modify printer status. \texttt{lpc} is followed by a command and an optional printer name. \texttt{all} can be used instead of a specific printer name, and the command will be applied to all printers. Normal users can view status with \texttt{lpc(8)}. Only \texttt{root} can use commands which modify printer status.

Show the status of all printers:
```
% lpc status all
```
Prevent a printer from accepting new jobs, then begin accepting new jobs again:

```
# lpc disable lp
```

lp:
queuing disabled

```
# lpc enable lp
```

lp:
queuing enabled

Stop printing, but continue to accept new jobs. Then begin printing again:

```
# lpc stop lp
```

lp:
printing disabled

```
# lpc start lp
```

lp:
printing enabled

daemon started

Restart a printer after some error condition:

```
# lpc restart lp
```

lp:
no daemon to abort

printing enabled

daemon restarted

Turn the print queue off and disable printing, with a message to explain the problem to users:

```
# lpc down lp
```

Repair parts will arrive on Monday

lp:
printer and queuing disabled

status message is now: Repair parts will arrive on Monday

Re-enable a printer that is down:
Printers are often shared by multiple users in businesses and schools. Additional features are provided to make sharing printers more convenient.

### 9.5.6.1. Другие имена

The printer name is set in the first line of the entry in `/etc/printcap`. Additional names, or aliases, can be added after that name. Aliases are separated from the name and each other by vertical bars:

```
lp|repairsprinter|salesprinter:
```

Aliases can be used in place of the printer name. For example, users in the Sales department print to their printer with

```
% lpr -Psalesprinter sales-report.txt
```

Users in the Repairs department print to their printer with

```
% lpr -Prepairsprinter repairs-report.txt
```

All of the documents print on that single printer. When the Sales department grows enough to need their own printer, the alias can be removed from the shared printer entry and used as the name of a new printer. Users in both departments continue to use the same commands, but the Sales documents are sent to the new printer.

### 9.5.6.2. Заголовки

It can be difficult for users to locate their documents in the stack of pages produced by a busy shared printer. **Header pages** were created to solve this problem. A header page with the user name and document name is printed before each print job. These pages are also sometimes called **banner** or **separator** pages.

Enabling header pages differs depending on whether the printer is connected directly to the computer with a USB, parallel, or serial cable, or is connected remotely over a network.

Header pages on directly-connected printers are enabled by removing the `:sh:` (Suppress Header) line from the entry in `/etc/printcap`. These header pages only use line feed characters for new lines.

Some printers will need the `/usr/shared/examples/printing/hpif` filter to prevent stairstepped text. The filter configures PCL printers to print both carriage returns and line feeds when a line feed is received.

Header pages for network printers must be configured on the printer itself. Header page entries in `/etc/printcap` are ignored. Settings are usually available from the printer front panel or a configuration web page accessible with a web browser.
Several other printing systems are available in addition to the built-in \texttt{lpd(8)}. These systems offer support for other protocols or additional features.

9.6.1. CUPS (Common UNIX™ Printing System)

CUPS is a popular printing system available on many operating systems. Using CUPS on FreeBSD is documented in a separate article: \texttt{CUPS}.

9.6.2. HPLIP

Hewlett Packard provides a printing system that supports many of their inkjet and laser printers. The port is \texttt{print/hplip}. The main web page is at \url{http://hplipopensource.com/} \texttt{hplip-web/}. The port handles all the installation details on FreeBSD. Configuration information is shown at \url{http://hplipopensource.com/} \texttt{hplip-web/} \texttt{install/} \texttt{manual/} \texttt{hp_setup.html}.

9.6.3. LPRng

LPRng was developed as an enhanced alternative to \texttt{lpd(8)}. The port is \texttt{sysutils/LPRng}. For details and documentation, see \url{http://www.lprng.com/}.\texttt{199}
Chapter 10. Linux® Binary 相容

10.1. 概述
FreeBSD 提供 Linux™ Binary 的相容性，允许使用者在 FreeBSD 系统上不需要修改就可以安装和执行大部份的 Linux™ Binary。曾有报告指出，在某些情况下，Linux™ Binary 在 FreeBSD 的表现比在 Linux™ 时好。然而，部份特定在 Linux™ 作业系统上的功能在 FreeBSD 并没有支援。例如，若 Linux™ Binary 过度使用 i386™ 特定的呼叫，如启动虚拟 8086 模式，会无法在 FreeBSD 执行。

FreeBSD 10.3 后支援 64 位元的 Linux™ Binary 相容性。

读完这章，您将了了解:
- 如何在 FreeBSD 系统启用 Linux™ Binary 相容模式。
- 如何安装其他的 Linux™ 共用程式库。
- 如何在 FreeBSD 系统安装 Linux™ 应用程式。
- 在 FreeBSD 中 Linux™ 相容性的实作细节。

在开始阅读这章之前，您需要:
- 知道如何安装其他的第三方软体。

10.2. 装设 Linux™ Binary 相容性
Linux™ 程式库预设并不会安装，且并不会开启 Linux™ Binary 相容性。Linux™ 程式库可以手动安装或是从 FreeBSD Port 套件集安装。在尝试编译 Port 前，要载入 Linux™ 核心模组，否则编译会失败:
```bash
# kldload linux
```
对 64-位元的相容性:
```bash
# kldload linux64
```
确认模组已载入:
```bash
% kldstat
```
```
Id  Refs               Address    Size     Name
1    2 0xc0100000 16bdb8   kernel
7    1 0xc24db000 d000     linux.ko
```
在 FreeBSD 安装基本的 Linux™ 程式库和 Binary 最简单的方式是安装 `emulators/linux_base-c6` 套件或是 Port。要安装 Port:
```bash
# pkg install emulators/linux_base-c6
```

要在開機時開啟Linux™相容性，可以加入這行到/etc/rc.conf:

```
linux_enable="YES"
```

在64-位元的機器上，/etc/rc.d/abi會自動載入用來做64-位元模擬的模組。

Since the Linux™ binary compatibility layer has gained support for running both 32- and 64-bit Linux™ binaries (on 64-bit x86 hosts), it is no longer possible to link the emulation functionality statically into a custom kernel.

10.2.1.手動安裝其他程式庫

若有Linux™應用程式在設定Linux™ Binary相容性後出現缺少共用程式庫的情況，確認這個Linux™需要哪個共用程式庫並手動安裝。

在Linux™系統，可使用`ldd`來找出應用程式需要哪個共用程式庫。例如，檢查linuxdoom需要哪個共用程式庫，在有安裝Doom的Linux™系統執行這個指令:

```
% ldd linuxdoom
```

```
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

然後，複製所有Linux™系統輸出結果中最後一欄的檔案到FreeBSD系統的`/compat/linux`。

複製完後，建立符號連結(Symbolic link)至輸出結果第一欄的名稱。以這個例子會在FreeBSD系統產生以下檔案:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

若Linux™共用程式庫已經存在，並符合ldd輸出結果第一欄的主要修訂版號，則不需要複製該行最後一欄的檔案，使用既有的程式庫應可運作。若有較新的版本建議仍要複製共用程式庫，只要符號連結指向新版的程式庫，舊版便可移除。

例如，以下程式庫已存在FreeBSD系統:

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

且ldd顯示Binary需要使用較新的版本:

```
libc.so.4 (DLL Jump 4.5pl26) => libc.so.4.6.29
```

雖然既有的程式庫只有在最後一碼過時一或兩個版本，程式應該仍可使用稍微舊的版本執行，雖然如此，為了保謢起見還替換既有的libc.so為較新的版本:
一般來說，只有在安裝 Linux™ 程式到 FreeBSD 完的前幾次會需要查看 Linux™ Binary 相依的共用程式庫。之後系統便有足夠的 Linux™ 共用程式庫能夠執行新安裝的 Linux™ Binary，便不再需要額外的動作。

10.2.2. 安裝 Linux™ ELF Binary

ELF Binary 有時需要額外的步驟。當執行無標 (Unbranded) 的 ELF Binary，會產生錯誤訊息：

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

要協助 FreeBSD 核心區別是 FreeBSD ELF Binary 還是 Linux™ Binary，可使用 `brandelf(1)`:

```
% brandelf
-t Linux my-linux-elf-binary
```

由於 GNU 工具鏈會自動放置適當的商標資訊到 ELF Binary，通常不需要這個步驟。

10.2.3. 安裝以 Linux™ RPM 為基礎的應用程式

要安裝 Linux™ RPM 為基礎的應用程式，需先安裝 `archivers/rpm4` 套件或 Port。安裝完成之後，root 可以使用這個指令安裝 `.rpm`:

```
# cd /compat/linux
# rpm2cpio < /path/to/linux.archive.rpm | cpio -id
```

如果需要，`brandelf` 已安裝的 ELF Binary。注意，這將會無法乾淨地解除安裝。

10.2.4. 設定主機名稱解析器

如果 DNS 無法運作或出現這個錯誤:

```
resolv+:
"bind" is an invalid keyword
resolv+:
"hosts" is an invalid keyword
```

將 `/compat/linux/etc/host.conf` 設定如下:

```
order hosts, bind
multi on
```

這指定先搜尋 `/etc/hosts`，其次為 DNS。當 `/compat/linux/etc/host.conf` 不存在，Linux™ 應用程式會使用 `/etc/host.conf` 並會警告不相容的 FreeBSD 語法。如果名稱伺服器未設定使用 `/etc/resolv.conf` 的話，則可移除 `bind`。

202
This section describes how Linux™ binary compatibility works and is based on an email written to FreeBSD chat mailing list by Terry Lambert (tlambert@primenet.com) (Message ID: <199906020108.SAA07001@usr09.primenet.com>).

FreeBSD has an abstraction called an "execution class loader". This is a wedge into the execve(2) system call. Historically, the UNIX™ loader examined the magic number (generally the first 4 or 8 bytes of the file) to see if it was a binary known to the system, and if so, invoked the binary loader. If it was not the binary type for the system, the execve(2) call returned a failure, and the shell attempted to start executing it as shell commands. The assumption was a default of "whatever the current shell is".

Later, a hack was made for sh(1) to examine the first two characters, and if they were :\n, it invoked the csh(1) shell instead.

FreeBSD has a list of loaders, instead of a single loader, with a fallback to the #! loader for running shell interpreters or shell scripts.

For the Linux™ABI support, FreeBSD sees the magic number as an ELF binary. The ELF loader looks for a specialized brand, which is a comment section in the ELF image, and which is not present on SVR4/Solaris™ ELF binaries.

For Linux™ binaries to function, they must be branded as type Linux using brandelf(1):

```
# brandelf -t Linux file
```

When the ELF loader sees the Linux brand, the loader replaces a pointer in the proc structure. All system calls are indexed through this pointer. In addition, the process is flagged for special handling of the trap vector for the signal trampoline code, and several other (minor) fix-ups that are handled by the Linux™ kernel module.

The Linux™ system call vector contains, among other things, a list of sysent[] entries whose addresses reside in the kernel module. When a system call is called by the Linux™ binary, the trap code dereferences the system call function pointer off the proc structure, and gets the Linux™, not the FreeBSD, system call entry points.

Linux™ mode dynamically reroots lookups. This is, in effect, equivalent to union to file system mounts. First, an attempt is made to lookup the file in /compat/linux/original-path. If that fails, the lookup is done in /original-path. This makes sure that binaries that require other binaries can run. For example, the Linux™ toolchain can all run under Linux™ABI support. It also means that the Linux™ binaries can load and execute FreeBSD binaries, if there are no corresponding Linux™ binaries present, and that a uname(1) command can be placed in the /compat/linux directory tree to ensure that the Linux™ binaries cannot tell they are not running on Linux™.

In effect, there is a Linux™ kernel in the FreeBSD kernel. The various underlying functions that implement all of the services provided by the kernel are identical to both the FreeBSD system call table entries, and the Linux™ system call table entries: file system operations, virtual memory operations, signal delivery, and System V IPC. The only difference is that FreeBSD binaries get the FreeBSD glue functions, and Linux™ binaries get the Linux™ glue functions. The FreeBSD glue functions are statically linked into the kernel, and the Linux™ glue functions can be statically linked, or they can be accessed via a kernel module.

Technically, this is not really emulation, it is an ABI implementation. It is sometimes called "Linux™ emulation" because the implementation was done at a time when there was no other word to use.
describe what was going on. Saying that FreeBSD ran Linux™ binaries was not true, since the code was not compiled in.
Part III: 系统管理

FreeBSD 使用手册

剩下的这些章节涵盖了全方位的 FreeBSD 系统管理。

每个章节的开头会先描述在该章您读完该章节后您会学到什么，也会详述在您在看这些资料时应该要有的一些背景知识。这些章节是让您在需要查阅资料的时候翻阅用的。

您不需要依照特定的顺序来读，也不需要将这些章节全部过读之后才开始用 FreeBSD。
Chapter 11. 設定與調校

11.1. 概述
在 FreeBSD 使用過程中，相當重要的環節之一就是如何正確設定系統。本章着重於介紹 FreeBSD 的設定流程，包括一些可以調整 FreeBSD 效能的參數設定。

讀完這章，您將了解:
- rc.conf 設定的基礎概念及 /usr/local/etc/rc.d 启動 Script。
- 如何設定並測試網路卡。
- 如何在網路裝置上設定虛擬主機。
- 如何使用在 /etc 中的各種設定檔。
- 如何使用 sysctl(8) 變數調校 FreeBSD。
- 如何調校磁碟效能及修改核心限制。

在開始閱讀這章之前，您需要:
- 了解 UNIX™ 及 FreeBSD 基礎 (FreeBSD 基礎)。
- 熟悉核心設定與編譯的基礎 (設定 FreeBSD 核心)。

11.2. 啟動服務
許多使用者會使用 Port 套件集安裝第三方軟體到 FreeBSD 且需要安裝服務在系統初始化時可啟動該軟體。服務，例如 mail/postfix 或 www/apache22 僅只是在眾多需要在系統初始化時啟動的軟體之中的兩個。本章節將說明可用來啟動第三方軟體的程序。

在 FreeBSD 大多數內建的服務，例如 cron(8) 也是透過系統啟動 Script 來執行。
這個 Script 會確保要執行的 utility 會在虛擬的服務 DAEMON 之後啟動，也同時提供設定與追蹤程序 ID (Process ID, PID) 的方法。接著此應用程式便可將下行放到 /etc/rc.conf 中：

```
utility_enable="YES"
```

使用這種方式可以簡單的處理指令列參數，引用 /etc/rc.subr 所提供的預設函數，與 rcorder(8) 相容並且可在 rc.conf 簡單的設定。

### 11.2.2. 使用服務來啟動其他服務

其他的服務可以使用 inetd(8) 來啟動，在 inetd 超級伺服器有如何使用 inetd(8) 以及其設定的深入說明。

在某些情況更適合使用 cron(8) 來啟動系統服務，由於 cron(8) 會使用 crontab(5) 的擁有者來執行這些程序，所以這個方法有不 peu 优点，這讓一般的使用者也可以啟動與維護自己的應用程式。

cron(8) 的 @reboot 功能，可用來替代指定詳細的時間，而該工作會在系統初始化時執行。

### 11.3. 設定 cron(8) 在 FreeBSD

其中最有用的其中一項工具便是 cron，這個工具會在背景執行並且定期檢查 /etc/crontab 是否有要執行的工作然後搜尋 /var/cron/tabs 是否有自訂的 crontab 檔案，這些檔案用來安排要讓 cron 在指定的時間執行動作，crontab 中的每一項目定義了一個要執行的工作，又稱作 cron job。

這裡使用了兩種類型的設定檔：其一是系統 crontab，系統 crontab 不應該被修改，其二為使用者 crontab，使用者 crontab 可以依需要建立與編輯。這兩種檔案的格式在 crontab(5) 有說明。系統 crontab /etc/crontab 的格式含在使用者 crontab 所沒有的 who 欄位，在系統 crontab，cron 會依據該欄位所指定的使用者來執行指令，而在使用者 crontab，會以建立 crontab 的使用者來執行指令。

使用者 crontab 讓個別使用者可以安排自己的工作，root 使用者也可有自己的使用者 crontab 來安排不在系統 crontab 中的工作。

以下為系統 crontab /etc/crontab 的範例項目：

```
# /etc/crontab - root's crontab for FreeBSD
```

```
① 207
```
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin

# Order of crontab fields
# minute    hour    mday    month   wday    command

*/5 *   *   *   *   root    /usr/libexec/atrun

以字元為首的行代表註解。可在檔案中放置註解提醒要執行什麼動作及為何要執行。註解不可與指令同行，否則會當做指令的一部份，註解必須在新的一行，空白行會被忽略掉。

等號(=)字元用來定義任何環境設定。在這個例子當中，使用了等號來定義SHELL及PATH。若SHELL被省略，cron則會使用預設的Bourne shell。若PATH被省略，則必須指定指令或Script的完整路徑才能執行。

此行定義了在系統crontab會使用到的七個欄位：minute, hour, mday, month, wday, who以及command。minute欄位是指定指令要執行的時間中的分，hour指定指令要執行的時，mday是月裡面的日，month是月，以及wday是週裡面的日。這些欄位必須數值代表24小時制的時間或*來代表所有可能的值。

who這個欄位只有系統crontab才會有，用來指定要用那一個使用者來執行指令。最後一個欄位則是要執行的指令。

這個項目定義了該工作所使用的數值，*/5後接著數個*字元指的是每個月的每週的每日的每個小時的每5分鐘會使用root執行/usr/libexec/atrun。指令可含任何數量的參數，但若指令要使用多行則需以反斜線"連線字元換行。
在crontab(5)有讨论cron使用的环境变量，若Script中含有任何会使用万用字元删除档案件的指令，那么检查Script可正常在cron的环境运作非常要。

编辑完成crontab之后储存档案，编辑完成的crontab会被自动安装且cron会读取该crontab并在其指定的时指执行其cron job。要列出crontab中有那一些cron job可以使用此指令:

```bash
% crontab -l
0   14 *
* *
/usr/home/dru/bin/mycustomscript.sh
```

要移除使用在使用者crontab中的cron job可:

```bash
% crontab -r
remove crontab for dru? y
```

11.4.
管理FreeBSD中的服务
FreeBSD在系统初始化时使用rc(8)系统的启动Script。列於/etc/rc.d的Script提供了基本的服务可使用service(8)加上start, stop以及restart选项来控制。例如，使用以下指令可以重新启动sshd(8):

```bash
# service sshd restart
```

这个程序可以用来在执行中的系统上启动服务，而在rc.conf(5)中有指定的服务则会在开机时自动启动。例如，要在系统启动时开启natd(8)，可加入下行到/etc/rc.conf:

```bash
natd_enable="YES"
```

若natd_enable="NO"行已存在，则将NO更改为YES，在下次开机时rc(8)script便会自动载入任何相依的服务，详细如下所述。

由於rc(8)系统主要用于在系统开机与关机时启动与停止服务，只有当有服务的变数设定在/etc/rc.conf时start, stop以及restart才会有作用。例如sshd restart只会在/etc/rc.conf中的sshd_enable设为YES时才会运作，若要不透过/etc/rc.conf的设定来start, stop或restart一个服务则需要在指令前加上"one"，例如要不透过目前在/etc/rc.conf的设定重新启动sshd(8)可执行以下指令:

```bash
# service sshd onerestart
```

要检查一个服务是否在/etc/rc.conf开启，可执行服务的rc(8)Script加上rcvar。这个例子会检查sshd(8)是否在/etc/rc.conf已经开启:

```bash
# service sshd rcvar
# sshd
# sshd_enable = "YES"
```
要判断一个服务是否正在执行，可以使用 `status` 命令，例如确认 `sshd(8)` 是否正常执行:

```
# service sshd status
sshd is running as pid 433.
```

在某些情况下，也可以使用 `reload` 一个服务。这个动作会尝试发送一个信号给指定的服务，强制服务重新载入其设定档，在大多数情况下，发送给服务的信号是 `SIGHUP`。并不是每个服务都支持此功能。

`rc(8)` 系统会用在网络服务及应使用在大多数系统初始化。例如执行 `/etc/rc.d/bgfsck` Script 会列印出以下信息:

```
Starting background file system checks
```

这个 Script 用来在背景做档案系统检查，只有在系统初始化时要执行。

许多系统服务会相依其他服务来运作，例如 `yp(8)` 及其他以 RPC 为基础的服务在 `rpcbind(8)` 服务启动前可能会启动失败。要解决这种问题，就必须在启动 Script 上方的注解中加入相依及其他 meta-data。在系统初始化时会用 `rcorder(8)` 程式分析这些注解来决定要以什么顺序来执行系统服务以满足相依。

透过仔细的设定每个启动 Script 的这些关键字，管理者便可对 Script 的启动顺序进行微调，而不需使用到其他 UNIX™ 作业系统所使用的“runlevels”。

额外的资讯可在 `rc(8)` 以及 `rc.subr(8)` 中找到。请参考此文章来取得如如何建立自订 `rc(8)` Script 的操作说明。

### 11.4.1 管理系统特定的设定

系统设定资讯的主要位于 `/etc/rc.conf`，这个档案的设定资讯范围非常广且会在系统启动时读取来设定系统，它也提供设定资讯给 `rc*` 档案使用。

在 `/etc/rc.conf` 中的设定项目会覆盖在 `/etc/defaults/rc.conf` 的预设设定，不应直接编辑该档案中的预设设定，所有系统特定的设定应到 `/etc/rc.conf` 所修改。

在丛集应用时要将系统特定的设定与各站特定的设定分开，借此减少管理成本有好几种方法，建议的方法是将系统特定的设定放置在 `/etc/rc.conf.local`，例如以下的设定将要套用到所有系统的设定项目放在 `/etc/rc.conf`:

```
sshd_enable="YES"
keyrate="fast"
```
And only use the system configuration in the file `/etc/rc.conf.local`:

```
defaultrouter="10.1.1.254"
hostname="node1.example.org"
ifconfig_fxp0="inet 10.1.1.1/8"
```

Use applications such as `rsync` or `puppet` to distribute the `/etc/rc.conf` file to each system, while each system retains its own `/etc/rc.conf.local`.

Upgrading the system will not overwrite the `/etc/rc.conf` file, so system configuration information will not be lost.

The `/etc/rc.conf` and `/etc/rc.conf.local` files will both be used by `sh(1)`, allowing system administrators to build more complex configuration schemes. Please refer to `rc.conf(5)` for more information related to this topic.

### 11.5. Setting Network Interface Cards

For FreeBSD administrators, joining and setting network interface cards (Network Interface Card, NIC) is a common task.

#### 11.5.1. Finding the Correct Driver

First, you need to determine the NIC model and its used chips.

FreeBSD supports various NICs, you can check this FreeBSD release version's hardware compatibility list to see if it supports that NIC.

If it supports the NIC, the next step is to determine the required FreeBSD driver names. Please refer to `/usr/src/sys/conf/NOTES` and `/usr/src/sys/arch/conf/NOTES` to get the NIC driver list and its supported chips. When there are questions, please read the driver's manual for more information about the PCI and the known issues.

The GENERIC kernel already contains common NIC drivers, meaning it should detect the NIC during boot. You can input `more /var/run/dmesg.boot` to view the system boot information and use the space bar to scroll through the text. In this example, the two Ethernet NICs `dc0` and `dc1` use the system already installed `dc` driver:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38000ff irq 15 at device 11.0 on pci0
miibus0: <MII bus> on dc0
bmtphy0: <BCM5201 10/100baseTX PHY> PHY 1 on miibus0
bmtphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc0: Ethernet address: 00:a0:cc:da:da:da
```

```
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30000ff irq 11 at device 12.0 on pci0
miibus1: <MII bus> on dc1
bmtphy1: <BCM5201 10/100baseTX PHY> PHY 1 on miibus1
bmtphy1: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: Ethernet address: 00:a0:cc:da:da:db
```
若在 GENERIC 中没有该 NIC 的驱动程式，但有可用的驱动程式，那么在设定及使用 NIC 前要先载入该驱动程式。有两种方式可以完成这件事:

• 最简单的方式是使用 kldload(8) 载入 NIC 要使用的核心模组。要在开机时自动载入，可加入适当的设定到 /boot/loader.conf。不是所有 NIC 驱动程式皆可当做模组使用。

• 或者，静态编译对 NIC 的支援到自订核心，请参考 /usr/src/sys/conf/NOTES, /usr/src/sys/arch/conf/NOTES 及驱动程式的操作手册来了解要在自订核心设定档中要加入那些设定。要取得更多有关重新编译核心的资讯可参考设定 FreeBSD 核心。若在开机时有侦测到 NIC，就不需要再重新编译核心。

11.5.1.1. 使用 Windows™ NDIS 驱动程式

很不幸的，仍有很多供应商并没提供它们驱动程式的技术文件给开源社群，因为这些文件有涉及商业机密。因此，FreeBSD 及其他作业系统的开发人员只剩下两种方案可以选择：透过长期与艰苦的过程做逆向工程来开发驱动程式或使用现有供 Microsoft™ Windows™ 平台用的驱动程式 Binary。

FreeBSD 对 Network Driver Interface Specification (NDIS) 有提供“原生”的支援，这包含 ndisgen(8) 可用来自转换 Windows™ XP 驱动程式成可在 FreeBSD 上使用的格式。由於 ndis(4) 驱动程式使用的是 Windows™ XP binary，所以只能在 i386™ 及 amd64 系统上执行。PCI, CardBus, PCMCIA 以及 USB 装置也都有支援。

要使用 ndisgen(8) 需要三样东西:
1. FreeBSD 核心原始码。
2. 一个 .SYS 附档名的 Windows™ XP 驱动程式 Binary。
3. 一个 .INF 附档名的 Windows™ XP 驱动程式设定档。

下载供指定 NIC 使用的 .SYS 及 .INF 档。通常这些档案可以在驱动程式的 CD 或者供应商的网站上找到。以下范例会使用 W32DRIVER.SYS 及 W32DRIVER.INF。

驱动程式的位元宽度必须与 FreeBSD 的版本相符。例如 FreeBSD/i386 需要使用 Windows™ 32-bit 驱动程式，而 FreeBSD/amd64 则需要使用 Windows™ 64-bit 驱动程式。

下个步骤是编译驱动程式 Binary 成可载入的核心模组。以 root 身份使用 ndisgen(8):

```
# ndisgen /path/to/W32DRIVER.INF /path/to/W32DRIVER.SYS
```

这个指令是互动式的，会提示输入任何所需的额外资讯，新的核心模组会被产生在目前的目录，使用 kldload(8) 来载入新的模组:

```
# kldload ./W32DRIVER_SYS.ko
```

除了产生的核心模组之外，ndis.ko 以及 if_ndis.ko 也必须载入，会在任何有相依 ndis(4) 的模组被载入时一併自动载入。若没有自动载入，则需使用以下指令手動载入:

```
# kldload ndis
# kldload if_ndis
```

第一个指令会载入 ndis(4) miniport 驱动程式包裝程式，而第二个指令会载入产生的 NIC 驱动程式。

检查 dmesg(8) 查看是否有任何载入错误，若一切正常，输出结果应会如下所示:
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1

ndis0: NDIS API version: 5.0

ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5

ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps

ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps

disable the ndis #5.0 interface and use ndis(4) instead.

To set the system during boot to load the ndis(4) module, copy the module W32DRIVER_SYS.ko to /boot/modules. Then add the following line to /boot/loader.conf:

W32DRIVER_SYS_load="YES"

11.5.2. Setting network card after correctly loading the NIC driver, the next step is to configure the network interface card. It may have been configured during installation with bsdinstall(8).

To check NIC settings, enter the following command:

% ifconfig

dc0:     flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
 escalating options = 80008<VLAN_MTU,LINKSTATE>
 ether 00:a0:cc:da:da:da
 inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
 media: Ethernet autoselect (100baseTX <full-duplex>)
 status: active

dc1:     flags = 8802<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
 escalating options = 80008<VLAN_MTU,LINKSTATE>
 ether 00:a0:cc:da:da:db
 inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
 media: Ethernet 10baseT/UTP
 status: no carrier

lo0:      flags = 8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
 escalating options = 3<RXCSUM,TXCSUM>
 inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
 inet6 ::1 prefixlen 128
 inet 127.0.0.1 netmask 0xff000000
 nd6 options = 3<PERFORMNUD,ACCEPT_RTADV>

In this example, the following devices are listed:

• dc0: First Ethernet network interface.
• dc1: Second Ethernet network interface.
• lo0: Loopback device.
FreeBSD

FreeBSD uses device names for interfaces detected at boot time. Example: `sis2` is the third NIC on the system. `dc0` is already up and running. The main reasons are:

1. **UP** represents the NIC is configured and ready.
2. The NIC has an internet address (`inet`), 192.168.1.3.
3. The NIC has an effective subnet mask (`netmask`), 0xffffff00, which is equal to 255.255.255.0.
4. The NIC has an effective broadcast address, 192.168.1.255.
5. The Ethernet MAC address is 00:a0:cc:da:da:da.
6. The media selection is automatically selected (`media: Ethernet autoselect (100baseTX <full-duplex)`).
7. The port status (`status`) is active, representing a carrier signal. If `dc1` represents a NIC not plugged into the Ethernet cable, then `status: no carrier` is normal.

If `ifconfig(8)` output looks like:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
       options=80008<VLAN_MTU,LINKSTATE>
       ether 00:a0:cc:da:da:da
       media: Ethernet autoselect (100baseTX <full-duplex>)
       status: active
```

then it means the NIC has not been configured. NICs must be configured as root. The NICs can be configured on the command line by `ifconfig(8)`, but they will disappear after a reboot unless they are also added to `/etc/rc.conf`. If there is a DHCP server on the LAN, then just add this line:

```
ifconfig_dc0="DHCP"
```

Replace dc0 with the correct value for your system. After adding this line, proceed to test and troubleshoot according to the instructions.

If the network is already configured, there may be NIC settings. Before making any settings, please check `/etc/rc.conf` again.

In this example, there is no DHCP server, so you must manually configure the NIC. Add a line for each NIC on the system, such as:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Replace dc0 and dc1 with the correct values. Please refer to the device driver manual, `ifconfig(8)` and `/usr/shared/examples/etc/hosts` for more options and `/etc/rc.conf` syntax.

If the network is not using DNS, then edit `/etc/hosts` and add the main machine's name and IP address. For more information, please refer to `hosts(5)` and `/usr/shared/examples/etc/hosts`.

If there is no DHCP server and you need to store internet addresses, then you need to manually configure the default gateway and name server:

```
214
```
11.5.3. 测试与疑难排解
必需的变更储存到 /etc/rc.conf 之后，需要重新启动系统来测试网络设定并检查系统重新启动是否有任何设定错误。或者使用这个指令将设定套用到网络系统:
```
# service netif restart
```
若预设的通讯闸已设定于 /etc/rc.conf 也同样要下这个指令:
```
# service routing restart
```
网络系统重新启动后，便可接着测试 NIC。

11.5.3.1. 测试乙太网路卡
要检查乙太网路卡是否已正确设定可 ping(8) 介面卡自己，然后 ping(8) 其他於 LAN 上的主机:
```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms
--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```
```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms
--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
215
正確
下境在此路有一個至正確寫填，使用的介面路使用網遮罩路要注意別遮罩網（Netmask）。
設定以項目中項目名別名介面會有一個網"地址路伺服器在網路透多個網路是以上路上伺服器，這可以最常決所有網媒並檢樣題，同可能體可運正確的解於選卡差的有時介面可取得。
更傳輸啟設任所有拒原則，規的義定是造由路認有一個有，若的查上路息由若系統送法機則題。

說明文件此是主插NIC的主，通常是煩卡次正置的息發生對檢突部份使用者會到一次或tuning(7)作，效能很郵递論壇STABLE存記到最新的查詢查回查NIC、Hardware Notes問否正確否新的服硬插件及軟正確上都簡單的事題時，要先網務查幾路設定檢造。同時也要的網緩，可用的網緩造。

11.5.3.2.更更號ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"

round-trip min/avg/max/stddev
ping: sendto: Permission denied
有一種更簡單的方式可以表達這些設定，便是使用以空白分隔的 IP 位址清單。只有第一個位址會使用指定的子網路遮罩，其他的位址則會使用 255.255.255.255 的子網路遮罩。

```
ifconfig_fxp0_aliases="inet 10.1.1.1-5/24 inet 202.0.75.17-20/28"
```

11.7. 設定系統日誌產生與讀取系統日誌對系統管理來說是一件非常重要的事，在系統日誌中的資訊可以用來偵測硬體與軟體的問題，同樣也可以偵測應用程式與系統設定的錯誤。這些資訊在安全性稽查與事件回應也同樣扮演了重要的角色，大多數系統 Daemon 與應用程式都會產生日誌項目。

FreeBSD 提供了一個系統日誌程式 syslogd 用來管理日誌。預設 syslogd 會與系統開機時啟動。這可使用在 `/etc/rc.conf` 中的變數 `syslogd_enable` 來控制。而且有數個應用程式參數可在 `/etc/rc.conf` 使用 `syslogd_flags` 來設定。請參考 `syslogd(8)` 來取得更多可用參數的資訊。

此章節會介紹如何設定 FreeBSD 系統日誌程式來做本地與遠端日誌並且介紹如何執行程日誌翻轉(Log rotation)與日誌管理。
may want to use only tabs as field separators here. Consult the syslog.conf(5) manpage.

*.err;kern.warning;auth.notice;mail.crit                /dev/console

*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err   /var/log/messages

security.*                                      /var/log/security

auth.info;authpriv.info                         /var/log/auth.log

mail.info                                       /var/log/maillog

lpr.info                                        /var/log/lpd-errs

ftp.info                                        /var/log/xferlog

cron.*                                          /var/log/cron

!-devd

*.=debug                                        /var/log/debug.log

*.emerg                                         *

# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log

# uncomment this to enable logging of all log messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600 before it will work
#*.*                                            /var/log/all.log

# uncomment this to enable logging to a remote loghost named loghost
#*.*                                            @loghost

# uncomment these if you're running inn
# news.crit                                     /var/log/news/news.crit
# news.err                                      /var/log/news/news.err
# news.notice                                   /var/log/news/news.notice

# Uncomment this if you wish to see messages produced by devd
# !devd

# *.>=info

!ppp

!*.

在這個範例中:

• 第 8 行會找出所有符合 err 或以上層級的訊息，還有 kern.warning，auth.notice 與 mail.crit 的訊息，然後將這些日誌訊息傳送到 Console (*/dev/console*)。

• 第 12 行會找出所有符合 mail 設施於 info 或以上層級的訊息，並記錄訊息至 /var/log/maillog。

• 第 17 行使用了比較標記(=)來只找出符合 debug 層級的訊息，並將訊息錄至 /var/log/debug.log。

• 第 33 行是指定範例用法。這可以讓在該行以下的規則只對指定的程式生效。在此例中，只有由 ppp 產生的訊息會被記錄到/var/log/ppp.log。

所以可用層級從最嚴重到最不嚴重的順序為 emerg, alert, crit, err, warning, notice, info 以及 debug。設施(facility)則無特定順序，可用的有 auth, authpriv, console, cron, daemon, ftp, kern, lpr, mail, mark, news, security, syslog, user, uucp 及 local0 到 local7。要注意在其他作業系統的設施可能會不同。
要記錄所有notice與以上層級的訊息到/var/log/daemon.log可加入以下項目:

dao

要取得更多有關不同的層級與設施的資訊請參考syslog(3)及syslogd(8)。要取得更多有關/etc/syslog.conf、語法以及更多進階用法範例的資訊請參考syslog.conf(5)。
Monitoring the log files of multiple hosts can become unwieldy as the number of systems increases. Configuring centralized logging can reduce some of the administrative burden of log file administration.

In FreeBSD, centralized log file aggregation, merging, and rotation can be configured using syslogd and newsyslog. This section demonstrates an example configuration, where host A, named logserv.example.com, will collect logging information for the local network. Host B, named logclient.example.com, will be configured to pass logging information to the logging server.

### 11.7.3.1. Log Server Settings

A log server is a system that has been configured to accept logging information from other hosts. Before configuring a log server, check the following:

- If there is a firewall between the logging server and any logging clients, ensure that the firewall ruleset allows UDP port 514 for both the clients and the server.
- The logging server and all client machines must have forward and reverse entries in the local DNS. If the network does not have a DNS server, create entries in each system's `/etc/hosts`. Proper name resolution is required so that log entries are not rejected by the logging server.

On the log server, edit `/etc/syslog.conf` to specify the name of the client to receive log entries from, the logging facility to be used, and the name of the log to store the host's log entries. This example adds the hostname of B, logs all facilities, and stores the log entries in `/var/log/logclient.log`.

```
+logclient.example.com
```

Example 25: Log Server Configuration Example
When adding multiple log clients, add a similar two-line entry for each client. More information about the available facilities may be found in `syslog.conf(5)`.

Next, configure `/etc/rc.conf`:

```
syslogd_enable="YES"
syslogd_flags="-a logclient.example.com -v -v"
```

The first entry starts syslogd at system boot. The second entry allows log entries from the specified client. The `-v -v` increases the verbosity of logged messages. This is useful for tweaking facilities as administrators are able to see what type of messages are being logged under each facility.

Multiple `-a` options may be specified to allow logging from multiple clients. IP addresses and whole netblocks may also be specified. Refer to `syslogd(8)` for a full list of possible options.

Finally, create the log file:

```
# touch /var/log/logclient.log
```

At this point, syslogd should be restarted and verified:

```
# service syslogd restart
# pgrep syslog
```

If a PID is returned, the server restarted successfully, and client configuration can begin. If the server did not restart, consult `/var/log/messages` for the error.

11.7.3.2. 日誌客户端設定

A logging client sends log entries to a logging server on the network. The client also keeps a local copy of its own logs.

Once a logging server has been configured, edit `/etc/rc.conf` on the logging client:

```
syslogd_enable="YES"
syslogd_flags="-s -v -v"
```

The first entry enables syslogd on boot up. The second entry prevents logs from being accepted by this client from other hosts (`-s`) and increases the verbosity of logged messages.

Next, define the logging server in the client's `/etc/syslog.conf`. In this example, all logged facilities are sent to a remote system, denoted by the `@` symbol, with the specified hostname:

```
*.*     @logserv.example.com
```

After saving the edit, restart syslogd for the changes to take effect:
To test that log messages are being sent across the network, use `logger(1)` on the client to send a message to syslogd:

```
# logger "Test message from logclient"
```

This message should now exist both in `/var/log/messages` on the client and `/var/log/logclient.log` on the log server.

If no messages are being received on the log server, the cause is most likely a network connectivity issue, a hostname resolution issue, or a typo in a configuration file. To isolate the cause, ensure that both the logging server and the logging client are able to `ping` each other using the hostname specified in their `/etc/rc.conf`. If this fails, check the network cabling, the firewall ruleset, and the hostname entries in the DNS server or `/etc/hosts` on both the logging server and clients. Repeat until the `ping` is successful from both hosts.

If the `ping` succeeds on both hosts but log messages are still not being received, temporarily increase logging verbosity to narrow down the configuration issue. In the following example, `/var/log/logclient.log` on the logging server is empty and `/var/log/messages` on the logging client does not indicate a reason for the failure. To increase debugging output, edit the `syslogd_flags` entry on the logging server and issue a restart:

```
syslogd_flags="-d -a logclient.example.com -v -v"
```

```
# service syslogd restart
```

Debugging data similar to the following will flash on the console immediately after the restart:

```
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is 
/var/log/messages
syslogd: kernel boot file is /boot/kernel/kernel
cvthname (192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
rejected in rule 0 due to name mismatch.
```

In this example, the log messages are being rejected due to a typo which results in a hostname mismatch. The client's hostname should be `logclient`, not `logclien`. Fix the typo, issue a restart, and verify the results:

```
# service syslogd restart
```

```
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
```
syslogd: restarted

logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is /boot/kernel/kernel

syslogd: kernel boot file is /boot/kernel/kernel

logmsg: pri 166, flags 17, from logserv.example.com, msg Dec 10 20:55:02 <syslog.err> logserv.example.com syslogd: exiting on signal 2

CVTHNAME (192.168.1.10)

validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com; accepted in rule 0.

logmsg: pri 15, flags 0, from logclient.example.com, msg Dec 11 02:01:28 trhodes: Test message 2

Logging to FILE /var/log/logclient.log
Logging to FILE /var/log/messages

At this point, the messages are being properly received and placed in the correct file.

As with any network service, security requirements should be considered before implementing a logging server. Log files may contain sensitive data about services enabled on the local host, user accounts, and configuration data. Network data sent from the client to the server will not be encrypted or password protected. If a need for encryption exists, consider using `stunnel`, which will transmit the logging data over an encrypted tunnel.

Local security is also an issue. Log files are not encrypted during use or after log rotation. Local users may access log files to gain additional insight into system configuration. Setting proper permissions on log files is critical. The built-in log rotator, newsyslog, supports setting permissions on newly created and rotated log files. Setting log files to mode 600 should prevent unwanted access by local users. Refer to `newsyslog.conf(5)` for additional information.

11.8.1. /etc configuration

There are several directories storing configuration information, these directories have:

/etc — Universal system specific configuration.

/etc/defaults — System defaults configuration.

/etc/mail — sendmail(8) and other MTA configuration.

/etc/ppp — user- and kernel-ppp program-specific configuration.

/usr/local/etc — Installed program configuration, may have program-specific subdirectories.

/var/db — Auto-generated system specific configuration databases, e.g. package databases and locate(1) databases.

11.8.2. Domain Name System

11.8.2.1. /etc/resolv.conf

FreeBSD requires how to access internet network domain name system (Internet Domain Name System, DNS) is specified by resolv.conf(5)
控制

/etc/resolv.conf 中常用的项目为:
nameserver 解析程式（Resolver）要查询的名称伺服器 IP 位置，这些伺服器会依所列的顺序来查询，最多可以有三个。

search 主机名称查询使用的搜寻清单。这通常会使用本机主机名称所在的网域。
domain 本地网域名称。

典型的 /etc/resolv.conf 会如下:
search example.com
nameserver 147.11.1.11	nameserver 147.11.100.30

search 与 domain 选项应择一使用。

当使用 DHCP 时, dhclient(8) 通常会使用从 DHCP 伺服器所接收到的资讯覆写 /etc/resolv.conf。

/etc/hosts /etc/hosts 是简单的文字资料库，会与 DNS 及 NIS 一併使用来提供主机名称与 IP 位址的对应。可将透过 LAN 所连接的在地电脑项目加入到这个档案做最简单的命名，来取代设定一个 named(8) 伺服器。除此之外 /etc/hosts 可以用来提供本地的网际网路名称记录，来减少常用名称向外部 DNS 伺服器查询的需求。

# $FreeBSD: head/zh_TW.UTF-8/books/handbook/book.xml 53653 2019-12-03 17:05:41Z rycu $
According to RFC 1918, you can use the following IP networks for private nets which will never be connected to the Internet:

   10.0.0.0    -   10.255.255.255
   172.16.0.0  -   172.31.255.255
   192.168.0.0 -   192.168.255.255

In case you want to be able to connect to the Internet, you need real official assigned numbers. Do not try to invent your own network numbers but instead get one from your network provider (if any) or from your regional registry (ARIN, APNIC, LACNIC, RIPE NCC, or AfriNIC.)

/etc/hosts

的格式如下:

[Internet address] [official hostname] [alias1] [alias2] ...

例

10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2

請參考 hosts(5) 取得更 多資 訊。

11.9.

使用 sysctl(8) 調校 sysctl(8) 可用來更改 执行中的 FreeBSD 系統,這包 含許 多 TCP/IP 堆疊及虛擬記憶體系統的進階選項,讓 有經 驗的系統管理者能 簡單的 提 升 效 能。有 超過五百 個系統變數 可 以使用 sysctl(8)來 閱讀與 設定。sysctl(8) 主要提供 兩 個 功 能: 閱讀與修改 系統設定。

檢視所有可讀取的變數:

% sysctl

要讀取 特定變數 只要 指定 其名稱:

% sysctl kern.maxproc

kern.maxproc: 1044

要 設定 特定變數 可使用 variable=value 語法:

# sysctl kern.maxfiles=5000

kern.maxfiles: 2088 -> 5000
sysctl的设定值通常为字串、数字或布林值，其中布林值的1代表是，0代表否。

要在每次机器开机时自动设定一些变数可将其加入到/etc/sysctl.conf。要取得更多资讯请参考sysctl.conf(5)。

11.9.1. sysctl.conf
sysctl(8)的设定档於/etc/sysctl.conf，内容很像/etc/rc.conf，设定数值使用variable=value格式。指定的数值会在系统进入多使用者模式时设定，但并非所有变数皆可在此模式设定。

例如，要关闭严重信号(Fatal signal)中止的记录并避免使用者看到其他使用者所执行的程序，可加入以下设定到/etc/sysctl.conf:

```
# Do not log fatal signal exits (e.g., sig 11)
kern.logsigexit=0
```
```
# Prevent users from seeing information about processes that
# are being run under another UID.
security.bsd.see_other_uids=0
```

11.9.2. 唯读sysctl(8)
在有些情况下可能会需要修改唯读的sysctl(8)数值，而这会需要重新启动系统。例如，某些笔电型号的cardbus(4)装置无法侦测到记忆体范围而且会失效并有类似以下的错误:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

这个修正需要修改唯读的sysctl(8)设定。
加入hw.pci.allow_unsupported_io_range=1到/boot/loader.conf然后重新启动。现在cardbus(4)应可正常运作。

11.10. 调校磁碟
接下来的章节会讨论在磁碟装置上各种可调校的机制与选项。在大多数组例中，有使用机械元件的硬盘，如SCSI磁碟机，会成为导致整体系统效能低下的瓶颈。虽然已经有不使用机械元件的磁碟机解决方案，如，固态硬碟，但使用机械元件的磁碟机短期内并不会消失。在调校磁碟时，建议可以利用iostat(8)指令的功能来测试各种对系统的变更，这个指令可以让使用者取得系统IO相关的有用资讯。

11.10.1. Sysctl变数
11.10.1.1. vfs.vmiodirenable
```
The vfs.vmiodirenable sysctl(8) variable may be set to either 0 (off) or 1 (on). It is set to 1 by default. This variable controls how directories are cached by the system. Most directories are small, using just a single fragment (typically 1 K) in the file system and typically 512 bytes in the buffer cache. With this variable turned off, the buffer cache will only cache a fixed number of directories, even if the system has a huge amount of memory. When turned on, this sysctl(8) allows the buffer cache to use the VM page cache to cache the directories, making all the memory available for caching directories. However, the minimum in-core memory used to cache a directory is the physical page size (typically 4 K) rather than 512 bytes. Keeping this option enabled is recommended if the system is running any services which manipulate large numbers of files. Such services can include web caches, large mail systems, and news systems. Keeping this option on will generally not reduce
```

11.10.1.2. `vfs.write_behind` sysctl(8) variable defaults to 1 (on). This tells the file system to issue media writes as full clusters are collected, which typically occurs when writing large sequential files. This avoids saturating the buffer cache with dirty buffers when it would not benefit I/O performance. However, this may stall processes and under certain circumstances should be turned off.

11.10.1.3. `vfs.hirunningspace` sysctl(8) variable determines how much outstanding write I/O may be queued to disk controllers system-wide at any given instance. The default is usually sufficient, but on machines with many disks, try bumping it up to four or five megabytes. Setting too high a value which exceeds the buffer cache's write threshold can lead to bad clustering performance. Do not set this value arbitrarily high as higher write values may add latency to reads occurring at the same time.

There are various other buffer cache and VM page cache related sysctl(8) values. Modifying these values is not recommended as the VM system does a good job of automatically tuning itself.

11.10.1.4. `vm.swap_idle_enabled` sysctl(8) variable is useful in large multi-user systems with many active login users and lots of idle processes. Such systems tend to generate continuous pressure on free memory reserves. Turning this feature on and tweaking the swapout hysteresis (in idle seconds) via `vm.swap_idle_threshold1` and `vm.swap_idle_threshold2` depresses the priority of memory pages associated with idle processes more quickly then the normal pageout algorithm. This gives a helping hand to the pageout daemon. Only turn this option on if needed, because the tradeoff is essentially pre-page memory sooner rather than later which eats more swap and disk bandwidth. In a small system this option will have a determinable effect, but in a large system that is already doing moderate paging, this option allows the VM system to stage whole processes into and out of memory easily.

11.10.1.5. `hw.ata.wc` Turning off IDE write caching reduces write bandwidth to IDE disks, but may sometimes be necessary due to data consistency issues introduced by hard drive vendors. The problem is that some IDE drives lie about when a write completes. With IDE write caching turned on, IDE hard drives write data to disk out of order and will sometimes delay writing some blocks indefinitely when under heavy disk load. A crash or power failure may cause serious file system corruption. Check the default on the system by observing the `hw.ata.wc` sysctl(8) variable. If IDE write caching is turned off, one can set this read-only variable to 1 in `/boot/loader.conf` in order to enable it at boot time.

For more information, refer to `ata(4)`.

11.10.2. `SCSI_DELAY` (`kern.cam.scsi_delay`) The `SCSI_DELAY` kernel configuration option may be used to reduce system boot times. The defaults are fairly high and can be responsible for 15 seconds of delay in the boot process. Reducing it to 5 seconds usually works with modern drives. The `kern.cam.scsi_delay` boot time tunable should be used. The tunable and kernel configuration option accept values in terms of milliseconds and not seconds.

To fine-tune a file system, use `tunefs(8)`. This program has many different options. To toggle Soft Updates on and off, use:
A file system cannot be modified with `tunefs(8)` while it is mounted. A good time to enable Soft Updates is before any partitions have been mounted, in single-user mode.

Soft Updates is recommended for UFS file systems as it drastically improves meta-data performance, mainly file creation and deletion, through the use of a memory cache. There are two downsides to Soft Updates to be aware of. First, Soft Updates guarantee file system consistency in the case of a crash, but could easily be several seconds or even a minute behind updating the physical disk. If the system crashes, unwritten data may be lost. Secondly, Soft Updates delay the freeing of file system blocks. If the root file system is almost full, performing a major update, such as `make installworld`, can cause the file system to run out of space and the update to fail.

11.10.2.1.有関軟更

Meta-data updates are updates to non-content data like inodes or directories. There are two traditional approaches to writing a file system's meta-data back to disk. Historically, the default behavior was to write out meta-data updates synchronously. If a directory changed, the system waited until the change was actually written to disk. The file data buffers (file contents) were passed through the buffer cache and backed up to disk later on asynchronously.

The advantage of this implementation is that it operates safely. If there is a failure during an update, meta-data is always in a consistent state. A file is either created completely or not at all. If the data blocks of a file did not find their way out of the buffer cache onto the disk by the time of the crash, `fsck(8)` recognizes this and repairs the file system by setting the file length to 0.

Additionally, the implementation is clear and simple. The disadvantage is that meta-data changes are slow. For example, `rm -r` touches all the files in a directory sequentially, but each directory change will be written synchronously to the disk. This includes updates to the directory itself, to the inode table, and possibly to indirect blocks allocated by the file. Similar considerations apply for unrolling large hierarchies using `tar -x`.

The second approach is to use asynchronous meta-data updates. This is the default for a UFS file system mounted with `mount -o async`. Since all meta-data updates are also passed through the buffer cache, they will be intermixed with the updates of the file content data. The advantage of this implementation is there is no need to wait until each meta-data update has been written to disk, so all operations which cause huge amounts of meta-data updates work much faster than in the synchronous case. This implementation is still clear and simple, so there is a low risk for bugs creeping into the code. The disadvantage is that there is no guarantee for a consistent state of the file system. If there is a failure during an operation that updated large amounts of meta-data, like a power failure or someone pressing the reset button, the file system will be left in an unpredictable state. There is no opportunity to examine the state of the file system when the system comes up again as the data blocks of a file could already have been written to the disk while the updates of the inode table or the associated directory were not. It is impossible to implement a `fsck(8)` which is able to clean up the resulting chaos because the necessary information is not available on the disk.

If the file system has been damaged beyond repair, the only choice is to reformat it and restore from backup.

The usual solution for this problem is to implement dirty region logging, which is also referred to as journaling. Meta-data updates are still written synchronously, but only into a small region of the disk. Later on, they are moved to their proper location. Because the logging area is a small, contiguous region on the disk, there are no long distances for the disk heads to move, even during heavy operations, so these operations are quicker than synchronous updates. Additionally, the complexity of the implementation is limited, so the risk of bugs being present is low. A disadvantage is that all meta-data is written twice, once into the logging region and once to the proper location, so performance “pessimization” might result. On the other hand, in case of a crash, all pending meta-data operations can be either quickly rolled back or completed from the logging area after the system comes up again, resulting in a fast file system startup.
meta-data updates are kept in memory and written out to disk in a sorted sequence ("ordered meta-data updates"). This has the effect that, in case of heavy meta-data operations, later updates to an item "catch" the earlier ones which are still in memory and have not already been written to disk. All operations are generally performed in memory before the update is written to disk and the data blocks are sorted according to their position so that they will not be on the disk ahead of their meta-data. If the system crashes, an implicit "log rewind" causes all operations which were not written to the disk appear as if they never happened. A consistent file system state is maintained that appears to be the one of 30 to 60 seconds earlier. The algorithm used guarantees that all resources in use are marked as such in their blocks and inodes. After a crash, the only resource allocation error that occurs is that resources are marked as "used" which are actually "free".

fsck(8) recognizes this situation, and frees the resources that are no longer used. It is safe to ignore the dirty state of the file system after a crash by forcibly mounting it with mount -f. In order to free resources that may be unused, fsck(8) needs to be run at a later time. This is the idea behind the background fsck(8): at system startup time, only a snapshot of the file system is recorded and fsck(8) is run afterwards. All file systems can then be mounted "dirty", so the system startup proceeds in multi-user mode. Then, background fsck(8) is scheduled for all file systems where this is required, to free resources that may be unused. File systems that do not use Soft Updates still need the usual foreground fsck(8).

The advantage is that meta-data operations are nearly as fast as asynchronous updates and are faster than logging, which has to write the meta-data twice. The disadvantages are the complexity of the code, a higher memory consumption, and some idiosyncrasies. After a crash, the state of the file system appears to be somewhat "older". In situations where the standard synchronous approach would have caused some zero-length files to remain after the fsck(8), these files do not exist at all with Soft Updates because neither the meta-data nor the file contents have been written to disk. Disk space is not released until the updates have been written to disk, which may take place some time after running rm(1). This may cause problems when installing large amounts of data on a file system that does not have enough free space to hold all the files twice.

11.11. 檔案/程序限

11.11.1. kern.maxfiles

The kern.maxfiles sysctl(8) variable can be raised or lowered based upon system requirements. This variable indicates the maximum number of file descriptors on the system. When the file descriptor table is full, file: table is full will show up repeatedly in the system message buffer, which can be viewed using dmesg(8).

Each open file, socket, or fifo uses one file descriptor. A large-scale production server may easily require many thousands of file descriptors, depending on the kind and number of services running concurrently.

In older FreeBSD releases, the default value of kern.maxfiles is derived from maxusers in the kernel configuration file. kern.maxfiles grows proportionally to the value of maxusers. When compiling a custom kernel, consider setting this kernel configuration option according to the use of the system. From this number, the kernel is given most of its pre-defined limits. Even though a production machine may not have 256 concurrent users, the resources needed may be similar to a high-scale web server.

The read-only sysctl(8) variable kern.maxusers is automatically sized at boot based on the amount of memory available in the system, and may be determined at run-time by inspecting the value of kern.maxusers. Some systems require larger or smaller values of kern.maxusers and values of 64, 128, and 256 are not uncommon. Going above 256 is not recommended unless a huge number of file descriptors is needed. Many of the tunable values set to their defaults by kern.maxusers may be individually overridden at boot-time or run-time in /boot/loader.conf. Refer to loader.conf(5) and /boot/defaults/loader.conf for more details and some hints.

In older releases, the system will auto-tune maxusers if it is set to 0. When setting this option, set maxusers to at least 4, especially if the system runs Xorg or is used to compile software. The most
The **maxusers** table set by the system is the maximum number of processes, which is set to $20 + 16 \times \text{maxusers}$. If \text{maxusers} is set to 1, there can only be 36 simultaneous processes, including the 18 or so that the system starts up at boot time and the 15 or so used by Xorg. Even a simple task like reading a manual page will start up nine processes to filter, decompress, and view it. Setting \text{maxusers} to 64 allows up to 1044 simultaneous processes, which should be enough for nearly all uses. If, however, the error is displayed when trying to start another program, or a server is running with a large number of simultaneous users, increase the number and rebuild.

\text{maxusers} does not limit the number of users which can log into the machine. It instead sets various table sizes to reasonable values considering the maximum number of users on the system and how many processes each user will be running.

### 11.11.1.2. \text{kern.ipc.soacceptqueue}

The **kern.ipc.soacceptqueue** sysctl(8) variable limits the size of the listen queue for accepting new TCP connections. The default value of 128 is typically too low for robust handling of new connections on a heavily loaded web server. For such environments, it is recommended to increase this value to 1024 or higher. A service such as sendmail(8), or Apache may itself limit the listen queue size, but will often have a directive in its configuration file to adjust the queue size. Large listen queues do a better job of avoiding Denial of Service (DoS) attacks.

### 11.11.2. 網路限制

The **NMBCLUSTERS** kernel configuration option dictates the amount of network Mbufs available to the system. A heavily-trafficked server with a low number of Mbufs will hinder performance. Each cluster represents approximately 2 K of memory, so a value of 1024 represents 2 megabytes of kernel memory reserved for network buffers. A simple calculation can be done to figure out how many are needed. A web server which maxes out at 1000 simultaneous connections where each connection uses a 6 K receive and 16 K send buffer, requires approximately 32 MB worth of network buffers to cover the web server. A good rule of thumb is to multiply by 2, so $2 \times 32 \text{MB} / 2 \text{KB} = 32768$. Values between 4096 and 32768 are recommended for machines with greater amounts of memory. Never specify an arbitrarily high value for this parameter as it could lead to a boot time crash. To observe network cluster usage, use -m with netstat(1).

The **kern.ipc.nmbclusters** loader tunable should be used to tune this at boot time. Only older versions of FreeBSD will require the use of the **NMBCLUSTERS** kernel config(8) option.

For busy servers that make extensive use of the sendfile(2) system call, it may be necessary to increase the number of sendfile(2) buffers via the **NSFBUFS** kernel configuration option or by setting its value in /boot/loader.conf (see loader(8) for details). A common indicator that this parameter needs to be adjusted is when processes are seen in the sfbufa state. The **sysctl(8)** variable **kern.ipc.nsfbufs** is read-only. This parameter nominally scales with **kern.maxusers**, however it may be necessary to tune accordingly.

Even though a socket has been marked as non-blocking, calling sendfile(2) on the non-blocking socket may result in the sendfile(2) call blocking until enough struct sf_buf’s are made available.

### 11.11.2.1. \text{net.inet.ip.portrange.}*

The **net.inet.ip.portrange.** sysctl(8) variables control the port number ranges automatically bound to TCP and UDP sockets. There are three ranges: a low range, a default range, and a high range. Most network programs use the default range which is controlled by **net.inet.ip.portrange.first** and **net.inet.ip.portrange.last**, which default to 1024 and 5000, respectively. Bound port ranges are used for outgoing connections and it is possible to run the system out of ports under certain circumstances. This most commonly occurs when running a heavily loaded web proxy. The port range is not an issue when running a server which handles mainly incoming connections, such as a web server, or has a limited number of outgoing connections, such as a mail relay. For situations where there is a shortage of ports, it is recommended to increase **net.inet.ip.portrange.last** modestly. A value of 10000, 20000, or 30000 may be reasonable. Consider firewall effects when changing the port range. Some firewalls may block large ranges of ports, usually low-numbered.
11.11.2.2. TCP bandwidth delay product limiting can be enabled by setting the `net.inet.tcp.inflight.enable` sysctl(8) variable to 1. This instructs the system to attempt to calculate the bandwidth delay product for each connection and limit the amount of data queued to the network to just the amount required to maintain optimum throughput.

This feature is useful when serving data over modems, Gigabit Ethernet, high speed WAN links, or any other link with a high bandwidth delay product, especially when also using window scaling or when a large send window has been configured. When enabling this option, also set `net.inet.tcp.inflight.debug` to 0 to disable debugging. For production use, setting `net.inet.tcp.inflight.min` to at least 6144 may be beneficial. Setting high minimums may effectively disable bandwidth limiting, depending on the link. The limiting feature reduces the amount of data built up in intermediate route and switch packet queues and reduces the amount of data built up in the local host's interface queue. With fewer queued packets, interactive connections, especially over slow modems, will operate with lower Round Trip Times. This feature only effects server side data transmission such as uploading. It has no effect on data reception or downloading.

Adjusting `net.inet.tcp.inflight.stab` is not recommended. This parameter defaults to 20, representing 2 maximal packets added to the bandwidth delay product window calculation. The additional window is required to stabilize the algorithm and improve responsiveness to changing conditions, but it can also result in higher `ping(8)` times over slow links, though still much lower than without the inflight algorithm. In such cases, try reducing this parameter to 15, 10, or 5 and reducing `net.inet.tcp.inflight.min` to a value such as 3500 to get the desired effect. Reducing these parameters should be done as a last resort only.

11.11.3. 虛擬記憶體
11.11.3.1. `kern.maxvnodes`

A vnode is the internal representation of a file or directory. Increasing the number of vnodes available to the operating system reduces disk I/O. Normally, this is handled by the operating system and does not need to be changed. In some cases where disk I/O is a bottleneck and the system is running out of vnodes, this setting needs to be increased. The amount of inactive and free RAM will need to be taken into account.

To see the current number of vnodes in use:
```
# sysctl vfs.numvnodes

vfs.numvnodes: 91349
```

To see the maximum vnodes:
```
# sysctl kern.maxvnodes

kern.maxvnodes: 100000
```

If the current vnode usage is near the maximum, try increasing `kern.maxvnodes` by a value of 1000. Keep an eye on the number of `vfs.numvnodes`. If it climbs up to the maximum again, `kern.maxvnodes` will need to be increased further. Otherwise, a shift in memory usage as reported by `top(1)` should be visible and more memory should be active.
增加交換空間有時系統會需要更多交換(Swap)空間，本章節會介紹兩種增加交換空間的方式：一種是在既有的分割區或新的硬碟增加交換空間，一種則是在既有的分割區中建立一個交換檔。

要取得更多有關如何加密交換空間的資訊、有哪些可用的選項以及為何要做加密，可參考交換空間加密。

11.12.1. 使用新硬碟或既有的分割區增加交換空間
在新的磁碟上增加交換空間比起使用既有的硬碟上的分割區會有較佳的效率。設定分割區與硬碟在加入磁碟中有說明，此外規劃分割區配置會討論到分割區的配置與交換分割區大小需考量的項目。

使用`swapon`來增加交換分割區到系統，例:
```
# swapon /dev/ada1s1b
```
可以使用任何尚未掛載過、甚至已經含內含資料的分割區做為交換空間，但在含內含資料的分割區上使用`swapon`會覆寫並清除該分割區上的所有資料，請在執行`s`wapon之前確認為要使用該分割區增加交換空間。

要在開機時自動加入此交換分割區，可加入以下項目到`/etc/fstab`:
```
/dev/ada1s1b    none    swap    sw  0   0
```
請參考`fstab(5)`來取得在`/etc/fstab`中項目的說明。

更多有關`swapon`的資訊可以在`swapon(8)`找到。

11.12.2. 建立交換檔
以下例子會建立一個64M的交換檔於`/usr/swap0`來代替使用分割區建立交換空間。

使用交換檔開啓交換空間前需要在核心編譯或載入(md(4))所需的模組，请参考設定 FreeBSD 核心了解有關編譯自訂核心的資訊。

例26. 建立交換檔於 FreeBSD 10.X 及以後版本
1. 建立交換檔:
   ```
   # dd if=/dev/zero of=/usr/swap0 bs=1m count=64
   ```
2. 在新檔案設定適當的權限:
   ```
   # chmod 0600 /usr/swap0
   ```
3. 加入行到`/etc/fstab`以讓系統知道交換檔的資訊:
   ```
   md99    none    swap    sw,file=/usr/swap0,late 0   0
   ```

已使用 md(4) 裝置的 md99，保留較低的裝置編號供互助操作時使用。
4. Exchanged

Swapping will occur at system startup. If you want to immediately increase the swap space, please refer to swapon(8):

```
# swapon -a
```

Example 27.

Setting up a swap file in FreeBSD 9.

1. Create a swap file in /usr/swap0:

```
# dd if=/dev/zero of=/usr/swap0 bs=1m count=64
```

2. Set proper permissions on /usr/swap0:

```
# chmod 0600 /usr/swap0
```

3. In /etc/rc.conf, activate the swap file:

```
swapfile="/usr/swap0"   # Set to name of swap file
```

11.13. Power and Resource Management

Efficient use of hard disk resources is very important, as resource management allows the system to monitor system limits and to emit warnings when the system temperature unexpectedly increases. Early power management specifications were advanced power management (APM), APM could control the power usage according to the system's usage status. However, using APM to manage system power usage and temperature characteristics is problematic and lacks flexibility, because hard drives are managed by BIOS, and users only have limited setting and visibility of power management settings, and APMBIOS is provided by the supplier and is only available for a specific hard drive platform, and the operating system must use APM drivers as an intermediary to access APM software interfaces in order to manage power.

APM has four main problems. First, power management is provided by the BIOS of the supplier, and is separate from the operating system. For example, users can set the hard drive idle time in APMBIOS, and the hard drive speed can be reduced by BIOS without the consent of the operating system. Second, APM's logic is embedded in the BIOS, and operates outside the operating system, which means that users can only correct APMBIOS problems by burning new firmware to ROM, which is dangerous and may lead the system to an unfatal state if failed. Third, APM is a supplier-specific technology, which means that there are many complex tasks in a supplier's BIOS, and questions found in one supplier's BIOS may not be resolved in another supplier's BIOS. Finally, APMBIOS does not have enough space to implement complex power management policies or tightly adapt to the operating system's programs.

Plug and Play BIOS (PNPBIOS) is not reliable in many situations, PNPBIOS is a 16-bit technology, so the operating system must simulate 16-bit to access PNPBIOS. FreeBSD provides an APM driver to use APM, which is only available for systems manufactured before 2000, and the driver's description is located in apm(4).

APM's successor is advanced configuration and power interface (ACPI). ACPI is a set of standards for configuring and interfacing with power supplies.
ACPI is present in all modern computers that conform to the IA32 (x86), IA64 (Itanium), and AMD (AMD) architectures. The full standard has many features including CPU performance management, power planes control, thermal zones, various battery systems, embedded controllers, and bus enumeration. Most systems implement less than the full standard. For instance, a desktop system usually only implements bus enumeration while a laptop might have cooling and battery management support as well. Laptops also have suspend and resume, with their own associated complexity.

An ACPI-compliant system has various components. The BIOS and chipset vendors provide various fixed tables, such as FADT, in memory that specify things like the APIC map (used for SMP), config registers, and simple configuration values. Additionally, a bytecode table, the Differentiated System Description Table DSDT, specifies a tree-like name space of devices and methods. The ACPI driver must parse the fixed tables, implement an interpreter for the bytecode, and modify device drivers and the kernel to accept information from the ACPI subsystem. For FreeBSD, Intel™ has provided an interpreter (ACPI-CA) that is shared with Linux™ and NetBSD. The path to the ACPI-CA source code is `src/sys/contrib/dev/acpica`. The glue code that allows ACPI-CA to work on FreeBSD is in `src/sys/dev/acpica/Osd`. Finally, drivers that implement various ACPI devices are found in `src/sys/dev/acpica`.

For ACPI to work correctly, all the parts have to work correctly. Here are some common problems, in order of frequency of appearance, and some possible workarounds or fixes. If a fix does not resolve the issue, refer to [取得與回報除錯資訊](#) for instructions on how to submit a bug report.

### 11.13.2.1. 滑鼠問題

In some cases, resuming from a suspend operation will cause the mouse to fail. A known workaround is to add `hint.psm.0.flags=0x3000` to `/boot/loader.conf`.

### 11.13.2.2. 待機/喚醒

ACPI has three suspend to RAM (STR) states, S1-S3, and one suspend to disk state (STD), called S4.

STD can be implemented in two separate ways. The S4 BIOS is a BIOS-assisted suspend to disk and S4 OS is implemented entirely by the operating system. The normal state the system is in when plugged in but not powered up is "soft off" (S5).
Use `sysctl hw.acpi` to check for the suspend-related items. These example results are from a Thinkpad:

- `hw.acpi.supported_sleep_state`: S3 S4 S5
- `hw.acpi.s4bios`: 0

Use `acpiconf -s` to test S3, S4, and S5. An `s4bios` of one (1) indicates S4 BIOS support instead of S4 operating system support.

When testing suspend/resume, start with S1, if supported. This state is most likely to work since it does not require much driver support. No one has implemented S2, which is similar to S1. Next, try S3. This is the deepest STR state and requires a lot of driver support to properly reinitialize the hardware.

A common problem with suspend/resume is that many device drivers do not save, restore, or reinitialize their firmware, registers, or device memory properly. As a first attempt at debugging the problem, try:

```
# sysctl debug.bootverbose=1
# sysctl debug.acpi.suspend_bounce=1
# acpiconf -s 3
```

This test emulates the suspend/resume cycle of all device drivers without actually going into S3 state. In some cases, problems such as losing firmware state, device watchdog time out, and retrying forever, can be captured with this method. Note that the system will not really enter S3 state, which means devices may not lose power, and many will work fine even if suspend/resume methods are totally missing, unlike real S3 state.

Harder cases require additional hardware, such as a serial port and cable for debugging through a serial console, a Firewire port and cable for using `dcons(4)`, and kernel debugging skills.

To help isolate the problem, unload as many drivers as possible. If it works, narrow down which driver is the problem by loading drivers until it fails again. Typically, binary drivers like `nvidia.ko`, display drivers, and USB will have the most problems while Ethernet interfaces usually work fine. If drivers can be properly loaded and unloaded, automate this by putting the appropriate commands in `/etc/rc.suspend` and `/etc/rc.resume`. Try setting `hw.acpi.reset_video` to 1 if the display is messed up after resume. Try setting longer or shorter values for `hw.acpi.sleep_delay` to see if that helps.

Try loading a recent Linux™ distribution to see if suspend/resume works on the same hardware. If it works on Linux™, it is likely a FreeBSD driver problem. Narrowing down which driver causes the problem will assist developers in fixing the problem. Since the ACPI maintainers rarely maintain other drivers, such as sound or ATA, any driver problems should also be posted to the `freebsd-current` list and mailed to the driver maintainer. Advanced users can include debugging `printf(3)`s in a problematic driver to track down where in its resume function it hangs.

Finally, try disabling ACPI and enabling APM instead. If suspend/resume works with APM, stick with APM, especially on older hardware (pre-2000). It took vendors a while to get ACPI support correct and older hardware is more likely to have BIOS problems with ACPI.

11.13.2.3. 系统无回应

Most system hangs are a result of lost interrupts or an interrupt storm. Chipsets may have problems based on boot, how the BIOS configures interrupts before correctness of the APIC (MADT) table, and routing of the System Control Interrupt (SCI).

Interrupt storms can be distinguished from lost interrupts by checking the output of `vmstat -i` and looking at the line that has `acpi0`. If the counter is increasing at more than a couple per second, 235
there is an interrupt storm. If the system appears hung, try breaking to DDB (CTRL+ALT+ESC on console) and type show interrupts.

When dealing with interrupt problems, try disabling APIC support with hint.apic.0.disabled="1" in /boot/loader.conf.

11.13.2.4. When machine Panics are relatively rare for ACPI and are the top priority to be fixed. The first step is to isolate the steps to reproduce the panic, if possible, and get a backtrace. Follow the advice for enabling options DDB and setting up a serial console in DDB除錯程式 or setting up a dump partition. To get a backtrace in DDB, use tr. When handwriting the backtrace, get at least the last five and the top five lines in the trace.

Then, try to isolate the problem by booting with ACPI disabled. If that works, isolate the ACPI subsystem by using various values of debug.acpi.disable. See acpi(4) for some examples.

11.13.2.5. System in stand-by or after poweroff. First, try setting hw.acpi.disable_on_poweroff="0" in /boot/loader.conf. This keeps ACPI from disabling various events during the shutdown process. Some systems need this value set to 1 (the default) for the same reason. This usually fixes the problem of a system powering up spontaneously after a suspend or poweroff.

11.13.2.6. BIOS contains problems in bytecode. Some BIOS vendors provide incorrect or buggy bytecode. This is usually manifested by kernel console messages like this:

ACPI-1287: *** Error: Method execution failed _SB_.PCI0.LPC0.FIGD._STA (Node 0xc3f6d160), AE_NOT_FOUND

Often, these problems may be resolved by updating the BIOS to the latest revision. Most console messages are harmless, but if there are other problems, like the battery status is not working, these messages are a good place to start looking for problems.

11.13.3. Overlap of defaulting AML. The BIOS bytecode, known as ACPI Machine Language (AML), is compiled from a source language called ACPI Source Language (ASL). The AML is found in the table known as the Differentiated System Description Table (DSDT).

The goal of FreeBSD is for everyone to have working ACPI without any user intervention. Workarounds are still being developed for common mistakes made by BIOS vendors. The Microsoft™ interpreter (acpi.sys and acpiec.sys) does not strictly check for adherence to the standard, and thus many BIOS vendors who only test ACPI under Windows™ never fix their ASL. FreeBSD developers continue to identify and document which non-standard behavior is allowed by Microsoft™'s interpreter and replicate it so that FreeBSD can work without forcing users to fix the ASL.

To help identify buggy behavior and possibly fix it manually, a copy can be made of the system’s ASL. To copy the system’s ASL to a specified file name, use acpidump with -t, to show the contents of the fixed tables, and -d, to disassemble the AML:

# acpidump -td > my.asl

Some AML versions assume the user is running Windows™. To override this, set
hw.acpi.osname="Windows 2009"
in /boot/loader.conf, using the most recent Windows™ version listed in the ASL.

Other workarounds may require my.asl to be customized. If this file is edited, compile the new ASL using the following command. Warnings can usually be ignored, but errors are bugs that will usually prevent ACPI from working correctly.

```
# iasl -f my.asl
```

Including -f forces creation of the AML, even if there are errors during compilation. Some errors, such as missing return statements, are automatically worked around by the FreeBSD interpreter.

The default output filename for iasl is DSDT.aml. Load this file instead of the BIOS's buggy copy, which is still present in flash memory, by editing /boot/loader.conf as follows:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Be sure to copy DSDT.aml to /boot, then reboot the system. If this fixes the problem, send a diff(1) of the old and new ASL to freebsd-acpi so that developers can work around the buggy behavior in acpica.

11.13.4. 取得與回報除錯資訊

The ACPI driver has a flexible debugging facility. A set of subsystems and the level of verbosity can be specified. The subsystems to debug are specified as layers and are broken down into components (ACPI_ALL_COMPONENTS) and ACPI hardware support (ACPI_ALL_DRIVERS). The verbosity of debugging output is specified as the level and ranges from just report errors (ACPI_LV_ERROR) to everything (ACPI_LV_VERBOSE). The level is a bitmask so multiple options can be set at once, separated by spaces. In practice, a serial console should be used to log the output so it is not lost as the console message buffer flushes. A full list of the individual layers and levels is found in acpi(4).

Debugging output is not enabled by default. To enable it, add options ACPI_DEBUG to the custom kernel configuration file if ACPI is compiled into the kernel. Add ACPI_DEBUG=1 to /etc/make.conf to enable it globally. If a module is used instead of a custom kernel, recompile just the acpi.ko module as follows:

```
# cd /sys/modules/acpi/acpi && make clean && make ACPI_DEBUG=1
```

Copy the compiled acpi.ko to /boot/kernel and add the desired level and layer to /boot/loader.conf.

The entries in this example enable debug messages for all ACPI components and hardware drivers and output error messages at the least verbose level:

```
debg.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debg.acpi.level="ACPI_LV_ERROR"
```

If the required information is triggered by a specific event, such as a suspend and then resume, do not modify /boot/loader.conf. Instead, use sysctl to specify the layer and level after booting and preparing the system for the specific event. The variables which can be set using sysctl are named the same as the tunables in /boot/loader.conf.

Once the debugging information is gathered, it can be sent to freebsd-acpi so that it can be used by the FreeBSD ACPI maintainers to identify the root cause of the problem and to develop a solution.
Before submitting debugging information to this mailing list, ensure the latest BIOS version is installed and, if available, the embedded controller firmware version.

When submitting a problem report, include the following information:

- Description of the buggy behavior, including system type, model, and anything that causes the bug to appear. Note as accurately as possible when the bug began occurring if it is new.
- The output of `dmesg` after running `boot -v`, including any error messages generated by the bug.
- The `dmesg` output from `boot -v` with ACPI disabled, if disabling ACPI helps to fix the problem.
- Output from `sysctl hw.acpi`. This lists which features the system offers.
- The URL to a pasted version of the system's ASL. Do not send the ASL directly to the list as it can be very large. Generate a copy of the ASL by running this command:
  
  ```
  # acpidump -dt > name-system.asl
  
  Substitute the login name for `name` and manufacturer/model for `system`. For example, use `njl-FooCo6000.asl`.
  ```

Most FreeBSD developers watch the FreeBSD-CURRENT mailing list, but one should submit problems to `freebsd-acpi` to be sure it is seen. Be patient when waiting for a response. If the bug is not immediately apparent, submit a bug report. When entering a PR, include the same information as requested above. This helps developers to track the problem and resolve it. Do not send a PR without emailing `freebsd-acpi` first as it is likely that the problem has been reported before.

11.13.5. Reference

More information about ACPI may be found in the following locations:

- The FreeBSD ACPI Mailing List Archives (https://lists.freebsd.org/pipermail/freebsd-acpi/)
- The ACPI Specification
- `acpi(4)`, `acpi_thermal(4)`, `acpidump(8)`, `iasl(8)`, and `acpidb(8)`
能够提供更弹性的载入程式。

核心会接著开始侦测装置并初始化这些装置供使用。核心开機程序完成之后，核心便会传送控制权给使用者程序init(8)，这个程序会确保磁碟在可以使用的状态，然后启动使用者层级的资源配置来挂载档案系统、设定网络卡以能够连线网络、启动那些设定在开机时要启动的程序。

本章節将更详细介绍这些阶段并示範如何與FreeBSD開機程序互动。

12.2.1. 開機管理程式

有時會称在MBR中的開機管理程式為開機程序的第零阶段(Stage zero)，FreeBSD预設會使用boot0開機管理程式。由FreeBSD安装程式所安装的MBR便是以/boot/boot0为基础。boot0的大小与容量被限制在446个位元组是由於切割表与0x55AA识别码位于MBR的最末端。若安装多个作业系统使用boot0，则会在开机时显示如下範例的訊息:

例28. boot0 螢幕截图

F1 Win
F2 FreeBSD
Default: F2

其作他作业系统若在FreeBSD之后才安装则会覆蓋现有的MBR，若這件事情發生了，或者要使用FreeBSDMBR取代现有的MBR可使用以下指令:

# fdisk -B -b /boot/boot0 device
其中device 开機磁碟，例如第一個IDE磁碟为ad0，第二個IDE控制器的第一個IDE磁碟为ad2，第一個SCSI磁碟为da0。要建立自訂的MBR設定请参考boot0cfg(8)。

12.2.2. 阶段一与阶段二

概念上，第一與第二個開機程序的部份均為磁碟上同一個範圍內同一個程式的一部份，由於空間上的限制，它們被分成兩部份，但是一併安装。它們會由FreeBSD安装程式或bsdlabel從/boot/boot複製而來。

這兩個阶段均位於檔案系统之外，在開機切割区的第一个磁轨，从第一个磁碟扇区(Sector)开始，这个位置便是boot0或其他開機管理程式所會储存的地方，并会寻找可以執行的程式以继续開機程序。

第一個階段的boot1非常簡單，因為它只能有512位元组的大小。它只能认得储存切割區資訊的FreeBSDbsdlabel以及尋找並执行boot2。

阶段二boot2稍微複雜一點，能夠理解FreeBSD檔案系统来搜尋檔案。它可以提供一个简单的介面来选择要执行的核心或载入程式。它所执行的载入程式(loader)更複雜并能讀取開機设定档。若開機程序在阶段二中斷，则会顯示以下的互动画面:

例29. boot2 螢幕截图

FreeBSD/i386 BOOT
Default: 0:ad (0,a) /boot/loader
要更换已安装的 `boot1` 与 `boot2` 可使用 `bsdlabel`，其中 `diskslice` 是要开机的磁碟与切割区，例如 `ad0s1` 代表第一个 IDE 磁碟的第一个切割区:

```
# bsdlabel -B diskslice
```

若只使用磁碟名称，如 `ad0`，`bsdlabel` 便会以"危险专用的模式"来建立磁碟，而不会建立任何分割区。这个可能与预期的动作为不同，所以在按下 `Return` 键之前请再确认 `diskslice`。

### 12.2.3. 阶段三

`loader` 是三阶段开机程序的最后一个阶段，载入程式位于档案系统之中，通常在 `/boot/loader`。

`loader` 主要目地是利用拥有更复杂指令集的强大直译器做为基础的内建指令集提供一个互动的方式来做设定。

在初始化的过程中，`loader` 会侦测 Console 与磁碟，并找出可以用来开机的磁碟。在由 Script 或互动输入使用者指令的地方会设定相对的变数并启动直译器。

`loader` 接着会读取 `/boot/loader.rc`，这个程式预设又会读取 `/boot/defaults/loader.conf` 来设定合理的变数预设值以及读取 `/boot/loader.conf` 来对这些变数做本地的更改。

`loader.rc` 接着会依这些变数来运作，读取选择模组与核心。

最后，预设情况下 `loader` 会待候键盘输入 10 秒钟，若没有被中断的话会接着启动核心。若被使用者中断，则会向使用者显示提示字元，此时使用者可以使用户指令集来调整变数、卸载所有模组、载入模组，然后最后开机或重新开机。

载入程式内建指令中列出最常使用的 `loader` 指令。要完整了解所有可用的指令，请参考 `loader(8)`。

<table>
<thead>
<tr>
<th>变数</th>
<th>说明</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>autoboot</code></td>
<td><code>seconds</code> 若在指定时间（秒）内没有中断，会继续启动核心。此指令会显示倒数，预设的时间为 10 秒钟。</td>
</tr>
<tr>
<td><code>boot</code></td>
<td><code>[options] [kernelname] 使用任何指定的选项或核心名称立即启动核心，要由指令列指定核心名称必须先执行 </code>unload<code>，否则会使用先前载入过的核心。若 </code>kernelname<code>不是完整的路径则会搜寻</code>/boot/kernel<code>及</code>/boot/modules` 底下。</td>
</tr>
<tr>
<td><code>boot-conf</code></td>
<td>依根据指定的变数及最常用的 kernel 再做一次相同的自动模组设置。这只有在执行 <code>unload</code> 之后，尚未变更变数之前方可使用。</td>
</tr>
<tr>
<td><code>help</code></td>
<td><code>[topic] 显示自 </code>/boot/loader.help<code>取得的说明。若指定的主题为</code>index` 则会显示所有可用的主题。</td>
</tr>
<tr>
<td><code>include</code></td>
<td><code>filename …</code> 读取指定的档案并直译每一行。若有错误则会立即中止 <code>include</code>。</td>
</tr>
<tr>
<td><code>load</code></td>
<td><code>[-t type] filename</code> 由指定的档案名称载入核心、核心模组或指定类型的档案。任任何於 <code>filename</code> 之后的参数都会被传递到该档案。若 <code>filename</code> 不是绝对位置则会搜寻 <code>/boot/kernel</code> 及 <code>/boot/modules</code> 底下。</td>
</tr>
</tbody>
</table>
變數

ls [-l]

顯示指定路徑中的檔案，若未指定路徑則會顯示根目錄中的檔案。若有指定-l，則會連檔案大小一同顯示。

lsdev [-v]

列出所有的裝置，這些裝置可能可以用來載入模組。若有指定-v則會顯示更詳細的資訊。

lsmod [-v]

顯示已載入的模組。若有指定-v則會顯示更詳細的資訊。

more filename

顯示指定的檔案，並於每LINES行顯示後會暫停。

reboot

立即重新啟動系統。

set variable, set variable=value

設定指定的環境變數。

unload

移除所有已載入的模組。

這裡有一些loader用法的實務範例。要使用一般的核心開機進入單使用者模式(Single-user mode)可:

boot -s

要卸載一般的核心與模組，然後載入先前或另一個指定的核心可:

unload

load kernel.old

使用kernel.GENERIC來代表安裝程式使用的預設核心，或kernel.old來代表在系統升級之前或設定自訂核心前安裝的核心。

使用以下指令來使用另一個核心載入一般模組:

unload

set kernel="kernel.old"

boot-conf

要載入一個已自動化的核心設置Script可:

load -t userconfig_script /boot/kernel.conf

12.2.4. 最終階段

由loader或會繞開loader的boot2載入核心之後，載入程式便會檢查是否有使用任開機標竿，並根據需要調整開機的方式。

開機時核心互動參數

<table>
<thead>
<tr>
<th>項目</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>核心初始化時，會詢問要掛載為根檔案系統的裝置。</td>
</tr>
<tr>
<td>-C</td>
<td>由CDROM做為根檔案系統開機。</td>
</tr>
<tr>
<td>-s</td>
<td>開機進入單使用者模式。</td>
</tr>
</tbody>
</table>
- core 启動時提供更多信息。

一旦核心完成開機程序後，便會傳送控制權給使用者程序 `init(8)`，該程序位於 `/sbin/init` 或在 loader 中的 `init_path` 變數所指的程式路徑。這是開機程序的最後一個階段。

開機程序會確保系統上檔案系統的一致性 (`Consistency`)，若 UFS 檔案系統不一致且 `fsck` 無法修時，`init` 會讓系統進入單使用者模式，讓系統管理者能夠直接解決問題，否則系統會開機進入多使用者模式。

### 12.2.4.1. 單使用者模式

使用者可以在開機時指定 `-s` 或在 loader 設定 `boot_single` 變數進入這個模式。也可以透過在多使用者模式執行 `shutdown now` 進入此模式。進入單使用者模式時會出現此訊息:

```
Enter full pathname of shell or RETURN for /bin/sh:
```

若使用者按下 Enter，系統便會進入預設的 Bourne shell。要指定使用其他的 Shell 則輸入該 Shell 的完整路徑。

單使用者模式通常用來修復因檔案系統不一致或開機設定檔發生錯誤造成無法開機，也可以用來重設忘記的 root 密碼，因為在單使用者模式會給予對本地系統及設定檔完整的存取權。在這個模式下沒有網路功能。

雖然單使用者模式對修復系統很有幫助，但若系統放於不安全的場所便會有安全上的風險。預設，開機進入單使用者模式後，任何能夠存取實體主機的使用者便擁有系統的完整控制權。

若在 `/etc/ttys` 系統 `console` 變更為 `insecure`，系統便會在初始化單使用者模式前先詢問 root 密碼。這可增加一定程度的安全性，但便無法在忘記 root 密碼時重設密碼。

例 30. 在 `/etc/ttys` 設定不安全的 Console

```
# name  getty                           type    status          comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off insecure
```

不安全 (`insecure`) console 代表對 Console 的實體安全性評估為不安全 (`insecure`)，所以只有知道 root 密碼的人可以使用單使用者模式。

### 12.2.4.2. 多使用者模式

若 `init` 正常找到檔案系統或在單使用者模式的使用者完成了操作並輸入 `exit` 畢業單使用者模式，系統便會進入多使用者模式，在這個模式便會開始系統的資源設置。

資源設置系統 (`Resource configuration system`) 會從 `/etc/defaults/rc.conf` 讀取設定預設值以及從 `/etc/rc.conf` 讀取系統特定的設定，接著會繼續掛載系統列於 `/etc/fstab` 的檔案系統，也會啟動網路服務、其他的系統 Daemon，然後執行本地已安裝軟體的啟動 Script。

要了解更多有關資源設置系統，請參考 `rc(8)` 以及查看位於 `/etc/rc.d` 的 Script。
設定開機
啟動畫面

FreeBSD系統開機會在Console顯示以一系列訊息來表示開機進度。開機啟動畫面(Boot splash screen)是另一種可以把所有開機偵測與服務啟動訊息隱藏的開機畫面,但即使開啟了啟動畫面,仍有數量的開機載入程式的訊息,如:開機選項選單以及倒數時間的提示,仍會在開機時顯示。在開機程序時可以按下鍵盤上的鍵來關閉顯示中的啟動畫面。

FreeBSD有兩種基本的環境可以使用,一種是預設的傳統虛擬Console指令列環境,在系統完成開機之後,便會顯示Console登入提示。另一種環境則是設定好的圖型環境,請參考X Window系統以取得更多相關如何安裝與設定圖型化顯示管理程式與圖型化登入管理程式的資訊。

系統開機之後,啟動畫面預設會作為螢幕保護程式,一段時間未使用便會顯示啟動畫面,並且會循環更改影像的亮度,從明亮到非常暗,然後再繼續循環。

啟動螢幕保護程式的設定可在/etc/rc.conf增加一行saver=來更改。有許多內建的螢幕保護程式可用,在splash(4)中有說明。saver=的選項只會套用至虛擬Console,對圖型化顯示管理程式並不會有任何影響。

透過安裝sysutils/bsd-splash-changer套件或Port,可在開機時顯示隨機挑選的啟動畫面。啟動畫面功能支援256色的點陣圖(.bmp)、ZSoft PCX (.pcx)或TheDraw (.bin)格式。.bmp,.pcx或.bin圖片必須放在根分割區,例如於/boot。啟動圖片檔必須使用320x200像素或更低的解析度以能夠在標準VGA介面卡上運作,要在預設256色、320x200像素或更低的解析度設定開機啟動圖片,可加入下行到/boot/loader.conf,並替換splash.bmp為實際要使用的點陣圖檔:

```
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

要使用PCX檔則可替換點陣圖檔:

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

若要改使用https://en.wikipedia.org/wiki/TheDraw格式的ASCII圖可:

```
splash_txt="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bin"
```

要使用較大的圖片來填滿整個顯示畫面支援的解析度最大可至1024x768像素,VESA模組也必須在系統開機時載入。若使用自訂的核心,請確定自訂核心設定檔中有含有VESA核心設定選項。要載入VESA模組來顯示啟動畫面可在/boot/loader.conf上述例子中提到的三行之前加入下行:

```
vesa_load="YES"
```

其他有用的loader.conf選項還有:

```
beastie_disable="YES"
```

這個會關閉開機選項選單的顯示,但倒數計時提示仍會在。即使關閉了開機選項選單,在倒數計時提示時244
輸入選擇的選項還會啟動對應的開機選項。

這個選項會替換預設與上色的小惡魔圖示一起顯示於開機選單右側的"FreeBSD"文字。

要取得更多資訊，请参考splash(4), loader.conf(5)以及vga(4)。

12.4. 裝置提示在一開始系統啟動時，開機loader(8)會讀取device.hints(5)，這個檔中儲存了核心開機資訊，即變數，有時我們又會稱其為"裝置提示(Device hints)"。這些"裝置提示(Device hints)"會傳送裝置驅動程式做裝置的設置使用。裝置提示也可在階段3開機載入程式提示時指定，如階段三中的示範，其變數也可以使用set增加、使用unset移除、使用show檢視，也可覆蓋設定在/boot/device.hints的變數，但在開機載入程式輸入的裝置提示並不是永久有效的，在下一次重新開機久後便會失效。

一旦系統開機後，便可使用kenv(1)來列出所有的變數。

/boot/device.hints的語法為一個變數一行，使用井字號"#"做為注解符號，每一行的構造如下:

```
hint.driver.unit.keyword = "value"
```

其中driver為裝置驅動程式名稱、unit為裝置驅動程式單位編號及keyword為提示關鍵字，關鍵字由以下選項所組成:

- **at**: 指定裝置所連結的匯流排(Bus)。
- **port**: 指定要使用的I/O開始位置。
- **irq**: 指定要使用的中斷請求編號。
- **drq**: 指定DMA頻道編號。
- **maddr**: 指定裝置所使用的實體記憶體位置。
- **flags**: 設定提供給裝置的各種旗標位元。
- **disabled**: 若設為1則可關閉該裝置。

由於裝置驅動程式可能會接受或請求更多未列於此處的提示，建議先閱讀驅動程式的操作手冊。要取得更多資訊請參考device.hints(5), kenv(1), loader.conf(5)以及loader(8)。

12.5. 關機程序在使用shutdown(8)控制關閉時，init(8)會嘗試執行/etc/rc.shutdown Script接著傳送TERM信號給所有的程序，然後傳送KILL信號給未在時問內中止的程序。

要在支援電源管理的架構與系統關閉FreeBSD主機電源，可使用shutdown -p now來立即關閉電源，要重新啟動FreeBSD系統可使用shutdown -r now。操作人必須為root或為operator的成員才可執行shutdown(8)，擁有這些身分的人也可使用halt(8)與reboot(8)，參考這些指令與shutdown(8)的操作手冊來取得更多資訊。

要修改群組成員可參考使用者與基礎帳號管理。
電源管理需要以載入acpi(4)模組或將其靜態編譯至自訂核心中。
在给安漏洞并准备重些案被意外，或是因可以商系统与网存经远取系统的情况能威脅经许远取系统的过程、未在并不尝未者，授权、意见对脑到电威胁。要策略的完何的意

13.2. 会取全性说明。(Mandatory Access Control, MAC)

全心原则便的其中的一个机密性是CIA何。资权，每的弱任个人的并侵系统中的整全心。必须如个上百系统绕著这个主题大到有确保全性例已经用虚拟个，何述13.1.
Blowfish全建程式設計, 其以及於是時, 且湊密機制來加密儲存使用強密碼方式。在安裝之後, 使用的權限稽查外的更細設定額使用者只能體提供了控制, 然後可以行需升權限議建(/

# visudo
# chsh -s /usr/sbin/nologin toor
# pw lock toor




啟組帳號到該, 然後設定可有不理這種。第一種, 也是兩 FreeBSD, 需要與其他使用者共用系統管理權限升級授權帳號。這種方式。在大多置上線操作程序行機科型應全性政作站, 含工置設置及組織策政程序。有涵蓋腦、裝遵以, 使其確保設定開始層系統上應建策全性發生足團隊。時能管理者,。並威脅
類字元四種碼密碼度為何有碼類密詳細複雜存碼義密小寫數, 符中,這些複雜可以有長度種控制密,了類碼複雜義密的可密插拔可使用內使用的方法之一。在帳號在本FreeBSD是系統安強度密碼碼的,這些更改訊多資解更代路証線來的認碼匙。自從東西有樣是某“:密訊FreeBSD遠說您同時要雙入(Two-factor authentication)應,重認證針對使用舉,然後該行如下:

```
algorithm Blowfish
```

設定是以下行湊演,且,以此為例,該使用者是號MD5符資料來加密密,密資料儲存於密碼號標示機制的一湊湊庫每雜要知使用者可以位湊演種道FreeBSD來加密其法密要響碼執任,OpenSSH

```
$6$pzIjSvCAn
```

```
$SH$A$512
```

```
密码        requisite       pam_passwdqc.so         min=disabled,disabled,disabled,12,10
```

```
/pw:passwd_format=blf:
```

Kerberos能在其網額路上做中會有說明。
這個檔案儲存之後，更改密碼的使用者將會看到如下的訊息:

```
% passwd
Changing local password for trhodes
Old Password: 
You can now choose the new password.
A valid password should be a mix of upper and lower case letters, digits and other characters. You can use a 12 character long password with characters from at least 3 of these 4 classes, or a 10 character long password containing characters from all the classes. Characters that form a common pattern are discarded by the check.
Alternatively, if no one else can see your terminal now, you can pick this as your password: “trait-useful&knob”.
Enter new password: 
```

若輸入了一個不符合密碼政策的密碼，則會被拒絕並顯示警告，然後使用者會有機會再重試，直到超過設定的允許重試數。

大多數密碼政策會讓密碼在多天過後過期。要在 FreeBSD 設定密碼年齡日期，可在 `/etc/login.conf` 中該使用者的登入類別設定 `passwordtime`。在 `default` 登入類別已設定範例:

```
#       :passwordtime=90d:
```

因此，要設定此登入類別的密碼在 90 天之後過期只需要移除註解號（#），然後儲存編輯結果並執行 `cap_mkdb /etc/login.conf`。

要在個別使用者設定限期，可將有效日期或到期的天數與使用者名稱傳給 `pw`:

```
# pw usermod -p 30-apr-2015 -n trhodes
```

如這個例子，有效日期的格式為天、月以及年。要取得更多資訊可參考 `pw(8)`。

### 13.2.5. 偵測 Root 工具 (Rootkit)

rootkit 指的是嘗試未經授權取得系統 root 存取權的軟體。一旦安裝之後，這個惡意軟體將可以光明正大地開啟給另一個給攻擊者進入的大門。

實際上，一旦系統已被 rootkit 滲透且執行了搜索動作之後，該系統就應該從頭重新安裝，因為即使是非常謹慎的資安或系統程式也可能會遺漏攻擊者留下的東西。

rootkit 對管理者而言唯一有幫助的是：一但偵測到，便代表某處已經被滲透，但這類型的應用程式躲藏的非常好，本節將會示範一個可以用來偵測 rootkit 的工具，`security/rkhunter`。

安裝此套件或 Port 之後，系統便可用以下指令檢查。該指令提供許多資訊且會需要手動按下 ENTER 確認:
這些值將儲存數目會是一個子會,一個在組任何一個值是因為事件,要檢案的查檔道攻碼校種子碼或常有格是Binary檔查數讓時也會需要使用這個種子更改過檢數。這這些或在可遠儲存、簽或USB磁碟。端名好在伺服器root。也因此,會這些資變更修議夠其訊來使用者有對獨會能Binary信,意的系統,地是的,這每的但讓關雖訊在本儲存原生IDS( Intrusion Detection System, IDS)。為入式系統要的,因的訊重能的資變更系統夠監體檔案與檢驗檢但想透是否要透滲,必須偵測了。避免也系統下雖量答當的程式然,大出疑上訊,然後找的資正那工具執與應了rkhunter及後全性題可以的分安裝、這些理或在析之軟擇通即處已體警的其他訊更以OpenSSH檢一些有查產rootkit藏關生。在及案、過程中,可能會該程序完成之後,目前會了狀含的訊已案、可上態查過息在畫面。這個疑顯示多多少案、可檢
機器的主機名稱、建立規格檔的日與時間，及建立此規格檔的使用者名稱皆會記錄在此報告當中，報告當中還會有在目錄中每個Binary的驗證碼、大小、時間及SHA256編碼。

要檢驗Binary簽名是否被變更過，可使用先前產生的規格檔比對目前目錄的內容，然後儲存結果到檔案。這個指令需要當初產生原規格檔所使用的種子碼:

```
# mtree -s 3483151339707503 -p /bin < /root/.bin_chksum_mtree > /root/.bin_chksum_output
# mtree: /bin checksum: 3427012225
```

這個動作應會產生與上次建立/bin規格檔時產生的校驗碼相同，若在此目錄的Binary沒有被變更過，那麼/root/.bin_chksum_output這個輸出檔將會是空的。要模擬變更，可以使用touch更改/root/.bin_chksum_output的日期然後再執行檢驗指令一次:

```
# touch /bin/cat
# mtree -s 3483151339707503 -p /bin < /root/.bin_chksum_mtree > /root/.bin_chksum_output
# more /root/.bin_chksum_output
```

```
cat
changed
modification time expected Fri Sep 27 06:32:55 2013 found Mon Feb  3 10:28:43 2014
```

建議對含有所有Binary以及設定檔的目錄建立規格，對含有所有敏感資料的目錄也是。通常會為/bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /etc 及 /usr/local/etc 建立規格。

也有更進階的IDS系統，例如security/aide。大多數情況mtree已可提供管理者所需的功能。將種子碼與校驗碼結果保存在惡意使用者無法存取的地方是非常重要的事。更多有關於mtree的資料可在mtree(8)找到。

13.2.7. 系統安全性調校

在FreeBSD,有許多系統功能可以使用sysctl調校，本節會涵蓋少數可以調校來避免阻斷服務（Denial of Service, DoS）攻擊的安全性功能。更多有關使用sysctl的資料包含：如如何暫時更改數值及如何在測試之後做永久更改可在使用sysctl(8)調校找到。

預設FreeBSD核心會使用安全性層級-1來開機，這又稱作"不安全模式"，因為不可變(Immutable)。
為使的場金鑰，安本節做全這個性密碼到產個程式。碼組與保數少登減無。過拿性密每生若使用擷碼登一次來性密用了。的密相同,若使用者提供的加密採單向代次根據短字成用次後字數的代次由兩,是一個介一個於母到疊五
OPIE重另碼個部份的資料很兩金鑰對"。且不密密碼UNIX™產，第三種是用
碼的型上設方式與詰碼全。加密使用了安
的的下,一擊使用一次,組密對位送
擊碼擊,然後使用該攻取系統。由設避免
(One-time Passwords In Everything, OPIE)
內建碼性密一次性密還有額設定求絕執求應動有理息拒傳的主息ICMP在網給所有在子網機通常是故中不可由於址由位位(One-time Passwords In Everything, OPIE)
封包,於
設定可度用在的行的接應閉查上看在系統預用程式接動RST傳作是會,預的層級將否檔可以置關且可以讀取或
。請參考
型是blackhole(4)。下何登或在不安的。會讀取在疊來更改密除之外,用
於設多與這些可用的安
黑hole(4)及
Xorg的成其他做會
會,

et.inet.blackhole
層級相多與這些可用的安

net.inet.udp.blackhole
及
tnet.inet.tcp.blackhole
為
net.inet.icmp.drop_redirect
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄

為
層級
為
封
丟棄
13.3.1. 初始化 OPIE

第一次要初始化 OPIE，要在安全的场所执行以下指令:
```
% opiepasswd -c
```

Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using telnet, xterm, or a dial-in, type ^C now or exit with no password. Then run opiepasswd without the -c parameter.

Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:

ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED

将设定采用假设指令在安全场所以�行的Console模式，如在使用者掌控之中的电脑或者透过SSH连线到一台在使用者掌控之中的电脑。

提示出现后，输入用来产生一次性登入金钥的秘密密码，应使用一个不容易被猜出来的密码，且应与使用者登入帐号所使用的密码不同，密码必须介於10到127个字元长度之间，然后请记住这个密码。

ID行会列出登入名称 (unfurl)、预设定的叠代次数 (499) 及预设定的种子码 (to4268)。在进行登入时，系统会记住这些参数并且显示出来，这也代表不需要另外纪录这些资讯。最后一行会列出根据这些参数与秘密密码所生产出来的一次性密钥，在下一次登入时便要使用这个一次性密钥。

13.3.2. 在不安全连线下做初始化

要在不安全的系统上初始化或更改秘密密码会需要某个可使用安全的连线的地方执行 opiekey，这可能是在某一信任的主机上的 Shell。

初始化需要设定叠代次数，100可能是不错的选择，种子码可以自行指定或随机产生，在不安全连线下要被初始化主机须使用 opiepasswd(1):
```
% opiepasswd
Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
otp-md5 498 to4268 ext
Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
otp-md5 499 to4269
Response: LINE PAP MILK NELL BUOY TROY
ID mark OTP key is 499 gr4269
```

要采用预设的种子码，可以直接按下Return。
13.3.3. 生成单组一次性密钥

在初始化 OPIE 之后进行登录会显示如下的提示信息:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.
```

```
FreeBSD/i386 (example.com) (ttypsa)
```

```
login: <username>
```

```
otp-md5 498 gr4269 ext
```

```
Password:
```

OPIE 的提示提供了一个很有用的功能,若在密码提示时按下 Return,便可以开启回应功能并显示输入的内容,这个功能在尝试手工输入列印出来的密码时很有用。

此时,要生成一次性密钥来回应登录时的提示,这必须在受信任且可安全执行 opiekey(1) 的系统上完成。这个指令有提供 Windows™, Mac OS™ 与 FreeBSD 版本,使用时需要叠代次数与种子码做为在指令列的参数,剪下在要登入主机在登入时所提示的讯息。

在信任的系统上执行:

```
% opiekey 498 to4268
```

```
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
```

```
Enter secret pass phrase:
```

```
GAME GAG WELT OUT DOWN CHAT
```

13.3.4. 生成多组一次性密钥

有时会无法储存信任的主机或没有安全的连线,在这种情况下,可以使用 opiekey(1) 来预先生成多个一次性密钥,例如:

```
255
```
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>

26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI

13.3.5. Limitation of use of UNIX™ Passwords
OPIE can be limited to the use of UNIX™ Passwords, depending on the location of the host.

The file /etc/opieaccess is used for this purpose. Refer to opieaccess(5) for more information on its use.

Here is a sample for /etc/opieaccess:

```
permit 192.168.0.0 255.255.0.0
```

This line grants access to the Source IP address (possibly for testing purposes) within the given range to users at any time.

If the /etc/opieaccess file does not match a rule, the default will be to refuse access by non-OPIE users.

13.4. TCP Wrapper
TCP Wrapper is a host-based access control system which extends the abilities of superuser.
It can be configured to provide logging support, return messages, and connection restrictions for the server daemons under the control of inetd. Refer to tcpd(8) for more information about TCP Wrapper and its features.

TCP Wrapper should not be considered a replacement for a properly configured firewall. Instead, TCP Wrapper should be used in conjunction with a firewall and other security enhancements in order to provide another layer of protection in the implementation of a security policy.

13.4.1. Initial setup
To enable TCP Wrapper in FreeBSD, add the following lines to /etc/rc.conf:

```
inetc_enable="YES"
inetc_flags="-Ww"
```

Then, properly configure /etc/hosts.allow.

Unlike other implementations of TCP Wrapper, the use of hosts.deny is deprecated in FreeBSD. All configuration options should be placed in /etc/hosts.allow.
In the simplest configuration, daemon connection policies are set to either permit or block, depending on the options in /etc/hosts.allow. The default configuration in FreeBSD is to allow all connections to the daemons started with inetd.

Basic configuration usually takes the form of `daemon : address : action`, where `daemon` is the daemon which inetd started, `address` is a valid hostname, IP address, or an IPv6 address enclosed in brackets ([ ]), and `action` is either `allow` or `deny`. TCP Wrapper uses a first rule match semantic, meaning that the configuration file is scanned from the beginning for a matching rule. When a match is found, the rule is applied and the search process stops.

For example, to allow POP3 connections via the `mail/qpopper` daemon, the following lines should be appended to `hosts.allow`:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

Whenever this file is edited, restart inetd:

```
# service inetd restart
```

13.4.2. 高级设置

TCP Wrapper provides advanced options to allow more control over the way connections are handled. In some cases, it may be appropriate to return a comment to certain hosts or daemon connections. In other cases, a log entry should be recorded or an email sent to the administrator. Other situations may require the use of a service for local connections only. This is all possible through the use of configuration options known as wildcards, expansion characters, and external command execution.

Suppose that a situation occurs where a connection should be denied yet a reason should be sent to the host who attempted to establish that connection. That action is possible with `twist`. When a connection attempt is made, `twist` executes a shell command or script. An example exists in `hosts.allow`:

```
# The rest of the daemons are protected.
ALL : ALL \
: severity auth.info \
: twist /bin/echo "You are not welcome to use %d from %h."
```

In this example, the message "You are not allowed to use `daemon name` from `hostname`." will be returned for any daemon not configured in `hosts.allow`. This is useful for sending a reply back to the connection initiator right after the established connection is dropped. Any message returned must be wrapped in quote (" ") characters.

It may be possible to launch a denial of service attack on the server if an attacker floods these daemons with connection requests. Another possibility is to use `spawn`. Like `twist`, `spawn` implicitly denies the connection and may be used to run external shell commands or scripts. Unlike `twist`, `spawn` will not send a reply back to the host who established the connection. For example, consider the following configuration:

```
# We do not allow connections from example.com:
```
This will deny all connection attempts from *.example.com and log the hostname, IP address, and the daemon to which access was attempted to /var/log/connections.log. This example uses the substitution characters %a and %h. Refer to hosts_access(5) for the complete list.

To match every instance of a daemon, domain, or IP address, use ALL. Another wildcard is PARANOID which may be used to match any host which provides an IP address that may be forged because the IP address differs from its resolved hostname. In this example, all connection requests to Sendmail which have an IP address that varies from its hostname will be denied:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```

Using the PARANOID wildcard will result in denied connections if the client or server has a broken DNS setup.

To learn more about wildcards and their associated functionality, refer to hosts_access(5).

When adding new configuration lines, make sure that any unneeded entries for that daemon are commented out in hosts.allow.

13.5. Kerberos

Kerberos is a network authentication protocol which was originally created by the Massachusetts Institute of Technology (MIT) as a way to securely provide authentication across a potentially hostile network. The Kerberos protocol uses strong cryptography so that both a client and server can prove their identity without sending any unencrypted secrets over the network. Kerberos can be described as an identity-verifying proxy system and as a trusted third-party authentication system. After a user authenticates with Kerberos, their communications can be encrypted to assure privacy and data integrity.

The only function of Kerberos is to provide the secure authentication of users and servers on the network. It does not provide authorization or auditing functions. It is recommended that Kerberos be used with other security methods which provide authorization and audit services.

The current version of the protocol is version 5, described in RFC 4120. Several free implementations of this protocol are available, covering a wide range of operating systems. MIT continues to develop their Kerberos package. It is commonly used in the US as a cryptography product, and has historically been subject to US export regulations. In FreeBSD, MITKerberos is available as the security/krb5 package or port. The Heimdal Kerberos implementation was explicitly developed outside of the US to avoid export regulations. The Heimdal Kerberos distribution is included in the base FreeBSD installation, and another distribution with more configurable options is available as security/heimdal in the Ports Collection.

In Kerberos users and services are identified as “principals” which are contained within an administrative grouping, called a “realm”. A typical user principal would be of the form user@REALM (realms are traditionally uppercase).

This section provides a guide on how to set up Kerberos using the Heimdal distribution included in FreeBSD.
The DNS domain (zone) will be example.org.

The Kerberos realm will be EXAMPLE.ORG.

Use real domain names when setting up Kerberos, even if it will run internally. This avoids DNS problems and assures inter-operation with other Kerberos realms.

13.5.1. Setting Heimdal KDC

The Key Distribution Center (KDC) is the centralized authentication service that Kerberos provides, the "trusted third party" of the system. It is the computer that issues Kerberos tickets, which are used for clients to authenticate to servers. Because the KDC is considered trusted by all other computers in the Kerberos realm, it has heightened security concerns. Direct access to the KDC should be limited.

While running a KDC requires few computing resources, a dedicated machine acting only as a KDC is recommended for security reasons.

To begin setting up a KDC, add these lines to /etc/rc.conf:

```plaintext
kdc_enable="YES"
kadmind_enable="YES"
```

Next, edit /etc/krb5.conf as follows:

```plaintext
[libdefaults]
default_realm = EXAMPLE.ORG

[realms]
EXAMPLE.ORG = {
kdc = kerberos.example.org
admin_server = kerberos.example.org
}

[domain_realm]
.example.org = EXAMPLE.ORG
```

In this example, the KDC will use the fully-qualified hostname kerberos.example.org. The hostname of the KDC must be resolvable in the DNS.

Kerberos can also use the DNS to locate KDCs, instead of a [realms] section in /etc/krb5.conf. For large organizations that have their own DNS servers, the above example could be trimmed to:

```plaintext
[libdefaults]
default_realm = EXAMPLE.ORG

[domain_realm]
.example.org = EXAMPLE.ORG
```

With the following lines being included in the example.org zone file:

```plaintext
_kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
```

259
In order for clients to be able to find the Kerberos services, they must have either a fully configured /etc/krb5.conf or a minimally configured /etc/krb5.conf and a properly configured DNS server.

Next, create the Kerberos database which contains the keys of all principals (users and hosts) encrypted with a master password. It is not required to remember this password as it will be stored in /var/heimdal/m-key; it would be reasonable to use a 45-character random password for this purpose. To create the master key, run kstash and enter a password:

```
# kstash
Master key: xxxxxxxxxxxxxxxxxxxxxxx
Verifying password - Master key: xxxxxxxxxxxxxxxxxxxxxxx
```

Once the master key has been created, the database should be initialized. The Kerberos administrative tool kadmin(8) can be used on the KDC in a mode that operates directly on the database, without using the kadmind(8) network service, as kadmin -l. This resolves the chicken-and-egg problem of trying to connect to the database before it is created. At the kadmin prompt, use init to create the realm's initial database:

```
# kadmin -l
kadmin> init EXAMPLE.ORG
```

Lastly, while still in kadmin, create the first principal using add. Stick to the default options for the principal for now, as these can be changed later with modify. Type ? at the prompt to see the available options.

```
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Next, start the KDC services by running service kdc start and service kadmind start. While there will not be any kerberized daemons running at this point, it is possible to confirm that the KDC is functioning by obtaining a ticket for the principal that was just created:

```
% kinit tillman
tillman@EXAMPLE.ORG
'Password: 
```

Confirm that a ticket was successfully obtained using klist:
The temporary ticket can be destroyed when the test is finished:

```
% kdestroy
```

13.5.2. 設定

伺服器使用 Kerberos

The first step in configuring a server to use Kerberos authentication is to ensure that it has the correct configuration in `/etc/krb5.conf`. The version from the KDC can be used as-is, or it can be regenerated on the new system.

Next, create `/etc/krb5.keytab` on the server. This is the main part of "Kerberizing" a service—it corresponds to generating a secret shared between the service and the KDC. The secret is a cryptographic key, stored in a "keytab". The keytab contains the server's host key, which allows it and the KDC to verify each others' identity. It must be transmitted to the server in a secure fashion, as the security of the server can be broken if the key is made public. Typically, the keytab is generated on an administrator's trusted machine using `kadmin`, then securely transferred to the server, e.g., with `scp(1)`; it can also be created directly on the server if that is consistent with the desired security policy. It is very important that the keytab is transmitted to the server in a secure fashion: if the key is known by some other party, that party can impersonate any user to the server!

Using `kadmin` on the server directly is convenient, because the entry for the host principal in the KDC database is also created using `kadmin`. Of course, `kadmin` is a kerberized service; a Kerberos ticket is needed to authenticate to the network service, but to ensure that the user running `kadmin` is actually present (and their session has not been hijacked), `kadmin` will prompt for the password to get a fresh ticket. The principal authenticating to the kadmin service must be permitted to use the `kadmin` interface, as specified in `kadmind.acl`. See the section titled "Remote administration" in `info heimdal` for details on designing access control lists. Instead of enabling remote kadmin access, the administrator could securely connect to the KDC via the local console or `ssh(1)`, and perform administration locally using `kadmin -l`.

After installing `/etc/krb5.conf`, use `add --random-key` in `kadmin`. This adds the server's host principal to the database, but does not extract a copy of the host principal key to a keytab. To generate the keytab, use `ext` to extract the server's host principal key to its own keytab:

```bash
# kadmin
kadmin> add --random-key host/myserver.example.org
```

Max ticket life [unlimited]:

Max renewable life [unlimited]:

Principal expiration time [never]:

Password expiration time [never]:

Attributes []:

```
# kadmin
kadmin> ext_keytab host/myserver.example.org
```

261
Note that `ext_keytab` stores the extracted key in `/etc/krb5.keytab` by default. This is good when being run on the server being kerberized, but the `--keytab` path/to/file argument should be used when the keytab is being extracted elsewhere:

```
# kadmin
kadmin> ext_keytab
   --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

The keytab can then be securely copied to the server using `scp(1)` or a removable media. Be sure to specify a non-default keytab name to avoid inserting unneeded keys into the system's keytab.

At this point, the server can read encrypted messages from the KDC using its shared key, stored in `krb5.keytab`. It is now ready for the Kerberos-using services to be enabled. One of the most common such services is `sshd(8)`, which supports Kerberos via the GSS-API. In `/etc/ssh/sshd_config`, add the line:

```
GSSAPIAuthentication yes
```

做完這個變更之後，必須重新啟動 `sshd(8)` 來使新的設定值生效:

```
service sshd restart
```

13.5.3. 設定客戶端使用 Kerberos

As it was for the server, the client requires configuration in `/etc/krb5.conf`. Copy the file in place (securely) or re-enter it as needed.

Test the client by using `kinit`, `klist`, and `kdestroy` from the client to obtain, show, and then delete a ticket for an existing principal. Kerberos applications should also be able to connect to Kerberos enabled servers. If that does not work but obtaining a ticket does, the problem is likely with the server and not with the client or the KDC. In the case of kerberized `ssh(1)`, GSS-API is disabled by default, so test using `ssh -o GSSAPIAuthentication=yes hostname`.

When testing a Kerberized application, try using a packet sniffer such as `tcpdump` to confirm that no sensitive information is sent in the clear.

Various Kerberos client applications are available. With the advent of a bridge so that applications using SASL for authentication can use GSS-API mechanisms as well, large classes of client applications can use Kerberos for authentication, from Jabber clients to IMAP clients.

Users within a realm typically have their Kerberos principal mapped to a local user account. Occasionally, one needs to grant access to a local user account to someone who does not have a matching Kerberos principal. For example, `tillman@example.org` may need access to the local user account `webdevelopers`. Other principals may also need access to that local account.

The `.k5login` and `.k5users` files, placed in a user's home directory, can be used to solve this problem. For example, if the following `.k5login` is placed in the home directory of `webdevelopers`, both principals listed will have access to that account without requiring a shared password:

```
tillman@example.org
jdoe@example.org
```

Refer to `ksu(1)` for more information about `.k5users`. 
The major difference between the MIT and Heimdal implementations is that `kadmin` has a different, but equivalent, set of commands and uses a different protocol. If the KDC is MIT, the Heimdal version of `kadmin` cannot be used to administer the KDC remotely, and vice versa.

Client applications may also use slightly different command line options to accomplish the same tasks. Following the instructions at [http://web.mit.edu/Kerberos/www/](http://web.mit.edu/Kerberos/www/) is recommended. Be careful of path issues: the MIT port installs into `/usr/local/` by default, and the FreeBSD system applications run instead of the MIT versions if `PATH` lists the system directories first.

When using MIT Kerberos as a KDC on FreeBSD, the following edits should also be made to `rc.conf`:

- `kerberos5_server = /usr/local/sbin/krb5kdc`
- `kadmind5_server = /usr/local/sbin/kadmind`
- `kerberos5_server_flags = ```
- `kerberos5_server_enable = YES`
- `kadmind5_server_enable = YES`

### Kerberos Tips, Tricks, and Troubleshooting

When configuring and troubleshooting Kerberos, keep the following points in mind:

- When using either Heimdal or MITKerberos from ports, ensure that the `PATH` lists the port's versions of the client applications before the system versions.
- If all the computers in the realm do not have synchronized time settings, authentication may fail. [NTP Time Synchronization](http://web.mit.edu/Kerberos/www/) describes how to synchronize clocks using NTP.
- If the hostname is changed, the `host/principal` must be changed and the keytab updated. This also applies to special keytab entries like the `HTTP/principal` used for Apache's `mod_auth_kerb`.
- All hosts in the realm must be both forward and reverse resolvable in DNS or, at a minimum, exist in `/etc/hosts`. CNAMEs will work, but the A and PTR records must be correct and in place. The error message for unresolvable hosts is not intuitive: `Kerberos5 refuses authentication because Read req failed: Key table entry not found`.
- Some operating systems that act as clients to the KDC do not set the permissions for `ksu` to be `setuid` root. This means that `ksu` does not work. This is a permissions problem, not a KDC error.
- With MITKerberos, to allow a principal to have a ticket life longer than the default lifetime of ten hours, use `modify_principal` at the `kadmin(8)` prompt to change the `maxlife` of both the principal in question and the `krbtgt` principal. The principal can then use `kinit -l` to request a ticket with a longer lifetime.
- When running a packet sniffer on the KDC to aid in troubleshooting while running `kinit` from a workstation, the Ticket Granting Ticket (TGT) is sent immediately, even before the password is typed. This is because the Kerberos server freely transmits a TGT to any unauthorized request. However, every TGT is encrypted in a key derived from the user's password. When a user types their password, it is not sent to the KDC, it is instead used to decrypt the TGT that `kinit` already obtained. If the decryption process results in a valid ticket with a valid time stamp, the user has valid Kerberos credentials. These credentials include a session key for establishing secure communications with the Kerberos server in the future, as well as the actual TGT, which is encrypted with the Kerberos server's own key. This second layer of encryption allows the Kerberos server to verify the authenticity of each TGT.
- Host principals can have a longer ticket lifetime. If the user principal has a lifetime of a week but the host being connected to has a lifetime of nine hours, the user cache will have an expired host principal and the ticket cache will not work as expected.
- When setting up `krb5.dict` to prevent specific bad passwords from being used as described in `kadmind(8)`, remember that it only applies to principals that have a password policy assigned to them.
The format used in krb5.dict is one string per line. Creating a symbolic link to /usr/shared/dict/words might be useful.

13.5.6. Kerberos

Since Kerberos is an all or nothing approach, every service enabled on the network must either be modified to work with Kerberos or be otherwise secured against network attacks. This is to prevent user credentials from being stolen and re-used. An example is when Kerberos is enabled on all remote shells but the non-Kerberized POP3 mail server sends passwords in plain text.

The KDC is a single point of failure. By design, the KDC must be as secure as its master password database. The KDC should have absolutely no other services running on it and should be physically secure. The danger is high because Kerberos stores all passwords encrypted with the same master key which is stored as a file on the KDC.

A compromised master key is not quite as bad as one might fear. The master key is only used to encrypt the Kerberos database and as a seed for the random number generator. As long as access to the KDC is secure, an attacker cannot do much with the master key.

If the KDC is unavailable, network services are unusable as authentication cannot be performed. This can be alleviated with a single master KDC and one or more slaves, and with careful implementation of secondary or fall-back authentication using PAM.

Kerberos allows users, hosts and services to authenticate between themselves. It does not have a mechanism to authenticate the KDC to the users, hosts, or services. This means that a trojanned kinit could record all user names and passwords. File system integrity checking tools like security/tripwire can alleviate this.

13.5.7. 相关资源与延伸资讯

• The Kerberos FAQ
• Designing an Authentication System: a Dialog in Four Scenes
• RFC 4120, The Kerberos Network Authentication Service (V5)
• MIT Kerberos home page
• Heimdal Kerberos home page

13.6. OpenSSL

OpenSSL is an open source implementation of the SSL and TLS protocols. It provides an encryption transport layer on top of the normal communications layer, allowing it to be intertwined with many network applications and services.

The version of OpenSSL included in FreeBSD supports the Secure Sockets Layer 3.0 (SSLv3) and Transport Layer Security 1.0/1.1/1.2 (TLSv1/TLSv1.1/TLSv1.2) network security protocols and can be used as a general cryptographic library. In FreeBSD 12.0-RELEASE and above, OpenSSL also supports Transport Layer Security 1.3 (TLSv1.3).

OpenSSL is often used to encrypt authentication of mail clients and to secure web based transactions such as credit card payments. Some ports, such as www/apache24 and databases/postgresql11-server, include a compile option for building with OpenSSL. If selected, the port will add support using OpenSSL from the base system. To instead have the port compile against OpenSSL from the security/openssl port, add the following to etc/make.conf:

DEFAULT_VERSIONS+= ssl=openssl

Another common use of OpenSSL is to provide certificates for use with software applications. Certificates can be used to verify the credentials of a company or individual. If a certificate has not...
been signed by an external Certificate Authority (CA), such as http://www.verisign.com, the application that uses the certificate will produce a warning. There is a cost associated with obtaining a signed certificate and using a signed certificate is not mandatory as certificates can be self-signed. However, using an external authority will prevent warnings and can put users at ease.

This section demonstrates how to create and use certificates on a FreeBSD system. Refer to 設定 LDAP 伺服器 for an example of how to create a CA for signing one's own certificates.

For more information about SSL, read the free OpenSSL Cookbook.

13.6.1. 產生憑證

To generate a certificate that will be signed by an external CA, issue the following command and input the information requested at the prompts. This input information will be written to the certificate. At the Common Name prompt, input the fully qualified name for the system that will use the certificate. If this name does not match the server, the application verifying the certificate will issue a warning to the user, rendering the verification provided by the certificate as useless.

```
# openssl req -new -nodes -out req.pem -keyout cert.key -sha256 -newkey rsa:2048
```

```
Generating a 2048 bit RSA private key
.............+++
.............................................................+++
writing new private key to 'cert.key'
-----
You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank. For some fields there will be a default value, if you enter '.' the field will be left blank.
-----
Country Name (2 letter code)             [AU]:US
State or Province Name (full name)       [Some-State]:PA
Locality Name (eg, city)                 []:Pittsburgh
Organization Name (eg, company)          [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section)    []:Systems Administrator
Common Name (eg, YOUR name)              []:localhost.example.org
Email Address []:trhodes@FreeBSD.org
Please enter the following 'extra' attributes to be sent with your certificate request
A challenge password []:
An optional company name []:Another Name
Other options, such as the expire time and alternate encryption algorithms, are available when creating a certificate. A complete list of options is described in openssl(1).
```

This command will create two files in the current directory. The certificate request, req.pem, can be sent to a CA who will validate the entered credentials, sign the request, and return the signed certificate, which is saved as cert.pem.
The second file, `cert.key`, is the private key for the certificate and should be stored in a secure location. If this falls in the hands of others, it can be used to impersonate the user or the server.

Alternately, if a signature from a CA is not required, a self-signed certificate can be created. First, generate the RSA key:

```
# openssl genrsa -rand -genkey -out cert.key 2048
```

The output might look like this:

```
0 semi-random bytes loaded
Generating RSA private key, 2048 bit long modulus
.............................................+++ 
.................................................................................................................+++ 
e is 65537 (0x10001) 
```

Use this key to create a self-signed certificate. Follow the usual prompts for creating a certificate:

```
# openssl req -new -x509 -days 365 -key cert.key -out cert.crt -sha256
```

You are about to be asked to enter information that will be incorporated into your certificate request. What you are about to enter is what is called a Distinguished Name or a DN. There are quite a few fields but you can leave some blank. If you enter '.', the field will be left blank.

```
-----
Country Name (2 letter code) [AU]: US
State or Province Name (full name) [Some-State]: PA
Locality Name (eg, city): Pittsburgh
Organization Name (eg, company): Internet Widgits Pty Ltd
Organizational Unit Name (eg, section): Systems Administrator
Common Name (e.g. server FQDN or YOUR name): localhost.example.org
Email Address: trhodes@FreeBSD.org
-----
```

This will create two new files in the current directory: a private key file `cert.key`, and the certificate itself, `cert.crt`. These should be placed in a directory, preferably under `/etc/ssl/`, which is readable only by root. Permissions of 0700 are appropriate for these files and can be set using `chmod`.

One use for a certificate is to encrypt connections to the Sendmail mail server in order to prevent the use of clear text authentication. Some mail clients will display an error if the user has not installed a local copy of the certificate. Refer to the documentation included with the software for more information on certificate installation.

In FreeBSD 10.0-RELEASE and above, it is possible to create a self-signed certificate for Sendmail automatically. To enable this, add the following lines to `/etc/rc.conf`:
This will automatically create a self-signed certificate, 
/etc/mail/certs/host.cert, a signing key, 
/etc/mail/certs/host.key, and a CA certificate, 
/etc/mail/certs/cacert.pem. The certificate will use 
the Common Name specified in sendmail_cert_cn. After saving the edits, restart Sendmail:

```
# service sendmail restart
```

If all went well, there will be no error messages in 
/var/log/maillog. For a simple test, connect to the 
mail server's listening port using telnet:

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^Z'.
Fri, 18 Apr 2014 11:50:32 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

If the STARTTLS line appears in the output, everything is working correctly.

13.7. VPN over IPsec

Internet Protocol Security (IPsec) is a set of protocols which sit on top of the Internet Protocol (IP) layer. It allows two or more hosts to communicate in a secure manner by authenticating and encrypting each IP packet of a communication session. The FreeBSD IPsec network stack is based on the [http://www.kame.net/](http://www.kame.net/) implementation and supports both IPv4 and IPv6 sessions.

IPsec is comprised of the following sub-protocols:

- Encapsulated Security Payload (ESP): this protocol protects the IP packet data from third party
interference by encrypting the contents using symmetric cryptography algorithms such as Blowfish and 3DES.

- **Authentication Header (AH)**: this protocol protects the IP packet header from third party interference and spoofing by computing a cryptographic checksum and hashing the IP packet header fields with a secure hashing function. This is then followed by an additional header that contains the hash, to allow the information in the packet to be authenticated.

- **IP Payload Compression Protocol (IPComp)**: this protocol tries to increase communication performance by compressing the IP payload in order to reduce the amount of data sent.

These protocols can either be used together or separately, depending on the environment.

IPsec supports two modes of operation. The first mode, **Transport Mode**, protects communications between two hosts. The second mode, **Tunnel Mode**, is used to build virtual tunnels, commonly known as Virtual Private Networks (VPNs). Consult `ipsec(4)` for detailed information on the IPsec subsystem in FreeBSD.

在FreeBSD 11版本之后的版本预设会开启IPsec功能，先前版本的FreeBSD可在自订核心设定档中加入以下选项然后依设定FreeBSD核心的指示来重新编译核心:

```plaintext
options   IPSEC
#IP security
device    crypto
```

If IPsec debugging support is desired, the following kernel option should also be added:

```plaintext
options   IPSEC_DEBUG  debug
```

for IP security

This rest of this chapter demonstrates the process of setting up an IPsecVPN between a home network and a corporate network. In the example scenario:

- Both sites are connected to the Internet through a gateway that is running FreeBSD.
- The gateway on each network has at least one external IP address. In this example, the corporate LAN's external IP address is `172.16.5.4` and the home LAN's external IP address is `192.168.1.12`.
- The internal addresses of the two networks can be either public or private IP addresses. However, the address space must not collide. For example, both networks cannot use `192.168.1.x`. In this example, the corporate LAN's internal IP address is `10.246.38.1` and the home LAN's internal IP address is `10.0.0.5`.

在FreeBSD上设定VPN

To begin, `security/ipsec-tools` must be installed from the Ports Collection. This software provides a number of applications which support the configuration.

The next requirement is to create two `gif(4)` pseudo-devices which will be used to tunnel packets and allow both networks to communicate properly. As root, run the following commands, replacing `internal` and `external` with the real IP addresses of the internal and external interfaces of the two gateways:

```plaintext
# ifconfig gif0 create
# ifconfig gif0 internal1 internal2
# ifconfig gif0 tunnel external1 external2
```

Verify the setup on each gateway, using `ifconfig`. Here is the output from Gateway 1:
Here is the output from Gateway 2:

```
inet 10.0.0.5 --> 10.246.38.1 netmask 0xffffff00
inet6 fe80::250:bfff:fe3a:c1f%gif0 prefixlen 64 scopeid 0x4
```

Once complete, both internal IP addresses should be reachable using `ping(8)`:

```
priv-net# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
64 bytes from 10.0.0.5: icmp_seq=0 ttl=64 time=42.786 ms
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=19.255 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=20.440 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=21.036 ms
--- 10.0.0.5 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.255/25.879/42.786/9.782 ms
```

```
corp-net# ping 10.246.38.1
PING 10.246.38.1 (10.246.38.1): 56 data bytes
64 bytes from 10.246.38.1: icmp_seq=0 ttl=64 time=28.106 ms
64 bytes from 10.246.38.1: icmp_seq=1 ttl=64 time=42.917 ms
64 bytes from 10.246.38.1: icmp_seq=2 ttl=64 time=127.525 ms
64 bytes from 10.246.38.1: icmp_seq=3 ttl=64 time=119.896 ms
--- 10.246.38.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.106/94.594/154.524/49.814 ms
```

As expected, both sides have the ability to send and receive ICMP packets from the privately configured addresses. Next, both gateways must be told how to route packets in order to correctly send traffic from either network. The following commands will achieve this goal:

```
corp-net# route add 10.0.0.0 10.0.0.5 255.255.255.0
corp-net# route add net 10.0.0.0: gateway 10.0.0.5
priv-net# route add 10.246.38.0 10.246.38.1 255.255.255.0
```
Priv-net# route add host 10.246.38.0: gateway 10.246.38.1

At this point, internal machines should be reachable from each gateway as well as from machines behind the gateways. Again, use `ping(8)` to confirm:

Corp-net# ping 10.0.0.8
PING 10.0.0.8 (10.0.0.8): 56 data bytes
64 bytes from 10.0.0.8: icmp_seq=0 ttl=63 time=92.391 ms
64 bytes from 10.0.0.8: icmp_seq=1 ttl=63 time=21.870 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=63 time=198.022 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=63 time=22.241 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=63 time=174.705 ms
--- 10.0.0.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.870/101.846/198.022/74.001 ms

Priv-net# ping 10.246.38.107
PING 10.246.38.1 (10.246.38.107): 56 data bytes
64 bytes from 10.246.38.107: icmp_seq=0 ttl=64 time=53.491 ms
64 bytes from 10.246.38.107: icmp_seq=1 ttl=64 time=23.395 ms
64 bytes from 10.246.38.107: icmp_seq=2 ttl=64 time=23.865 ms
64 bytes from 10.246.38.107: icmp_seq=3 ttl=64 time=21.145 ms
64 bytes from 10.246.38.107: icmp_seq=4 ttl=64 time=36.708 ms
--- 10.246.38.107 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.145/31.721/53.491/12.179 ms

Setting up the tunnels is the easy part. Configuring a secure link is a more in depth process. The following configuration uses pre-shared (PSK) RSA keys. Other than the IP addresses, the `/usr/local/etc/racoon/racoon.conf` on both gateways will be identical and look similar to:

```
path pre_shared_key /usr/local/etc/racoon/psk.txt; # location of pre-shared key file
log debug; # log verbosity setting: set to 'notify' when testing and debugging is complete
padding # options are not to be changed
{
  maximum_length 20;
  randomize off;
  strict_check off;
  exclusive_tail off;
}
timer # timing options. change as needed
270
```
counter 5;
interval 20 sec;
persend 1;

# natt_keepalive 15 sec;
phase1 30 sec;
phase2 15 sec;

listen
address [port] that racoon will listen on

isakmp 172.16.5.4 [500];
isakmp_natt 172.16.5.4 [4500];

remote 192.168.1.12 [500]
exchange_mode main,aggressive;
doi ipsec_doi;

situation identity_only;
my_identifier address 172.16.5.4;
peers_identifier address 192.168.1.12;
lifetime time 8 hour;
passive off;
proposal_check obey;

# nat_traversal off;
genenerate_policy off;

proposal
encryption_algorithm blowfish;
hash_algorithm md5;
authentication_method pre_shared_key;
lifetime time 30 sec;
dh_group 1;

sainfo (address 10.246.38.0/24 any address 10.0.0.0/24 any)
$network/$netmask $type address $network/$netmask $type
($type being any or esp)
pfs_group 1;
lifetime time 36000 sec;
encryption_algorithm blowfish,3des;
authentication_algorithm        hmac_md5,hmac_sha1;
compression_algorithm   deflate;
}

For descriptions of each available option, refer to the manual page for racoon.conf.

The Security Policy Database (SPD) needs to be configured so that FreeBSD and racoon are able to encrypt and decrypt network traffic between the hosts. This can be achieved with a shell script, similar to the following, on the corporate gateway. This file will be used during system initialization and should be saved as /usr/local/etc/racoon/setkey.conf.

```
flush;
spdflush;
#
# To the home network
 spdadd 10.246.38.0/24 10.0.0.0/24 any -P out ipsec esp/tunnel/172.16.5.4-192.168.1.12/use;
 spdadd 10.0.0.0/24 10.246.38.0/24 any -P in ipsec esp/tunnel/192.168.1.12-172.16.5.4/use;
```

Once in place, racoon may be started on both gateways using the following command:

```
# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf -l /var/log/racoon.log
```

The output should be similar to the following:

```
corp-net# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf
Foreground mode.
2006-01-30 01:35:48: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:35:55: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:36:04: INFO: ISAKMP-SA established 172.16.5.4[500]-192.168.1.12[500]
spi:623b9b3bd2492452:7deab82d54ff704a
2006-01-30 01:36:05: INFO: initiate new phase 2 negotiation: 172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-172.16.5.4[0] spi=28496098(0x1b2d0e2)
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-192.168.1.12[0] spi=47784998(0x2d92426)
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-172.16.5.4[0] spi=124397467(0x76a279b)
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-192.168.1.12[0] spi=175852902(0xa7b4d66)
```

To ensure the tunnel is working properly, switch to another console and use tcpdump(1) to view network traffic using the following command. Replace em0 with the network interface card as required:

```
272
```
Data similar to the following should appear on the console. If not, there is an issue and debugging the returned data will be required.

At this point, both networks should be available and seem to be part of the same network. Most likely both networks are protected by a firewall. To allow traffic to flow between them, rules need to be added to pass packets. For the `ipfw(8)` firewall, add the following lines to the firewall configuration file:

```
ipfw add 00201 allow log esp from any to any
ipfw add 00202 allow log ah from any to any
ipfw add 00203 allow log ipencap from any to any
ipfw add 00204 allow log udp from any 500 to any
```

The rule numbers may need to be altered depending on the current host configuration.

For users of `pf(4)` or `ipf(8)`, the following rules should do the trick:

```
pass in quick proto esp from any to any
pass in quick proto ah from any to any
pass in quick proto ipencap from any to any
pass in quick proto udp from any port = 500 to any port = 500
pass in quick on gif0 from any to any
pass out quick proto esp from any to any
pass out quick proto ah from any to any
pass out quick proto ipencap from any to any
pass out quick proto udp from any port = 500 to any port = 500
pass out quick on gif0 from any to any
```

Finally, to allow the machine to start support for the VPN during system initialization, add the following lines to `/etc/rc.conf`:

```
ipsec_enable="YES"
ipsec_program="/usr/local/sbin/setkey"
ipsec_file="/usr/local/etc/racoon/setkey.conf" # allows setting up spd policies on boot
```
OpenSSH 是一套网络工具，可安全地存取远程主机，此外，透过 SSH 连线可以建立 TCP/IP 通道或安全的封包传送。OpenSSH 会对所有传输的资料做加密，有效地避免窃听 (Eavesdropping)、连线劫持 (Connection hijacking) 与其他网络层的攻击。

OpenSSH 由 OpenBSD 专案所维护且在 FreeBSD 预设会安装，它可同时相容 SSH 版本 1 与 2 通讯协定。

当以未加密的方式在网络上传送资料时，任何在客户端与伺服器之间的网络窃听程式 (Network sniffer) 皆可窃取使用者/密码资讯或者在连线阶段传送的资料，OpenSSH 提供了多种认证与加密方式来避免这种事情发生。更多有关 OpenSSH 的资讯可於 http://www.openssh.com/ 获得。

本节会简单介绍如何使用内建的客户端工具安全地存取其他系统及安全的传输档案到 FreeBSD 系统，然后会说明如何设定在 FreeBSD 系统上的 SSH 伺服器。更多资讯可於本章节所提及的操作手册 (Man page) 获得。

13.8.1. 使用 SSH 客户端工具

要登入一台 SSH 伺服器，可使用 `ssh` 然后指定在伺服器上存储的使用者名称与 IP 位址或伺服器的主机名。若这是第一次连线到指定的伺服器，会提示该使用者伺服器的指纹做第一次检验：

```bash
# ssh user@example.com
The authenticity of host 'example.com (10.0.0.1)' can't be established.
Are you sure you want to continue connecting (yes/no)? yes
Permanently added 'example.com' (ECDSA) to the list of known hosts.
Password for user@example.com: user_password
```

SSH 会在客户端连线时利用金钥指纹 (Key fingerprint) 系统来验证伺服器的真实，当使用者在第一次连线时输入 yes 接受了这个金钥指纹，便会将该金钥的复本储存在使用者家目录的 `.ssh/known_hosts`，未尝试登入时便会以这个存好的金钥来验证，若伺服器的金钥与储存的金钥不同将会显示警告资讯。若出现这个警告时，使用者应在继续连线之前检查金钥变动的原因为。

最近版本的 OpenSSH 预设只会接受 SSHv2 的连线。客户端预设会尽可能使用版本 2 的通讯协定，若伺服器不支援版本 2 的通讯协定便会向下相容版本 1 的协定。要强制 `ssh` 只能使用指定的通讯协定，可使用 `-1` 或 `-2`，其他的选项在 `ssh(1)` 中有说明。

使用 `scp(1)` 可从远程主机安全的複製一个档案，以下範例会複製在远程主机的 COPYRIGHT 到本地主机的目前目录：

```bash
# scp user@example.com:/COPYRIGHT COPYRIGHT
Password for user@example.com: ********
COPYRIGHT            100% |
*****************************
|  4735
00:00 #
```

由於这个主机的指纹已验证过，在提示使用者输入密码之前伺服器的金钥已自动检查。
傳給 scp 的參數與傳給 cp 的參數相似。第一個參數是要複製的檔案,第二個參數是目的地，由於檔案是透過網路取得,檔案參數需要用 user@host:<path_to_remote_file> 格式。注意,在 scp 要遞迴複製目錄是使用 -r,如同 cp 使用 -R。

要開啓可互動的連線來複製檔案可使用 sftp,請参考 sftp(1)來取得在 sftp 連線時可用的指令清單。

13.8.1.1.以金鑰為基礎的認證

除了使用密碼之外,客戶端可以設定成使用金鑰來連線到遠端的主機。要產生 RSA 證金鑰可使用 ssh-keygen。要產生成對的公鑰與私鑰,可指定金鑰的類型並依提示操作。建議使用容易記住但較難猜出的密碼來保護這個金鑰。

```
% ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):①
Enter same passphrase again:②
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:SHA256:54Xm9Uvtv6H4NOo6yjP/YCfODryvUU7yWHzMqeXwhq8 user@host.example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|                 |
|        . o..    |
|       .S*+*o    |
|      . O=Oo . . |
|       = Oo= oo..|
|      .oB.* +.oo.|
|       =OE**.o..=|
+----[SHA256]-----+
```

在此輸入密碼,密碼不可含空白或符號。

再輸入一次密碼驗證。

私鑰會儲存於 ~/.ssh/id_rsa 而公鑰會儲存於 ~/.ssh/id_rsa.pub。公鑰必須複製到遠端主機的 ~/.ssh/authorized_keys 來讓以金鑰為基礎的認証可以運作。

許多人認為金鑰的設計是安全的並在產生金鑰時未使用密碼,這樣的行為其實很危險。管理者可以手動查看私鑰來檢查金鑰對是否受密碼保護,如果私鑰檔案中包含 ENCRYPTED 字詞,則代表金鑰的擁有者有使用密碼。此外,要更進一步保護最終使用者的安全,可在公鑰檔案中放入 from,例如,在 ssh-rsa 前加上 from="192.168.10.5" 將只允許指定的使用者由該 IP 位址登入。

不同版本 OpenSSH 的選項與檔案會不同,要避免發生問題請參考 ssh-keygen(1)。

若使用了密碼,在每次連線到伺服器時都會提示使用者輸入密碼。要將 SSH 金鑰載入到記憶體並讓每次連線時不必再輸入密碼,可使用 ssh-agent(1)與 ssh-add(1)。
SSH Agent

SSH agent is available to manage, as long as you load your private key into it. SSH agent is available to execute other applications, such as Shell or window management.

To use SSH agent in Shell, use Shell as an argument to start it.

Execute `ssh-add` to add the identity, and then enter the private key's password. The user will be able to connect to any machine that has installed the corresponding public key, for example:

```
% ssh-agent
% ssh-add
Enter passphrase for key '/usr/home/user/.ssh/id_rsa':
```

```
Identity added: /usr/home/user/.ssh/id_rsa (/usr/home/user/.ssh/id_rsa)
%
```

```
① Input the key's password.
```

To use SSH agent in Xorg, you can add a setting item to `.xinitrc`, which will allow SSH agent to provide service for all programs executed in Xorg.

 `.xinitrc` example:

```
exec ssh-agent startxfce4
```

This will run in the background of Xorg every time it is started, and will execute `ssh-agent` before executing XFCE. However, every time Xorg is restarted, you need to run `ssh-add` to load all SSH keys.

13.8.1.2 SSH Tunnel

OpenSSH can build a tunnel (Tunnel) to encrypt other protocols over an encrypted connection.

The following command will tell `ssh` to build a tunnel for `telnet`:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
% 
```

This example uses the following options:

- `-2` forces `ssh` to use version 2 of the protocol.
- `-N` represents no command line, only builds a tunnel. If this option is omitted, `ssh` will initialize a normal connection.
- `-f` forces `ssh` to run in the background.
- `-L` indicates that this is a local tunnel, using `localport:remotehost:remoteport` format.

`user@foo.example.com` is the remote SSH server's login name.

SSH tunnels can build a listening socket on localhost at `localport`, and then send any connection at `localport` to the remote host. In this example, the client's port 5023 will be sent to the remote machine's port 23, as Port 23 is used by `telnet`, so this will build an encrypted `telnet` connection.

This method can be used to wrap many insecure TCP protocols, such as SMTP, POP3, and FTP, as shown in the example.
例31. 建立供SMTP使用的安全通道

% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com 's password: ******

% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTP

這可配合ssh-keygen與另一個使用者帳號來建立一個更無縫的SSH通道路環境，可使用金鑰來代替手動輸入密碼，然後該通路便可以另一個使用者執行。

例32. 安全存取POP3伺服器

在這個例子中有一個SSH伺服器會接受來自外部的連線，在同個網段下有一個郵件伺服器執行POP3伺服器。要使用較安全的方式檢查有沒新郵件可建立一個SSH連線到SSH伺服器然後透過通道連線到郵件伺服器:

% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com 's password: ******

但通道啟動並執行後，指定郵件客户端將POP3請求傳送到localhost的Port 2110,這個連線將會被安全的透過通道轉送到mail.example.com。

例33. 跳過防火牆

有些防火牆會同時過濾傳入與傳出的連線。例如，防火牆很可能會限制來自遠端主機只能存取Port 22與80來只讓SSH與網頁瀏覽器連線，這會使得Port使用22或80以外的服務無法存取。

這問題的解決方是建立一個SSH連線到在防火牆防護之外主機然後使用該連線的通道連線到想想要使用的服務:

% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org 's password: ******

在這個例子中，串流Ogg Vorbis客戶端現在可以指向localhost Port 8888,連線將會被轉送到music.example.com於Port 8000,成功地跳過防火牆。

13.8.2. 開啟SSH伺服器

除了提供內建的SSH客戶端工具外，還可以設定FreeBSD系統為一個SSH伺服器，以接受來自其他SSH客戶端的連線。

要查看sshd是否正在運作，可使用service(8)指令:
若服务 sshd 未执行，请加入下行到 /etc/rc.conf。

```
sshd_enable="YES"
```

这会让下次系统开机关动 OpenSSH 的Daemon程式 sshd。若要立即启动：

```
# service sshd start
```

在 FreeBSD 系统第一次启动 sshd 时会自动产生系统的主机金钥且会显示在 Console 上，这个金钥可供使用者在第一次连线到伺服器时验证用。

请参考 sshd(8) 可取得在启动 sshd 时可用选项及更多完整有关认证、登入程序与各种设定档的资讯。

现在，sshd 应可供所有在系统上有使用者名称及密码的使用者使用。

13.8.3. SSH 服务器安全性

在 FreeBSD 广泛使用 sshd 做为远程管理基础设施的同时，所有暴露在公网路上的系统也会时常受到暴力攻击（Brute force attack）与路过的攻击（Drive by attack）。在本节会介绍一些可用来避免这些攻击的参数。

使用在 OpenSSH 伺服器设定档的 AllowUsers 关键字限制可以登入到 SSH 伺服器的使用者及来源是一个不错的方式。

例如要只允许来自 192.168.1.32 的 root 登入，可加入下行到 /etc/ssh/sshd_config：

```
AllowUsers root@192.168.1.32
```

要允许来自任何地方的 admin 登入，可只列出使用者名称，不指定 IP 位址：

```
AllowUsers admin
```

有多位使用者也应列在同一行，例如：

```
AllowUsers root@192.168.1.32 admin
```

在对 /etc/ssh/sshd_config 做完变更后，执行以下指令告诉 sshd 重新载入设定档：

```
# service sshd reload
```

在使用了这个关键字时，列出每一位置需要登入此主机的使用者很重要，任何未被在该行指定的使用者将无法登入。同时，在 OpenSSH 伺服器设定档使用的关键字是区分大小写的，若关键字未正确的拼写（含其大小写），则会将被忽略，永远要记得测试对这个档案所做的更改来确保伺服器有如预期的方式运作。请参考 sshd_config(5) 来检查拼写及可用的关键字。
此外，使用者可能被强制要透过公钥与私钥使用双重认证（Two factor authentication）。当需要时，使用者可以透过使用 `ssh-keygen(1)` 生成一组金钥然后将公钥传送给管理者，这个金钥档会如以上在客户端章节所述被放在 `authorized_keys`。要强制使用者只能使用这个金钥，可能需要设定以下选项:

```
AuthenticationMethods publickey
```

请不要将 `/etc/ssh/sshd_config` 以及 `/etc/ssh/ssh_config` 搞混（注意在第一档档名有多出个d），第一个档案用来设定服务器，而第二个档案用来设定客户端。请参考 `ssh_config(5)` 来取得可用的客户端设定清单。

### 13.9.1. 开启 ACL 支援

ACLs are enabled by the mount-time administrative flag, `acls`, which may be added to `/etc/fstab`. The mount-time flag can also be automatically set in a persistent manner using `tunefs(8)` to modify a superblock ACLs flag in the file system header. In general, it is preferred to use the superblock flag for several reasons:

- The superblock flag cannot be changed by a remount using `mount -u` as it requires a complete `umount` and fresh `mount`. This means that ACLs cannot be enabled on the root file system after boot. It also means that ACL support on a file system cannot be changed while the system is in use.
- Setting the superblock flag causes the file system to always be mounted with ACLs enabled, even if there is not an `fstab` entry or if the devices re-order. This prevents accidental mounting of the file system without ACL support.

It is desirable to discourage accidental mounting without ACLs enabled because nasty things can happen if ACLs are enabled, then disabled, then re-enabled without flushing the extended attributes. In general, once ACLs are enabled on a file system, they should not be disabled, as the resulting file protections may not be compatible with those intended by the users of the system, and re-enabling ACLs may re-attach the previous ACLs to files that have since had their permissions changed, resulting in unpredictable behavior.

File systems with ACLs enabled will show a plus (+) sign in their permission settings:
In this example, directory1, directory2, and directory3 are all taking advantage of ACLs, whereas public_html is not.

13.9.2. 使用 ACL

File system ACLs can be viewed using `getfacl`. For instance, to view the ACL settings on test:

```
% getfacl
```

```
#file:test
#owner:1001
#group:1001
user::rw-
group::r--
other::r--
```

To change the ACL settings on this file, use `setfacl`. To remove all of the currently defined ACLs from a file or file system, include `-k`. However, the preferred method is to use `-b` as it leaves the basic fields required for ACLs to work.

```
% setfacl
```

```
-test
```

To modify the default ACL entries, use `-m`:

```
% setfacl
```

```
-u:trhodes:rwx,group:web:r--,o::---
test
```

In this example, there were no pre-defined entries, as they were removed by the previous command. This command restores the default options and assigns the options listed. If a user or group is added which does not exist on the system, an `Invalid argument` error will be displayed.

Refer to `getfacl(1)` and `setfacl(1)` for more information about the options available for these commands.

13.10. 監視第三方安全性問題

In recent years, the security world has made many improvements to how vulnerability assessment is handled. The threat of system intrusion increases as third party utilities are installed and configured for virtually any operating system available today.

Vulnerability assessment is a key factor in security. While FreeBSD releases advisories for the base system, doing so for every third party utility is beyond the FreeBSD Project’s capability. There is a way to mitigate third party vulnerabilities and warn administrators of known security issues. A FreeBSD add on utility known as pkg includes options explicitly for this purpose.

280
pkg polls a database for security issues. The database is updated and maintained by the FreeBSD
Security Team and ports developers.

Please refer to instructions for installing pkg.

Installation provides periodic(8) configuration files for maintaining the pkg audit database, and
provides a programmatic method of keeping it updated. This functionality is enabled if
daily_status_security_pkgaudit_enable is set to YES in periodic.conf(5).

Ensure that daily security run emails, which are sent to root's email account, are being read.

After installation, and to audit third party utilities as part of the Ports Collection at any time, an
administrator may choose to update the database and view known vulnerabilities of installed
packages by invoking:

```
# pkg audit -F
```

pkg displays messages any published vulnerabilities in installed packages:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <https://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-
0001020eed82.html>
```

1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.

By pointing a web browser to the displayed URL, an administrator may obtain more information
about the vulnerability. This will include the versions affected, by FreeBSD port version, along with
other web sites which may contain security advisories.

pkg is a powerful utility and is extremely useful when coupled with ports-mgmt/portmaster.

13.11. FreeBSD 安全報告

Like many producers of quality operating systems, the FreeBSD Project has a security team which is
responsible for determining the End-of-Life (EoL) date for each FreeBSD release and to provide
security updates for supported releases which have not yet reached their EoL. More information
about the FreeBSD security team and the supported releases is available on the
FreeBSD security page.

One task of the security team is to respond to reported security vulnerabilities in the FreeBSD
operating system. Once a vulnerability is confirmed, the security team verifies the steps necessary
to fix the vulnerability and updates the source code with the fix. It then publishes the details as a
"Security Advisory". Security advisories are published on the FreeBSD website and mailed to the
freebsd-security-notifications, freebsd-security, and freebsd-announce mailing lists.

This section describes the format of a FreeBSD security advisory.

13.11.1. 安全報告的格式

Here is an example of a FreeBSD security advisory:

```
============================================================================
```
FreeBSD-SA-14:04.bind                                       Security Advisory
The FreeBSD Project
Topic:          BIND remote denial of service vulnerability
Category:       contrib
Module:         bind
Announced:      2014-01-14
Credits:        ISC
Affects:        FreeBSD 8.x and FreeBSD 9.x
2014-01-14 19:38:37 UTC (stable/8, 8.4-STABLE)
2014-01-14 19:42:28 UTC (releng/8.4, 8.4-RELEASE-p7)
CVE Name:       CVE-2014-0591
For general information regarding FreeBSD Security Advisories, including descriptions of the fields above, security branches, and the following sections, please visit <URL:http://security.FreeBSD.org/>.

I.   Background
BIND 9 is an implementation of the Domain Name System (DNS) protocols. The named(8) daemon is an Internet Domain Name Server.

II.  Problem Description
Because of a defect in handling queries for NSEC3-signed zones, BIND can crash with an "INSIST" failure in name.c when processing queries possessing certain properties. This issue only affects authoritative nameservers with at least one NSEC3-signed zone. Recursive-only servers are not at risk.

III. Impact
An attacker who can send a specially crafted query could cause named(8)
to crash, resulting in a denial of service.

IV. Workaround
No workaround is available, but systems not running authoritative DNS service with at least one NSEC3-signed zone using named(8) are not vulnerable.

V. Solution
Perform one of the following:
1) Upgrade your vulnerable system to a supported FreeBSD stable or release / security branch (releng) dated after the correction date.
2) To update your vulnerable system via a source code patch:
The following patches have been verified to apply to the applicable FreeBSD release branches.
a) Download the relevant patch from the location below, and verify the detached PGP signature using your PGP utility.

[FreeBSD 8.3, 8.4, 9.1, 9.2-RELEASE and 8.4-STABLE]
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-release.patch
# gpg --verify bind-release.patch.asc

[FreeBSD 9.2-STABLE]
# gpg --verify bind-stable-9.patch.asc

b) Execute the following commands as root:
# cd /usr/src
# patch < /path/to/patch

Recompile the operating system using buildworld and installworld as described in <URL:https://www.FreeBSD.org/handbook/makeworld.html>.

Restart the applicable daemons, or reboot the system.
To update your vulnerable system via a binary patch:

Systems running a RELEASE version of FreeBSD on the i386 or amd64 platforms can be updated via the `freebsd-update(8)` utility:

```
# freebsd-update fetch
# freebsd-update install
```

VI. Correction details

The following list contains the correction revision numbers for each affected branch.

<table>
<thead>
<tr>
<th>Branch/path</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>stable/8/</td>
<td>r260646</td>
</tr>
<tr>
<td>releng/8.3/</td>
<td>r260647</td>
</tr>
<tr>
<td>releng/8.4/</td>
<td>r260647</td>
</tr>
<tr>
<td>stable/9/</td>
<td>r260646</td>
</tr>
<tr>
<td>releng/9.1/</td>
<td>r260647</td>
</tr>
<tr>
<td>releng/9.2/</td>
<td>r260647</td>
</tr>
</tbody>
</table>

To see which files were modified by a particular revision, run the following command, replacing NNNNNN with the revision number, on a machine with Subversion installed:

```
# svn diff -cNNNNN --summarize svn://svn.freebsd.org/base
```

Or visit the following URL, replacing NNNNNN with the revision number:

```
URL: https://svnweb.freebsd.org/base?view=revision&revision=NNNNNN
```

VII. References

```
URL: https://kb.isc.org/article/AA-01078
URL: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0591
```

The latest revision of this advisory is available at

```
URL: http://security.FreeBSD.org/advisories/FreeBSD-SA-14:04.bind.asc
```

---

-----BEGIN PGP SIGNATURE-----
284
Every security advisory uses the following format:

• Each security advisory is signed by the PGP key of the Security Officer. The public key for the Security Officer can be verified at OpenPGP.

• The name of the security advisory always begins with FreeBSD-SA- (for FreeBSD Security Advisory), followed by the year in two digit format (14:), followed by the advisory number for that year (04.), followed by the name of the affected application or subsystem (bind). The advisory shown here is the fourth advisory for 2014 and it affects BIND.

• The Topic field summarizes the vulnerability.

• The Category refers to the affected part of the system which may be one of core, contrib, or ports. The core category means that the vulnerability affects a core component of the FreeBSD operating system. The contrib category means that the vulnerability affects software included with FreeBSD, such as BIND. The ports category indicates that the vulnerability affects software available through the Ports Collection.

• The Module field refers to the component location. In this example, the bind module is affected; therefore, this vulnerability affects an application installed with the operating system.

• The Announced field reflects the date the security advisory was published. This means that the security team has verified that the problem exists and that a patch has been committed to the FreeBSD source code repository.

• The Credits field gives credit to the individual or organization who noticed the vulnerability and reported it.

• The Affects field explains which releases of FreeBSD are affected by this vulnerability.

• The Corrected field indicates the date, time, time offset, and releases that were corrected. The section in parentheses shows each branch for which the fix has been merged, and the version number of the corresponding release from that branch. The release identifier itself includes the version number and, if appropriate, the patch level. The patch level is the letter p followed by a number, indicating the sequence number of the patch, allowing users to track which patches have already been applied to the system.

• The CVE Name field lists the advisory number, if one exists, in the public cve.mitre.org security vulnerabilities database.

• The Background field provides a description of the affected module.

• The Problem Description field explains the vulnerability. This can include information about the flawed code and how the utility could be maliciously used.
The Impact field describes what type of impact the problem could have on a system.

The Workaround field indicates if a workaround is available to system administrators who cannot immediately patch the system.

The Solution field provides the instructions for patching the affected system. This is a step by step tested and verified method for getting a system patched and working securely.

The Correction Details field displays each affected Subversion branch with the revision number that contains the corrected code.

The References field offers sources of additional information regarding the vulnerability.

13.12. 程序追蹤

Process accounting is a security method in which an administrator may keep track of system resources used and their allocation among users, provide for system monitoring, and minimally track a user's commands.

Process accounting has both positive and negative points. One of the positives is that an intrusion may be narrowed down to the point of entry. A negative is the amount of logs generated by process accounting, and the disk space they may require. This section walks an administrator through the basics of process accounting.

If more fine-grained accounting is needed, refer to [安全事件稽查].

13.12.1. 開啟並使用程序追蹤

Before using process accounting, it must be enabled using the following commands:

```bash
# sysrc accounting_enable=yes
# service accounting start
```

The accounting information is stored in files located in `/var/account`, which is automatically created, if necessary, the first time the accounting service starts. These files contain sensitive information, including all the commands issued by all users. Write access to the files is limited to root, and read access is limited to root and members of the wheel group. To also prevent members of wheel from reading the files, change the mode of the `/var/account` directory to allow access only by root.

Once enabled, accounting will begin to track information such as CPU statistics and executed commands. All accounting logs are in a non-human readable format which can be viewed using `sa`.

If issued without any options, `sa` prints information relating to the number of per-user calls, the total elapsed time in minutes, total CPU and user time in minutes, and the average number of I/O operations. Refer to `sa(8)` for the list of available options which control the output.

To display the commands issued by users, use `lastcomm`. For example, this command prints out all usage of `ls` by trhodes on the ttyp1 terminal:

```bash
# lastcomm ls trhodes ttyp1
```

Many other useful options exist and are explained in `lastcomm(1)`, `acct(5)`, and `sa(8)`.

13.13. 限制資源

FreeBSD provides several methods for an administrator to limit the amount of system resources an individual may use. Disk quotas limit the amount of disk space available to users. Quotas are discussed in [磁碟配額].

286
Limits to other resources, such as CPU and memory, can be set using either a flat file or a command to configure a resource limits database. The traditional method defines login classes by editing /etc/login.conf. While this method is still supported, any changes require a multi-step process of editing this file, rebuilding the resource database, making necessary changes to /etc/master.passwd, and rebuilding the password database. This can become time consuming, depending upon the number of users to configure.

rctl can be used to provide a more fine-grained method for controlling resource limits. This command supports more than user limits as it can also be used to set resource constraints on processes and jails.

This section demonstrates both methods for controlling resources, beginning with the traditional method.

13.13.1. 設定登入類別

In the traditional method, login classes and the resource limits to apply to a login class are defined in /etc/login.conf. Each user account can be assigned to a login class, where default is the default login class. Each login class has a set of login capabilities associated with it. A login capability is a name=value pair, where name is a well-known identifier and value is an arbitrary string which is processed accordingly depending on the name.

Whenever /etc/login.conf is edited, the /etc/login.conf.db must be updated by executing the following command:

```
# cap_mkdb /etc/login.conf
```

Resource limits differ from the default login capabilities in two ways. First, for every limit, there is a soft and hard limit. A soft limit may be adjusted by the user or application, but may not be set higher than the hard limit. The hard limit may be lowered by the user, but can only be raised by the superuser. Second, most resource limits apply per process to a specific user.

登入類別限制資源類型 lists the most commonly used resource limits. All of the available resource limits and capabilities are described in detail in login.conf(5).

<table>
<thead>
<tr>
<th>限制</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>coredumpsize</td>
<td>The limit on the size of a core file generated by a program is subordinate to other limits on disk usage, such as filesize or disk quotas. This limit is often used as a less severe method of controlling disk space consumption. Since users do not generate core files and often do not delete them, this setting may save them from running out of disk space should a large program crash.</td>
</tr>
<tr>
<td>cputime</td>
<td>The maximum amount of CPU time a user's process may consume. Offending processes will be killed by the kernel. This is a limit on CPU time consumed, not the percentage of the CPU as displayed in some of the fields generated by top and ps.</td>
</tr>
<tr>
<td>filesize</td>
<td>The maximum size of a file the user may own. Unlike disk quotas, this limit is enforced on individual files, not the set of all files a user owns.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>限制</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>coredumpsize</td>
<td>The limit on the size of a core file generated by a program is subordinate to other limits on disk usage, such as filesize or disk quotas. This limit is often used as a less severe method of controlling disk space consumption. Since users do not generate core files and often do not delete them, this setting may save them from running out of disk space should a large program crash.</td>
</tr>
<tr>
<td>cputime</td>
<td>The maximum amount of CPU time a user's process may consume. Offending processes will be killed by the kernel. This is a limit on CPU time consumed, not the percentage of the CPU as displayed in some of the fields generated by top and ps.</td>
</tr>
<tr>
<td>filesize</td>
<td>The maximum size of a file the user may own. Unlike disk quotas, this limit is enforced on individual files, not the set of all files a user owns.</td>
</tr>
</tbody>
</table>
maxproc
The maximum number of foreground and background processes a user can run. This limit may not be larger than the system limit specified by `kern.maxproc`. Setting this limit too small may hinder a user's productivity as some tasks, such as compiling a large program, start lots of processes.

memorylocked
The maximum amount of memory a process may request to be locked into main memory using `mlock(2)`. Some system-critical programs, such as `amd(8)`, lock into main memory so that if the system begins to swap, they do not contribute to disk thrashing.

memoryuse
The maximum amount of memory a process may consume at any given time. It includes both core memory and swap usage. This is not a catch-all limit for restricting memory consumption, but is a good start.

openfiles
The maximum number of files a process may have open. In FreeBSD, files are used to represent sockets and IPC channels, so be careful not to set this too low. The system-wide limit for this is defined by `kern.maxfiles`.

sbsize
The limit on the amount of network memory a user may consume. This can be generally used to limit network communications.

stacksize
The maximum size of a process stack. This alone is not sufficient to limit the amount of memory a program may use, so it should be used in conjunction with other limits.

There are a few other things to remember when setting resource limits:
• Processes started at system startup by `/etc/rc` are assigned to the `daemon` login class.
• Although the default `/etc/login.conf` is a good source of reasonable values for most limits, they may not be appropriate for every system. Setting a limit too high may open the system up to abuse, while setting it too low may put a strain on productivity.
• `Xorg` takes a lot of resources and encourages users to run more programs simultaneously.
• Many limits apply to individual processes, not the user as a whole. For example, setting `openfiles` to 50 means that each process the user runs may open up to 50 files. The total amount of files a user may open is the value of `openfiles` multiplied by the value of `maxproc`. This also applies to memory consumption.

For further information on resource limits and login classes and capabilities in general, refer to `cap_mkdb(1)`, `getrlimit(2)`, and `login.conf(5)`.
Once the system has rebooted into the new kernel, rctl may be used to set rules for the system. Rule syntax is controlled through the use of a subject, subject-id, resource, and action, as seen in this example rule:

```
user:trhodes:maxproc:deny=10/user
```

In this rule, the subject is `user`, the subject-id is `trhodes`, the resource, `maxproc`, is the maximum number of processes, and the action is `deny`, which blocks any new processes from being created. This means that the user, `trhodes`, will be constrained to no greater than 10 processes. Other possible actions include logging to the console, passing a notification to `devd(8)`, or sending a `sigterm` to the process.

Some care must be taken when adding rules. Since this user is constrained to 10 processes, this example will prevent the user from performing other tasks after logging in and executing a `screen` session. Once a resource limit has been hit, an error will be printed, as in this example:

```
% man test
/usr/bin/man: Cannot fork: Resource temporarily unavailable
eval: Cannot fork: Resource temporarily unavailable
```

As another example, a jail can be prevented from exceeding a memory limit. This rule could be written as:

```
# rctl -a jail:httpd:memoryuse:deny=2G/jail
```

Rules will persist across reboots if they have been added to `/etc/rctl.conf`. The format is a rule, without the preceding command. For example, the previous rule could be added as:

```
# Block jail from using more than 2G memory:
jail:httpd:memoryuse:deny=2G/jail
```

To remove a rule, use `rctl` to remove it from the list:

```
# rctl -r user:trhodes:maxproc:deny=10/user
```

A method for removing all rules is documented in `rctl(8)`. However, if removing all rules for a single user is required, this command may be issued:

```
# rctl -r user:trhodes
```

Many other resources exist which can be used to exert additional control over various subjects. See `rctl(8)` to learn about them.

13.14. 使用 Sudo 分享管理权限

系统管理者通常会要能够授予额外的权限给其他使用者，以便这些使用者可以执行需权限的工作。要让团队成员可以存取FreeBSD系统的权限对所有管理者都会带来挑战，这些团队成员通常只需要比一般使用者多出一些存取权限。
權限便可作業，但他們總是會告訴管理者若沒超級使用者的存取權便無法完成其工作。幸好，有工具可以管理這類的需求，這樣便不需提供這麼大的權限給一般使用者。

到目前為止，安全性章節已說明了如何允許已授權的使用者存取以及嘗試防止未經授權的存取，而現在有另一個問題，是由已授權的使用者擁有權限存取系統資源造成的。在很多的情況下，使用者會需要存取應用程式啟動Script的權限，或是管理者團隊需要維護系統，以往會使用標準的使用者與群組、檔案權限，甚至是最Direct指令來管理存取權，但當應用程式需要更多的存取權，更多的使用者需要使用系統資源時，便需要更好的解決方案，目前最常用來解決此問題的應用程式便是Sudo。

Sudo讓管理者可以對系統指令的存取設定更嚴格的限制並提供進階的記錄功能。如同其他工具，它可自Port套件組取得，在其中的security/sudo，或使用pkg(8)工具取得，若要使用pkg(8)工具可:

```
# pkg install sudo
```

安裝完成之後，可用安裝的visudo以文字編輯器開啟設定檔，強烈建議使用visudo來編輯設定檔，因它有內建的語法檢查程式可在檔案儲存之前檢查是否有誤。

設定檔由個小節所組成，透過這些小節可做常廣泛的設定，在以下的範例中，網站應用程式維護人員user1需要啓動、停止與重新啓動名稱為webservice的網站應用程式。要授權此使用者執行這些工作權限，可加入此行到/usr/local/etc/sudoers的最後:

```
user1   ALL=(ALL)       /usr/sbin/service webservice *
```

現在使用者可使用此指令來啓動webservice:

```
%sudo /usr/sbin/service webservice start
```

雖然這項設定可以讓一位使用者存取webservice服務，但在大部份組織中會有一整個網站小組負責管理該服務，因此也可以一行來授予整個群組存取權，以下步驟會建立一個網站群組、加入使用者到這個群組，然後讓該群組中的所有成員能夠管理服

```
# pw groupadd -g 6001 -n webteam
```

```
# pw groupmod -m user1 -n webteam
```

最後，在/usr/local/etc/sudoers中的這行設定可以讓webteam群組的所有成員可以管理webservice:

```
%webteam   ALL=(ALL)       /usr/sbin/service webservice *
```

與su(1)不同的是Sudo只需要一般使用者的密碼，這有一個使用者不需要共用密碼的優點，在大多數安全稽查都會發現共用密碼的問題且這種情況只有壞處可言。

使用Sudo允許使用者執行應用程式只需要輸入使用者自己的密碼，這更安全且提供比su(1)更佳的控制權，因為su(1)只要輸入root密碼之後該使用者便可取得所有的root權限。

大多數組織已正導入或已導入雙重認證（Two factor authentication）
In this context, users can use the NOPASSWD variable provided by Sudo to manage the service without entering a password. For example:

```
%webteam   ALL=(ALL)       NOPASSWD: /usr/sbin/service webservice *
```

This setting will allow all members of the webteam group to manage the service without entering a password.

Logging using Sudo has another advantage: it can enable recording of connection stages. By default, Sudo records all commands initiated in a recording file for later inspection. To enable this feature, you can add the following setting:

```
Defaults iolog_dir=/var/log/sudo-io/%{user}
```

This logging will be automatically created after the setting is enabled, and it is recommended to use the default permission to ensure security. This setting also records the Sudo replay administrator. To change the setting, read and cancel the comment in the sudoers file.

Once this setting is added to the sudoers file, all user setting items can be added to the recording. Apart from the webteam setting, the following example is added:

```
%webteam ALL=(ALL) NOPASSWD: LOG_INPUT: LOG_OUTPUT: /usr/sbin/service webservice *
```

From this point onwards, all changes to the webservice application state by members of the webteam will be recorded. To list previous and current connection stage recordings:

```
# sudoreplay -l
```

In the output results, you can re-run the specified connection stage recording by searching for the TSID= item. For example:

```
# sudoreplay user1/00/00/02
```

Although all connection stages will be recorded, any administrator can remove the connection stage, making it unknown what they did, so it is very useful to use an intrusion detection system (IDS) or similar software to check every day, so that other administrators are notified when changes are made.

Sudoreplay has a very large expansion space, please refer to the manual for more information.
以下列出将术语：本章使用的 FreeBSD 及它们关系统有，系互其他部分的相与协助 Jail 更作用解 FreeBSD 为容。14.2. Jail 关术语

读完这章，您解：来皆的使用者说系统管理者强来及。大的对进是工具指令 FreeBSD Jail 术语及管理的概将 Jail 提供，本章述环境执之外的系统己及使用者有自 Jail 自的四：素环境存可业是一种作。系统的一取方式，的型的 Jail 微细这个更网控制案系统、使用者多数来调校过虚拟化存透子系统取来档及模扩展路，可使用的程序系统所共 Jail 主系统 chroot 及机。享行的程序以余，程序地源环境仅限网存在一部份执路制及子案系统可方。其传 chroot 统用环境许改 Jail 的概念。在 chroot 传进了的理找到，以再安让确保不法 chroot 环境全离已经务逃这个方的方想的方案。被是服的适合阶功，只间用在作。弹随有，著时复杂简单推移作，不需要多可性限许或多工但。

chroot 着透渗让执行的个系统攻务环境行的作者被并不会有档立所取该一个在环境存 chroot 案或资 chroot 渗的环境。也因此，源建透，会目上环境根概念之更改程序的，全。这可以创将一个安。在造录的隔一系建在立 Jail 功便能、定性以 FreeBSD 4.X 稳全性率效的。强化它安及可使用并持续之一的方式。这些是就工具维护设定系统安以全性装、护及之可以用来强化 FreeBSD 更通常可以工具开发来让作，由系统管理者能。这些多困项强化难的许系统安于工系统管理是一够。这些是就工具管理基础，不不论从内部或外部。
chroot(2)

(環境)
指程序在“chroot”中執行的環境。包含的資源如：部份可見的檔案系統、可用的使用者及群組ID、網路介面及其他IPC機制等。

jail(8)
(指令)
允許在Jail環境下執行程序的系統管理工具。

主要機(系統、程序、使用者等)
Jail環境的控制系統。主要機系統可以存取所有可用的硬體資源，並能控制Jail環境內外的程序。主要機系統與Jail最大的差別在於：在主要機系統中的超級使用者程序並不像在Jail環境那樣受到限制。

託管(主要機、程序、使用者等)
存取資源受到FreeBSD Jail限制的託管程序、使用者或其他實體。

14.3.建立和控制Jail
部份管理者將Jail分成兩種類型：“完整的”Jail，它像一個真正的FreeBSD系統以及“服務的”Jail，專門用於某個應用程式或服務，可能使用管理權限執行。但這些只是概念上區分，建立Jail的程序並不受這個概念的影響。當要建立一個“完整的”Jail，Userland有兩個來源選項：使用預先編譯的Binary (如安裝媒體上提供的Binary)或從原始碼編譯。

要從安裝媒體安裝Userland，需要先建立根目錄供Jail使用。這個動作可以透過設定DESTDIR來到適當的位置來完成。

啟動Shell並定義DESTDIR:
```
# sh
# export DESTDIR=/here/is/the/jail
```

當使用安裝ISO時，可依照mdconfig(8)中的說明掛載安裝媒體:
```
# mount -t cd9660 /dev/`mdconfig -f cdimage.iso` /mnt
# cd /mnt/usr/freebsd-dist/
```

或者自鏡像站下載Tarball壓縮檔:
```
# sh
# export DESTRELEASE=12.0-RELEASE
# export DESTARCH=`uname -m`
# export SOURCEURL=http://ftp.freebsd.org/pub/FreeBSD/releases/$DESTARCH/$DESTRELEASE/
# for set in base ports; do fetch $SOURCEURL/$set.txz ; done
```

從安裝媒體上的Tarball中取出Binary並放到宣告的位置，至少需要取出Base set的部份，若需要也可完整安裝。

只安裝基礎系統(Base system):
```
# tar -xf base.txz -C $DESTDIR
```
安裝全不含核心:

```bash
# for set in base ports; do
tar -xf $set.txz -C $DESTDIR;
done
```

依jail(8)操作手冊說明的程序建立Jail:

```bash
# setenv D /here/is/the/jail
# mkdir -p $D
# cd /usr/src
# make buildworld
# make installworld DESTDIR=$D
# make distribution DESTDIR=$D
# mount -t devfs devfs $D/dev
```

① 選擇Jail的位置是建立Jail最好的起點，這是在Jail主機上儲存Jail的實體位置。

較好的選擇是/usr/jail/jailname，其中jailname是用來辨識Jail的主機名稱。

對"完整的"Jail來說，便是複製FreeBSD基礎系統預設安裝的每一個檔案。

② 若您已經使用make world或make buildworld重新編譯您的Userland，您可以跳過這個步驟並安裝您已存在的Userland到新的Jail。

③ 這個指令將會在檔案系統中Jail所在的實體位置產生樹狀目錄及必要的Binary、程式庫、操作手冊與相關檔案。

④ make的distribution目標會安裝所有需要的設定檔。簡單來說，它會安裝所有/usr/src/etc/中可安裝的檔案到Jail環境的/etc目錄:

$D/etc/。

⑤ 在Jail中掛載devfs(8)檔案系統並非必要的動作。從另一個角度來說，任何或大部份的應用程式會依該程式的目的會需要存取至少一個裝置，在Jail中控制存取的裝置非常重要，不恰當的設定可能會讓攻擊者可以在Jail中做不軌的事。

對devfs(8)的控制是透過Ruleset，在devfs(8)及devfs.conf(5)操作手冊中有詳細說明。

Jail安裝完成之後，便可使用jail(8)工具來啟動。

jail(8)工具需要四個必要參數，在概述有說明。其他參數也可能需要指定，例如要使用特定使用者的身分來執行要Jail的程序。

command參數依Jail的類型而定，對一個虛擬系統來說，/etc/rc是不錯的選擇，因為該檔案可以模彷真實FreeBSD的啟動順序。對於服務型的Jail來說，則看在Jail中要執行的服務或應用程式來決定。

Jail通常會需要隨著開機執行，使用FreeBSD rc機制可讓以簡單的達成這件事。
在 `rc.conf` 中設定開機時啟動 Jail:

```
jail_enable="YES"   # Set to NO to disable starting of any jails
```

預設要啟動的 Jail 可在 `jail.conf(5)` 設定，會把 Jail 當作是一個完全虛擬的系統，然後執行 Jail 中的 `/etc/rc` Script。

針對服務型的 Jail 則需透過設定 `exec.start` 選項來適當更改 Jail 的預設啟動指令。

要取得完整的可用選項清單，請參考 `jail.conf(5)` 操作手冊。

若 Jail 項目已經在 `jail.conf` 中設定好，可以手動用 `service(8)` 來啟動或停止某個 Jail 項目:

```
# service jail start www
# service jail stop www
```

Jail 可以使用 `jexec(8)` 來關機。先使用 `jls(8)` 來識別 Jail 的 JID，然後使用 `jexec(8)` 在該 Jail 中執行關機 Script。

```
# jls
JID  IP Address      Hostname                      Path
3  192.168.0.10    www                           /usr/jail/www
# jexec 3 /etc/rc.shutdown
```

更多有關 Jail 的資訊可在 `jail(8)` 操作手冊取得。

### 14.4.調校與管理

還有許多選項可以對所有 Jail 做設定，以及各種可讓 Jail 與主機 FreeBSD 系統結合的方

法來提供更高層級的應用程式使用。本節將介紹:

- Some of the options available for tuning the behavior and security restrictions implemented by a jail installation.
- Some of the high-level applications for jail management, which are available through the FreeBSD Ports Collection, and can be used to implement overall jail-based solutions.

### 14.4.1.在 FreeBSD 中調校 Jail

Fine tuning of a jail's configuration is mostly done by setting `sysctl(8)` variables. A special subtree of `sysctl` exists as a basis for organizing all the relevant options: the `security.jail.*` hierarchy of FreeBSD kernel options. Here is a list of the main jail-related sysctls, complete with their default value. Names should be self-explanatory, but for more information about them, please refer to the `jail(8)` and `sysctl(8)` manual pages.

- `security.jail.set_hostname_allowed`: 1
- `security.jail.socket_unixiproute_only`: 1
- `security.jail.sysvipc_allowed`: 0
- `security.jail.enforce_statfs`: 2
- `security.jail.allow_raw_sockets`: 0
- `security.jail.chflags_allowed`: 0
These variables can be used by the system administrator of the host system to add or remove some of the limitations imposed by default on the root user. Note that there are some limitations which cannot be removed. The root user is not allowed to mount or unmount file systems from within a jail. The root inside a jail may not load or unload devfs rulesets, set firewall rules, or do many other administrative tasks which require modifications of in-kernel data, such as setting the securelevel of the kernel.

The base system of FreeBSD contains a basic set of tools for viewing information about the active jails, and attaching to a jail to run administrative commands. The jls(8) and jexec(8) commands are part of the base FreeBSD system, and can be used to perform the following simple tasks:

- Print a list of active jails and their corresponding jail identifier (JID), IP address, hostname and path.
- Attach to a running jail, from its host system, and run a command inside the jail or perform administrative tasks inside the jail itself. This is especially useful when the root user wants to cleanly shut down a jail. The jexec(8) utility can also be used to start a shell in a jail to do administration in it; for example:

  # jexec 1 tcsh

14.4.2. 在 FreeBSD Port 套件中的高層級管理工具

Among the many third-party utilities for jail administration, one of the most complete and useful is sysutils/ezjail. It is a set of scripts that contribute to jail(8) management. Please refer to the handbook section on ezjail for more information.

14.4.3. 持續 Jail 的修補與更新

Jails should be kept up to date from the host operating system as attempting to patch userland from within the jail may likely fail as the default behavior in FreeBSD is to disallow the use of chflags(1) in a jail which prevents the replacement of some files. It is possible to change this behavior but it is recommended to use freebsd-update(8) to maintain jails instead. Use -b to specify the path of the jail to be updated.

  # freebsd-update -b /here/is/the/jail fetch
  # freebsd-update -b /here/is/the/jail install

14.5. 更新多個 Jail

The management of multiple jails can become problematic because every jail has to be rebuilt from scratch whenever it is upgraded. This can be time consuming and tedious if a lot of jails are created and manually updated.

This section demonstrates one method to resolve this issue by safely sharing as much as is possible between jails using read-only mount_nullfs(8) mounts, so that updating is simpler. This makes it more attractive to put single services, such as HTTP, DNS, and SMTP, into individual jails. Additionally, it provides a simple way to add, remove, and upgrade jails.

Simpler solutions exist, such as ezjail, which provides an easier method of administering FreeBSD jails but is less versatile than this setup. ezjail is covered in more detail in 使用 ezjail 管理 Jail.
Create a simple and easy to understand jail structure that does not require running a full installworld on each and every jail.

- Make it easy to add new jails or remove existing ones.
- Make it easy to update or upgrade existing jails.
- Make it possible to run a customized FreeBSD branch.
- Be paranoid about security, reducing as much as possible the possibility of compromise.
- Save space and inodes, as much as possible.

This design relies on a single, read-only master template which is mounted into each jail and one read-write device per jail. A device can be a separate physical disc, a partition, or a vnode backed memory device. This example uses read-write nullfs mounts.

The file system layout is as follows:

- The jails are based under the `/home` partition.
- Each jail will be mounted under the `/home/j` directory.
- The template for each jail and the read-only partition for all of the jails is `/home/j/mroot`.
- A blank directory will be created for each jail under the `/home/j` directory.
- Each jail will have a `/s` directory that will be linked to the read-write portion of the system.
- Each jail will have its own read-write system that is based upon `/home/j/skel`.
- The read-write portion of each jail will be created in `/home/js`.

14.5.1. 建立範本

This section describes the steps needed to create the master template. It is recommended to first update the host FreeBSD system to the latest -RELEASE branch using the instructions in 更新 FreeBSD. Additionally, this template uses the `sysutils/cpdup` package or portsnap will be used to download the FreeBSD Ports Collection.

1. First, create a directory structure for the read-only file system which will contain the FreeBSD binaries for the jails. Then, change directory to the FreeBSD source tree and install the read-only file system to the jail template:
   ```
   # mkdir /home/j /home/j/mroot
   # cd /usr/src
   # make installworld DESTDIR=/home/j/mroot
   ```

2. Next, prepare a FreeBSD Ports Collection for the jails as well as a FreeBSD source tree, which is required for mergemaster:
   ```
   # cd /home/j/mroot
   # mkdir usr/ports
   # portsnap -p /home/j/mroot/usr/ports fetch extract
   # cpdup /usr/src /home/j/mroot/usr/src
   ```

3. Create a skeleton for the read-write portion of the system:
   ```
   # mkdir /home/j/skel /home/j/skel/home /home/j/skel/usr/X11R6
   ```
4. Use mergemaster to install missing configuration files. Then, remove the extra directories that mergemaster creates:

```
# mergemaster -t /home/j/skel/var/tmp/temproot -D /home/j/skel -i
# cd /home/j/skel
# rm -R bin boot lib libexec mnt proc rescue sbin sys usr dev
```

5. Now, symlink the read-write file system to the read-only file system. Ensure that the symlinks are created in the correct locations as the creation of directories in the wrong locations will cause the installation to fail.

```
# cd /home/j/mroot
# mkdir s
# ln -s s/etc etc
# ln -s s/home home
# ln -s s/root root
# ln -s ../s/usr-local usr/local
# ln -s ../s/usr-X11R6 usr/X11R6
# ln -s ../../s/distfiles usr/ports/distfiles
# ln -s s/tmp tmp
# ln -s s/var var
```

6. As a last step, create a generic `/home/j/skel/etc/make.conf` containing this line:

```
WRKDIRPREFIX?= /s/portbuild
```
This makes it possible to compile FreeBSD ports inside each jail. Remember that the ports directory is part of the read-only system. The custom path for `WRKDIRPREFIX` allows builds to be done in the read-write portion of every jail.
To prevent fsck from checking nullfs mounts during boot and dump from backing up the read-only nullfs mounts of the jails, the last two columns are both set to 0.

2. Configure the jails in /etc/rc.conf:
   - jail_enable="YES"
   - jail_set_hostname_allow="NO"
   - jail_list="ns mail www"
   - jail_ns_hostname="ns.example.org"
   - jail_ns_ip="192.168.3.17"
   - jail_ns_rootdir="/usr/home/j/ns"
   - jail_ns_devfs_enable="YES"
   - jail_mail_hostname="mail.example.org"
   - jail_mail_ip="192.168.3.18"
   - jail_mail_rootdir="/usr/home/j/mail"
   - jail_mail_devfs_enable="YES"
   - jail_www_hostname="www.example.org"
   - jail_www_ip="62.123.43.14"
   - jail_www_rootdir="/usr/home/j/www"
   - jail_www_devfs_enable="YES"

   The jail name rootdir variable is set to /usr/home instead of /home because the physical path of /home on a default FreeBSD installation is /usr/home. The jail name rootdir variable must not be set to a path which includes a symbolic link, otherwise the jails will refuse to start.

3. Create the required mount points for the read-only file system of each jail:
   ```
   # mkdir /home/j/ns /home/j/mail /home/j/www
   ```

4. Install the read-write template into each jail using sysutils/cpdup:
   ```
   # mkdir /home/js
   # cpdup /home/j/skel /home/js/ns
   # cpdup /home/j/skel /home/js/mail
   # cpdup /home/j/skel /home/js/www
   ```

5. In this phase, the jails are built and prepared to run. First, mount the required file systems...
for each jail, and then start them:

```
# mount -a
# service jail start
```

The jails should be running now. To check if they have started correctly, use `jls`. Its output should be similar to the following:

```
# jls
JID  IP Address      Hostname                      Path
3  192.168.3.17    ns.example.org                /home/j/ns
2  192.168.3.18    mail.example.org              /home/j/mail
1  62.123.43.14    www.example.org               /home/j/www
```

At this point, it should be possible to log onto each jail, add new users, or configure daemons. The JID column indicates the jail identification number of each running jail. Use the following command to perform administrative tasks in the jail whose JID is 3:

```
# jexec 3 tcsh
```

The design of this setup provides an easy way to upgrade existing jails while minimizing their downtime. Also, it provides a way to roll back to the older version should a problem occur.

1. The first step is to upgrade the host system. Then, create a new temporary read-only template in `/home/j/mroot2`.

```
# mkdir /home/j/mroot2
# cd /usr/src
# make installworld DESTDIR=/home/j/mroot2
# cd /home/j/mroot2
# cpdup /usr/src usr/src
```

The `installworld` creates a few unnecessary directories, which should be removed:

```
# chflags -R 0 var
# rm -R etc var root usr/local tmp
```

2. Recreate the read-write symlinks for the master file system:

```
# ln -s s/etc etc
# ln -s s/root root
```

3. After the upgrades, you may want to run `jls` again to check the status of the jails. You should see the updated configuration.
3. Next, stop the jails:
   
   # service jail stop

4. Unmount the original file systems as the read-write systems are attached to the read-only system (/s):
   
   # umount /home/j/ns/s
   # umount /home/j/ns
   # umount /home/j/mail/s
   # umount /home/j/mail
   # umount /home/j/www/s
   # umount /home/j/www

5. Move the old read-only file system and replace it with the new one. This will serve as a backup and archive of the old read-only file system should something go wrong. The naming convention used here corresponds to when a new read-only file system has been created. Move the original FreeBSD Ports Collection over to the new file system to save some space and inodes:
   
   # cd /home/j
   # mv mroot mroot.200601
   # mv mroot2 mroot
   # mv mroot.200601/usr/ports mroot/usr

6. At this point the new read-only template is ready, so the only remaining task is to remount the file systems and start the jails:
   
   # mount -a
   # service jail start

   Use jls to check if the jails started correctly. Run mergemaster in each jail to update the configuration files.

14.6. 使用 ezjail 管理 Jail

Creating and managing multiple jails can quickly become tedious and error-prone. Dirk Engling's ezjail automates and greatly simplifies many jail tasks. A basejail is created as a template. Additional jails use mount_nullfs(8) to share many of the basejail directories without using additional disk space. Each additional jail takes only a few megabytes of disk space before
Applications are installed. Upgrading the copy of the userland in the basejail automatically upgrades all of the other jails. Additional benefits and features are described in detail on the ezjail web site, https://erdgeist.org/.

14.6.1. Installing ezjail consists of adding a loopback interface for use in jails, installing the port or package, and enabling the service.

1. To keep jail loopback traffic off the host's loopback network interface lo0, a second loopback interface is created by adding an entry to /etc/rc.conf:

   cloned_interfaces="lo1"

   The second loopback interface lo1 will be created when the system starts. It can also be created manually without a restart:

   # service netif cloneup
   Created clone interfaces: lo1.

   Jails can be allowed to use aliases of this secondary loopback interface without interfering with the host.

   Inside a jail, access to the loopback address 127.0.0.1 is redirected to the first IP address assigned to the jail. To make the jail loopback correspond with the new lo1 interface, that interface must be specified first in the list of interfaces and IP addresses given when creating a new jail.

   Give each jail a unique loopback address in the 127.0.0.0/8 netblock.

2. Install sysutils/ezjail:

   # cd /usr/ports/sysutils/ezjail
   # make install clean

3. Enable ezjail by adding this line to /etc/rc.conf:

   ezjail_enable="YES"

4. The service will automatically start on system boot. It can be started immediately for the current session:

   # service ezjail start
With ezjail installed, the basejail directory structure can be created and populated. This step is only needed once on the jail host computer.

In both of these examples, `-p` causes the ports tree to be retrieved with `portsnap(8)` into the basejail. That single copy of the ports directory will be shared by all the jails. Using a separate copy of the ports directory for jails isolates them from the host. The ezjailFAQ explains in more detail: [http://erdgeist.org/arts/software/ezjail/#FAQ](http://erdgeist.org/arts/software/ezjail/#FAQ).

1. **To Populate the Jail with FreeBSD-RELEASE**
   For a basejail based on the FreeBSD RELEASE matching that of the host computer, use `install`. For example, on a host computer running FreeBSD 10-STABLE, the latest RELEASE version of FreeBSD -10 will be installed in the jail:
   ```bash
   # ezjail-admin install -p
   ```

2. **To Populate the Jail with `installworld`**
   The basejail can be installed from binaries created by `buildworld` on the host with `ezjail-admin update`. In this example, FreeBSD 10-STABLE has been built from source. The jail directories are created. Then `installworld` is executed, installing the host's `/usr/obj` into the basejail.
   ```bash
   # ezjail-admin update -i -p
   ```
   The host's `/usr/src` is used by default. A different source directory on the host can be specified with `-s` and a path, or set with `ezjail_sourcetree` in `/usr/local/etc/ezjail.conf`.

- The basejail's ports tree is shared by other jails. However, downloaded distfiles are stored in the jail that downloaded them. By default, these files are stored in `/var/ports/distfiles` within each jail. `/var/ports` inside each jail is also used as a work directory when building ports.

- The FTP protocol is used by default to download packages for the installation of the basejail. Firewall or proxy configurations can prevent or interfere with FTP transfers. The HTTP protocol works differently and avoids these problems. It can be chosen by specifying a full URL for a particular download mirror in `/usr/local/etc/ezjail.conf`:
  ```bash
  ```
  See [FTP station](http://ftp.FreeBSD.org) for a list of sites.

14.6.3. **建立並啟動新的Jail**
New jails are created with `ezjail-admin create`. In these examples, the `lo1` loopback interface is used as described above.

Procedure: Create and Start a New Jail

---

**Note:**
- The basejail's ports tree is shared by other jails. However, downloaded distfiles are stored in the jail that downloaded them. By default, these files are stored in `/var/ports/distfiles` within each jail. `/var/ports` inside each jail is also used as a work directory when building ports.
- The FTP protocol is used by default to download packages for the installation of the basejail. Firewall or proxy configurations can prevent or interfere with FTP transfers. The HTTP protocol works differently and avoids these problems. It can be chosen by specifying a full URL for a particular download mirror in `/usr/local/etc/ezjail.conf`:
  ```bash
  ```
  See [FTP station](http://ftp.FreeBSD.org) for a list of sites.
1. Create the jail, specifying a name and the loopback and network interfaces to use, with their IP addresses. In this example, the jail is named `dnsjail`.

   ```bash
   # ezjail-admin create dnsjail 'lo1|127.0.1.1,em0|192.168.1.50'
   ```

Most network services run in jails without problems. A few network services, most notably `ping(8)`, use raw network sockets. In jails, raw network sockets are disabled by default for security. Services that require them will not work.

Occasionally, a jail genuinely needs raw sockets. For example, network monitoring applications often use `ping(8)` to check the availability of other computers. When raw network sockets are actually needed in a jail, they can be enabled by editing the `ezjail` configuration file for the individual jail, `/usr/local/etc/ezjail/jailname`. Modify the parameters entry:

   ```bash
   export jail_jailname_parameters="allow.raw_sockets=1"
   ```

Do not enable raw network sockets unless services in the jail actually require them.

2. Start the jail:

   ```bash
   # ezjail-admin start dnsjail
   ```

3. Use a console on the jail:

   ```bash
   # ezjail-admin console dnsjail
   ```

The jail is operating and additional configuration can be completed. Typical settings added at this point include:

1. Set the root Password

   Connect to the jail and set the root user's password:

   ```bash
   # ezjail-admin console dnsjail
   # passwd
   ```

   Changing local password for root

   New Password:

   Retype New Password:

2. Time Zone Configuration

   The jail's time zone can be set with `tzsetup(8)`. To avoid spurious error messages, the `adjkerntz(8)` entry in `/etc/crontab` can be commented or removed. This job attempts to update the computer's hardware clock with time zone changes, but jails are not allowed to access that hardware.
3. DNS Servers
Enter domain name server lines in /etc/resolv.conf so DNS works in the jail.

4. Edit /etc/hosts
Change the address and add the jail name to the localhost entries in /etc/hosts.

5. Configure /etc/rc.conf
Enter configuration settings in /etc/rc.conf. This is much like configuring a full computer. The host name and IP address are not set here. Those values are already provided by the jail configuration.

With the jail configured, the applications for which the jail was created can be installed.

Some ports must be built with special options to be used in a jail. For example, both of the network monitoring plugin packages net-mgmt/nagios-plugins and net-mgmt/monitoring-plugins have a JAIL option which must be enabled for them to work correctly inside a jail.

14.6.4. 更新 Jail
14.6.4.1. 更新作業系统
Because the basejail's copy of the userland is shared by the other jails, updating the basejail automatically updates all of the other jails. Either source or binary updates can be used.

To build the world from source on the host, then install it in the basejail, use:

```
# ezjail-admin update -b
```

If the world has already been compiled on the host, install it in the basejail with:

```
# ezjail-admin update -i
```

Binary updates use `freebsd-update(8)`. These updates have the same limitations as if `freebsd-update(8)` were being run directly. The most important one is that only -RELEASE versions of FreeBSD are available with this method.

Update the basejail to the latest patched release of the version of FreeBSD on the host. For example, updating from RELEASE-p1 to RELEASE-p2.

```
# ezjail-admin update -u
```

To upgrade the basejail to a new version, first upgrade the host system as described in 执行主要及次要版本升级. Once the host has been upgraded and rebooted, the basejail can then be upgraded.

`freebsd-update(8)` has no way of determining which version is currently installed in the basejail, so the original version must be specified. Use `file(1)` to determine the original version in the basejail:

```
# file /usr/jails/basejail/bin/sh
/usr/jails/basejail/bin/sh: ELF 64-bit LSB executable, x86-64, version 1 (FreeBSD), 305
```
Now use this information to perform the upgrade from 9.3-RELEASE to the current version of the host system:

```
# ezjail-admin update -U -s 9.3-RELEASE
```

After updating the basejail, `mergemaster(8)` must be run to update each jail's configuration files. How to use `mergemaster(8)` depends on the purpose and trustworthiness of a jail. If a jail's services or users are not trusted, then `mergemaster(8)` should only be run from within that jail:

```
例 34.  在不信任的Jail做mergemaster(8)
```

Delete the link from the jail's `/usr/src` into the basejail and create a new `/usr/src` in the jail as a mountpoint. Mount the host computer's `/usr/src` read-only on the jail's new `/usr/src` mountpoint:

```
# rm /usr/jails/jailname/usr/src
# mkdir /usr/jails/jailname/usr/src
# mount -t nullfs -o ro /usr/src /usr/jails/jailname/usr/src
```

Get a console in the jail:

```
# ezjail-admin console jailname
```

Inside the jail, run `mergemaster`. Then exit the jail console:

```
# cd /usr/src
# mergemaster -U
# exit
```

Finally, unmount the jail's `/usr/src`:

```
# umount /usr/jails/jailname/usr/src
```

例 35.  在信任的Jail做mergemaster(8)

If the users and services in a jail are trusted, `mergemaster(8)` can be run from the host:

```
# mergemaster -U -D /usr/jails/jailname
```

14.6.4.2. 更新 Port

The ports tree in the basejail is shared by the other jails. Updating that copy of the ports tree gives...
The base jail ports tree is updated with `portsnap(8)`:

```
# ezjail-admin update -P
```

14.6.5.

**Control Jail**

14.6.5.1.

Stop and start jails automatically.

Jails can be manually stopped and restarted with `stop` and `start`:

```
# ezjail-admin stop sambajail
```

```
Stopping jails: sambajail.
```

```
# ezjail-admin start sambajail
```

By default, jails are started automatically when the host computer starts. Autostarting can be disabled with `config`:

```
# ezjail-admin config -r norun seldomjail
```

This takes effect the next time the host computer is started. A jail that is already running will not be stopped.

Enabling autostart is very similar:

```
# ezjail-admin config -r run oftenjail
```

14.6.5.2.

**Save and Restore Jail**

Use `archive` to create a `.tar.gz` archive of a jail. The file name is composed from the name of the jail and the current date. Archive files are written to the archive directory, `/usr/jails/ezjail_archives`. A different archive directory can be chosen by setting `ezjail_archivedir` in the configuration file.

The archive file can be copied elsewhere as a backup, or an existing jail can be restored from it with `restore`. A new jail can be created from the archive, providing a convenient way to clone existing jails.

Stop and archive a jail named `wwwserver`:

```
# ezjail-admin stop wwwserver
```

```
Stopping jails: wwwserver.
```

```
# ezjail-admin archive wwwserver
```

```
# ls /usr/jails/ezjail-archives/
wwwserver-201407271153.13.tar.gz
```

Create a new jail named `wwwserver-clone` from the archive created in the previous step. Use the `em1` interface and assign a new IP address to avoid conflict with the original:
Putting the BINDDNS server in a jail improves security by isolating it. This example creates a simple caching-only name server.

• The jail will be called `dns1`.
• The jail will use IP address `192.168.1.240` on the host's `re0` interface.
• The upstream ISP's DNS servers are at `10.0.0.62` and `10.0.0.61`.
• The basejail has already been created and a ports tree installed as shown in 初始設定.

Example 36. In Jail execute BIND

Create a cloned loopback interface by adding a line to `/etc/rc.conf`:
```
cloned_interfaces="lo1"
```

Immediately create the new loopback interface:
```
# service netif cloneup
Created clone interfaces: lo1.
```

Create the jail:
```
# ezjail-admin create dns1 'lo1|127.0.2.1,re0|192.168.1.240'
```

Start the jail, connect to a console running on it, and perform some basic configuration:
```
# ezjail-admin start dns1
# ezjail-admin console dns1
# passwd
Changing local password for root
New Password:
Retype New Password:
```

Temporarily set the upstream DNS servers in `/etc/resolv.conf` so ports can be downloaded:
```
nameserver 10.0.0.62
308
```
Still using the jail console, install `dns/bind99`.

```
# make -C /usr/ports/dns/bind99 install clean
```

Configure the name server by editing `/usr/local/etc/namedb/named.conf`.

Create an Access Control List (ACL) of addresses and networks that are permitted to send DNS queries to this name server. This section is added just before the `options` section already in the file:

```bash
// or cause huge amounts of useless Internet traffic.

acl "trusted" {
  192.168.1.0/24;
  localhost;
  localnets;
};

options {
  ...
}
```

Use the jail IP address in the `listen-on` setting to accept DNS queries from other computers on the network:

```
listen-on   { 192.168.1.240; };
```

A simple caching-only DNS name server is created by changing the `forwarders` section. The original file contains:

```bash
/*
forwarders {
  127.0.0.1;
};
*/
```

Uncomment the section by removing the `//` and `/` lines. Enter the IP addresses of the upstream DNS servers. Immediately after the `forwarders` section, add references to the `trusted` ACL defined earlier:

```
forwarders {
  10.0.0.62;
  10.0.0.61;
};
```
Enable the service in /etc/rc.conf:

```
named_enable="YES"
```

Start and test the name server:

```
# service named start
wrote key file "/usr/local/etc/namedb/rndc.key"
Starting named.
```

```
# /usr/local/bin/dig @192.168.1.240 freebsd.org
Got answer
```

shows that the new DNS server is working. A long delay followed by a response including

```
connection timed out
no servers could be reached
```

shows a problem. Check the configuration settings and make sure any local firewalls allow the new DNS access to the upstream DNS servers.

The new DNS server can use itself for local name resolution, just like other local computers. Set the address of the DNS server in the client computer's /etc/resolv.conf:

```
nameserver 192.168.1.240
```

A local DHCP server can be configured to provide this address for a local DNS server, providing automatic configuration on DHCP clients.
Chapter 15. 強制存取控制 (MAC)

15.1. 概述

FreeBSD supports security extensions based on the POSIX™.1e draft. These security mechanisms include file system Access Control Lists (存取控制清單) and Mandatory Access Control (MAC). MAC allows access control modules to be loaded in order to implement security policies. Some modules provide protections for a narrow subset of the system, hardening a particular service. Others provide comprehensive labeled security across all subjects and objects. The mandatory part of the definition indicates that enforcement of controls is performed by administrators and the operating system. This is in contrast to the default security mechanism of Discretionary Access Control (DAC) where enforcement is left to the discretion of users.

This chapter focuses on the MAC framework and the set of pluggable security policy modules FreeBSD provides for enabling various security mechanisms.

読完這章,您將了解:
• The terminology associated with the MAC framework.
• The capabilities of MAC security policy modules as well as the difference between a labeled and non-labeled policy.
• The considerations to take into account before configuring a system to use the MAC framework.
• Which MAC security policy modules are included in FreeBSD and how to configure them.
• How to implement a more secure environment using the MAC framework.
• How to test the MAC configuration to ensure the framework has been properly implemented.

在開始閱讀這章之前,您需要:
• 了解 UNIX™及 FreeBSD 基礎 (FreeBSD 基礎).
• Have some familiarity with security and how it pertains to FreeBSD (安全性).

Improper MAC configuration may cause loss of system access, aggravation of users, or inability to access the features provided by Xorg. More importantly, MAC should not be relied upon to completely secure a system. The MAC framework only augments an existing security policy. Without sound security practices and regular security checks, the system will never be completely secure.

The examples contained within this chapter are for demonstration purposes and the example settings should not be implemented on a production system.

Implementing any security policy takes a good deal of understanding, proper design, and thorough testing.

While this chapter covers a broad range of security issues relating to the MAC framework, the development of new MAC security policy modules will not be covered. A number of security policy modules included with the MAC framework have specific characteristics which are provided for both testing and new module development. Refer to mac_test(4), mac_stub(4) and mac_none(4) for more information on these security policy modules and the various mechanisms they provide.

15.2. 關鍵詞

The following key terms are used when referring to the MAC framework:
• compartment: a set of programs and data to be partitioned or separated, where users are given explicit access to specific component of a system. A compartment represents a grouping, such as a work group, department, project, or topic. Compartments make it possible to implement a need-to-know-basis security policy.
integrity: the level of trust which can be placed on data. As the integrity of the data is elevated, so does the ability to trust that data.

level: the increased or decreased setting of a security attribute. As the level increases, its security is considered to elevate as well.

label: a security attribute which can be applied to files, directories, or other items in the system. It could be considered a confidentiality stamp. When a label is placed on a file, it describes the security properties of that file and will only permit access by files, users, and resources with a similar security setting. The meaning and interpretation of label values depends on the policy configuration. Some policies treat a label as representing the integrity or secrecy of an object while other policies might use labels to hold rules for access.

multilabel: this property is a file system option which can be set in single-user mode using tunefs(8), during boot using fstab(5), or during the creation of a new file system. This option permits an administrator to apply different MAC labels on different objects. This option only applies to security policy modules which support labeling.

single label: a policy where the entire file system uses one label to enforce access control over the flow of data. Whenever multilabel is not set, all files will conform to the same label setting.

object: an entity through which information flows under the direction of a subject. This includes directories, files, fields, screens, keyboards, memory, magnetic storage, printers or any other data storage or moving device. An object is a data container or a system resource. Access to an object effectively means access to its data.

subject: any active entity that causes information to flow between objects such as a user, user process, or system process. On FreeBSD, this is almost always a thread acting in a process on behalf of a user.

policy: a collection of rules which defines how objectives are to be achieved. A policy usually documents how certain items are to be handled. This chapter considers a policy to be a collection of rules which controls the flow of data and information and defines who has access to that data and information.

high-watermark: this type of policy permits the raising of security levels for the purpose of accessing higher level information. In most cases, the original level is restored after the process is complete. Currently, the FreeBSD MAC framework does not include this type of policy.

low-watermark: this type of policy permits lowering security levels for the purpose of accessing information which is less secure. In most cases, the original security level of the user is restored after the process is complete. The only security policy module in FreeBSD to use this is mac_lomac(4).

sensitivity: usually used when discussing Multilevel Security (MLS). A sensitivity level describes how important or secret the data should be. As the sensitivity level increases, so does the importance of the secrecy, or confidentiality, of the data.

15.3. MAC label

A MAC label is a security attribute which may be applied to subjects and objects throughout the system. When setting a label, the administrator must understand its implications in order to prevent unexpected or undesired behavior of the system. The attributes available on an object depend on the loaded policy module, as policy modules interpret their attributes in different ways. The security label on an object is used as a part of a security access control decision by a policy. With some policies, the label contains all of the information necessary to make a decision. In other policies, the labels may be processed as part of a larger rule set.

There are two types of label policies: single label and multi label. By default, the system will use single label. The administrator should be aware of the pros and cons of each in order to implement policies which meet the requirements of the system's security model.

A single label security policy only permits one label to be used for every subject or object. Since a single label policy enforces one set of access permissions across the entire system, it provides lower administration overhead, but decreases the flexibility of policies which support labeling. However, in many environments, a single label policy may be all that is required.
A single label policy is somewhat similar to DAC as root configures the policies so that users are placed in the appropriate categories and access levels. A notable difference is that many policy modules can also restrict root. Basic control over objects will then be released to the group, but root may revoke or modify the settings at any time.

When appropriate, a multi label policy can be set on a UFS file system by passing multilabel to tunefs(8). A multi label policy permits each subject or object to have its own independent MAC label. The decision to use a multi label or single label policy is only required for policies which implement the labeling feature, such as biba, lomac, and mls. Some policies, such as seeotheruids, portacl, and partition, do not use labels at all.

Using a multi label policy on a partition and establishing a multi label security model can increase administrative overhead as everything in that file system has a label. This includes directories, files, and even device nodes.

The following command will set multilabel on the specified UFS file system. This may only be done in single-user mode and is not a requirement for the swap file system:

```
# tunefs -l enable /
```

Some users have experienced problems with setting the multilabel flag on the root partition. If this is the case, please review MAC架构问题排除.

Since the multi label policy is set on a per-file system basis, a multi label policy may not be needed if the file system layout is well designed. Consider an example security MAC model for a FreeBSD web server. This machine uses the single label, biba/high, for everything in the default file systems. If the web server needs to run at biba/low to prevent write up capabilities, it could be installed to a separate UFS /usr/local file system set at biba/low.

15.3.1. 标签设定

Virtually all aspects of label policy module configuration will be performed using the base system utilities. These commands provide a simple interface for object or subject configuration or the manipulation and verification of the configuration.

All configuration may be done using setfmac, which is used to set MAC labels on system objects, and setpmac, which is used to set the labels on system subjects. For example, to set the biba MAC label to high on test:

```
# setfmac biba/high test
```

If the configuration is successful, the prompt will be returned without error. A common error is Permission denied which usually occurs when the label is being set or modified on a restricted object. Other conditions may produce different failures. For instance, the file may not be owned by the user attempting to relabel the object, the object may not exist, or the object may be read-only.

A mandatory policy will not allow the process to relabel the file, maybe because of a property of the file, a property of the process, or a property of the proposed new label value. For example, if a user running at low integrity tries to change the label of a high integrity file, or a user running at low integrity tries to change the label of a low integrity file to a high integrity label, these operations will fail.

The system administrator may use setpmac to override the policy module's settings by assigning a different label to the invoked process:

```
# setpmac biba/high test
```

Permission denied
For currently running processes, such as sendmail, `getpmac` is usually used instead. This command takes a process ID (PID) in place of a command name. If users attempt to manipulate a file not in their access, subject to the rules of the loaded policy modules, the `Operation not permitted` error will be displayed.

15.3.2. 預先定義的標籤

A few FreeBSD policy modules which support the labeling feature offer three predefined labels: `low`, `equal`, and `high`, where:

- `low` is considered the lowest label setting an object or subject may have. Setting this on objects or subjects blocks their access to objects or subjects marked high.
- `equal` sets the subject or object to be disabled or unaffected and should only be placed on objects considered to be exempt from the policy.
- `high` grants an object or subject the highest setting available in the Biba and MLS policy modules.

Such policy modules include `mac_biba(4)`, `mac_mls(4)` and `mac_lomac(4)`. Each of the predefined labels establishes a different information flow directive. Refer to the manual page of the module to determine the traits of the generic label configurations.

15.3.3. 數值標籤

The Biba and MLS policy modules support a numeric label which may be set to indicate the precise level of hierarchical control. This numeric level is used to partition or sort information into different groups of classification, only permitting access to that group or a higher group level. For example:

```
biba/10:2+3+6(5:2+3-20:2+3+4+5+6)
```

may be interpreted as "Biba Policy Label/Grade 10: Compartments 2, 3 and 6: (grade 5 ...

In this example, the first grade would be considered the effective grade with effective compartments, the second grade is the low grade, and the last one is the high grade. In most configurations, such fine-grained settings are not needed as they are considered to be advanced configurations.

System objects only have a current grade and compartment. System subjects reflect the range of available rights in the system, and network interfaces, where they are used for access control. The grade and compartments in a subject and object pair are used to construct a relationship known as dominance, in which a subject dominates an object, the object dominates the subject, neither dominates the other, or both dominate each other. The "both dominate" case occurs when the two labels are equal. Due to the information flow nature of Biba, a user has rights to a set of compartments that might correspond to projects, but objects also have a set of compartments. Users may have to subset their rights using `su` or `setpmac` in order to access objects in a compartment from which they are not restricted.

15.3.4. 使用者標籤

Users are required to have labels so that their files and processes properly interact with the security policy defined on the system. This is configured in `/etc/login.conf` using login classes. Every policy module that uses labels will implement the user class setting.
To set the user class default label which will be enforced by MAC, add a label entry. An example label entry containing every policy module is displayed below. Note that in a real configuration, the administrator would never enable every policy module. It is recommended that the rest of this chapter be reviewed before any configuration is implemented.

```
default:
copyright=/etc/COPYRIGHT:
welcome=/etc/motd:
setenv=MAIL=/var/mail/$,BLOCKSIZE=K:
path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:
manpath=/usr/shared/man /usr/local/man:
nologin=/usr/sbin/nologin:
cputime=1h30m:
datasize=8M:
vmemoryuse=100M:
stacksize=2M:
memorylocked=4M:
memoryuse=8M:
filesize=8M:
coredumpsize=8M:
openfiles=24:
maxproc=32:
priority=0:
requirehome:
passwordtime=91d:
umask=022:
ignoretime@:
label=partition/13,mls/5,biba/10(5-15),lomac/10[
```

While users can not modify the default value, they may change their label after they login, subject to the constraints of the policy. The example above tells the Biba policy that a process's minimum integrity is 5, its maximum is 15, and the default effective label is 10. The process will run at 10 until it chooses to change label, perhaps due to the user using `setpmac`, which will be constrained by Biba to the configured range.

After any change to `login.conf`, the login class capability database must be rebuilt using `cap_mkdb`.

Many sites have a large number of users requiring several different user classes. In depth planning is required as this can become difficult to manage.

### 15.3.5. 網路介面標籤

Labels may be set on network interfaces to help control the flow of data across the network. Policies using network interface labels function in the same way that policies function with respect to objects. Users at high settings in Biba, for example, will not be permitted to access network interfaces with a label of low.
This example will set the MAC label of `biba/equal` on the `bge0` interface. When using a setting similar to `biba/high(low-high)`, the entire label should be quoted to prevent an error from being returned.

Each policy module which supports labeling has a tunable which may be used to disable the MAC label on network interfaces. Setting the label to `equal` will have a similar effect. Review the output of `sysctl`, the policy manual pages, and the information in the rest of this chapter for more information on those tunables.

### 15.4. 规划安全架构

Before implementing any MAC policies, a planning phase is recommended. During the planning stages, an administrator should consider the implementation requirements and goals, such as:

- How to classify information and resources available on the target systems.
- Which information or resources to restrict access to along with the type of restrictions that should be applied.
- Which MAC modules will be required to achieve this goal.

A trial run of the trusted system and its configuration should occur before a MAC implementation is used on production systems. Since different environments have different needs and requirements, establishing a complete security profile will decrease the need of changes once the system goes live.

Consider how the MAC framework augments the security of the system as a whole. The various security policy modules provided by the MAC framework could be used to protect the network and file systems or to block users from accessing certain ports and sockets. Perhaps the best use of the policy modules is to load several security policy modules at a time in order to provide a MLS environment. This approach differs from a hardening policy, which typically hardens elements of a system which are used only for specific purposes. The downside to MLS is increased administrative overhead.

The overhead is minimal when compared to the lasting effect of a framework which provides the ability to pick and choose which policies are required for a specific configuration and which keeps performance overhead down. The reduction of support for unneeded policies can increase the overall performance of the system as well as offer flexibility of choice. A good implementation would consider the overall security requirements and effectively implement the various security policy modules offered by the framework.

A system utilizing MAC guarantees that a user will not be permitted to change security attributes at will. All user utilities, programs, and scripts must work within the constraints of the access rules provided by the selected security policy modules and control of the MAC access rules is in the hands of the system administrator.

It is the duty of the system administrator to carefully select the correct security policy modules. For an environment that needs to limit access control over the network, the `mac_portacl(4)`, `mac_ifoff(4)`, and `mac_biba(4)` policy modules make good starting points. For an environment where strict confidentiality of file system objects is required, consider the `mac_bsdextended(4)` and `mac_mls(4)` policy modules.

Policy decisions could be made based on network configuration. If only certain users should be permitted access to `ssh(1)`, the `mac_portacl(4)` policy module is a good choice. In the case of file systems, access to objects might be considered confidential to some users, but not to others. As an example, a large development team might be broken off into smaller projects where developers in project A might not be permitted to access objects written by developers in project B. Yet both projects might need to access objects created by developers in project C. Using the different security policy modules provided by the MAC framework, users could be divided into these groups.
and then given access to the appropriate objects. Each security policy module has a
unique way of dealing with the overall security of a system. Module selection should be based on a well thought out
security policy which may require revision and reimplementation. Understanding the different security policy
modules offered by the MAC framework will help administrators choose the best policies for their situations.

The rest of this chapter covers the available modules, describes their use and configuration, and in some cases,
provides insight on applicable situations.

Implementing MAC is much like implementing a firewall since care must be taken to prevent being completely locked out of the system. The ability to revert back to a previous configuration should be considered and the implementation of MAC over a remote connection should be done with extreme caution.

15.5. Available MAC Management Policies

The default FreeBSD kernel includes options MAC. This means that every module included with the MAC framework can be loaded with kldload as a run-time kernel module. After testing the module, add the module name to /boot/loader.conf so that it will load during boot. Each module also provides a kernel option for those administrators who choose to compile their own custom kernel. FreeBSD includes a group of policies that will cover most security requirements. Each policy is summarized below. The last three policies support integer settings in place of the three default labels.

15.5.1. MAC See Other UIDs Policy

Module name: mac_seeotheruids.ko

Kernel configuration line: options MAC_SEEOTHERUIDS

Boot option: mac_seeotheruids_load="YES"

The mac_seeotheruids(4) module extends the security.bsd.see_other_uids and security.bsd.see_other_gids sysctl tunables. This option does not require any labels to be set before configuration and can operate transparently with other modules. After loading the module, the following sysctl tunables may be used to control its features:

- security.mac.seeotheruids.enabled enables the module and implements the default settings which deny users the ability to view processes and sockets owned by other users.
- security.mac.seeotheruids.specificgid_enabled allows specified groups to be exempt from this policy. To exempt specific groups, use the security.mac.seeotheruids.specificgid=XXX sysctl tunable, replacing XXX with the numeric group ID to be exempted.
- security.mac.seeotheruids.primarygroup_enabled is used to exempt specific primary groups from this policy. When using this tunable, security.mac.seeotheruids.specificgid_enabled may not be set.

15.5.2. MAC BSD Extended Policy

Module name: mac_bsdextended.ko

Kernel configuration line: options MAC_BSDEXTENDED

Boot option: mac_bsdextended_load="YES"

The mac_bsdextended(4) module enforces a file system firewall. It provides an extension to the standard file system permissions model, permitting an administrator to create a firewall-like...
ruleset to protect files, utilities, and directories in the file system hierarchy. When access to a file system object is attempted, the list of rules is iterated until either a matching rule is located or the end is reached. This behavior may be changed using `security.mac.bsdextended.firstmatch_enabled`. Similar to other firewall modules in FreeBSD, a file containing the access control rules can be created and read by the system at boot time using an `rc.conf(5)` variable.

The rule list may be entered using `ugidfw(8)` which has a syntax similar to `ipfw(8)`. More tools can be written by using the functions in the `libugidfw(3)` library.

After the `mac_bsdextended(4)` module has been loaded, the following command may be used to list the current rule configuration:

```
# ugidfw list
0 slots, 0 rules
```

By default, no rules are defined and everything is completely accessible. To create a rule which blocks all access by users but leaves `root` unaffected:

```
# ugidfw add subject not uid root new object not uid root mode n
```

While this rule is simple to implement, it is a very bad idea as it blocks all users from issuing any commands. A more realistic example blocks `user1` all access, including directory listings, to `user2`'s home directory:

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

Instead of `user1`, `not uid user2` could be used in order to enforce the same access restrictions for all users. However, the `root` user is unaffected by these rules.

Extreme caution should be taken when working with this module as incorrect use could block access to certain parts of the file system.

15.5.3. MAC Interface Silencing

Module name: `mac_ifoff.ko`

Kernel configuration line: `options MAC_IFOFF`

Boot option: `mac_ifoff_load="YES"`

The `mac_ifoff(4)` module is used to disable network interfaces on the fly and to keep network interfaces from being brought up during system boot. It does not use labels and does not depend on any other MAC modules.

Most of this module's control is performed through these `sysctl` tunables:

- `security.mac.ifoff.lo_enabled` enables or disables all traffic on the loopback, `lo(4)`, interface.
- `security.mac.ifoff.bpfrecv_enabled` enables or disables all traffic on the Berkeley Packet Filter interface, `bpf(4)`.
- `security.mac.ifoff.other_enabled` enables or disables traffic on all other interfaces.

One of the most common uses of `mac_ifoff(4)` is network monitoring in an environment where...
network traffic should not be permitted during the boot sequence. Another use would be to write a script which uses an application such as `security/aide` to automatically block network traffic if it finds new or altered files in protected directories.

15.5.4. MAC Port Access Control

**Module name:** `mac_portacl.ko`

**Kernel configuration line:** `MAC_PORTACL`

**Boot option:** `mac_portacl_load=YES`

The `mac_portacl(4)` module is used to limit binding to local TCP and UDP ports, making it possible to allow non-root users to bind to specified privileged ports below 1024.

Once loaded, this module enables the MAC policy on all sockets. The following tunables are available:

- `security.mac.portacl.enabled` enables or disables the policy completely.
- `security.mac.portacl.port_high` sets the highest port number that `mac_portacl(4)` protects.
- `security.mac.portacl.suser_exempt`, when set to a non-zero value, exempts the root user from this policy.
- `security.mac.portacl.rules` specifies the policy as a text string of the form `rule[,rule,...]`, with as many rules as needed, and where each rule is of the form `idtype:id:protocol:port`. The `idtype` is either `uid` or `gid`. The `protocol` parameter can be `tcp` or `udp`. The `port` parameter is the port number to allow the specified user or group to bind to. Only numeric values can be used for the user ID, group ID, and port parameters.

By default, ports below 1024 can only be used by privileged processes which run as root. For `mac_portacl(4)` to allow non-privileged processes to bind to ports below 1024, set the following tunables as follows:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0
# sysctl net.inet.ip.portrange.reservedhigh=0
```

To prevent the root user from being affected by this policy, set `security.mac.portacl.suser_exempt` to a non-zero value.

```
# sysctl security.mac.portacl.suser_exempt=1
```

To allow the www user with UID 80 to bind to port 80 without ever needing root privilege:

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

This next example permits the user with the UID of 1001 to bind to TCP ports 110 (POP3) and 995 (POP3s):

```
```
15.5.5. MAC Partition

Module name: mac_partition.ko

Kernel configuration line: options MAC_PARTITION

Boot option: mac_partition_load="YES"

The mac_partition(4) policy drops processes into specific "partitions" based on their MAC label. Most configuration for this policy is done using setpmac(8). One sysctl tunable is available for this policy:

- security.mac.partition.enabled enables the enforcement of MAC process partitions.

When this policy is enabled, users will only be permitted to see their processes, and any others within their partition, but will not be permitted to work with utilities outside the scope of this partition. For instance, a user in the insecure class will not be permitted to access top as well as many other commands that must spawn a process.

This example adds top to the label set on users in the insecure class. All processes spawned by users in the insecure class will stay in the partition/13 label.

```
# setpmac partition/13 top
```

This command displays the partition label and the process list:

```
# ps Zax
```

This command displays another user's process partition label and that user's currently running processes:

```
# ps -ZU trhodes
```

Users can see processes in root's label unless the mac_seeotheruids(4) policy is loaded.

15.5.6. MAC Multi-Level Security

Module name: mac_mls.ko

Kernel configuration line: options MAC_MLS

Boot option: mac_mls_load="YES"

The mac_mls(4) policy controls access between subjects and objects in the system by enforcing a strict information flow policy. In MLS environments, a "clearance" level is set in the label of each subject or object, along with compartments. Since these clearance levels can reach numbers greater than several thousand, it would be a daunting task to thoroughly configure every subject or object. To ease this administrative overhead, three labels are included in this policy:

- mls/low
- mls/equal
- mls/high

where:

- Anything labeled with mls/low will have a low clearance level and not be permitted to access information of a higher level. This label also prevents objects of a higher clearance level from writing or passing information to a lower level.
MLS provides:

- A hierarchical security level with a set of non-hierarchical categories.
- Fixed rules of no read up, no write down. This means that a subject can have read access to objects on its own level or below, but not above. Similarly, a subject can have write access to objects on its own level or above, but not beneath.
- Secrecy, or the prevention of inappropriate disclosure of data.
- A basis for the design of systems that concurrently handle data at multiple sensitivity levels without leaking information between secret and confidential.

The following sysctl tunables are available:

- `security.mac.mls.enabled` is used to enable or disable the MLS policy.
- `security.mac.mls.ptys_equal` labels all `pty(4)` devices as `mls/equal` during creation.
- `security.mac.mls.revocation_enabled` revokes access to objects after their label changes to a label of a lower grade.
- `security.mac.mls.max_compartments` sets the maximum number of compartment levels allowed on a system.

To manipulate MLS labels, use `setfmac(8)`. To assign a label to an object:

```
# setfmac mls/5 test
```

To get the MLS label for the file `test`:

```
# getfmac test
```

Another approach is to create a master policy file in `/etc/` which specifies the MLS policy information and to feed that file to `setfmac`.

When using the MLS policy module, an administrator plans to control the flow of sensitive information. The default block read up block write down sets everything to a low state. Everything is accessible and an administrator slowly augments the confidentiality of the information.

Beyond the three basic label options, an administrator may group users and groups as required to block the information flow between them. It might be easier to look at the information in clearance levels using descriptive words, such as classifications of Confidential, Secret, and Top Secret. Some administrators instead create different groups based on project levels. Regardless of the classification method, a well thought out plan must exist before implementing a restrictive policy.

Some example situations for the MLS policy module include an e-commerce web server, a file server holding critical company information, and financial institution environments.

15.5.7. MAC Biba 模組

**Module name:** mac_biba.ko

**Kernel configuration line:**

```
options MAC_BIBA
```

**Boot option:**

```
mac_biba_load="YES"
```
The `mac_biba(4)` module loads the MAC Biba policy. This policy is similar to the MLS policy with the exception that the rules for information flow are slightly reversed. This is to prevent the downward flow of sensitive information whereas the MLS policy prevents the upward flow of sensitive information.

In Biba environments, an "integrity" label is set on each subject or object. These labels are made up of hierarchical grades and non-hierarchical components. As a grade ascends, so does its integrity. Supported labels are `biba/low`, `biba/equal`, and `biba/high`, where:

- `biba/low` is considered the lowest integrity an object or subject may have. Setting this on objects or subjects blocks their write access to objects or subjects marked as `biba/high`, but will not prevent read access.
- `biba/equal` should only be placed on objects considered to be exempt from the policy.
- `biba/high` permits writing to objects set at a lower label, but does not permit reading that object. It is recommended that this label be placed on objects that affect the integrity of the entire system.

Biba provides:

- Hierarchical integrity levels with a set of non-hierarchical integrity categories.
- Fixed rules are "no write up, no read down", the opposite of MLS. A subject can have write access to objects on its own level or below, but not above. Similarly, a subject can have read access to objects on its own level or above, but not below.
- Integrity by preventing inappropriate modification of data.
- Integrity levels instead of MLS sensitivity levels.

The following tunables can be used to manipulate the Biba policy:

- `security.mac.biba.enabled` is used to enable or disable enforcement of the Biba policy on the target machine.
- `security.mac.biba.ptys_equal` is used to disable the Biba policy on `pty(4)` devices.
- `security.mac.biba.revocation_enabled` forces the revocation of access to objects if the label is changed to dominate the subject.

To access the Biba policy setting on system objects, use `setfmac` and `getfmac`:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

Integrity, which is different from sensitivity, is used to guarantee that information is not manipulated by untrusted parties. This includes information passed between subjects and objects. It ensures that users will only be able to modify or access information they have been given explicit access to. The `mac_biba(4)` security policy module permits an administrator to configure which files and programs a user may see and invoke while assuring that the programs and files are trusted by the system for that user.

During the initial planning phase, an administrator must be prepared to partition users into grades, levels, and areas. The system will default to a high label once this policy module is enabled, and it is up to the administrator to configure the different grades and levels for users. Instead of using clearance levels, a good planning method could include topics. For instance, only allow developers modification access to the source code repository, source code compiler, and other development utilities. Other users would be grouped into other categories such as testers, designers, or end users and would only be permitted read access.
A lower integrity subject is unable to write to a higher integrity subject and a higher integrity subject cannot list or read a lower integrity object. Setting a label at the lowest possible grade could make it inaccessible to subjects. Some prospective environments for this security policy module would include a constrained web server, a development and test machine, and a source code repository. A less useful implementation would be a personal workstation, a machine used as a router, or a network firewall.

15.5.8. MAC Low-watermark

Module name: mac_lomac.ko
Kernel configuration line: options MAC_LOMAC
Boot option: mac_lomac_load="YES"

Unlike the MAC Biba policy, the mac_lomac(4) policy permits access to lower integrity objects only after decreasing the integrity level to not disrupt any integrity rules. The Low-watermark integrity policy works almost identically to Biba, with the exception of using floating labels to support subject demotion via an auxiliary grade compartment. This secondary compartment takes the form [auxgrade]. When assigning a policy with an auxiliary grade, use the syntax lomac/10[2], where 2 is the auxiliary grade.

This policy relies on the ubiquitous labeling of all system objects with integrity labels, permitting subjects to read from low integrity objects and then downgrading the label on the subject to prevent future writes to high integrity objects using [auxgrade]. The policy may provide greater compatibility and require less initial configuration than Biba.

Like the Biba and MLS policies, setfmac and setpmac are used to place labels on system objects:

# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]

The auxiliary grade low is a feature provided only by the MACLOMAC policy.

15.6. User Lock Down

This example considers a relatively small storage system with fewer than fifty users. Users will have login capabilities and are permitted to store data and access resources.

For this scenario, the mac_bsdextended(4) and mac_seeotheruids(4) policy modules could co-exist and block access to system objects while hiding user processes. Begin by adding the following line to /boot/loader.conf:

mac_seeotheruids_load="YES"

The mac_bsdextended(4) security policy module may be activated by adding this line to /etc/rc.conf:

ugidfw_enable="YES"

Default rules stored in /etc/rc.bsdextended will be loaded at system initialization. However, the default entries may need modification. Since this machine is expected only to service users, everything may be left commented out except the last two lines in order to force the loading of user...
This section demonstrates the steps that are needed to implement the Nagios network monitoring system in a MAC environment. This is meant as an example which still requires the administrator to test that the implemented policy meets the security requirements of the network before using in a production environment.

15.7.1. 建立不安全的使用者类别

Begin the procedure by adding the following user class to /etc/login.conf:

```
insecure:
:copyright=/etc/COPYRIGHT:
:welcome=/etc/motd:
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
:manpath=/usr/shared/man /usr/local/man:
:nologin=/usr/sbin/nologin:
:cputime=1h30m:
:datasize=8M:
:vmemoryuse=100M:
:stacksize=2M:
:memorylocked=4M:
:memoryuse=8M:
:filesize=8M:
:coredumpsize=8M:
:openfiles=24:
:maxproc=32:
:priority=0:
:requirehome:
:passwordtime=91d:
:umask=022:
:ignoretime@:
```
Then, add the following line to the default user class section:

```
# cap_mkdb /etc/login.conf
```

15.7.2. 設定
使用者

Set the root user to the default class using:

```
# pw usermod root -L default
```

All user accounts that are not root will now require a login class. The login class is required, otherwise users will be refused access to common commands. The following sh script should do the trick:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' /etc/passwd`; do
    pw usermod $x -L default;
# endfor
```

Next, drop the nagios and www accounts into the insecure class:

```
# pw usermod nagios -L insecure
# pw usermod www -L insecure
```

15.7.3. 建立關聯檔 (Context File)

A contexts file should now be created as /etc/policy.contexts:

```
# This is the default BIBA policy for this system.
# System:
/var/run(/.*)?          biba/equal
/dev/(/.*)?         biba/equal
/var                biba/equal
/var/spool(/.*)?        biba/equal
/var/log(/.*)?          biba/equal
```

325
This policy enforces security by setting restrictions on the flow of information. In this specific configuration, users, including root, should never be allowed to access Nagios. Configuration files and processes that are a part of Nagios will be completely self contained or jailed. This file will be read after running setfsmac on every file system. This example sets the policy on the root file system:

```
# setfsmac -ef /etc/policy.contexts /
```

Next, add these edits to the main section of `/etc/mac.conf`:

```
default_labels file ?biba
default_labels ifnet ?biba
default_labels process ?biba
default_labels socket ?biba
```

To finish the configuration, add the following lines to `/boot/loader.conf`:

```
mac_biba_load="YES"
mac_seeotheruids_load="YES"
security.mac.biba.trust_all_interfaces=1
```

And the following line to the network card configuration stored in `/etc/rc.conf`. If the primary network configuration is done via DHCP, this may need to be configured manually after every system boot:

```
maclabel biba/equal
```
First, ensure that the web server and Nagios will not be started on system initialization and reboot. Ensure that root cannot access any of the files in the Nagios configuration directory. If root can list the contents of /var/spool/nagios, something is wrong. Instead, a "permission denied" error should be returned. If all seems well, Nagios, Apache, and Sendmail can now be started:

```
# cd /etc/mail && make stop && 
setpmac biba/equal make start 
&& 
setpmac biba/10 (10-10) apachectl start 
&& 
setpmac biba/10 (10-10) /usr/local/etc/rc.d/nagios.sh forcestart 
```

Double check to ensure that everything is working properly. If not, check the log files for error messages. If needed, use `sysctl(8)` to disable the `mac_biba(4)` security policy module and try starting everything again as usual.

The root user can still change the security enforcement and edit its configuration files. The following command will permit the degradation of the security policy to a lower grade for a newly spawned shell:

```
# setpmac biba/10 csh 
```

To block this from happening, force the user into a range using `login.conf(5)`. If `setpmac(8)` attempts to run a command outside of the compartment's range, an error will be returned and the command will not be executed. In this case, set root to `biba/high(high-high)`.

### 15.8. MAC 架構疑難排解

This section discusses common configuration errors and how to resolve them. The `multilabel` flag does not stay enabled on the root (`/`) partition.

The following steps may resolve this transient error:

1. Edit `/etc/fstab` and set the root partition to `ro` for read-only.
2. Reboot into single user mode.
3. Run `tunefs -l enable` on `/`.
4. Reboot the system.
5. Run `mount -urw /` and change the `ro` back to `rw` in `/etc/fstab` and reboot the system again.
6. Double-check the output from `mount` to ensure that `multilabel` has been properly set on the root file system.

After establishing a secure environment with MAC, Xorg no longer starts. This could be caused by the MAC partition policy or by a mislabeling in one of the MAC labeling policies. To debug, try the following:

1. Check the error message. If the user is in the `insecure` class, the partition policy may be the culprit. Try setting the user's class back to the `default` class and rebuild the database with `cap_mkdb`. If this does not alleviate the problem, go to step two.
b. Double-check that the label policies are set correctly for the user, Xorg, and the /dev entries.

c. If neither of these resolve the problem, send the error message and a description of the environment to the FreeBSD general questions mailing list.

The _secure_path: unable to stat .login_conf error appears. This error can appear when a user attempts to switch from the root user to another user in the system. This message usually occurs when the user has a higher label setting than that of the user they are attempting to become. For instance, if joe has a default label of biba/low and root has a label of biba/high, root cannot view joe's home directory. This will happen whether or not root has used su to become joe as the Biba integrity model will not permit root to view objects set at a lower integrity level.

The system no longer recognizes root. When this occurs, whoami returns 0 and su returns who are you?. This can happen if a labeling policy has been disabled by sysctl(8) or the policy module was unloaded. If the policy is disabled, the login capabilities database needs to be reconfigured. Double check /etc/login.conf to ensure that all label options have been removed and rebuild the database with cap_mkdb.

This may also happen if a policy restricts access to master.passwd. This is usually caused by an administrator altering the file under a label which conflicts with the general policy being used by the system. In these cases, the user information would be read by the system and access would be blocked as the file has inherited the new label. Disable the policy using sysctl(8) and everything should return to normal.
The FreeBSD operating system includes support for security event auditing. Event auditing supports reliable, fine-grained, and configurable logging of a variety of security-relevant system events, including logins, configuration changes, and file and network access. These log records can be invaluable for live system monitoring, intrusion detection, and postmortem analysis. FreeBSD implements Sun™'s published Basic Security Module (BSM) Application Programming Interface (API) and file format, and is interoperable with the Solaris™ and Mac OS™ X audit implementations.

This chapter focuses on the installation and configuration of event auditing. It explains audit policies and provides an example audit configuration.

Read this chapter to learn:

• What event auditing is and how it works.
• How to configure event auditing on FreeBSD for users and processes.
• How to review the audit trail using the audit reduction and review tools.

Before starting this chapter, you should:

• Be familiar with the basics of kernel configuration/compilation (FreeBSD Core).
• Have some familiarity with security and how it pertains to FreeBSD (Security).

The audit facility has some known limitations. Not all security-relevant system events are auditable and some login mechanisms, such as Xorg-based display managers and third-party daemons, do not properly configure auditing for user login sessions.

The security event auditing facility is able to generate very detailed logs of system activity. On a busy system, trail file data can be very large when configured for high detail, exceeding gigabytes a week in some configurations. Administrators should take into account the disk space requirements associated with high volume audit configurations. For example, it may be desirable to dedicate a file system to /var/audit so that other file systems are not affected if the audit file system becomes full.

16.2. Keywords

The following terms are related to security event auditing:

• **event**: an auditable event is any event that can be logged using the audit subsystem. Examples of security-relevant events include the creation of a file, the building of a network connection, or a user logging in. Events are either "attributable", meaning that they can be traced to an authenticated user, or "non-attributable". Examples of non-attributable events are any events that occur before authentication in the login process, such as bad password attempts.

• **class**: a named set of related events which are used in selection expressions. Commonly used classes of events include "file creation" (fc), "exec" (ex), and "login_logout" (lo).

• **record**: an audit log entry describing a security event. Records contain a record event type, information on the subject (user) performing the action, date and time information, information on any objects or arguments, and a success or failure condition.

• **trail**: a log file consisting of a series of audit records describing security events. Trails are in roughly chronological order with respect to the time events completed. Only authorized processes are allowed to commit records to the audit trail.
• **selection expression**: a string containing a list of prefixes and audit event class names used to match events.

• **preselection**: the process by which the system identifies which events are of interest to the administrator. The preselection configuration uses a series of selection expressions to identify which classes of events to audit for which users, as well as global settings that apply to both authenticated and unauthenticated processes.

• **reduction**: the process by which records from existing audit trails are selected for preservation, printing, or analysis. Likewise, the process by which undesired audit records are removed from the audit trail. Using reduction, administrators can implement policies for the preservation of audit data. For example, detailed audit trails might be kept for one month, but after that, trails might be reduced in order to preserve only login information for archival purposes.

16.3. **稽査設定**

User space support for event auditing is installed as part of the base FreeBSD operating system. Kernel support is available in the `GENERIC` kernel by default, and `auditd(8)` can be enabled by adding the following line to `/etc/rc.conf`:

```
auditd_enable="YES"
```

Then, start the audit daemon:

```
# service auditd start
```

Users who prefer to compile a custom kernel must include the following line in their custom kernel configuration file:

```
options AUDIT
```

16.3.1. **事項選取方法**

Selection expressions are used in a number of places in the audit configuration to determine which events should be audited. Expressions contain a list of event classes to match. Selection expressions are evaluated from left to right, and two expressions are combined by appending one onto the other.

**預設稽査事件類別** summarizes the default audit event classes:

<table>
<thead>
<tr>
<th>類別名稱</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>Match all event classes.</td>
</tr>
<tr>
<td>aa</td>
<td>authentication and authorization</td>
</tr>
<tr>
<td>ad</td>
<td>administrative</td>
</tr>
<tr>
<td>ap</td>
<td>application</td>
</tr>
<tr>
<td>cl</td>
<td>file close</td>
</tr>
</tbody>
</table>

...
<table>
<thead>
<tr>
<th>Class</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ex</td>
<td>exec</td>
<td>Audit program execution. Auditing of command line arguments and environmental variables is controlled via audit_control(5) using the argv and envv parameters to the policy setting.</td>
</tr>
<tr>
<td>fa</td>
<td>file attribute access</td>
<td>Audit the access of object attributes such as stat(1) and pathconf(2).</td>
</tr>
<tr>
<td>fc</td>
<td>file create</td>
<td>Audit events where a file is created as a result.</td>
</tr>
<tr>
<td>fd</td>
<td>file delete</td>
<td>Audit events where file deletion occurs.</td>
</tr>
<tr>
<td>fm</td>
<td>file attribute modify</td>
<td>Audit events where file attribute modification occurs, such as by chown(8), chflags(1), and flock(2).</td>
</tr>
<tr>
<td>fr</td>
<td>file read</td>
<td>Audit events in which data is read or files are opened for reading.</td>
</tr>
<tr>
<td>fw</td>
<td>file write</td>
<td>Audit events in which data is written or files are written or modified.</td>
</tr>
<tr>
<td>io</td>
<td>ioctl</td>
<td>Audit use of the ioctl system call.</td>
</tr>
<tr>
<td>ip</td>
<td>ipc</td>
<td>Audit various forms of Inter-Process Communication, including POSIX pipes and System V IPC operations.</td>
</tr>
<tr>
<td>lo</td>
<td>login_logout</td>
<td>Audit login(1) and logout(1) events.</td>
</tr>
<tr>
<td>na</td>
<td>non attributable</td>
<td>Audit non-attributable events.</td>
</tr>
<tr>
<td>no</td>
<td>invalid class</td>
<td>Match no audit events.</td>
</tr>
<tr>
<td>nt</td>
<td>network</td>
<td>Audit events related to network actions such as connect(2) and accept(2).</td>
</tr>
<tr>
<td>ot</td>
<td>other</td>
<td>Audit miscellaneous events.</td>
</tr>
<tr>
<td>pc</td>
<td>process</td>
<td>Audit process operations such as exec(3) and exit(3).</td>
</tr>
</tbody>
</table>

These audit event classes may be customized by modifying the audit_class and audit_event configuration files. Each audit event class may be combined with a prefix indicating whether successful/failed operations are matched, and whether the entry is adding or removing matching for the class and type.
Audit neither successful nor failed events in this class.

Do not audit successful events in this class.

Do not audit failed events in this class.

If no prefix is present, both successful and failed instances of the event will be audited.

The following example selection string selects both successful and failed login/logout events, but only successful execution events:

`lo,+ex`

16.3.2.

設定檔

The following configuration files for security event auditing are found in `/etc/security`:

- `audit_class`: contains the definitions of the audit classes.
- `audit_control`: controls aspects of the audit subsystem, such as default audit classes, minimum disk space to leave on the audit log volume, and maximum audit trail size.
- `audit_event`: textual names and descriptions of system audit events and a list of which classes each event is in.
- `audit_user`: user-specific audit requirements to be combined with the global defaults at login.
- `audit_warn`: a customizable shell script used by `auditd(8)` to generate warning messages in exceptional situations, such as when space for audit records is running low or when the audit trail file has been rotated.

Audit configuration files should be edited and maintained carefully, as errors in configuration may result in improper logging of events.

In most cases, administrators will only need to modify `audit_control` and `audit_user`. The first file controls system-wide audit properties and policies and the second file may be used to fine-tune auditing by user.

16.3.2.1. The `audit_control` File

A number of defaults for the audit subsystem are specified in `audit_control`:

- `dir:/var/audit`
- `dist:off`
- `flags:lo,aa`
- `minfree:5`
- `naflags:lo,aa`
- `policy:cnt,argv`
- `filesz:2M`
- `expire-after:10M`

The `dir` entry is used to set one or more directories where audit logs will be stored. If more than one directory entry appears, they will be used in order as they fill. It is common to configure audit so that audit logs are stored on a dedicated file system, in order to prevent interference between the...
Audit subsystem and other subsystems if the file system fills. If the `dist` field is set to `on` or `yes`, hard links will be created to all trail files in `/var/audit/dist`.

The `flags` field sets the system-wide default preselection mask for attributable events. In the example above, successful and failed login/logout events as well as authentication and authorization are audited for all users.

The `minfree` entry defines the minimum percentage of free space for the file system where the audit trail is stored.

The `naflags` entry specifies audit classes to be audited for non-attributed events, such as the login/logout process and authentication and authorization.

The `policy` entry specifies a comma-separated list of policy flags controlling various aspects of audit behavior. The `cnt` indicates that the system should continue running despite an auditing failure (this flag is highly recommended). The other flag, `argv`, causes command line arguments to the `execve(2)` system call to be audited as part of command execution.

The `filesz` entry specifies the maximum size for an audit trail before automatically terminating and rotating the trail file. A value of `0` disables automatic log rotation. If the requested file size is below the minimum of 512k, it will be ignored and a log message will be generated.

The `expire-after` field specifies when audit log files will expire and be removed.

16.3.2.2. The `audit_user` File

The administrator can specify further audit requirements for specific users in `audit_user`. Each line configures auditing for a user via two fields: the `alwaysaudit` field specifies a set of events that should always be audited for the user, and the `neveraudit` field specifies a set of events that should never be audited for the user.

The following example entries audit login/logout events and successful command execution for `root` and file creation and successful command execution for `www`.

```
root:lo,+ex:no
www:fc,+ex:no
```

Audit trails consist of a series of audit records made up of tokens, which `praudit` prints sequentially, one per line. Each token is of a specific type, such as header (an audit record header) or path (a file path from a name lookup). The following is an example of an `execve` event:
This audit represents a successful `execve` call, in which the command `finger doug` has been run. The `exec arg` token contains the processed command line presented by the shell to the kernel. The `path` token holds the path to the executable as looked up by the kernel. The `attribute` token describes the binary and includes the file mode. The `subject` token stores the audit user ID, effective user ID and group ID, real user ID and group ID, process ID, session ID, port ID, and login address. Notice that the audit user ID and real user ID differ as the user `robert` switched to the `root` account before running this command, but it is audited using the original authenticated user. The `return` token indicates the successful execution and the `trailer` concludes the record.

XML output format is also supported and can be selected by including `-x`. Since audit logs may be very large, a subset of records can be selected using `auditreduce`. This example selects all audit records produced for the user `trhodes` stored in `AUDITFILE`:

```
# auditreduce -u trhodes /var/audit/AUDITFILE | praudit
```

Members of the `audit` group have permission to read audit trails in `/var/audit`. By default, this group is empty, so only the `root` user can read audit trails. Users may be added to the `audit` group in order to delegate audit review rights. As the ability to track audit log contents provides significant insight into the behavior of users and processes, it is recommended that the delegation of audit review rights be performed with caution.

16.4.1. 使用Audit Pipes

Audit pipes are cloning pseudo-devices which allow applications to tap the live audit record stream. This is primarily of interest to authors of intrusion detection and system monitoring applications. However, the audit pipe device is a convenient way for the administrator to allow live monitoring without running into problems with audit trail file ownership or log rotation interrupting the event stream. To track the live audit event stream:

```
# praudit /dev/auditpipe
```

By default, audit pipe device nodes are accessible only to the `root` user. To make them accessible to the members of the `audit` group, add a `devfs` rule to `/etc/devfs.rules`:

```
add path 'auditpipe*' mode 0440 group audit
```

See `devfs.rules(5)` for more information on configuring the devfs file system.

It is easy to produce audit event feedback cycles, in which the viewing of each audit event results in the generation of more audit events. For example, if all network I/O is audited, and `praudit` is run from an SSH session, a continuous stream of audit events will be generated at a high rate, as each event being printed...
will generate another event. For this reason, it is advisable to run `praudit` on an audit pipe device from sessions without fine-grained I/O auditing.

16.4.2. Audit Trail

Audit trails are written to by the kernel and managed by the audit daemon, `auditd(8)`. Administrators should not attempt to use `newsyslog.conf(5)` or other tools to directly rotate audit logs. Instead, `audit` should be used to shut down auditing, reconfigure the audit system, and perform log rotation. The following command causes the audit daemon to create a new audit log and signal the kernel to switch to using the new log. The old log will be terminated and renamed, at which point it may then be manipulated by the administrator:

```
# audit -n
```

If `auditd(8)` is not currently running, this command will fail and an error message will be produced. Adding the following line to `/etc/crontab` will schedule this rotation every twelve hours:

```
0     */12       *       *       *       root    /usr/sbin/audit -n
```

The change will take effect once `/etc/crontab` is saved.

Automatic rotation of the audit trail file based on file size is possible using `filesz` in `audit_control` as described in `The audit_control File`.

As audit trail files can become very large, it is often desirable to compress or otherwise archive trails once they have been closed by the audit daemon. The `audit_warn` script can be used to perform customized operations for a variety of audit-related events, including the clean termination of audit trails when they are rotated. For example, the following may be added to `/etc/security/audit_warn` to compress audit trails on close:

```
# Compress audit trail files on close.
#
if [...]
  gzip -9 $2
fi
```

Other archiving activities might include copying trail files to a centralized server, deleting old trail files, or reducing the audit trail to remove unneeded records. This script will be run only when audit trail files are cleanly terminated, so will not be run on trails left unterminated following an improper shutdown.
Chapter 17. 儲存設備

17.1. 概述
本章涵蓋如何在 FreeBSD 下使用磁碟及儲存媒體，這包含 SCSI 及 IDE 磁碟、CD 及 DVD 媒體、記憶體磁碟及 USB 儲存裝置。

讀完這章，您將了解:
• 如何在 FreeBSD 系統加入額外的硬碟。
• 如何在 FreeBSD 增加磁碟分割區的大小。
• 如何設定 FreeBSD 使用 USB 儲存裝置。
• 如何在 FreeBSD 系統使用 CD 及 DVD 媒體。
• 如何使用在 FreeBSD 下可用的備份程式。
• 如何設定記憶體磁碟。
• 什麼是檔案系統快照 (Snapshot) 以及如何有效使用。
• 如何使用配額 (Quota) 來限制磁碟空間使用量。
• 如何加密磁碟及交換空間來防範攻擊者。
• 如何設定高可用性 (Highly available) 的儲存網路。

在開始閱讀這章之前，您需要:
• 了解如何設定並安裝新的 FreeBSD 核心。
在新磁碟的新分割區上建立檔案系統:

```bash
# newfs -U /dev/ada1p1
```

建立一個空的目錄做來做為掛載點（mountpoint），一個在原有磁碟的檔案系統上可用來掛載新磁碟的位置:

```bash
# mkdir /newdisk
```

最後，將磁碟項目加入到 `/etc/fstab`, 讓啟動時會自動掛載新的磁碟:

```bash
/dev/ada1p1 /newdisk    ufs rw  2   2
```

新的磁碟也可手動掛載，無須重新啟動系統:

```bash
# mount /newdisk
```

17.3. 重設大小與擴增磁碟

磁碟的容量可以增加且不需要更動任何已存在資料。這時常會用在虛擬機器, 當虛擬磁碟太小且需要增加時。有時磁碟映像檔會被寫入到USB隨身碟, 但卻沒有使用全部的容量。此節我們將說明如何重設大小或擴增磁碟內容來使用增加的容量。

要取得要重設大小的磁碟的代號可以查看 `/var/run/dmesg.boot`。在本例中, 在系統上只有一個SATA磁碟, 该磁碟会以 `ada0` 表示。

列出在磁碟上的分割區來查看目前的設定:

```bash
# gpart show ada0
```

若磁碟已使用GPT分割表格式做格式化，可能會顯示為"已損壞（corrupted）"因為GPT備份分割區已不存在於磁碟結尾。使用 `gpart` 來修正備份分割區:

```bash
# gpart recover ada0
```

現在在磁碟上的額外空間已經可以被新的分割區使用, 或者可以拿來擴充既有的分割區:
分割區只能在連續的未使用空間上重設大小。在這個例子中，磁碟上的最後一個分割區是交換(Swap)分割區，而第二個分割區才是需要重設大小的分割區。由於交換分割區中只會有暫存的資料，所以此時可以安全的卸載、刪除，然後在重設第二個分割區大小之後再重建最後一個分割區。

停用交換分割區:
```
# swapoff /dev/ada0p3
```

刪除ada0磁碟上的第三個分割區，可使用-i參數來指定分割區。
```
# gpart delete -i 3 ada0
```

```
ada0p3 deleted
```

```
# gpart show ada0
```

```
34  102399933  ada0  GPT  (48G)
34        128     1  freebsd-boot  (64k)
162   79691648     2  freebsd-ufs  (38G)
79691810   22708157        - free -  (10G)
```

在掛載的檔案系統上修改分割區表可能會造成資料遺失。最好的方式是在未掛載檔案系統的情況下(使用Live CD-ROM或USB裝置)執行以下步驟。雖然如此，若仍要這樣做的话，在關閉GEOM安全性功能之後可以在掛載的檔案系統上修改分割區表:
```
# sysctl kern.geom.debugflags=16
```

重設分割區大小並保留要用來重建交換分割區的空間，要重設大小的分割區可以用-i來指定，而要重設的大小可用-s來指定，若要對齊分割區可以使用-a。這個動作只會修改分割區大小，分割區中的檔案系統需在另一步驟擴增。
```
# gpart resize -i 2 -s 47G -a 4k ada0
```

```
ada0p2 resized
```

```
# gpart show ada0
```

```
34  102399933  ada0  GPT  (48G)
34        128     1  freebsd-boot  (64k)
162   98566144     2  freebsd-ufs  (47G)
98566306    3833661        - free -  (1.8G)
```

重建交換分割區並且啟動，若不使用-s指定大小則會使用所有剩餘的空間：
gpart add -t freebsd-swap -a 4k ada0
ada0p3 added

gpart show ada0
34  102399933  ada0  GPT (48G)
34        128     1  freebsd-boot (64k)
162   98566144     2  freebsd-ufs (47G)
98566306    3833661     3  freebsd-swap (1.8G)

swapon /dev/ada0p3

# growfs /dev/ada0p2
Device is mounted read-write; resizing will result in temporary write suspension for /
It's strongly recommended to make a backup before growing the file system.
OK to grow file system on /dev/ada0p2, mounted on /, from 38GB to 47GB? [Yes/No] Yes
super-block backups (for fsck -b #) at:
80781312, 82063552, 83345792, 84628032, 85910272, 87192512, 88474752,
89756992, 91039232, 92321472, 93603712, 94885952, 96168192, 97450432

若檔案系統使用ZFS，重設大小需執行online子指令並使用-e來觸發作:

zpool online -e zroot /dev/ada0p2

現在分割區與檔案系統已透過重設大小來使用新增加的磁碟空間。

17.4. USB 儲存裝置

USB儲存裝置許多外部儲存裝置的解決方案，例如硬碟、USB隨身碟及CD與DVD燒錄機皆使用通用序列匯流排(Ubiversal Serial Bus, USB)，FreeBSD提供了對USB 1.x, 2.0及3.0裝置的支援。

部份硬體尚不相容USB 3.0，包含Haswell (Lynx point)晶片組，若FreeBSD開機出現failed with error 19訊息，請在系統BIOS關閉xHCI/USB3。

對USB儲存裝置的支援已內建於GENERIC核心，若為自訂的核心請確定在核心設定檔有下列幾行設定:

device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device pass # Passthrough device (direct ATA/SCSI access)
device uhci # provides USB 1.x support
device ohci # provides USB 1.x support
device ehci # provides USB 2.0 support
device xhci # provides USB 3.0 support
device usb  # USB Bus (required)
FreeBSD
使用 umass(4) 驅動程式透過 SCSI 子系統來存取 USB 儲存裝置，因此任何在系統的 USB 裝置都會以 SCSI 裝置呈現，若 USB 裝置是 CD 或 DVD 燒錄機，請不要在自訂核心設定檔中引用 device atapicam。

本節後續的部份示範如何檢查 FreeBSD 能夠辦識 USB 儲存裝置及如何設定該裝置。

17.4.1. 裝置設定
要測試 USB 設定，請先插入 USB 裝置，然後使用 dmesg 來確認系統訊息緩衝區中有出現該磁碟機，該訊息如下:

```
umass0: <STECH Simple Drive, class 0/0, rev 2.00/1.04, addr 3> on usbus0
```

不同的裝置會有不同的廠牌、裝置節點 (da0)、速度與大小。

當 USB 裝置可以做為 SCSI 檢視時，便可使用 camcontrol 來列出連接到系統的 USB 儲存裝置:
```
# camcontrol devlist
```

或者，可以使用 usbconfig 來列出裝置，請參考 usbconfig(8) 來取得更多有關此指令的資訊。
```
# usbconfig
ugen0.3: <Simple Drive STECH> at usbus0, cfg=0 md=HOST spd=HIGH (480Mbps) pwr=ON (2mA)
```

若該裝置尚未被格式化，請參考加入磁碟中有關於如何在 USB 磁碟格式化與建立分割區的說明。若磁碟中有檔案系統，可由 root 依据掛載與卸載檔案系統中的說明掛載磁碟。

要允許未被信任的使用者掛載任意媒體，可開啟 vfs.usermount，詳細說明如下。從安全性的角度來看這並不是安全的，大多的檔案系統並不會防範惡意裝置。

要讓裝置可讓一般使用者挂載，其中一個解決方案便是使用 pw(8) 讓所有裝置的使用者成為 operator 群組。接著，將下列幾行加入 /etc/devfs.rules 來確保 operator 能夠讀取與寫入裝置:
```
[localrules=5]
```
若系统也同时安装了内建SCSI磁碟，请更改第二行如下:
add path 'da[3-9]*' mode 0660 group operator
这会从operator群组中排除前三個SCSI磁碟(da0到da2)，接著取代3为内部SCSI磁碟的编号。请参考devfs.rules(5)来取得更多与此档文件相关的资讯。

接着，在/etc/rc.conf开启规则:
```
devfs_system_ruleset="localrules"
```
然后，加入以下行到/etc/sysctl.conf指示系统允许正常使用者挂载档案系统:
```
vfs.usermount=1
```
这样只会在下次重新开机时生效，可使用sysctl来立即设定这个变量:
```
# sysctl vfs.usermount=1
```
最后一个是建立要挂载档案系统的目录，要挂载档案系统的使用者需要拥有这个目录。其中一种办法是让root建立由該使用者拥有的子目录/mnt/username。在下面的例子，将username替换为該使用者的登入名称并将usergroup替换为該使用者的主要群组:
```
# mkdir /mnt/username
# chown username:usergroup /mnt/username
```
如果已经插入USB随身碟，并已出现/dev/da0s1装置。若装置使用FAT格式的档案系统，则使用者可使用以下指令挂载该档案系统:
```
% mount -t msdosfs -o -m=644,-M=755 /dev/da0s1 /mnt/username
```
在装置可以被拔除前，必须先卸载:
```
% umount /mnt/username
```
装置移除之后，系统信息缓衝区会显示如下的信息:
```
umass0: at uhub3, port 2, addr 3
```
```
da0 at umass-sim0 bus 0 scbus4 target 0 lun 0
```
```
da0: <STECH Simple Drive 1.04> s/n WD-WXE508CAN263 detached
```
17.4.2. 自動掛載可移除的媒體可以取消在 `/etc/auto_master` 中的下行來自動掛載 USB 裝置:

```
/media/media-nosuid
```

然後加入這些行到 `/etc/devd.conf`:

```
notify 100 {
  match "system" "GEOM";
  match "subsystem" "DEV";
  action "/usr/sbin/automount -c";
}
```

若 `autofs(5)` 及 `devd(8)` 已經正執行，則需重新載入設定:

```
# service automount restart
# service devd restart
```

要設定讓 `autofs(5)` 在開機時啟動可以加入此行到 `/etc/rc.conf`:

```
autofs_enable="YES"
```

`autofs(5)` 需要開啓 `devd(8)`，預設已經開啓。

立即啓動服務:

```
# service automount start
# service automountd start
# service autounmountd start
# service devd start
```

可以被自動掛載的檔案系統會在 `/media/` 中以目錄呈現，會以檔案系統的標籤來命名目錄，若標籤遺失，則以裝置節點命名。

檔案系統會在第一次存取時自動掛載，並在一段時間未使用後自動卸載。自動掛載的磁碟也可手動卸載:

```
# automount -fu
```

這個機制一般會用在記憶卡與 USB 隨身碟，也可用在任何 Block 裝置，包含光碟機或 iSCSI LUN。
Compact Disc (CD) media provide a number of features that differentiate them from conventional disks. They are designed so that they can be read continuously without delays to move the head between tracks. While CD media do have tracks, these refer to a section of data to be read continuously, and not a physical property of the disk. The ISO 9660 file system was designed to deal with these differences.

The FreeBSD Ports Collection provides several utilities for burning and duplicating audio and data CDs. This chapter demonstrates the use of several command line utilities. For CD burning software with a graphical utility, consider installing the `sysutils/xcdroast` or `sysutils/k3b` packages or ports.

### 17.5.1. Supported Devices

The GENERIC kernel provides support for SCSI, USB, and ATAPICD readers and burners. If a custom kernel is used, the options that need to be present in the kernel configuration file vary by the type of device.

For a SCSI burner, make sure these options are present:

```
device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device passthrough # Passthrough device (direct ATA/SCSI access)
device cd   # needed for CD and DVD burners
```

For a USB burner, make sure these options are present:

```
device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device passthrough # Passthrough device (direct ATA/SCSI access)
device cd   # needed for CD and DVD burners
device uhci # provides USB 1.x support
device ohci # provides USB 1.x support
device ehci # provides USB 2.0 support
device xhci # provides USB 3.0 support
device usb  # USB Bus (required)
device umass    # Disks/Mass storage - Requires scbus and da
```

For an ATAPI burner, make sure these options are present:

```
device ata  # Legacy ATA/SATA controllers
device scbus    # SCSI bus (required for ATA/SCSI)
device passthrough # Passthrough device (direct ATA/SCSI access)
device cd   # needed for CD and DVD burners
```

On FreeBSD versions prior to 10.x, this line is also needed in the kernel configuration file if the burner is an ATAPI device:

```
343
```
Alternately, this driver can be loaded at boot time by adding the following line to /boot/loader.conf:

```
atapicam_load="YES"
```

This will require a reboot of the system as this driver can only be loaded at boot time.

To verify that FreeBSD recognizes the device, run `dmesg` and look for an entry for the device. On systems prior to 10.x, the device name in the first line of the output will be `acd0` instead of `cd0`.

```
% dmesg | grep cd
```

```
cd0 at ahcich1 bus 0 scbus1 target 0 lun 0
cd0: <HL-DT-ST DVDRAM GU70N LT20> Removable CD-ROM SCSI-0 device
cd0: Serial Number M3OD3S34152
cd0: 150.000MB/s transfers (SATA 1.x, UDMA6, ATAPI 12bytes, PIO 8192bytes)
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

17.5.2. 燒錄 CD

In FreeBSD, `cdrecord` can be used to burn CDs. This command is installed with the `sysutils/cdrtools` package or port.

While `cdrecord` has many options, basic usage is simple. Specify the name of the ISO file to burn and, if the system has multiple burner devices, specify the name of the device to use:

```
# cdrecord dev=device imagefile.iso
```

To determine the device name of the burner, use `-scanbus` which might produce results like this:

```
# cdrecord -scanbus
```

```
ProDVD-ProBD-Clone 3.00 (amd64-unknown-freebsd10.0)
Copyright (C) 1995-2010 Jörg Schilling
Using libscg version 'schily-0.9'
scsibus0: 0,0,0     0
  'SEAGATE ' 'ST39236LW       ' '0004'
  Disk
0,1,0     1
  'SEAGATE ' 'ST39173W        ' '5958'
  Disk
0,2,0     2
  *0,3,0     3
  'iomega  ' 'jaz 1GB         ' 'J.86'
  Removable Disk
0,4,0     4
  'NEC     ' 'CD-ROM DRIVE:466' '1.26'
  Removable CD-ROM
0,5,0     5
  *0,6,0     6
  *0,7,0     7
```

344
Locate the entry for the CD burner and use the three numbers separated by commas as the value for dev. In this case, the Yamaha burner device is 1,5,0, so the appropriate input to specify that device is dev=1,5,0. Refer to the manual page for cdrecord for other ways to specify this value and for information on writing audio tracks and controlling the write speed.

Alternately, run the following command to get the device address of the burner:

```
# camcontrol devlist
```

Use the numeric values for scbus, target, and lun. For this example, 1,0,0 is the device name to use.

17.5.3. 写入资料到一个ISO档

In order to produce a data CD, the data files that are going to make up the tracks on the CD must be prepared before they can be burned to the CD. In FreeBSD, sysutils/cdrtools installs mkisofs, which can be used to produce an ISO 9660 file system that is an image of a directory tree within a UNIX™ file system. The simplest usage is to specify the name of the ISO file to create and the path to the files to place into the ISO 9660 file system:

```
# mkisofs -o imagefile.iso /path/to/tree
```

This command maps the file names in the specified path to names that fit the limitations of the standard ISO 9660 file system, and will exclude files that do not meet the standard for ISO file systems.

A number of options are available to overcome the restrictions imposed by the standard. In particular, -R enables the Rock Ridge extensions common to UNIX™ systems and -J enables Joliet extensions used by Microsoft™ systems.

For CDs that are going to be used only on FreeBSD systems, -U can be used to disable all filename restrictions. When used with -R, it produces a file system image that is identical to the specified FreeBSD tree, even if it violates the ISO 9660 standard.

The last option of general use is -b. This is used to specify the location of a boot image for use in producing an "El Torito" bootable CD. This option takes an argument which is the path to a boot image from the top of the tree being written to the CD. By default, mkisofs creates an ISO image in "floppy disk emulation" mode, and thus expects the boot image to be exactly 1200, 1440 or 2880 KB in size. Some boot loaders, like the one used by the FreeBSD distribution media, do not use emulation mode. In this case, -no-emul-boot should be used. So, if /tmp/myboot holds a bootable FreeBSD system with the boot image in /tmp/myboot/boot/cdboot, this command would produce /tmp/bootable.iso:
mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot

The resulting ISO image can be mounted as a memory disk with:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

One can then verify that `/mnt` and `/tmp/myboot` are identical.

There are many other options available for `mkisofs` to fine-tune its behavior. Refer to `mkisofs(8)` for details.

It is possible to copy a data CD to an image file that is functionally equivalent to the image file created with `mkisofs`. To do so, use `dd` with the device name as the input file and the name of the ISO to create as the output file:

```
# dd if=/dev/cd0 of=file.iso bs=2048
```

The resulting image file can be burned to CD as described in `燒錄CD`.

17.5.4. 使用資料CD

Once an ISO has been burned to a CD, it can be mounted by specifying the file system type, the name of the device containing the CD, and an existing mount point:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Since `mount` assumes that a file system is of type `ufs`, an `Incorrect super block` error will occur if `-t cd9660` is not included when mounting a data CD.

While any data CD can be mounted this way, disks with certain ISO 9660 extensions might behave oddly. For example, Joliet disks store all filenames in two-byte Unicode characters. If some non-English characters show up as question marks, specify the local charset with `-C`. For more information, refer to `mount_cd9660(8)`.

In order to do this character conversion with the help of `-C`, the kernel requires the `cd9660_iconv.ko` module to be loaded. This can be done either by adding this line to `loader.conf`:

```
cd9660_iconv_load="YES"
```

and then rebooting the machine, or by directly loading the module with `kldload`.

Occasionally, `Device not configured` will be displayed when trying to mount a data CD. This usually means that the CD drive has not detected a disk in the tray, or that the drive is not visible on the bus. It can take a couple of seconds for a CD drive to detect media, so be patient.

Sometimes, a SCSICD drive may be missed because it did not have enough time to answer the bus reset. To resolve this, a custom kernel can be created which increases the default SCSI delay. Add the following option to the custom kernel configuration file and rebuild the kernel using the instructions in `編譯與安裝自訂核心`.
This tells the SCSI bus to pause 15 seconds during boot, to give the CD drive every possible chance to answer the bus reset.

It is possible to burn a file directly to CD, without creating an ISO 9660 file system. This is known as burning a raw data CD and some people do this for backup purposes. This type of disk can not be mounted as a normal data CD. In order to retrieve the data burned to such a CD, the data must be read from the raw device node.

For example, this command will extract a compressed tar file located on the second CD device into the current working directory:

```
# tar xzvf /dev/cd1
```

In order to mount a data CD, the data must be written using `mkisofs`.

17.5.5. 複製音楽CD

To duplicate an audio CD, extract the audio data from the CD to a series of files, then write these files to a blank CD. Duplicating an Audio CD describes how to duplicate and burn an audio CD. If the FreeBSD version is less than 10.0 and the device is ATAPI, the `atapicam` module must be first loaded using the instructions in the manual page for ATAPI.

**Procedure: Duplicating an Audio CD**

1. The `sysutils/cdrtools` package or port installs `cdda2wav`. This command can be used to extract all of the audio tracks, with each track written to a separate WAV file in the current working directory:

   ```bash
   % cdda2wav -v all -B -O wav
   ```

   A device name does not need to be specified if there is only one CD device on the system. Refer to the `cdda2wav` manual page for instructions on how to specify a device and to learn more about the other options available for this command.

2. Use `cdrecord` to write the .wav files:

   ```bash
   % cdrecord -v dev=2,0 -dao -useinfo * .wav
   ```

   Make sure that `2,0` is set appropriately, as described in 燒錄CD.

17.6. 建立與使用DVD媒體

Compared to the CD, the DVD is the next generation of optical media storage technology. The DVD can hold more data than any CD and is the standard for video publishing. Five physical recordable formats can be defined for a recordable DVD:
• **DVD-R**: This was the first DVD recordable format available. The DVD-R standard is defined by the DVD Forum. This format is write once.

• **DVD-RW**: This is the rewritable version of the DVD-R standard. A DVD-RW can be rewritten about 1000 times.

• **DVD-RAM**: This is a rewritable format which can be seen as a removable hard drive. However, this media is not compatible with most DVD-ROM drives and DVD-Video players as only a few DVD writers support the DVD-RAM format. Refer to 使用 DVD-RAM for more information on DVD-RAM use.

• **DVD+RW**: This is a rewritable format defined by the DVD+RW Alliance. A DVD+RW can be rewritten about 1000 times.

• **DVD+R**: This format is the write once variation of the DVD+RW format.

A single layer recordable DVD can hold up to 4,700,000,000 bytes which is actually 4.38 GB or 4485 MB as 1 kilobyte is 1024 bytes.

A distinction must be made between the physical media and the application. For example, a DVD-Video is a specific file layout that can be written on any recordable DVD physical media such as DVD-R, DVD+R, or DVD-RW. Before choosing the type of media, ensure that both the burner and the DVD-Video player are compatible with the media under consideration.

17.6.1. To perform DVD recording, use `growisofs(1)`. This command is part of the `sysutils/dvd+rw-tools` utilities which support all DVD media types. These tools use the SCSI subsystem to access the devices, therefore `ATAPI/CAM support` must be loaded or statically compiled into the kernel. This support is not needed if the burner uses the USB interface. Refer to `USB储存裝置` for more details on USB device configuration.

DMA access must also be enabled for ATAPI devices, by adding the following line to `/boot/loader.conf`:

```
hw.ata.atapi_dma="1"
```

Before attempting to use dvd+rw-tools, consult the `Hardware Compatibility Notes`.

For a graphical user interface, consider using `sysutils/k3b` which provides a user friendly interface to `growisofs(1)` and many other burning tools.

17.6.2. Since `growisofs(1)` is a front-end to `mkisofs`, it will invoke `mkisofs(8)` to create the file system layout and perform the write on the DVD. This means that an image of the data does not need to be created before the burning process.

To burn to a DVD+R or a DVD-R the data in `/path/to/data`, use the following command:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

In this example, `-J -R` is passed to `mkisofs(8)` to create an ISO 9660 file system with Joliet and Rock Ridge extensions. Refer to `mkisofs(8)` for more details.

For the initial session recording, `-Z` is used for both single and multiple sessions. Replace `/dev/cd0`, with the name of the DVD device. Using `-dvd-compat` indicates that the disk will be closed and that
the recording will be unappendable. This should also provide better media compatibility with DVD-ROM drives.

To burn a pre-mastered image, such as `imagefile.iso`, use:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

The write speed should be detected and automatically set according to the media and the drive being used. To force the write speed, use `-speed=...`

Refer to `growisofs(1)` for example usage.

In order to support working files larger than 4.38GB, an UDF/ISO-9660 hybrid file system must be created by passing `-udf -iso-level 3` to `mkisofs(8)` and all related programs, such as `growisofs(1)`. This is required only when creating an ISO image file or when writing files directly to a disk. Since a disk created this way must be mounted as an UDF file system with `mount_udf(8)`, it will be usable only on an UDF aware operating system. Otherwise it will look as if it contains corrupted files.

To create this type of ISO file:

```
% mkisofs
```

To burn files directly to a disk:

```
# growisofs -dvd-compat -udf -iso-level 3 -Z /dev/cd0 -J -R /path/to/data
```

When an ISO image already contains large files, no additional options are required for `growisofs(1)` to burn that image on a disk.

Be sure to use an up-to-date version of `sysutils/cdrtools`, which contains `mkisofs(8)`, as an older version may not contain large files support. If the latest version does not work, install `sysutils/cdrtools-devel` and read its `mkisofs(8)`.

17.6.3. 燒錄 DVD-Video

A DVD-Video is a specific file layout based on the ISO 9660 and micro-UDF (M-UDF) specifications. Since DVD-Video presents a specific data structure hierarchy, a particular program such as `multimedia/dvdauthor` is needed to author the DVD.

If an image of the DVD-Video file system already exists, it can be burned in the same way as any other image. If `dvdauthor` was used to make the DVD and the result is in `/path/to/video`, the following command should be used to burn the DVD-Video:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

`-dvd-video` is passed to `mkisofs(8)` to instruct it to create a DVD-Video file system layout. This option implies the `-dvd-compat` `growisofs(1)` option.

17.6.4. 使用 DVD+RW

Unlike CD-RW, a virgin DVD+RW needs to be formatted before first use. It is recommended to let `growisofs(1)` take care of this automatically whenever appropriate. However, it is possible to use `dvd+rw-format` to format the DVD+RW:
**dvd+rw-format /dev/cd0**

Only perform this operation once and keep in mind that only virgin DVD+RW medias need to be formatted. Once formatted, the DVD+RW can be burned as usual.

To burn a totally new file system and not just append some data onto a DVD+RW, the media does not need to be blanked first. Instead, write over the previous recording like this:

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

The DVD+RW format supports appending data to a previous recording. This operation consists of merging a new session to the existing one as it is not considered to be multi-session writing. `growisofs(1)` will grow the ISO 9660 file system present on the media.

For example, to append data to a DVD+RW, use the following:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

The same `mkisofs(8)` options used to burn the initial session should be used during next writes.

To blank the media, use:

```
# growisofs -Z /dev/cd0=/dev/zero
```

17.6.5. **使用 DVD-RW**

A DVD-RW accepts two disc formats: incremental sequential and restricted overwrite. By default, DVD-RW discs are in sequential format.

A virgin DVD-RW can be directly written without being formatted. However, a non-virgin DVD-RW in sequential format needs to be blanked before writing a new initial session.

To blank a DVD-RW in sequential mode:

```
# dvd+rw-format -blank=full /dev/cd0
```

A full blanking using `-blank=full` will take about one hour on a 1x media. A fast blanking can be performed using `-blank`, if the DVD-RW will be recorded in Disk-At-Once (DAO) mode. To burn the DVD-RW in DAO mode, use the command:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
```

Since `growisofs(1)` automatically attempts to detect fast blanked media and engage DAO write, `-use-the-force-luke=dao` should not be required. One should instead use restricted overwrite mode with any DVD-RW as this format.
is more flexible than the default of incremental sequential.

To write data on a sequential DVD-RW, use the same instructions as for the other DVD formats:

```
$ growisofs -Z /dev/cd0 -J -R /path/to/data
```

To append some data to a previous recording, use `-M` with `growisofs(1)`. However, if data is appended on a DVD-RW in incremental sequential mode, a new session will be created on the disc and the result will be a multi-session disc.

A DVD-RW in restricted overwrite format does not need to be blanked before a new initial session. Instead, overwrite the disc with `-Z`. It is also possible to grow an existing ISO 9660 file system written on the disc with `-M`. The result will be a one-session DVD.

To put a DVD-RW in restricted overwrite format, the following command must be used:

```
$ dvd+rw-format /dev/cd0
```

To change back to sequential format, use:

```
$ dvd+rw-format -blank=full /dev/cd0
```

17.6.6. 多阶段烧录 (Multi-Session)

Few DVD-ROM drives support multi-session DVDs and most of the time only read the first session. DVD+R, DVD-R and DVD-RW in sequential format can accept multiple sessions. The notion of multiple sessions does not exist for the DVD+RW and the DVD-RW restricted overwrite formats.

Using the following command after an initial non-closed session on a DVD+R, DVD-R, or DVD-RW in sequential format, will add a new session to the disc:

```
$ growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Using this command with a DVD+RW or a DVD-RW in restricted overwrite mode will append data while merging the new session to the existing one. The result will be a single-session disc. Use this method to add data after an initial write on these types of media.

Since some space on the media is used between each session to mark the end and start of sessions, one should add sessions with a large amount of data to optimize media space. The number of sessions is limited to 154 for a DVD+R, about 2000 for a DVD-R, and 127 for a DVD+R Double Layer.

17.6.7. 取得更多资讯

To obtain more information about a DVD, use `dvd+rw-mediainfo /dev/cd0` while the disc in the specified drive.

More information about dvd+rw-tools can be found in `growisofs(1)`, on the dvd+rw-tools web site, and in the cdwrite mailing list archives.

When creating a problem report related to the use of dvd+rw-tools, always include the output of `dvd+rw-mediainfo`. 

351
DVD-RAM writers can use either a SCSI or ATAPI interface. For ATAPI devices, DMA access has to be enabled by adding the following line to `/boot/loader.conf`:

```
hw.ata.atapi_dma="1"
```

A DVD-RAM can be seen as a removable hard drive. Like any other hard drive, the DVD-RAM must be formatted before it can be used. In this example, the whole disk space will be formatted with a standard UFS2 file system:

```
# dd if=/dev/zero of=/dev/acd0 bs=2k count=1
# bsdlabel -Bw acd0
# newfs /dev/acd0
```

The DVD device, `acd0`, must be changed according to the configuration.

Once the DVD-RAM has been formatted, it can be mounted as a normal hard drive:

```
# mount /dev/acd0 /mnt
```

Once mounted, the DVD-RAM will be both readable and writeable.

---

This section explains how to format a 3.5 inch floppy disk in FreeBSD.

**Procedure: Steps to Format a Floppy**

A floppy disk needs to be low-level formatted before it can be used. This is usually done by the vendor, but formatting is a good way to check media integrity. To low-level format the floppy disk on FreeBSD, use `fdformat(1)`. When using this utility, make note of any error messages, as these can help determine if the disk is good or bad.

1. To format the floppy, insert a new 3.5 inch floppy disk into the first floppy drive and issue:

   ```
   # /usr/sbin/fdformat -f 1440 /dev/fd0
   ```

2. After low-level formatting the disk, create a disk label as it is needed by the system to determine the size of the disk and its geometry. The supported geometry values are listed in `/etc/disktab`.

   ```
   # /sbin/bsdlabel -B -w /dev/fd0 fd1440
   ```

3. The floppy is now ready to be high-level formatted with a file system. The floppy's file system can be either UFS or FAT, where FAT is generally a better choice for floppies.

   To format the floppy with FAT, issue:

   ```
   # fdformat -f 1440 /dev/fd0
   ```

```

352
The disk is now ready for use. To use the floppy, mount it with:
```
# /sbin/newfs_msdos /dev/fd0
```

rdump(8) from early versions of AT&T UNIX™, circa 1975, is a tool for restoring dumps to a target system. It can be used to restore dumps from a source system to a target system. The command to run rdump is:
```
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com
```

When restoring a dump, the target system is specified with the `-t` option. For example, to restore the dump to a different machine:
```
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com
```

The target system can be specified as an email address or a hostname. The email address is used to connect to the target system, and the hostname is used to identify the machine to which the dump is being restored.

```bash
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com
```

The output of the command is displayed on the screen, indicating the progress of the restoration process. Once the restoration process is complete, the target system is ready for use.

```bash
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com
```

The command also allows for the specification of a directory for the restored dump. For example, to restore the dump to a directory named `home`:
```
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```

While the restoration process is ongoing, the target system can be used to perform other tasks. If the restoration process is interrupted, the target system can be used to continue the restoration process. Once the restoration process is complete, the target system can be used to continue the task.

```bash
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```

The output of the command is displayed on the screen, indicating the progress of the restoration process. Once the restoration process is complete, the target system is ready for use.

```bash
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```

The command also allows for the specification of a directory for the restored dump. For example, to restore the dump to a directory named `home`:
```
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```

The output of the command is displayed on the screen, indicating the progress of the restoration process. Once the restoration process is complete, the target system is ready for use.

```bash
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```

The command also allows for the specification of a directory for the restored dump. For example, to restore the dump to a directory named `home`:
```
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```

The output of the command is displayed on the screen, indicating the progress of the restoration process. Once the restoration process is complete, the target system is ready for use.

```bash
# /usr/sbin/rdump -v -t targetuser@targetmachine.example.com -d /home
```
這個例子會設定 RSH，以便透過 SSH 連線寫入備份到遠端系統的磁帶機:

```
例 38. 在 ssh 使用 dump 透 RSH 設定
# env RSH=/usr/bin/ssh /sbin/dump -0uan -f
targetuser@targetmachine.example.com:/dev/sa0 /usr
```

17.8.2. 目錄備份

系統已有內建數個工具可在需要時用來備份與還原指定的檔案與目錄。

要備份一個目錄中的所有檔案最好選擇 tar(1)，這個工具最早可以追溯自 AT&T UNIX™ 版本 6，因此預設會做一個遞迴備份到一個磁帶機，可以使用參數來指定備份檔案的名稱。

這個例子會建立目前目錄的壓縮備份並儲存至 /tmp/mybackup.tgz，在建立備份檔案時，要確認備份檔案不要儲存到與目前備份目錄相同的目錄。

```
例 39. 使用 tar 備份目前目錄
# tar czvf /tmp/mybackup.tgz .
```

要還原整個備份，先 cd 進入要放置還原檔的目錄並指定備份的名稱。注意，這個動作會覆寫任何在該還原目錄中任何較新版的檔案，當不確定時，可先還原到一個暫時的目錄或指定備份檔案中的檔案做還原。

```
例 40. 使用 tar 還原目前目錄
# tar xzvf /tmp/mybackup.tgz
```

除此之外還有多可用的參數在 tar(1) 中會有說明。本工具也支援使用排除模式 (Exclude pattern) 來指定那些檔案應該在備份指定目錄或自備份還原檔案時排除。

要使用指定的檔案與目錄清單做備份使用 cpio(1) 是不錯的選擇。它並不像 tar，cpio 並不知道如何走訪目錄樹，所以必須提供檔案的清單才能做備份。

```
例 41. 使用 ls 與 cpio 來製作目前目錄的遞迴備份
# ls -R | cpio -ovF /tmp/mybackup.cpio
```

有一個備份工具嘗試整合 tar 與 cpio 所提供的功能，便是 pax(1)。經歷數年，各種版本的 tar 與 cpio 變的有一些無法相容。POSIX™ 開發出 pax，嘗試讀取與寫入各種版本的 cpio and tar 格式並加入自己的新格式。

以先前的例子改使用 pax 會是:

```
354
```
使用
備份目前目錄

# pax -wf /tmp/mybackup.pax .

17.8.3.
使用資料磁帶備份

隨著磁帶的技術持續發展，當今的備份系統將異地備份與本地可移除媒體做了結合。
FreeBSD支援任使用SCSI的磁帶機，如LTO或DAT，並有限制的支援SATA與USB磁帶機。

SCSI磁帶機在FreeBSD會使用sa(4)驅動程式及/dev/sa0, /dev/nsa0與/dev/esa0裝置，實體裝置名稱為/dev/sa0，當使用/dev/nsa0時，備份程式在寫入檔案之後不會倒帶，這可允許寫入超過一個檔案到磁帶，而使用/dev/esa0時，當關閉裝置後便會退出磁帶。

在FreeBSD中會使用mt來做磁帶機的控制操作，例如在磁帶中搜尋檔案或寫入磁帶控制記號到磁帶。
例如，要保留磁帶上的前三個檔案，可以在寫入新檔案前跳過這些檔案：

```bash
# mt -f /dev/nsa0 fsf 3
```

這個工具尚支援許多操作，請參考mt(1)了解詳情。

要使用tar寫入单一檔案到磁帶，可指定磁帶裝置的名稱及要備份的檔案：

```bash
# tar cvf /dev/sa0 file
```

要從磁帶上的tar封存档還原檔案到目前目錄可：

```bash
# tar xvf /dev/sa0
```

要備份一個UFS檔案系統可使用dump。以下例子會備份/usr並在完成時不倒帶：

```bash
# dump -0aL -b64 -f /dev/nsa0 /usr
```

要以互動的方式從磁帶上的dump檔案還原到目前目錄：

```bash
# restore -i -f /dev/nsa0
```

17.8.4.
第三方備份工具

FreeBSD Port套件提供了許多第三方工具可用於排程建立備份，簡化磁帶備份並讓備份更簡單方便。許多這類應用程式是以客戶端/伺服器為基礎，可用來自動化單一系統或網路上所有電腦的備份。

較熱門的工具包含Amanda, Bacula, rsync以及duplicity。
緊急

除了正規の備份外、建言を下のこと步を超え緊急の準備計画の一一部として、以下のような手順を設立することが推奨されます。

1. gpart show
2. more /etc/fstab
3. dmesg

これらの結果を印刷するためのバックアップを作成してください。緊急時においては、これらのファイルが重要になるかもしれません。さらに、安謹に管理して、インストールメディアを再開する際、Live CD (Rescue shell) を選択することが推奨されます。

FreeBSD/i386 11.2-RELEASE のインストールメディアには救急シェルが含まれていません。それを取得するには、次の URL から Livefs CD イメージをダウンロードし、ISO に書き込みます。


その後、救急シェルの下のファイルをバックアップし、それをメモクリップと印刷し保存してください。これは緊張が高まると、誤ってバックアップが破壊されることを防ぐことができます。

要するに、新しくバックアップを保存し、実体機器と磁気ドライブの間に明確な距離をとることで、安全性は向上します。

17.9.1.4. 記憶体ディスク

In addition to physical disks, FreeBSD also supports the creation and use of memory disks. One possible use for a memory disk is to access the contents of an ISO file system without the overhead of first burning it to a CD or DVD, then mounting the CD/DVD media.

In FreeBSD, the md(4) driver is used to provide support for memory disks. The GENERIC kernel includes this driver. When using a custom kernel configuration file, ensure it includes this line:

device md

17.9.1.5. 連接と解連接既存の映像

To mount an existing file system image, use mdconfig to specify the name of the ISO file and a free unit number. Then, refer to that unit number to mount it on an existing mount point. Once mounted, the files in the ISO will appear in the mount point. This example attaches diskimage.iso to the memory device /dev/md0 and mounts that memory device on /mnt:

```
# mdconfig -f diskimage.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Notice that -t cd9660 was used to mount an ISO format. If a unit number is not specified with -u, mdconfig will automatically allocate an unused memory device and output the name of the allocated unit, such as md4. Refer to mdconfig(8) for more details about this command and its options.

When a memory disk is no longer in use, its resources should be released back to the system. First, unmount the file system, then use mdconfig to detach the disk from the system and release its resources. To continue this example:
To determine if any memory disks are still attached to the system, type `mdconfig -l`.

### 17.9.2. 建立以檔案或記憶體為基礎的磁碟

FreeBSD also supports memory disks where the storage to use is allocated from either a hard disk or an area of memory. The first method is commonly referred to as a file-backed file system and the second method as a memory-backed file system. Both types can be created using `mdconfig`.

To create a new memory-backed file system, specify a type of `swap` and the size of the memory disk to create. Then, format the memory disk with a file system and mount as usual. This example creates a 5M memory disk on unit 1. That memory disk is then formatted with the UFS file system before it is mounted:

```
# mdconfig -a -t swap -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors)
block size 16384, fragment size 2048 using 4 cylinder groups of 1.27MB, 81 blks, 192 inodes.
with soft updates super-block backups (for fsck -b) at: 160, 2752, 5344, 7936
```

```
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity Mounted on
/dev/md1        4718    4  4338     0%    /mnt
```

To create a new file-backed memory disk, first allocate an area of disk to use. This example creates an empty 5MB file named `newimage`:

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
```

Next, attach that file to a memory disk, label the memory disk and format it with the UFS file system, mount the memory disk, and verify the size of the file-backed disk:

```
# mdconfig -f newimage -u 0
# bsdlabel -w md0 auto
# newfs -U md0a
/dev/md0a: 5.0MB (10224 sectors)
block size 16384, fragment size 2048 using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
```

```
# mount /dev/md0a /mnt
```

It takes several commands to create a file- or memory-backed file system using `mdconfig`. FreeBSD also comes with `mdmfs` which automatically configures a memory disk, formats it with the UFS file system, and mounts it. For example, after creating `newimage` with `dd`, this one command is equivalent to running the `bsdlabel`, `newfs`, and `mount` commands shown above:

```
# mdmfs -F newimage -s 5m md0 /mnt
```

To instead create a new memory-based memory disk with `mdmfs`, use this one command:

```
# mdmfs -s 5m md1 /mnt
```

If the unit number is not specified, `mdmfs` will automatically select an unused memory device. For more details about `mdmfs`, refer to `mdmfs(8)`.

17.10. 檔案系統快照
FreeBSD offers a feature in conjunction with Soft Updates: file system snapshots. UFS snapshots allow a user to create images of specified file systems, and treat them as a file. Snapshot files must be created in the file system that the action is performed on, and a user may create no more than 20 snapshots per file system. Active snapshots are recorded in the superblock so they are persistent across unmount and remount operations along with system reboots. When a snapshot is no longer required, it can be removed using `rm(1)`. While snapshots may be removed in any order, all the used space may not be acquired because another snapshot will possibly claim some of the released blocks.

The un-alterable snapshot file flag is set by `mksnap_ffs(8)` after initial creation of a snapshot file. `unlink(1)` makes an exception for snapshot files since it allows them to be removed.

Snapshots are created using `mount(8)`. To place a snapshot of `/var` in the file `/var/snapshot/snap`, use the following command:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

Alternatively, use `mksnap_ffs(8)` to create the snapshot:

```
# mksnap_ffs /var /var/snapshot/snap
```

One can find snapshot files on a file system, such as `/var`, using `find(1)`:  

```
# find /var -flags snapshot
```

Once a snapshot has been created, it has several uses:

- Some administrators will use a snapshot file for backup purposes, because the snapshot can be transferred to CDs or tape.
The file system integrity checker, `fsck(8)`, may be run on the snapshot. Assuming that the file system was clean when it was mounted, this should always provide a clean and unchanging result.

Running `dump(8)` on the snapshot will produce a dump file that is consistent with the file system and the timestamp of the snapshot. `dump(8)` can also take a snapshot, create a dump image, and then remove the snapshot in one command by using `-L`.

The snapshot can be mounted as a frozen image of the file system. To mount the snapshot `/var/snapshot/snap` run:

```
# mdconfig -a -t vnode -o readonly -f /var/snapshot/snap -u 4
# mount -r /dev/md4 /mnt
```

The frozen `/var` is now available through `/mnt`. Everything will initially be in the same state it was during the snapshot creation time. The only exception is that any earlier snapshots will appear as zero length files. To unmount the snapshot, use:

```
# umount /mnt
# mdconfig -d -u 4
```

For more information about softupdates and file system snapshots, including technical papers, visit Marshall Kirk McKusick's website at http://www.mckusick.com/.

17.11. 磁碟配額

磁碟配額可以用來限制使用者或群組成員能夠在個別檔案系統上使用的磁碟空間或檔案數量。這個可以避免一個使用者或群組成員耗盡所有磁碟的可用空間。

本節將說明如何設定 UFS 檔案系統的磁碟配額。要在 ZFS 檔案系統上設定配額，請參考資料集、使用者以及群組配額。

17.11.1. 開啟磁碟配額

查看 FreeBSD 核心是否支援磁碟配額:

```
% sysctl kern.features.ufs_quota
```

在本例中，數值 1 代表支援磁碟配額，若為 0，則需加入下列設定到自訂核心設定檔然後依照設定 FreeBSD 核心的指示重新編譯核心:

```
options QUOTA
```

接著，在 `/etc/rc.conf` 開啟磁碟配額:

```
quota_enable="YES"
```

正常在開機時，會使用 `quotacheck(8)` 檢查每個檔案系統的配額完整性，這個程式會確保在配額資料庫中的資料正確的反映在檔案系統上的資料。

這是一個耗時間的程序，會明顯的影響系統開機的時間，要跳過這個步驟可以加入此變數到 `/etc/rc.conf`:

```
359
```
最後，編輯/etc/fstab 來開啓在各檔案系統上的磁碟配額。要開啓在檔案系統上對每個使用者的配額要加入 userquota 選項到 /etc/fstab 中。要開啓配額的檔案系統的項目中。例如:

/dev/da1s2g   /home    ufs rw,userquota 1 2

要開啓群組配額，則使用 groupquota。要同時開啓使用者及群組配額，可使用逗號隔開選項:

/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2

預設配額檔案會儲存在檔案系統的根目錄的 quota.user 及 quota.group，請參考 fstab(5)來取得更多資訊，較不建議指定其他位置來儲存配額檔案。

設定完成之後，重新啟動系統，/etc/rc 會自動執行適當的指令對所有在 /etc/fstab 中開啓配額的檔案系統建立初始的配額檔案。

在一般的操作中，並不需要手動執行 quotacheck(8), quotaon(8) 是, 尽管如此，仍應閱讀這些指令的操作手冊來熟悉這些指令的操作。

17.11.2. 設定配額限制
要確認配額已經開啓，可執行:

# quota -v

每個有開啓配額的檔案系統應該會有一行磁碟用量及目前配額限制的摘要。
現在我們已準備好可以使用 edquota 分配配額限制。

有數個選項可以強制限制使用者或群組對磁碟空間的使用量及可以建立多少檔案。可以用磁碟空間 (block 配額), 檔案數量 (inode 配額) 或同時使用來分配。

每種限制又可進一步細分為兩個類型：硬性 (Hard) 及軟性 (Soft) 限制。

硬性限制無法被超額使用。一旦使用者超出硬性限制，該使用者在該檔案系統將無法再使用任何空間。例如來說，若一個使用者在一個檔案系統上 有 500 KB 的硬性限制，且目前已經使用了 490 KB，該使用者只能再使用 10 KB 的空間，若嘗試使用 11 KB 的空間將會失敗。

軟性限制在有限的時間內可以被超額使用，即為寬限期 (Grace period)，預設為一週。若一個使用者超出限制並超出寬限期，則軟性限制將轉為硬性限制並且將不允許再使用空間。當使用者使用的空間回降到低於軟性限制內，寬限期就會被重置。

在下面的例子中，會編輯 test 的配額。當執行 edquota 時，將會使用 EDITOR 指定的編輯器來編輯配額限制。預設的編輯器為 vi。

# edquota -u test

Quotas for user test:

/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
inodes

in use: 0, limits (soft = 50, hard = 60)

正常每個開啓配額的檔案系統會有兩行需要設定，一行代表區塊限制(Block limit)而另ㄧ行代表節點限制(inode limit)，更改行內的值來修改配額限制。

例如來說，要在/usr 提高區塊的軟性限制到 500 及硬性限制到 600，可更改行內的值如下:

/usr: kbytes in use: 65, limits (soft = 500, hard = 600)

新的配額限制將在離開編輯器後生效。

有時會想想要對一羣使用者設定配額限制，這時可以透過指定想想要的配額給第一個使用者，若然後使用-p來複製配額到指定範圍的使用者ID (UID)。以下指定將複製配額限制給UID 10,000到19,999的使用者:

# edquota -p test 10000-19999

要取得更多資訊，請參考edquota(8)。

17.11.3. 檢查配額限制與磁碟使用狀況

要檢查各別使用者或群組的配額與磁碟用量可使用quota(1)。使用者僅可查看自己及所屬群組的配額，只有超級使用者可以檢視所有使用者及群組的配額。要取得某個有開啓配額的檔案系統的所有配額及磁碟用量摘要，可使用repquota(8)。

正常に，使用者未使用任何磁碟空間的檔案系統並不會顯示在quota的輸出結果中，即使該使用者或群組在該檔案系統設定配額限制，使用-v可以顯示這些檔案系統。以下是使用使用quota -v查詢某個使用者在兩個檔案系統上的配額限制的範例輸出。

Disk quotas for user test (uid 1002):

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>usage</th>
<th>quota</th>
<th>limit</th>
<th>grace</th>
<th>files</th>
<th>quota</th>
<th>limit</th>
<th>grace</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr</td>
<td>65*</td>
<td>50</td>
<td>75</td>
<td>5days</td>
<td>7</td>
<td>50</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>/usr/var</td>
<td>0</td>
<td>50</td>
<td>75</td>
<td></td>
<td>0</td>
<td>50</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

在這個例子當中，使用者在/usr的軟性限制50 KB已經超出15 KB並已經過了5天寬限期。星號*代表該使用者目前已超出配額限制。
FreeBSD offers excellent online protections against unauthorized data access. File permissions and Mandatory Access Control (MAC) help prevent unauthorized users from accessing data while the operating system is active and the computer is powered up. However, the permissions enforced by the operating system are irrelevant if an attacker has physical access to a computer and can move the computer's hard drive to another system to copy and analyze the data.

Regardless of how an attacker may have come into possession of a hard drive or powered-down computer, the GEOM-based cryptographic subsystems built into FreeBSD are able to protect the data on the computer's file systems against even highly-motivated attackers with significant resources. Unlike encryption methods that encrypt individual files, the built-in gbde and geli utilities can be used to transparently encrypt entire file systems. No cleartext ever touches the hard drive's platter.

This chapter demonstrates how to create an encrypted file system on FreeBSD. It first demonstrates the process using gbde and then demonstrates the same example using geli.

17.12.1. 使用gbde做磁碟加密

The objective of the gbde(4) facility is to provide a formidable challenge for an attacker to gain access to the contents of a cold storage device. However, if the computer is compromised while up and running and the storage device is actively attached, or the attacker has access to a valid passphrase, it offers no protection to the contents of the storage device. Thus, it is important to provide physical security while the system is running and to protect the passphrase used by the encryption mechanism.

This facility provides several barriers to protect the data stored in each disk sector. It encrypts the contents of a disk sector using 128-bit AES in CBC mode. Each sector on the disk is encrypted with a different AES key. For more information on the cryptographic design, including how the sector keys are derived from the user-supplied passphrase, refer to gbde(4).

FreeBSD provides a kernel module for gbde which can be loaded with this command:

```
# kldload geom_bde
```

If using a custom kernel configuration file, ensure it contains this line:

```
options GEOM_BDE
```

The following example demonstrates adding a new hard drive to a system that will hold a single encrypted partition that will be mounted as /private.

**Procedure: Encrypting a Partition with gbde**

1. **Add the New Hard Drive**
   - Install the new drive to the system as explained in [加入磁碟](#). For the purposes of this example, a new hard drive partition has been added as /dev/ad4s1c and /dev/ad0s1* represents the existing standard FreeBSD partitions.

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   /dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
   /dev/ad0s1a     /dev/ad0s1d     /dev/ad4
   ```

362
2. Create a Directory to Hold gbde Lock Files

```bash
# mkdir /etc/gbde
```

The gbde lock file contains information that gbde requires to access encrypted partitions. Without access to the lock file, gbde will not be able to decrypt the data contained in the encrypted partition without significant manual intervention which is not supported by the software. Each encrypted partition uses a separate lock file.

3. Initialize the gbde Partition

A gbde partition must be initialized before it can be used. This initialization needs to be performed only once. This command will open the default editor, in order to set various configuration options in a template. For use with the UFS file system, set the sector_size to 2048:

```bash
# gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c.lock
```

Once the edit is saved, the user will be asked twice to type the passphrase used to secure the data. The passphrase must be the same both times. The ability of gbde to protect data depends entirely on the quality of the passphrase. For tips on how to select a secure passphrase that is easy to remember, see [http://world.std.com/~reinhold/diceware.htm](http://world.std.com/~reinhold/diceware.htm).

This initialization creates a lock file for the gbde partition. In this example, it is stored as `/etc/gbde/ad4s1c.lock`. Lock files must end in ".lock" in order to be correctly detected by the `/etc/rc.d/gbde` startup script.

5. Attach the Encrypted Partition to the Kernel

```bash
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

This command will prompt to input the passphrase that was selected during the initialization of the encrypted partition. The new encrypted device will appear in `/dev` as `/dev/device_name.bde`:

```bash
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Create a File System on the Encrypted Device

Once the encrypted device has been attached to the kernel, a file system can be created on the device. This example creates a UFS file system with soft updates enabled. Be sure to specify the partition which has a ".bde" extension:
6. Mount the Encrypted Partition

Create a mount point and mount the encrypted file system:

```bash
# mkdir /private
# mount /dev/ad4s1c.bde /private
```

7. Verify That the Encrypted File System is Available

The encrypted file system should now be visible and available for use:

```
% df -H
Filesystem        Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a      1037M    72M   883M     8%    /
/devfs            1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f       8.1G    55K   7.5G     0%    /home
/dev/ad0s1e      1037M   1.1M   953M     0%    /tmp
/dev/ad4s1c.bde   150G   4.1K   138G     0%    /private
```

After each boot, any encrypted file systems must be manually re-attached to the kernel, checked for errors, and mounted, before the file systems can be used. To configure these steps, add the following lines to `/etc/rc.conf`:

```bash
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
gbde_lockdir="/etc/gbde"
```

This requires that the passphrase be entered at the console at boot time. After typing the correct passphrase, the encrypted partition will be mounted automatically. Additional gbde boot options are available and listed in `rc.conf(5)`.

---

**17.12.2. 使用 geli 做磁碟加密**

An alternative cryptographic GEOM class is available using `geli`. This control utility adds some features and uses a different scheme for doing cryptographic work. It provides the following features:
• Utilizes the crypto(9) framework and automatically uses cryptographic hardware when it is available.
• Supports multiple cryptographic algorithms such as AES, Blowfish, and 3DES.
• Allows the root partition to be encrypted. The passphrase used to access the encrypted root partition will be requested during system boot.
• Allows the use of two independent keys.
• It is fast as it performs simple sector-to-sector encryption.
• Allows backup and restore of master keys. If a user destroys their keys, it is still possible to get access to the data by restoring keys from the backup.
• Allows a disk to attach with a random, one-time key which is useful for swap partitions and temporary file systems.

More features and usage examples can be found in geli(8).

The following example describes how to generate a key file which will be used as part of the master key for the encrypted provider mounted under /private. The key file will provide some random data used to encrypt the master key. The master key will also be protected by a passphrase. The provider's sector size will be 4kB. The example describes how to attach to the geli provider, create a file system on it, mount it, work with it, and finally, how to detach it.

Procedure: Encrypting a Partition with geli

1. Load geli Support

Support for geli is available as a loadable kernel module. To configure the system to automatically load the module at boot time, add the following line to /boot/loader.conf:

```
geom_eli_load="YES"
```

To load the kernel module now:

```
# kldload geom_eli
```

For a custom kernel, ensure the kernel configuration file contains these lines:

```
options GEOM_ELI
device crypto
```

2. Generate the Master Key

The following commands generate a master key (/root/da2.key) that is protected with a passphrase. The data source for the key file is /dev/random and the sector size of the provider (/dev/da2.eli) is 4kB as a bigger sector size provides better performance:

```
# dd if=/dev/random of=/root/da2.key bs=64 count=1
# geli init -s 4096 -K /root/da2.key /dev/da2
```

Enter new passphrase:

```
Reenter new passphrase:
```

365
It is not mandatory to use both a passphrase and a key file as either method of securing the master key can be used in isolation.

If the key file is given as "-", standard input will be used. For example, this command generates three key files:

```
# cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
```

3. Attach the Provider with the Generated Key

To attach the provider, specify the key file, the name of the disk, and the passphrase:

```
# geli attach -k /root/da2.key /dev/da2
```

Enter passphrase:

This creates a new device with an `.eli` extension:

```
# ls /dev/da2*
/dev/da2  /dev/da2.eli
```

4. Create the New File System

Next, format the device with the UFS file system and mount it on an existing mount point:

```
# dd if=/dev/random of=/dev/da2.eli bs=1m
# newfs /dev/da2.eli
# mount /dev/da2.eli /private
```

The encrypted file system should now be available for use:

```
# df -H
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a    248M    89M   139M    38%    /
/devfs         1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
/dev/ad0s1d    989M   1.5M   909M     0%    /tmp
/dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
/dev/da2.eli   150G   4.1K   138G     0%    /private
```

Once the work on the encrypted partition is done, and the `/private` partition is no longer needed, it is prudent to put the device into cold storage by unmounting and detaching the geli encrypted partition from the kernel:

```
# umount /private
# geli detach da2.eli
```
A script is provided to simplify the mounting of geli-encrypted devices at boot time. For this example, add these lines to /etc/rc.conf:

```
geli_devices="da2"
geli_da2_flags="-k /root/da2.key"
```

This configures /dev/da2 as a geli provider with a master key of /root/da2.key. The system will automatically detach the provider from the kernel before the system shuts down. During the startup process, the script will prompt for the passphrase before attaching the provider. Other kernel messages might be shown before and after the password prompt. If the boot process seems to stall, look carefully for the password prompt among the other messages. Once the correct passphrase is entered, the provider is attached. The file system is then mounted, typically by an entry in /etc/fstab. Refer to 挂载与卸载档系统 for instructions on how to configure a file system to mount at boot time.

17.13. 交换空间加密

Like the encryption of disk partitions, encryption of swap space is used to protect sensitive information. Consider an application that deals with passwords. As long as these passwords stay in physical memory, they are not written to disk and will be cleared after a reboot. However, if FreeBSD starts swapping out memory pages to free space, the passwords may be written to the disk unencrypted. Encrypting swap space can be a solution for this scenario.

This section demonstrates how to configure an encrypted swap partition using gbde(8) or geli(8) encryption. It assumes that /dev/ada0s1b is the swap partition.

17.13.1. 設定已加密的交换空间

Swap partitions are not encrypted by default and should be cleared of any sensitive data before continuing. To overwrite the current swap partition with random garbage, execute the following command:

```
# dd if=/dev/random of=/dev/ada0s1b bs=1m
```

To encrypt the swap partition using gbde(8), add the .bde suffix to the swap line in /etc/fstab:

```
# Device        Mountpoint  FStype  Options     Dump    Pass#
/dev/ada0s1b.bde    none        swap    sw      0   0
```

To instead encrypt the swap partition using geli(8), use the .eli suffix:

```
# Device        Mountpoint  FStype  Options     Dump    Pass#
/dev/ada0s1b.eli    none        swap    sw      0   0
```

By default, geli(8) uses the AES algorithm with a key length of 128 bits. Normally the default settings will suffice. If desired, these defaults can be altered in the options field in /etc/fstab. The possible flags are:

- aalgo
  - Data integrity verification algorithm used to ensure that the encrypted data has not been tampered with. See geli(8) for a list of supported algorithms.
Encryption algorithm used to protect the data. See geli(8) for a list of supported algorithms.

keylen
The length of the key used for the encryption algorithm. See geli(8) for the key lengths that are supported by each encryption algorithm.

sectorsize
The size of the blocks data is broken into before it is encrypted. Larger sector sizes increase performance at the cost of higher storage overhead. The recommended size is 4096 bytes.

This example configures an encrypted swap partition using the Blowfish algorithm with a key length of 128 bits and a sectorsize of 4 kilobytes:

```
# Device        Mountpoint  FStype  Options             Dump    Pass#
/dev/ada0s1b.eli    none        swap    sw,ealgo=blowfish,keylen=128,sectorsize=4096    0   0
```

Once the system has rebooted, proper operation of the encrypted swap can be verified using swapinfo.

If gbde(8) is being used:
```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ada0s1b.bde   542720        0   542720     0%
```

If geli(8) is being used:
```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ada0s1b.eli   542720        0   542720     0%
```

High availability is one of the main requirements in serious business applications and highly-available storage is a key component in such environments. In FreeBSD, the Highly Available Storage (HAST) framework allows transparent storage of the same data across several physically separated machines connected by a TCP/IP network. HAST can be understood as a network-based RAID1 (mirror), and is similar to the DRBD® storage system used in the GNU/Linux™ platform. In combination with other high-availability features of FreeBSD like CARP, HAST makes it possible to build a highly-available storage cluster that is resistant to hardware failures.

The following are the main features of HAST:

• Can be used to mask I/O errors on local hard drives.
• File system agnostic as it works with any file system supported by FreeBSD.
• Efficient and quick resynchronization as only the blocks that were modified during the downtime of a node are synchronized.
• Can be used in an already deployed environment to add additional redundancy.
Together with CARP, Heartbeat, or other tools, it can be used to build a robust and durable storage system.

After reading this section, you will know:
• What HAST is, how it works, and which features it provides.
• How to set up and use HAST on FreeBSD.
• How to integrate CARP and devd(8) to build a robust storage system.

Before reading this section, you should:
• 了解 UNIX™ 及 FreeBSD 基礎 (FreeBSD 基礎).
• Know how to configure network interfaces and other core FreeBSD subsystems (設定與調校).
• Have a good understanding of FreeBSD networking (網路通訊).

The HAST project was sponsored by The FreeBSD Foundation with support from http://www.omc.net/ and http://www.transip.nl/.

17.14.1. HAST 运作模式

HAST provides synchronous block-level replication between two physical machines: the primary, also known as the master node, and the secondary, or slave node. These two machines together are referred to as a cluster.

Since HAST works in a primary-secondary configuration, it allows only one of the cluster nodes to be active at any given time. The primary node, also called active, is the one which will handle all the I/O requests to HAST-managed devices. The secondary node is automatically synchronized from the primary node.

The physical components of the HAST system are the local disk on primary node, and the disk on the remote, secondary node.

HAST operates synchronously on a block level, making it transparent to file systems and applications. HAST provides regular GEOM providers in /dev/hast/ for use by other tools or applications. There is no difference between using HAST-provided devices and raw disks or partitions.

Each write, delete, or flush operation is sent to both the local disk and to the remote disk over TCP/IP. Each read operation is served from the local disk, unless the local disk is not up-to-date or an I/O error occurs. In such cases, the read operation is sent to the secondary node.

HAST tries to provide fast failure recovery. For this reason, it is important to reduce synchronization time after a node's outage. To provide fast synchronization, HAST manages an on-disk bitmap of dirty extents and only synchronizes those during a regular synchronization, with an exception of the initial sync.

There are many ways to handle synchronization. HAST implements several replication modes to handle different synchronization methods:
• memsync: This mode reports a write operation as completed when the local write operation is finished and when the remote node acknowledges data arrival, but before actually storing the data. The data on the remote node will be stored directly after sending the acknowledgement. This mode is intended to reduce latency, but still provides good reliability. This mode is the default.
• fullsync: This mode reports a write operation as completed when both the local write and the remote write complete. This is the safest and the slowest replication mode.
• async: This mode reports a write operation as completed when the local write completes. This is the fastest and the most dangerous replication mode. It should only be used when replicating to a distant node where latency is too high for other modes.
The HAST framework consists of several components:

• The hastd(8) daemon which provides data synchronization. When this daemon is started, it will automatically load geom_gate.ko.

• The userland management utility, hastctl(8).

• The hast.conf(5) configuration file. This file must exist before starting hastd.

Users who prefer to statically build GEOM_GATE support into the kernel should add this line to the custom kernel configuration file, then rebuild the kernel using the instructions in FreeBSD "{core}:

```
options GEOM_GATE
```

The following example describes how to configure two nodes in master-slave/primary-secondary operation using HAST to replicate the data between the two. The nodes will be called hasta, with an IP address of 172.16.0.1, and hastb, with an IP address of 172.16.0.2. Both nodes will have a dedicated hard drive /dev/ad6 of the same size for HAST operation. The HAST pool, sometimes referred to as a resource or the GEOM provider in /dev/hast/, will be called test.

Configuration of HAST is done using /etc/hast.conf. This file should be identical on both nodes. The simplest configuration is:

```
resource test {
  on hasta {
    local /dev/ad6
    remote 172.16.0.2
  }
  on hastb {
    local /dev/ad6
    remote 172.16.0.1
  }
}
```

For more advanced configuration, refer to hast.conf(5).

It is also possible to use host names in the remote statements if the hosts are resolvable and defined either in /etc/hosts or in the local DNS.

Once the configuration exists on both nodes, the HAST pool can be created. Run these commands on both nodes to place the initial metadata onto the local disk and to start hastd(8):

```
# hastctl create test
# service hastd onestart
```

It is not possible to use GEOM providers with an existing file system or to convert an existing storage to a HAST-managed pool. This procedure needs to store some metadata on the provider and there will not be enough required space available on an existing provider.
A HAST node's primary or secondary role is selected by an administrator, or software like Heartbeat, using hastctl(8). On the primary node, hasta, issue this command:

```
# hastctl role primary test
```

Run this command on the secondary node, hastb:

```
# hastctl role secondary test
```

Verify the result by running hastctl on each node:

```
# hastctl status test
```

Check the status line in the output. If it says degraded, something is wrong with the configuration file. It should say complete on each node, meaning that the synchronization between the nodes has started. The synchronization completes when hastctl status reports 0 bytes of dirty extents.

The next step is to create a file system on the GEOM provider and mount it. This must be done on the primary node. Creating the file system can take a few minutes, depending on the size of the hard drive. This example creates a UFS file system on /dev/hast/test:

```
# newfs -U /dev/hast/test
# mkdir /hast/test
# mount /dev/hast/test /hast/test
```

Once the HAST framework is configured properly, the final step is to make sure that HAST is started automatically during system boot. Add this line to /etc/rc.conf:

```
hastd_enable="YES"
```

The goal of this example is to build a robust storage system which is resistant to the failure of any given node. If the primary node fails, the secondary node is there to take over seamlessly, check and mount the file system, and continue to work without missing a single bit of data.

To accomplish this task, the Common Address Redundancy Protocol (CARP) is used to provide for automatic failover at the IP layer. CARP allows multiple hosts on the same network segment to share an IP address. Set up CARP on both nodes of the cluster according to the documentation available in 共用位址備援協定(CARP). In this example, each node will have its own management IP address and a shared IP address of 172.16.0.254. The primary HAST node of the cluster must be the master CARP node.

The HAST pool created in the previous section is now ready to be exported to the other hosts on the network. This can be accomplished by exporting it through NFS or Samba, using the shared IP address 172.16.0.254. The only problem which remains unresolved is an automatic failover should the primary node fail.

In the event of CARP interfaces going up or down, the FreeBSD operating system generates a devd(8) event, making it possible to watch for state changes on the CARP interfaces. A state change on the CARP interface is an indication that one of the nodes failed or came back online. These state change events make it possible to run a script which will automatically handle the HAST failover.
To catch state changes on the CARP interfaces, add this configuration to /etc/devd.conf on each node:

```
notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_UP";
    action "/usr/local/sbin/carp-hast-switch master";
}
```

```
notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_DOWN";
    action "/usr/local/sbin/carp-hast-switch slave";
}
```

If the systems are running FreeBSD 10 or higher, replace carp0 with the name of the CARP-configured interface.

Restart devd(8) on both nodes to put the new configuration into effect:

```
# service devd restart
```

When the specified interface state changes by going up or down, the system generates a notification, allowing the devd(8) subsystem to run the specified automatic failover script, /usr/local/sbin/carp-hast-switch. For further clarification about this configuration, refer to devd.conf(5).

Here is an example of an automated failover script:

```
#!/bin/sh
# Original script by Freddie Cash <fjwcash@gmail.com>
# Modified by Michael W. Lucas <mwlucas@BlackHelicopters.org>
# and Viktor Petersson <vpetersson@wireload.net>
# The names of the HAST resources, as listed in /etc/hast.conf
resources="test"
# delay in mounting HAST resource after becoming master
# make your best guess
delay=3
# logging
```
case "$1" as:
  master
    logger -p $log -t $name "Switching to primary provider for ${resources}."
    sleep ${delay}
    # Wait for any "hastd secondary" processes to stop
    for disk in ${resources}; do
      while $(pgrep -lf "hastd: ${disk} (secondary)" > /dev/null 2>&1); do
        sleep 1
      done
      # Switch role for each disk
      hastctl role primary ${disk}
      if [ $? -ne 0 ]; then
        logger -p $log -t $name "Unable to change role to primary for resource ${disk}."
        exit 1
      fi
    done
    # Wait for the /dev/hast/* devices to appear
    for disk in ${resources}; do
      for I in $(jot 60); do
        [ -c "/dev/hast/${disk}" ] && break
      done
      sleep 0.5
    done
    if [ ! -c "/dev/hast/${disk}" ]; then
      logger -p $log -t $name "GEOM provider /dev/hast/${disk} did not appear."
      exit 1
    fi
    logger -p $log -t $name "Role for HAST resources ${resources} switched to primary."
    logger -p $log -t $name "Mounting disks."
    for disk in ${resources}; do
      mkdir -p /hast/${disk}
      # Perform disk mounting operations
    done
```bash
fsck -p -y -t ufs /dev/hast/${disk}
mount /dev/hast/${disk} /hast/${disk}
done
slave)
logger -p $log -t $name "Switching to secondary provider for ${resources}."

# Switch roles for the HAST resources
for disk in ${resources}; do
  if ! mount | grep -q "^/dev/hast/${disk} on ";
  then
    umount -f /hast/${disk}
  fi
  sleep $delay
  hastctl role secondary ${disk} 2>&1
  if [ $? -ne 0 ]; then
    logger -p $log -t $name "Unable to switch role to secondary for resource ${disk}."
    exit 1
  fi
  logger -p $log -t $name "Role switched to secondary for resource ${disk}."
  done
esac
```

In a nutshell, the script takes these actions when a node becomes master:

- Promotes the HAST pool to primary on the other node.
- Checks the file system under the HAST pool.
- Mounts the pool.

When a node becomes secondary:

- Unmounts the HAST pool.
- Degrades the HAST pool to secondary.

This is just an example script which serves as a proof of concept. It does not handle all the possible scenarios and can be extended or altered in any way, for example, to start or stop required services.

For this example, a standard UFS file system was used. To reduce the time needed for recovery, a journal-enabled UFS or ZFS file system can be used instead.

HAST should generally work without issues. However, as with any other software product, there may be times when it does not work as supposed. The sources of the problems may be different, but the rule of thumb is to ensure that the time is synchronized between the nodes of the cluster. When troubleshooting HAST, the debugging level of `hastd(8)` should be increased by starting `hastd` with `-d`. This argument may be specified multiple times to further increase the debugging level. Consider also using `-F`, which starts `hastd` in the foreground.

### 17.14.3.1. 自

**Split-brain 情況** 復

Split-brain occurs when the nodes of the cluster are unable to communicate with each other, and both are configured as primary. This is a dangerous condition because it allows both nodes to make incompatible changes to the data. This problem must be corrected manually by the system administrator.

The administrator must either decide which node has more important changes, or perform the merge manually. Then, let HAST perform full synchronization of the node which has the broken data. To do this, issue these commands on the node which needs to be resynchronized:

```
# hastctl role init test
# hastctl create test
# hastctl role secondary test
```
Chapter 18. GEOM: 模組化磁碟

18.1. 概述
在 FreeBSD 中, GEOM 允許對類別做存取與控制,例如: 主開機記錄 (Master Boot Record) 與 BSD 標籤, 透過提供者, 或在 /dev 中的磁碟裝置。透過支援各種 RAID 的配置, GEOM 透明的提供了對作業系統與作業系統工具的存取。

This chapter covers the use of disks under the GEOM framework in FreeBSD. This includes the major RAID control utilities which use the framework for configuration. This chapter is not a definitive guide to RAID configurations and only GEOM-supported RAID classifications are discussed.

读后這章,您將了解:
• What type of RAID support is available through GEOM.
• How to use the base utilities to configure, maintain, and manipulate the various RAID levels.
• How to mirror, stripe, encrypt, and remotely connect disk devices through GEOM.
• How to troubleshoot disks attached to the GEOM framework.

在開始閱讀這章之前,您需要:
• Understand how FreeBSD treats disk devices (儲存設備).
• 了解如何設定並安裝新的核心 (設定 FreeBSD 核心)。

18.2. RAID0 - 串連 (Striping)
串連會合併數個磁碟成單一個磁碟區 (Volume), 可以透過使用硬體 RAID 控制器來做到串連。GEOM 磁碟子系統提供了軟體支援的磁碟串連, 也就是所謂的 RAID0, 而不需要 RAID 磁碟控制器。

在 RAID0 串連中, 資料會被切割成數個資料區塊 (Block) 寫入到磁碟陣列中的每一個磁碟機。如下圖所示, 取代以往等候系統寫入 256k 到一個磁碟的同時, RAID0 可以同時寫入 64k 到磁碟陣列中的四個磁碟的每個磁碟, 這可提供優異的 I/O 效能, 若使用多個磁碟控制器可增加更多的效能。

在 RAID0 串連中的每個磁碟必須要相同大小, 因為 I/O 的請求是平行交錯讀取或寫入到多個磁碟的。
RAID0 does not provide any redundancy. This means that if a disk in the array fails, all data on that disk will be lost. If the data is important, please create backup strategies, regularly store backups to remote systems or devices.

The process for creating a software, GEOM-based RAID0 on a FreeBSD system using commodity disks is as follows. Once the stripe is created, refer to `gstripe(8)` for more information on how to control an existing stripe.

**Procedure: Creating a Stripe of Unformatted ATA Disks**

1. Load the `geom_stripe.ko` module:
   ```bash
   # kldload geom_stripe
   ```

2. Ensure that a suitable mount point exists. If this volume will become a root partition, then temporarily use another mount point such as `/mnt`.

3. Determine the device names for the disks which will be striped, and create the new stripe device. For example, to stripe two unused and unpartitioned ATA disks with device names `/dev/ad2` and `/dev/ad3`:
   ```bash
   # gstripe label -v st0 /dev/ad2 /dev/ad3
   ```
   Metadata value stored on /dev/ad2.
   Metadata value stored on /dev/ad3.
   Done.

4. Write a standard label, also known as a partition table, on the new volume and install the default bootstrap code:
   ```bash
   # bsdlabel -wB /dev/stripe/st0
   ```

5. This process should create two other devices in `/dev/stripe` in addition to `st0`. Those include `st0a` and `st0c`. At this point, a UFS file system can be created on `st0a` using `newfs`:
   ```bash
   # newfs -U /dev/stripe/st0a
   ```
   Many numbers will Glide across the screen, and after a few seconds, the process will be complete. The volume has been created and is ready to be mounted.

6. To manually mount the created disk stripe:
   ```bash
   # mount /dev/stripe/st0a /mnt
   ```

7. To mount this striped file system automatically during the boot process, place the volume information in `/etc/fstab`. In this example, a permanent mount point, named `stripe`, is created:
   ```bash
   # mkdir /stripe
   # echo "/dev/stripe/st0a /stripe ufs rw 2 2" > /etc/fstab
   ```
The `geom_stripe.ko` module must also be automatically loaded during system initialization, by adding a line to `/boot/loader.conf`:

```bash
# sysrc -f /boot/loader.conf geom_stripe_load=YES
```

### 18.3. RAID1 - Mirroring

RAID1 or mirroring is a storage technique that writes the same data to over one disk drive. Mirroring is often used to protect data in case of disk drive failure, as each drive in the mirror will have the exact same data. When a drive in the mirror fails, the mirror continues to operate, using the remaining drives to provide the data. The computer will continue to execute, waiting for managers to swap out the faulty hard drive, without interrupting the operation.

Two common situations are illustrated in these examples. The first creates a mirror out of two new drives and uses it as a replacement for an existing single drive. The second example creates a mirror on a single new drive, copies the old drive's data to it, then inserts the old drive into the mirror. While this procedure is slightly more complicated, it only requires one new drive.

Traditionally, the two drives in a mirror are identical in model and capacity, but `gmirror(8)` does not require that. Mirrors created with dissimilar drives will have a capacity equal to that of the smallest drive in the mirror. Extra space on larger drives will be unused. Drives inserted into the mirror later must have at least as much capacity as the smallest drive already in the mirror.

The mirroring procedures shown here are non-destructive, but as with any major disk operation, make a full backup first.

While `dump(8)` is used in these procedures to copy file systems, it does not work on file systems with soft updates journaling. See `tunefs(8)` for information on detecting and disabling soft updates journaling.

#### 18.3.1. Metadata

Many disk systems store metadata at the end of each disk. Old metadata should be erased before reusing the disk for a mirror. Most problems are caused by two particular types of leftover metadata: GPT partition tables and old metadata from a previous mirror. GPT metadata can be erased with `gpart(8)`. This example erases both primary and backup GPT partition tables from disk `ada8`:

```bash
# gpart destroy -F ada8
```

A disk can be removed from an active mirror and the metadata erased in one step using `gmirror(8)`. Here, the example disk `ada8` is removed from the active mirror `gm4`:

```bash
# gmirror remove gm4 ada8
```

If the mirror is not running, but old mirror metadata is still on the disk, use `gmirror clear` to remove it:

```bash
# gmirror clear
```
gmirror clears ada8

gmirror(8) stores one block of metadata at the end of the disk. Because GPT partition schemes also store metadata at the end of the disk, mirroring entire GPT disks with gmirror(8) is not recommended. MBR partitioning is used here because it only stores a partition table at the start of the disk and does not conflict with the mirror metadata.

18.3.2. 使用兩個新磁碟建立鏡像

In this example, FreeBSD has already been installed on a single disk, ada0. Two new disks, ada1 and ada2, have been connected to the system. A new mirror will be created on these two disks and used to replace the old single disk.

The geom_mirror.ko kernel module must either be built into the kernel or loaded at boot- or run-time. Manually load the kernel module now:

```
# gmirror load
```

Create the mirror with the two new drives:

```
# gmirror label -v gm0 /dev/ada1 /dev/ada2
```

gm0 is a user-chosen device name assigned to the new mirror. After the mirror has been started, this device name appears in /dev/mirror/.

MBR and bsdlabel partition tables can now be created on the mirror with gpart(8). This example uses a traditional file system layout, with partitions for /, swap, /var, /tmp, and /usr. A single / and a swap partition will also work.

Partitions on the mirror do not have to be the same size as those on the existing disk, but they must be large enough to hold all the data already present on ada0.

```
# gpart create -s MBR mirror/gm0
# gpart add -t freebsd -a 4k mirror/gm0
# gpart show mirror/gm0
```

```
63  156301423  mirror/gm0  MBR (74G)
63         63                    - free - (31k)
126  156301299                 1  freebsd (74G)
156301425         61                    - free - (30k)
```

```
# gpart create -s BSD mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-swap -a 4k -s 4g mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k -s 1g mirror/gm0s1
# gpart add -t freebsd-ufs -a 4k       mirror/gm0s1
# gpart show mirror/gm0s1
```

```
379
```
Make the mirror bootable by installing bootcode in the MBR and bsdlabel and setting the active slice:

```
# gpart bootcode -b /boot/mbr mirror/gm0
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Format the file systems on the new mirror, enabling soft-updates.

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
```

File systems from the original ada0 disk can now be copied onto the mirror with `dump(8)` and `restore(8)`.

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - / | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/tmp
# mount /dev/mirror/gm0s1f /mnt/usr
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /tmp | (cd /mnt/tmp && restore -rf -)
```

Edit `/mnt/etc/fstab` to point to the new mirror file systems:

```
<table>
<thead>
<tr>
<th>Device</th>
<th>Mountpoint</th>
<th>FStype</th>
<th>Options</th>
<th>Dump</th>
<th>Pass#</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/mirror/gm0s1a</td>
<td>/</td>
<td>ufs</td>
<td>rw</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>none</td>
<td></td>
<td>swap</td>
<td>sw</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>/dev/mirror/gm0s1d</td>
<td>/var</td>
<td>ufs</td>
<td>rw</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>none</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/mirror/gm0s1e</td>
<td>/tmp</td>
<td>ufs</td>
<td>rw</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>none</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/mirror/gm0s1f</td>
<td>/usr</td>
<td>ufs</td>
<td>rw</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>none</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
If the `geom_mirror.ko` kernel module has not been built into the kernel, `/mnt/boot/loader.conf` is edited to load the module at boot:

```
geom_mirror_load="YES"
```

Reboot the system to test the new mirror and verify that all data has been copied. The BIOS will see the mirror as two individual drives rather than a mirror. Because the drives are identical, it does not matter which is selected to boot.

See **疑難排解** if there are problems booting. Powering down and disconnecting the original `ada0` disk will allow it to be kept as an offline backup.

In use, the mirror will behave just like the original single drive.

### 18.3.3. 使用既存磁碟建立鏡像

In this example, FreeBSD has already been installed on a single disk, `ada0`. A new disk, `ada1`, has been connected to the system. A one-disk mirror will be created on the new disk, the existing system copied onto it, and then the old disk will be inserted into the mirror. This slightly complex procedure is required because `gmirror` needs to put a 512-byte block of metadata at the end of each disk, and the existing `ada0` has usually had all of its space already allocated.

Load the `geom_mirror.ko` kernel module:

```
# gmirror load
```

Check the media size of the original disk with `diskinfo`:

```
# diskinfo -v ada0 | head -n3
/dev/ada0
512
# sectorsize
1000204821504
# mediasize in bytes (931G)
```

Create a mirror on the new disk. To make certain that the mirror capacity is not any larger than the original `ada0` drive, `gnop(8)` is used to create a fake drive of the exact same size. This drive does not store any data, but is used only to limit the size of the mirror. When `gmirror(8)` creates the mirror, it will restrict the capacity to the size of `gzero.nop`, even if the new `ada1` drive has more space. Note that the `1000204821504` in the second line is equal to `ada0`’s media size as shown by `diskinfo` above.

```
# geom zero load
# gnop create -s 1000204821504 gzero
# gmirror label -v gm0 gzero.nop ada1
# gmirror forget gm0
```

Since `gzero.nop` does not store any data, the mirror does not see it as connected. The mirror is told to “forget” unconnected components, removing references to `gzero.nop`. The result is a mirror device containing only a single disk, `ada1`.

After creating `gm0`, view the partition table on `ada0`. This output is from a 1 TB drive. If there is some unallocated space at the end of the drive, the contents may be copied directly from `ada0` to the new mirror.
However, if the output shows that all of the space on the disk is allocated, as in the following listing,

```
# gpart show ada0
63  1953525105        ada0  MBR (931G)
63  1953525105           1  freebsd [active (931G)]
```

In this case, the partition table must be edited to reduce the capacity by one sector on mirror/gm0.
The procedure will be explained later.

In either case, partition tables on the primary disk should be first copied using `gpart backup` and `gpart restore`.

```
# gpart backup ada0 > table.ada0
# gpart backup ada0s1 > table.ada0s1
```

These commands create two files, `table.ada0` and `table.ada0s1`. This example is from a 1 TB drive:

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525105 [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
382
```

If no free space is shown at the end of the disk, the size of both the slice and the last partition must be reduced by one sector. Edit the two files, reducing the size of both the slice and last partition by one. These are the last numbers in each listing.

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525104 [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
382
```
If at least one sector was unallocated at the end of the disk, these two files can be used without modification.

Now restore the partition table into `mirror/gm0`:

```bash
# gpart restore mirror/gm0 < table.ada0
# gpart restore mirror/gm0s1 < table.ada0s1
```

Check the partition table with `gpart show`. This example has `gm0s1a` for `/`, `gm0s1d` for `/var`, `gm0s1e` for `/usr`, `gm0s1f` for `/data1`, and `gm0s1g` for `/data2`.

```bash
# gpart show mirror/gm0
```

```bash
=] 63  1953525104  mirror/gm0  MBR  
[931G]
63  1953525042           1  freebsd  [active]{931G}
1953525105          62              - free -  {31k}
# gpart show mirror/gm0s1
```

```bash
=] 0  1953525042  mirror/gm0s1  BSD  {931G}
0     2097152             1  freebsd-ufs  {1.0G}
2097152    16777216             2  freebsd-swap  {8.0G}
18874368    41943040             4  freebsd-ufs  {20G}
60817408    20971520             5  freebsd-ufs  {10G}
81788928   629145600             6  freebsd-ufs  {300G}
710934528  1242590514             7  freebsd-ufs  {592G}
1953525042          63                - free -  {31k}
```
Adjust /etc/fstab to use the new partitions on the mirror. Back up this file first by copying it to /etc/fstab.orig.

```
# cp /etc/fstab /etc/fstab.orig
```

Edit /etc/fstab, replacing /dev/ada0 with mirror/gm0.

```
Device        Mountpoint  FStype  Options Dump    Pass#
/dev/mirror/gm0s1a  /       ufs rw  1   1
/dev/mirror/gm0s1b  none        swap    sw  0   0
/dev/mirror/gm0s1d  /var        ufs rw  2   2
/dev/mirror/gm0s1e  /usr        ufs rw  2   2
/dev/mirror/gm0s1f  /data1      ufs rw  2   2
/dev/mirror/gm0s1g  /data2      ufs rw  2   2
```

If the geom_mirror.ko kernel module has not been built into the kernel, edit /boot/loader.conf to load it at boot:

```
geom_mirror_load="YES"
```

File systems from the original disk can now be copied onto the mirror with dump(8) and restore(8).

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - /    | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/usr
# mount /dev/mirror/gm0s1f /mnt/data1
# mount /dev/mirror/gm0s1g /mnt/data2
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /data1 | (cd /mnt/data1 && restore -rf -)
# dump -C16 -b64 -0aL -f - /data2 | (cd /mnt/data2 && restore -rf -)
```

Restart the system, booting from ada1. If everything is working, the system will boot from mirror/gm0, which now contains the same data as ada0 had previously. See 疑難排解 if there are problems booting.

At this point, the mirror still consists of only the single ada1 disk. After booting from mirror/gm0 successfully, the final step is inserting ada0 into the mirror.

When ada0 is inserted into the mirror, its former contents will be overwritten by...
data from the mirror. Make certain that mirror/gm0 has the same contents as ada0 before adding ada0 to the mirror. If the contents previously copied by dump(8) and restore(8) are not identical to what was on ada0, revert /etc/fstab to mount the file systems on ada0, reboot, and start the whole procedure again.

```
# gmirror insert gm0 ada0
GEOM_MIRROR: Device gm0: rebuilding provider ada0
Synchronization between the two disks will start immediately. Use gmirror status to view the progress.
```

```
# gmirror status
Name    Status  Components
mirror/gm0  DEGRADED  ada1 (ACTIVE)
ada0 (SYNCHRONIZING, 64%)
```

After a while, synchronization will finish.

```
GEOM_MIRROR: Device gm0: rebuilding provider ada0 finished.
# gmirror status
Name    Status  Components
mirror/gm0  COMPLETE  ada1 (ACTIVE)
ada0 (ACTIVE)
```

mirror/gm0 now consists of the two disks ada0 and ada1, and the contents are automatically synchronized with each other. In use, mirror/gm0 will behave just like the original single drive.

18.3.4. 疑難排解
If the system no longer boots, BIOS settings may have to be changed to boot from one of the new mirrored drives. Either mirror drive can be used for booting, as they contain identical data.

If the boot stops with this message, something is wrong with the mirror device:
```
Mounting from ufs:/dev/mirror/gm0s1a failed with error 19.
```

Loader variables:
```
vfs.root.mountfrom = ufs:/dev/mirror/gm0s1a
evf.root.mountfrom.options = rw
```

Manual root filesystem specification:
```
<fstype>:<device>[
(options)
Mount <device> using filesystem <fstype> and with the specified (optional) option list.
eg. ufs:/dev/da0s1a
```
zfs:tank
cd9660:/dev/acd0 ro

which is equivalent to: mount -t cd9660 -o ro /dev/acd0 /

?               List valid disk boot devices
Yield 1 second (for background tasks)

<empty line>    Abort manual input

mountroot>

Forgetting to load the geom_mirror.ko module in /boot/loader.conf can cause this problem. To fix it, boot from a FreeBSD installation media and choose Shell at the first prompt. Then load the mirror module and mount the mirror device:

```
# gmirror load
# mount /dev/mirror/gm0s1a /mnt
```

Edit /mnt/boot/loader.conf, adding a line to load the mirror module:

```
geom_mirror_load="YES"
```

Save the file and reboot.

Other problems that cause error 19 require more effort to fix. Although the system should boot from ada0, another prompt to select a shell will appear if /etc/fstab is incorrect. Enter ufs:/dev/ada0s1a at the boot loader prompt and press Enter. Undo the edits in /etc/fstab then mount the file systems from the original disk (ada0) instead of the mirror. Reboot the system and try the procedure again.

```
Enter full pathname of shell or RETURN for /bin/sh:
```

```
# cp /etc/fstab.orig /etc/fstab
# reboot
```

18.3.5.
自磁碟故障復原

The benefit of disk mirroring is that an individual disk can fail without causing the mirror to lose any data. In the above example, if ada0 fails, the mirror will continue to work, providing data from the remaining working drive, ada1.

To replace the failed drive, shut down the system and physically replace the failed drive with a new drive of equal or greater capacity. Manufacturers use somewhat arbitrary values when rating drives in gigabytes, and the only way to really be sure is to compare the total count of sectors shown by diskinfo -v. A drive with larger capacity than the mirror will work, although the extra space on the new drive will not be used.

After the computer is powered back up, the mirror will be running in a "degraded" mode with only one drive. The mirror is told to forget drives that are not currently connected:

```
# gmirror forget gm0
```

386
Any old metadata should be cleared from the replacement disk using the instructions in Metadata. Then the replacement disk, ada4 for this example, is inserted into the mirror:

```
# gmirror insert gm0 /dev/ada4
```

Resynchronization begins when the new drive is inserted into the mirror. This process of copying mirror data to a new drive can take a while. Performance of the mirror will be greatly reduced during the copy, so inserting new drives is best done when there is low demand on the computer. Progress can be monitored with `gmirror status`, which shows drives that are being synchronized and the percentage of completion. During resynchronization, the status will be `DEGRADED`, changing to `COMPLETE` when the process is finished.

18.4. RAID3 - 位元級串連與獨立奇偶校驗

RAID3 is a method used to combine several disk drives into a single volume with a dedicated parity disk. In a RAID3 system, data is split up into a number of bytes that are written across all the drives in the array except for one disk which acts as a dedicated parity disk. This means that disk reads from a RAID3 implementation access all disks in the array. Performance can be enhanced by using multiple disk controllers. The RAID3 array provides a fault tolerance of 1 drive, while providing a capacity of 1 - 1/n times the total capacity of all drives in the array, where n is the number of hard drives in the array. Such a configuration is mostly suitable for storing data of larger sizes such as multimedia files.

At least 3 physical hard drives are required to build a RAID3 array. Each disk must be of the same size, since I/O requests are interleaved to read or write to multiple disks in parallel. Also, due to the nature of RAID3, the number of drives must be equal to 3, 5, 9, 17, and so on, or 2^n + 1.

This section demonstrates how to create a software RAID3 on a FreeBSD system.

18.4.1. 建立 Dedicated RAID3 陣列

In FreeBSD, support for RAID3 is implemented by the `graid3(8)` GEOM class. Creating a dedicated RAID3 array on FreeBSD requires the following steps.

1. First, load the `geom_raid3.ko` kernel module by issuing one of the following commands:
   ```
   # graid3 load
   ```
   or:
   ```
   # kldload geom_raid3
   ```

2. Ensure that a suitable mount point exists. This command creates a new directory to use as the mount point:
   ```
   # mkdir /multimedia
   ```

3. Determine the device names for the disks which will be added to the array, and create the
   ```
   ```
new RAID3 device. The final device listed will act as the dedicated parity disk. This example uses three unpartitioned ATA drives: ada1 and ada2 for data, and ada3 for parity.

```bash
# graid3 label -v gr0 /dev/ada1 /dev/ada2 /dev/ada3
```

Metadata value stored on /dev/ada1.
Metadata value stored on /dev/ada2.
Metadata value stored on /dev/ada3.

Done.

4. Partition the newly created gr0 device and put a UFS file system on it:

```bash
# gpart create -s GPT /dev/raid3/gr0
# gpart add -t freebsd-ufs /dev/raid3/gr0
# newfs -j /dev/raid3/gr0p1
```

Many numbers will glide across the screen, and after a bit of time, the process will be complete. The volume has been created and is ready to be mounted:

```bash
# mount /dev/raid3/gr0p1 /multimedia/
```

The RAID3 array is now ready to use.

Additional configuration is needed to retain this setup across system reboots.

1. The geom_raid3.ko module must be loaded before the array can be mounted. To automatically load the kernel module during system initialization, add the following line to /boot/loader.conf:

   ```conf
   geom_raid3_load="YES"
   ```

2. The following volume information must be added to /etc/fstab in order to automatically mount the array's file system during the system boot process:

   ```bash
   /dev/raid3/gr0p1    /multimedia ufs rw  2   2
   ```

Some motherboards and expansion cards add some simple hardware, usually just a ROM, that allows the computer to boot from a RAID array. After booting, access to the RAID array is handled by software running on the computer's main processor. This "hardware-assisted software RAID" gives RAID arrays that are not dependent on any particular operating system, and which are functional even before an operating system is loaded.

Several levels of RAID are supported, depending on the hardware in use. See `graid(8)` for a complete list.
18.5.1. 建立陣列

Software RAID devices often have a menu that can be entered by pressing special keys when the computer is booting. The menu can be used to create and delete RAID arrays. `graid(8)` can also create arrays directly from the command line.

`graid label` is used to create a new array. The motherboard used for this example has an Intel software RAID chipset, so the Intel metadata format is specified. The new array is given a label of `gm0`, it is a mirror (RAID1), and uses drives `ada0` and `ada1`.

Some space on the drives will be overwritten when they are made into a new array. Back up existing data first!

```bash
# graid label Intel gm0 RAID1 ada0 ada1
GEOM_RAID: Intel-a29ea104: Array Intel-a29ea104 created.
GEOM_RAID: Intel-a29ea104: Disk ada0 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:0-ada0 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Array started.
GEOM_RAID: Intel-a29ea104: Volume gm0 state changed from STARTING to OPTIMAL.
Intel-a29ea104 created
GEOM_RAID: Intel-a29ea104: Provider raid/r0 for volume gm0 created.
```

A status check shows the new mirror is ready for use:

```bash
# graid status
Name   Status  Components
raid/r0  OPTIMAL  ada0 (ACTIVE (ACTIVE))
ada1 (ACTIVE (ACTIVE))
```

The array device appears in `/dev/raid/`. The first array is called `r0`. Additional arrays, if present, will be `r1`, `r2`, and so on.

The BIOS menu on some of these devices can create arrays with special characters in their names. To avoid problems with those special characters, arrays are given simple numbered names like `r0`. To show the actual labels, like `gm0` in the example above, use `sysctl(8)`: `# sysctl kern.geom.raid.name_format=1`.

18.5.2. 多磁碟區

Some software RAID devices support more than one volume on an array. Volumes work like partitions, allowing space on the physical drives to be split and used in different ways. For example, Intel software RAID devices support two volumes. This example creates a 40 G mirror for safely storing the operating system, followed by a 20 G RAID0 (stripe) volume for fast temporary storage:
Volumes appear as additional entries in /dev/raid. An array with two volumes will show r0 and r1.

See graid(8) for the number of volumes supported by different software RAID devices.

18.5.3. 轉換單一磁碟為鏡像

Under certain specific conditions, it is possible to convert an existing single drive to a graid(8) array without reformatting. To avoid data loss during the conversion, the existing drive must meet these minimum requirements:

• The drive must be partitioned with the MBR partitioning scheme. GPT or other partitioning schemes with metadata at the end of the drive will be overwritten and corrupted by the graid(8) metadata.
• There must be enough unpartitioned and unused space at the end of the drive to hold the graid(8) metadata. This metadata varies in size, but the largest occupies 64 M, so at least that much free space is recommended.

If the drive meets these requirements, start by making a full backup. Then create a single-drive mirror with that drive:

# graid label Intel gm0 RAID1 ada0 NONE

graid(8) metadata was written to the end of the drive in the unused space. A second drive can now be inserted into the mirror:

# graid insert raid/r0 ada1

Data from the original drive will immediately begin to be copied to the second drive. The mirror will operate in degraded status until the copy is complete.

18.5.4. 插入新磁碟到陣列

Drives can be inserted into an array as replacements for drives that have failed or are missing. If there are no failed or missing drives, the new drive becomes a spare. For example, inserting a new drive into a working two-drive mirror results in a two-drive mirror with one spare drive, not a three-drive mirror.

In the example mirror array, data immediately begins to be copied to the newly-inserted drive. Any existing information on the new drive will be overwritten.

# graid insert raid/r0 ada1

GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NONE to NEW.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NEW to REBUILD.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 rebuild start at 0.
Individual drives can be permanently removed from a array and their metadata erased:

```
# graid remove raid/r0 ada1
```

GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from ACTIVE to OFFLINE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-[unknown] state changed from ACTIVE to NONE.
GEOM_RAID: Intel-a29ea104: Volume gm0 state changed from OPTIMAL to DEGRADED.

An array can be stopped without removing metadata from the drives. The array will be restarted when the system is booted.

```
# graid stop raid/r0
```

Array status can be checked at any time. After a drive was added to the mirror in the example above, data is being copied from the original drive to the new drive:

```
# graid status
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>raid/r0</td>
<td>DEGRADED</td>
<td>ada0 (ACTIVE (ACTIVE))</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ada1 (ACTIVE (REBUILD 28%))</td>
</tr>
</tbody>
</table>

Some types of arrays, like RAID0 or CONCAT, may not be shown in the status report if disks have failed. To see these partially-failed arrays, add `-ga`:

```
# graid status -ga
```

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel-e2d07d9a</td>
<td>BROKEN</td>
<td>ada6 (ACTIVE (ACTIVE))</td>
</tr>
</tbody>
</table>

Arrays are destroyed by deleting all of the volumes from them. When the last volume present is deleted, the array is stopped and metadata is removed from the drives:

```
# graid delete raid/r0
```

Drives may unexpectedly contain `graid(8)` metadata, either from previous use or manufacturer testing. `graid(8)` will detect these drives and create an array, interfering with access to the individual drive. To remove the unwanted metadata:
1. Boot the system. At the boot menu, select 2 for the loader prompt. Enter:

```
OK
set kern.geom.raid.enable = 0
OK boot
```

The system will boot with `graid(8)` disabled.

2. Back up all data on the affected drive.

3. As a workaround, `graid(8)` array detection can be disabled by adding `kern.geom.raid.enable=0` to `/boot/loader.conf`.

To permanently remove the `graid(8)` metadata from the affected drive, boot a FreeBSD installation CD-ROM or memory stick, and select Shell. Use status to find the name of the array, typically `raid/r0`:

```
# graid status
Name   Status  Components
raid/r0  OPTIMAL  ada0
( ACTIVE ( ACTIVE )
ada1
( ACTIVE ( ACTIVE )
```

Delete the volume by name:

```
# graid delete raid/r0
```

If there is more than one volume shown, repeat the process for each volume. After the last array has been deleted, the volume will be destroyed.

Reboot and verify data, restoring from backup if necessary. After the metadata has been removed, the `kern.geom.raid.enable=0` entry in `/boot/loader.conf` can also be removed.

18.6. GEOM Gate Network

GEOM provides a simple mechanism for providing remote access to devices such as disks, CDs, and file systems through the use of the GEOM Gate network daemon, ggated. The system with the device runs the server daemon which handles requests made by clients using ggatec. The devices should not contain any sensitive data as the connection between the client and the server is not encrypted.

Similar to NFS, which is discussed in 網路檔案系統 (NFS), ggated is configured using an exports file. This file specifies which systems are permitted to access the exported resources and what level of access they are offered. For example, to give the client 192.168.1.5 read and write access to the fourth slice on the first SCSI disk, create `/etc/gg.exports` with this line:

```
192.168.1.5 RW /dev/da0s4d
```
Before exporting the device, ensure it is not currently mounted. Then, start `ggated`:

```
# ggated
```

Several options are available for specifying an alternate listening port or changing the default location of the exports file. Refer to `ggated(8)` for details.

To access the exported device on the client machine, first use `ggatec` to specify the IP address of the server and the device name of the exported device. If successful, this command will display a `ggate` device name to mount. Mount that specified device name on a free mount point. This example connects to the `/dev/da0s4d` partition on `192.168.1.1`, then mounts `/dev/ggate0` on `/mnt`:

```
# ggatec create -o rw 192.168.1.1 /dev/da0s4d
# mount /dev/ggate0 /mnt
```

The device on the server may now be accessed through `/mnt` on the client. For more details about `ggatec` and a few usage examples, refer to `ggatec(8)`.

The mount will fail if the device is currently mounted on either the server or any other client on the network. If simultaneous access is needed to network resources, use NFS instead.

When the device is no longer needed, unmount it with `umount` so that the resource is available to other clients.

18.7. 磁碟装置签

During system initialization, the FreeBSD kernel creates device nodes as devices are found. This method of probing for devices raises some issues. For instance, what if a new disk device is added via USB? It is likely that a flash device may be handed the device name of `da0` and the original `da0` shifted to `da1`. This will cause issues mounting file systems if they are listed in `/etc/fstab` which may also prevent the system from booting.

One solution is to chain SCSI devices in order so a new device added to the SCSI card will be issued unused device numbers. But what about USB devices which may replace the primary SCSI disk? This happens because USB devices are usually probed before the SCSI card. One solution is to only insert these devices after the system has been booted. Another method is to use only a single ATA drive and never list the SCSI devices in `/etc/fstab`.

A better solution is to use `glabel` to label the disk devices and use the labels in `/etc/fstab`. Because `glabel` stores the label in the last sector of a given provider, the label will remain persistent across reboots. By using this label as a device, the file system may always be mounted regardless of what device node it is accessed through.

`glabel` can create both transient and permanent labels. Only permanent labels are consistent across reboots. Refer to `glabel(8)` for more information on the differences between labels.

18.7.1. 标签类型与例子

Permanent labels can be a generic or a file system label. Permanent file system labels can be created with `tunefs(8)` or `newfs(8)`. These types of labels are created in a sub-directory of `/dev`, and will be named according to the file system type. For example, UFS2 file system labels will be created in `/dev/ufs`. Generic permanent labels can be created with `glabel label`. These are not file...
Temporary labels are destroyed at the next reboot. These labels are created in /dev/label and are suited to experimentation. A temporary label can be created using `glabel create`.

To create a permanent label for a UFS2 file system without destroying any data, issue the following command:

```
# tunefs -L home /dev/da3
```

A label should now exist in /dev/ufs which may be added to /etc/fstab:

```
/dev/ufs/home       /home            ufs     rw              2      2
```

The file system must not be mounted while attempting to run `tunefs`.

Now the file system may be mounted:

```
# mount /home
```

From this point on, so long as the `geom_label.ko` kernel module is loaded at boot with `/boot/loader.conf` or the `GEOM_LABEL` kernel option is present, the device node may change without any ill effect on the system.

File systems may also be created with a default label by using the `-L` flag with `newfs`. Refer to `newfs(8)` for more information.

The following command can be used to destroy the label:

```
# glabel destroy home
```

The following example shows how to label the partitions of a boot disk.

By permanently labeling the partitions on the boot disk, the system should be able to continue to boot normally, even if the disk is moved to another controller or transferred to a different system. For this example, it is assumed that a single ATA disk is used, which is currently recognized by the system as `ad0`. It is also assumed that the standard FreeBSD partition scheme is used, with `/`, `/var`, `/usr` and `/tmp`, as well as a swap partition.

Reboot the system, and at the `loader(8)` prompt, press 4 to boot into single user mode. Then enter the following commands:

```
# glabel label rootfs /dev/ad0s1a
GEOM_LABEL: Label for provider /dev/ad0s1a is label/rootfs

# glabel label var /dev/ad0s1d
GEOM_LABEL: Label for provider /dev/ad0s1d is label/var

# glabel label usr /dev/ad0s1f
GEOM_LABEL: Label for provider /dev/ad0s1f is label/usr
```
The system will continue with multi-user boot. After the boot completes, edit /etc/fstab and replace the conventional device names, with their respective labels. The final /etc/fstab will look like this:

| Device                | Mountpoint | FStype | Options   | Dump | Pass#
|-----------------------|------------|--------|-----------|------|--------
| /dev/label/swap       | none       | swap   | sw        | 0    | 0      |
| /dev/label/rootfs     | /          | ufs    | rw        | 1    | 1      |
| /dev/label/tmp        | /tmp       | ufs    | rw        | 2    | 2      |
| /dev/label/usr        | /usr       | ufs    | rw        | 2    | 2      |
| /dev/label/var        | /var       | ufs    | rw        | 2    | 2      |

The system can now be rebooted. If everything went well, it will come up normally and mount will show:

```
# mount
/dev/label/rootfs on / (ufs, local)
devfs on /dev (devfs, local)
/dev/label/tmp on /tmp (ufs, local, soft-updates)
/dev/label/usr on /usr (ufs, local, soft-updates)
/dev/label/var on /var (ufs, local, soft-updates)
```

The glabel(8) class supports a label type for UFS file systems, based on the unique file system id, ufsid. These labels may be found in /dev/ufsid and are created automatically during system startup. It is possible to use ufsid labels to mount partitions using /etc/fstab. Use glabel status to receive a list of file systems and their corresponding ufsid labels:

```
% glabel status
Name  Status  Components
ufsid/486b6fc38d330916     N/A  ad4s1d
ufsid/486b6fc16926168e     N/A  ad4s1f
```

In the above example, ad4s1d represents /var, while ad4s1f represents /usr. Using the ufsid values shown, these partitions may now be mounted with the following entries in /etc/fstab:

```
/dev/ufsid/486b6fc38d330916        /var        ufs        rw        2      2
/dev/ufsid/486b6fc16926168e        /usr        ufs        rw        2      2
```

Any partitions with ufsid labels can be mounted in this way, eliminating the need to manually create permanent labels, while still enjoying the benefits of device name independent mounting.
Support for journals on UFS file systems is available on FreeBSD. The implementation is provided through the GEOM subsystem and is configured using `gjournal`. Unlike other file system journaling implementations, the `gjournal` method is block based and not implemented as part of the file system. It is a GEOM extension.

Journaling stores a log of file system transactions, such as changes that make up a complete disk write operation, before meta-data and file writes are committed to the disk. This transaction log can later be replayed to redo file system transactions, preventing file system inconsistencies. This method provides another mechanism to protect against data loss and inconsistencies of the file system. Unlike Soft Updates, which tracks and enforces meta-data updates, and snapshots, which create an image of the file system, a log is stored in disk space specifically for this task. For better performance, the journal may be stored on another disk. In this configuration, the journal provider or storage device should be listed after the device to enable journaling on.

The GENERIC kernel provides support for `gjournal`. To automatically load the `geom_journal.ko` kernel module at boot time, add the following line to `/boot/loader.conf`:

```
geom_journal_load="YES"
```

If a custom kernel is used, ensure the following line is in the kernel configuration file:

```
options GEOM_JOURNAL
```

Once the module is loaded, a journal can be created on a new file system using the following steps. In this example, `da4` is a new SCSI disk:

```
# gjournal load
# gjournal label /dev/da4
```

This will load the module and create a `/dev/da4.journal` device node on `/dev/da4`.

A UFS file system may now be created on the journaled device, then mounted on an existing mount point:

```
# newfs -O 2 -J /dev/da4.journal
# mount /dev/da4.journal /mnt
```

In the case of several slices, a journal will be created for each individual slice. For instance, if `ad4s1` and `ad4s2` are both slices, then `gjournal` will create `ad4s1.journal` and `ad4s2.journal`.

Journaling may also be enabled on current file systems by using `tunefs`. However, always make a backup before attempting to alter an existing file system. In most cases, `gjournal` will fail if it is unable to create the journal, but this does not protect against data loss incurred as a result of misusing `tunefs`. Refer to `gjournal(8)` and `tunefs(8)` for more information about these commands.

It is possible to journal the boot disk of a FreeBSD system. Refer to the article “Implementing UFS Journaling on a Desktop PC” for detailed instructions.
Chapter 19. ZFS

ZFS (Zettabyte File System or Zettabyte File System) is designed to overcome many of the problems found in previous designs. Originally developed by Sun, it later moved to OpenZFS.

ZFS's design goals are:

- Data Integrity: All data will have a checksum when written, and then combined and written. When reading data back, it will calculate the checksum again. If the checksum does not match the initial write, the error can be detected using data redundancy (Data Redundancy).

- Storage Pool: Real storage devices are first added to a shared storage pool (Pool). The storage pool is accessible by all file systems, and new storage can be added to increase the storage space.

- Efficiency: Provides multiple caching mechanisms to improve performance. The advanced, memory-based caching mechanism uses ARC. The second layer, which is disk-based caching, uses L2ARC, disk-based synchronized writing can use ZIL.

Complete functions and terminology are detailed in the ZFS features and terminology section.

19.1. What Makes ZFS Different

ZFS is different from any previous file system because it is not just a file system, but ZFS's unique advantages come from combining file system (Volume Manager) and file system (Volume Manager) roles, allowing the file system to observe the lower layer's structure changes.

Traditional systems on a single disk could only build a file system, so if there were two disks, they would need to build two separate file systems. To solve this problem, traditional systems would use RAID to make a virtual disk on the actual disk, which is composed of multiple physical disks, and the operating system could then place the file system on this virtual disk, just as it would use GEOM software RAID solution, with UFS file system placed on RAID Transform.

ZFS combined Volume Manager and file system to solve this problem, allowing multiple file systems to share a storage pool (Pool).

ZFS has several different properties that can be used on different file systems, compared to a single file system, when creating multiple file systems and datasets (Dataset) there are many benefits.

19.2. Quick Start Guide

Here is a boot mechanism that allows FreeBSD to load ZFS storage pools at system initialization. To activate this feature, add the following line to /etc/rc.conf:

```
zfs_enable="YES"
```

Then start the service:

```
# service zfs start
```

This section will assume three SCSI disks, with names da0, da1, and da2.

19.2.1. Single Disk Storage Pool

To create a simple, no redundancy storage pool:

```bash
397
```
要检视这个新的储存池，可以查看 df 的输出结果:

```
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235230  1628718    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032846 48737598     2%    /usr
example      17547136       0 17547136     0%    /example
```

这个输出结果说明 example 储存池已建立且被挂载，现在已经可以作为档案系统存取，可以在上面建立档案且使用者可以浏览:

```
# cd /example
# ls
# touch testfile
# ls -al
total 4
drwxr-xr-x   2 root  wheel    3 Aug 29 23:15 .
drwxr-xr-x  21 root  wheel  512 Aug 29 23:12 ..
-rw-r--r-- 1 root  wheel    0 Aug 29 23:15 testfile
```

但是，这个储存池并未运用到任何 ZFS 功能，若要在这个储存池上建立一个有开启压缩功能的资料集:

```
# zfs create example/compressed
# zfs set compression=gzip example/compressed
```

example/compressed 资料集现在是一个 ZFS 压缩的档案系统，可以尝试複製较大的档案到 /example/compressed。

压缩功能也可以使用以下指令关闭:

```
# zfs set compression=off example/compressed
```

要卸载档案系统，使用 zfs umount 然后再次使用 df 确认:

```
# zfs umount example/compressed
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235232  1628716    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032864 48737580     2%    /usr
```
要重新掛載檔案系統以便再次使用,使用 `zfs mount` 然後以 `df` 檢查:

```bash
# zfs mount example/compressed
# df
```

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Avail</th>
<th>Capacity</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/ad0s1a</td>
<td>2026030</td>
<td>235234</td>
<td>1628714</td>
<td>13%</td>
<td>/</td>
</tr>
<tr>
<td>devfs</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>100%</td>
<td>/dev</td>
</tr>
<tr>
<td>/dev/ad0s1d</td>
<td>54098308</td>
<td>1032864</td>
<td>48737580</td>
<td>2%</td>
<td>/usr</td>
</tr>
<tr>
<td>example</td>
<td>17547008</td>
<td>0</td>
<td>17547008</td>
<td>0%</td>
<td>/example</td>
</tr>
<tr>
<td>example/compressed</td>
<td>17547008</td>
<td>0</td>
<td>17547008</td>
<td>0%</td>
<td>/example/compressed</td>
</tr>
<tr>
<td>example/data</td>
<td>17547008</td>
<td>0</td>
<td>17547008</td>
<td>0%</td>
<td>/example/data</td>
</tr>
</tbody>
</table>

儲存池與檔案系統也可以從 `mount` 的結果查詢到:

```bash
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
example on /example (zfs, local)
example/compressed on /example/compressed (zfs, local)
```

在建立之後, ZFS 的資料集可如同其他檔案系統一般使用,且有多額外功能可在每個資料集上設定。例如, 建立一個預計存放重要資料的新檔案系統 `data`, 要設定每個資料區塊 (Data block) 要保留兩份備份:

```bash
# zfs create example/data
# zfs set copies=2 example/data
```

現在, 可以使用 `df` 指令來查看資料與空間的使用率:

```bash
# df
```

<table>
<thead>
<tr>
<th>Filesystem</th>
<th>1K-blocks</th>
<th>Used</th>
<th>Avail</th>
<th>Capacity</th>
<th>Mounted on</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev/ad0s1a</td>
<td>2026030</td>
<td>235234</td>
<td>1628714</td>
<td>13%</td>
<td>/</td>
</tr>
<tr>
<td>devfs</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>100%</td>
<td>/dev</td>
</tr>
<tr>
<td>/dev/ad0s1d</td>
<td>54098308</td>
<td>1032864</td>
<td>48737580</td>
<td>2%</td>
<td>/usr</td>
</tr>
<tr>
<td>example</td>
<td>17547008</td>
<td>0</td>
<td>17547008</td>
<td>0%</td>
<td>/example</td>
</tr>
<tr>
<td>example/compressed</td>
<td>17547008</td>
<td>0</td>
<td>17547008</td>
<td>0%</td>
<td>/example/compressed</td>
</tr>
<tr>
<td>example/data</td>
<td>17547008</td>
<td>0</td>
<td>17547008</td>
<td>0%</td>
<td>/example/data</td>
</tr>
</tbody>
</table>

注意, 從這個可以發現每個在儲存池上的檔案系統都擁有相同的可用空間, 這為什麼要在這些範例使用 `df` 的原因, 為了要顯示檔案系統只會用它們所需要使用到的空間, 且均取自同一個儲存池。ZFS 淘汰了磁碟區 (Volume) 與分割區 (Partition) 的概念, 且允許多個檔案系統共用相同的儲存池。不需要使用時可摧毀檔案系統後再摧毀儲存池:
zfs destroy example/compressed
zfs destroy example/data
zpool destroy example

19.2.2. RAID-Z
磁碟損壞時，要避免資料因磁碟故障造成遺失，便是使用RAID。
ZFS在它的儲存池設計中支援了這項功能。
RAID-Z儲存池需要使用三個或多的磁碟，但可以提供比鏡像(Mirror)儲存池更多的可用空間。

這個例子會建立一個RAID-Z儲存池，並指定要加入這個儲存池的磁碟:
```
# zpool create storage raidz da0 da1 da2
```

Sun™建議用在RAID-Z設定的裝置數在三到九個之間。
若需要由10個或多的磁碟組成單一儲存池的環境，可考慮分成較小的RAID-Z群組。
若只有兩個可用的磁碟且需要做備援(Redundancy)，可考慮使用ZFS鏡像(Mirror)。
請參考zpool(8)取得更多詳細資訊。

先前的例子已經建立了storage儲存池(zpool)，現在這個例子會在該儲存池中建立一個新的檔案系統，名稱為home:
```
# zfs create storage/home
```
可以設定啟動壓縮及保留目錄及檔案額外備份的功能:
```
# zfs set copies=2 storage/home
# zfs set compression=gzip storage/home
```
要讓這個空間作為使用者的新家目錄位置，需複製使用者資料到這個目錄並建立適合的符號連結(Symbolic link):
```
# cp -rp /home/* /storage/home
# rm -rf /home /usr/home
# ln -s /storage/home /home
# ln -s /storage/home /usr/home
```
現在使用者的資料會儲存在新建立的/storage/home，可以加入新使用者並登入該使用者來測試。

試著建立檔案系統快照(Snapshot)，稍後可用來還原(Rollback):
```
# zfs snapshot storage/home@08-30-08
```
快照只可以使用整個檔案系統製作，無法使用各別目錄或檔案。

字元用來區隔檔案系統名稱(File system)或磁碟區(Volume)名稱，若有重要的目錄意外被刪除，檔案系統可以備份然後還原到先前目錄還存在時的快照(Snapshot)。
要列出所有可用的快照，可在档案系统的 .zfs/snapshot 目录下执行 ls，举例来说，要查看先前已做的快照：

```
# ls /storage/home/.zfs/snapshot
```

也可以写一个 script 来对使用者资料做例行性的快照，但随著时间快照可能消耗大量的磁碟空间。先前的快照可以使用指令移除：

```
# zfs destroy storage/home@08-30-08
```

在测试之后，便可以将 /storage/home 成为真正的 /home 使用此指令:

```
# zfs set mountpoint=/home storage/home
```

执行 df mount 来确认系统现在是否以把档案系统做为真正的 /home:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
storage on /storage (zfs, local)
storage/home on /home (zfs, local)
```

```
# df
Filesystem   1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a    2026030  235240  1628708    13%    /
devfs                1       1        0   100%    /dev
/dev/ad0s1d   54098308 1032826 48737618     2%    /usr
storage       26320512       0 26320512     0%    /storage
storage/home  26320512       0 26320512     0%    /home
```

这个动作完成 RAID-Z 最后的设定，有关建立的档案系统每日状态更新可以做为 periodic(8) 的一部分在每晚上执行。加入此行到 /etc/periodic.conf:

```
daily_status_zfs_enable="YES"
```

19.2.3. 复原 RAID-Z

每个多体 RAID 都有监控其状态 (state) 的方式，而 RAID-Z 装置的状态可以使用这个指令来查看：

```
# zpool status -x
```

If all storage pools are online and healthy, information will be displayed:

```
all pools are healthy
```

If there is a problem, the disk may show an offline (Offline) status, at which time the storage pool's status will be:

```
pool: storage
state: DEGRADED
status: One or more devices has been taken offline by the administrator.
Sufficient replicas exist for the pool to continue functioning in a degraded state.
action: Online the device using `zpool online` or replace the device with `zpool replace`.
```

```
scrub: none requested
config:
NAME        STATE     READ WRITE CKSUM
storage     DEGRADED     0     0     0
raidz1    DEGRADED     0     0     0
da0     ONLINE       0     0     0
da1     OFFLINE      0     0     0
da2     ONLINE       0     0     0
```

This indicates that the device has been taken offline by the administrator:

```
# zpool offline storage da1
```

Now the system can be powered off and replaced with da1. When the system recovers online, it can replace the storage pool with the failed disk:

```
# zpool replace storage da1
```

to here, you can check the status again, this time without the `-x` parameter to display all storage pools:

```
# zpool status storage
```

```
pool: storage
state: ONLINE
scrub: resilver completed with 0 errors on Sat Aug 30 19:44:11 2008
```

```
config:
NAME        STATE     READ WRITE CKSUM
storage     ONLINE       0     0     0
raidz1      ONLINE       0     0     0
da0      ONLINE       0     0     0
da1      OFFLINE      0     0     0
da2      ONLINE       0     0     0
```

```
errors: No known data errors
```
在這個例子中，所有的磁碟均已正常運作。

19.2.4. 資料檢驗

ZFS 使用校驗碼 (Checksum) 來檢驗資料的完整性 (Integrity)，會在建立檔案系統時便自動開啟。

校驗碼可以關閉，但不建議！校驗碼只會使用非常少的儲存空間來確保資料的完整性。若關閉校驗碼會使許多 ZFS 功能無法正常運作，且關閉校驗碼對並不會明顯的改善效能。

檢驗校驗碼這個動作即所謂的清潔 (Scrub)，可以使用以下指令來檢驗 storage 儲存池的資料完整性:

```
# zpool scrub storage
```

清潔所需要的時間依儲存的資料量而定，較大的資料量相對會需要花費較長的時間來檢驗。清潔會對 I/O 有非常密集的操作且一次只能進行一個清潔動作。在清潔完成之後，可以使用 `status` 來查看狀態:

```
# zpool status storage
```

pool: storage
state: ONLINE
scrub: scrub completed with 0 errors on Sat Jan 26 19:57:37 2013
cfg:
NAME        STATE     READ WRITE CKSUM
storage     ONLINE       0     0     0
raidz1    ONLINE       0     0     0
da0     ONLINE       0     0     0
da1     ONLINE       0     0     0
da2     ONLINE       0     0     0
errors: No known data errors

查詢結果會顯示上次完成清潔的時間來協助追蹤是否要再做清潔。

定期清潔可以協助保護資料不會默默損壞且確保儲存池的完整性。

請參考 `zfs(8)` 及 `zpool(8)` 來取得其他 ZFS 選項。
19.3.1. 建立與摧毀儲存池

建立 ZFS 儲存池 (zpool) 要做幾個涉及長遠規畫的決定，因為建立儲存池之後便無法再更改儲存池的結構。最重要的決定是要使用那一種型態的 vdev 將實體磁碟設為同一群組。請參考 vdev 型態的清單來取得有關可用選項的詳細資訊。大部份的 vdev 型態不允許在建立儲存池後再加入額外的磁碟，鏡像 (Mirror) 是可以允許加入額外的磁碟到 vdev 的其中一個外，另一個則是串連 (Stripe)，可以加入額外的磁碟到 vdev 來升級為鏡像。

雖然可以加入額外的 vdev 來擴充儲存池，但儲存池的配置在建立之後便無法更改，若要更改，則必須先備份資料，把儲存池摧毀後再重新建立。

建立一個簡單的鏡像儲存池:

```
# zpool create mypool mirror /dev/ada1 /dev/ada2
# zpool status
```

```
pool: mypool
state: ONLINE
scan: none requested
config:
  NAME        STATE     READ WRITE CKSUM
  mypool      ONLINE       0     0     0
  mirror-0  ONLINE       0     0     0
  ada1    ONLINE       0     0     0
  ada2    ONLINE       0     0     0
errors: No known data errors
```

可以一次建立數個 vdev，磁碟群組間使用 vdev 型態關鍵字來區隔，在這個例子使用 mirror:

```
# zpool create mypool mirror /dev/ada1 /dev/ada2 mirror /dev/ada3 /dev/ada4
```

```
pool: mypool
state: ONLINE
scan: none requested
config:
  NAME        STATE     READ WRITE CKSUM
  mypool      ONLINE       0     0     0
  mirror-0  ONLINE       0     0     0
  ada1    ONLINE       0     0     0
  ada2    ONLINE       0     0     0
  ada3    ONLINE       0     0     0
  ada4    ONLINE       0     0     0
errors: No known data errors
```
儲存池也可以不使用整個磁碟而改使用分割區（Partition）來建立。把 ZFS 放到不同的分割區可讓同一個磁碟有其他的分割區可做其他用途，尤其是有 Bootcode 與檔案系統要用來開機的分割區，這讓磁碟可以用來開機也同樣可以做為儲存池的一部份。在 FreeBSD 用分割區來取代整個磁碟並不會對效能有影響。使用分割區也讓管理者可以對磁碟容量做少算的預備，使用比完成容量少的容量，未來若要替換的磁碟號稱與原磁碟相同，但實際上卻比較小時，也可符合這個較小的分割區容量，以使用替換的磁碟。

使用分割區建立一個 RAID-Z2 儲存池:

```
# zpool status
pool: mypool
state: ONLINE
scan: none requested
config:
  NAME        STATE     READ WRITE CKSUM
  mypool      ONLINE       0     0     0
  raidz2-0   ONLINE       0     0     0
  ada0p3      ONLINE       0     0     0
  ada1p3      ONLINE       0     0     0
  ada2p3      ONLINE       0     0     0
  ada3p3      ONLINE       0     0     0
  ada4p3      ONLINE       0     0     0
  ada5p3      ONLINE       0     0     0
errors: No known data errors
```

不需使用的儲存池可以摧毀，來讓磁碟可以再使用。摧毀一個儲存池要先卸載所有該儲存池的資料集。若資料集在使用中，卸載的操作會失敗且儲存池不會被摧毀。儲存池的摧毀可以使用 `-f` 來強制執行，但這可能造成那些有開啓這些資料集之中檔案的應用程式無法辨識的行為。
也可用來加入額外的磁碟到鏡像群組，來增加備援與讀取效
率。若使用的磁碟已分割區，可以複製該磁碟的分割區配置到另
一個，使用 `gpart backup` 與 `gpart restore` 可讓這件事變的很簡單。

加入 `ada1p3` 來升級單一磁碟串連 `(stripe) vdev ada0p3 採用鏡像型態 `(mirror):

```
# zpool status
pool: mypool
state: ONLINE
scan: none requested
cfg:
NAME        STATE     READ WRITE CKSUM
mypool      ONLINE       0     0     0
ada0p3    ONLINE       0     0     0
errors: No known data errors
# zpool attach mypool ada0p3 ada1p3
Make sure to wait until resilver is done before rebooting.
If you boot from pool 'mypool', you may need to update boot code on newly attached disk 'ada1p3'.
Assuming you use GPT partitioning and 'da0' is your new boot disk you may use the following command:
gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
bootcode written to ada1
# zpool status
pool: mypool
state: ONLINE
status: One or more devices is currently being resilvered. The pool will continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scan: resilver in progress since Fri May 30 08:19:19 2014
527M scanned out of 781M at 47.9M/s, 0h0m to go
527M resilvered, 67.53% done
cfg:
NAME        STATE     READ WRITE CKSUM
mypool      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
ada0p3  ONLINE       0     0     0
```
若不想选择加入磁碟到既有的vdev，对RAID-Z来说，可以选择另一种方式，便是加入另一个vdev到储存池。额外的vdev可以提供更高的效能，分散写入资料到vdev之间，每个vdev会负责自己的备份。也可以混合使用不同的vdev型态，但并不建议，例如混合使用mirror与RAID-Z，加入一个无备份的vdev到一个含有mirror或RAID-Z vdev的储存池会让资料损坏的风险扩大整著储存池，因为会分散写入资料，若在无备份的磁碟上发生故障的结果便是遗失大半写到储存池的资料区块。

在每个vdev间的资料是串连的，例如，有两个mirror vdev，便跟RAID 10一样在两个mirror间分散写入资料，且会做空间的分配，因此vdev会在同时达到全满100%的用量。若vdev间的可用空间量不同则会影响到效能，因为资料量会不成比例的写入到使用量较少的vdev。

当连接额外的装置到一个可以开机的储存池，要记得更新Bootcode。

连接第二个mirror群组(ada2p3及ada3p3)到既有的mirror:

```
# zpool status
pool: mypool
state: ONLINE
scan: resilvered 781M in 0h0m with 0 errors on Fri May 30 08:19:35 2014
cfg:
<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mirror-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada1p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
errors: No known data errors
# zpool add mypool mirror ada2p3 ada3p3
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada2
```
bootcode written to ada2
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada3
bootcode written to ada3
# zpool status

pool: mypool
state: ONLINE
scan: scrub repaired 0
in 0h0m with 0 errors on Fri May 30 08:29:51 2014

config:
NAME        STATE     READ WRITE CKSUM
mypool      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
ada0p3  ONLINE       0     0     0
ada1p3  ONLINE       0     0     0
ada2p3  ONLINE       0     0     0
mirror-1  ONLINE       0     0     0
ada3p3  ONLINE       0     0     0
errors: No known data errors

現在已無法從儲存池上移除vdev, 磁碟只能夠在有足夠備援空間的情況下從mirror移除, 若在mirror群組中只剩下一個磁碟, 便會取消mirror然後還原為stripe, 若剩下的那個磁碟故障, 便會影響到整個儲存池。

從一個三方mirror群組移除一個磁碟:
# zpool status
pool: mypool
state: ONLINE
scan: scrub repaired 0
in 0h0m with 0 errors on Fri May 30 08:29:51 2014

config:
NAME        STATE     READ WRITE CKSUM
mypool      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
ada0p3  ONLINE       0     0     0
ada1p3  ONLINE       0     0     0
ada2p3  ONLINE       0     0     0
mirror-1  ONLINE       0     0     0
ada3p3  ONLINE       0     0     0
errors: No known data errors
# zpool detach mypool ada2p3
# zpool status
# zpool status

pool: mypool
state: ONLINE
scan: scrub repaired 0
in 2h25m with 0 errors on Sat Sep 14 04:25:50 2013

cfg:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>raidz2-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada1p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada2p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada3p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada4p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada5p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

data:

errors: No known data errors

19.3.4. 清除錯誤

當偵測到錯誤發生，讀取、寫入或校驗碼（Checksum）的計數便會增加。使用 `zpool clear mypool` 可以清除錯誤訊息及重置計數。清空錯誤狀態對當儲存池發生錯誤要使用自動化 Script 通知的管理者來說很重要，因為在舊的錯誤尚未清除前不會回報後續的錯誤。

19.3.5. 更換運作中的裝置

可能有一些情況會需要更換磁碟為另一個磁碟，當要更換運作中的磁碟，此程序會維持舊有的磁碟在更換的過程為上線的狀態，儲存池不會進入降級（Degraded）的狀態，以減少資料遺失的風險。

`zpool replace` 會複製所有舊磁碟的資料到新磁碟，操作完成之後舊磁碟便會與 vdev 409
中斷連線。若新磁碟容量較舊磁碟大，也可以會增加儲存池來使用新的空間，请参考扩展儲存池。

更換儲存池中正在运作的狀態:

```
# zpool status
pool: mypool
state: ONLINE
scan: none requested
config:
  NAME        STATE     READ WRITE CKSUM
  mypool      ONLINE       0     0     0
  mirror-0  ONLINE       0     0     0
  ada0p3  ONLINE       0     0     0
  ada1p3  ONLINE       0     0     0

errors: No known data errors
```

```
# zpool replace mypool ada1p3 ada2p3
Make sure to wait until resilver is done before rebooting.
```

如果您从pool'zroot'引导，您可能需要更新新建连接磁碟'ada2p3'的boot code。

假设您使用GPT分割表且'da0'是您的新boot磁碟，您可能需要使用以下命令:

```
gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
```

```
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada2
# zpool status
pool: mypool
state: ONLINE
status: One or more devices is currently being resilvered. The pool will continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scan: resilver in progress since Mon Jun  2 14:21:35 2014
604M scanned out of 781M at 46.5M/s, 0h0m to go
604M resilvered, 77.39% done
```

```
config:
  NAME             STATE     READ WRITE CKSUM
  mypool           ONLINE       0     0     0
  mirror-0       ONLINE       0     0     0
  ada0p3       ONLINE       0     0     0
  ada1p3       ONLINE       0     0     0
```

```
410
```
Replacing

Online       0     0     0
ada1p3     ONLINE       0     0     0
ada2p3     ONLINE       0     0     0
(resilvering)

errors: No known data errors

# zpool status

pool: mypool
state: ONLINE
scan: resilvered 781M
in 0h0m with 0 errors on Mon Jun  2 14:21:52 2014

cfg:

NAME        STATE     READ WRITE CKSUM

mypool      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
ada0p3  ONLINE       0     0     0
ada2p3  ONLINE       0     0     0
errors: No known data errors

19.3.6.

故障處理

當儲存池中的磁碟故障，該故障硬碟所屬的vdev便會進入降級(Degraded)狀態，所有的資料仍可使用,但效能可能會降低,因為遺失的資料必須從可用的備援資料計算才取得。要將vdev恢復完整運作的狀態必須更換故障的實體裝置。然後ZFS便會開始修復(Resilver,古代鏡子的修復稱Resilver)作業，會從可用的備援資料計算出故障磁碟中的資料並寫入到替換的裝置上。完成後vdev便會重新返回線上(Online)的狀態。

若vdev沒有任何備援資料或有多個裝置故障，沒有足夠的備援資料可以補償，儲存池便會進入故障(Faulted)的狀態。

更換故障的磁碟時，故障磁碟的名稱會更換為裝置的GUID,若替換裝置要使用相同的裝置名稱,則在zpool replace不須加上新裝置名稱參數。

使用zpool replace更換故障的磁碟:

# zpool status

pool: mypool
state: DEGRADED
status: One or more devices could not be opened. Sufficient replicas exist for the pool to continue functioning in a degraded state.

action: Attach the missing device and online it using 'zpool online'.

see: http://illumos.org/msg/ZFS-8000-2Q

scan: none requested

cfg:

NAME                    STATE     READ WRITE CKSUM
<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>DEGRADED</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mirror-0</td>
<td>DEGRADED</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>replacing-1</td>
<td>UNAVAIL</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15732067398082357289</td>
<td>UNAVAIL</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

errors: No known data errors

# zpool replace mypool 316502962686821739 ada2p3

# zpool status

pool: mypool
state: DEGRADED
status: One or more devices is currently being resilvered. The pool will continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
scan: resilver in progress since Mon Jun  2 14:52:21 2014
641M scanned out of 781M at 49.3M/s, 0h0m to go
640M resilvered, 82.04% done

cfg:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mirror-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada2p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

errors: No known data errors

# zpool status

pool: mypool
state: ONLINE
scan: resilvered 781M in 0h0m with 0 errors on Mon Jun  2 14:52:38 2014

cfg:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mirror-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada2p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
清潔儲存池建議儲存池要定期清潔（Scrub），最好是每個月清潔一次。

作業對磁碟操作非常密集，在執行時會降低磁碟的效能。在排程時避免在使用高峰的時期，或使用`vfs.zfs.scrub_delay`來調整`scrub`的相對優先權來避免影響其他的作業。

```
# zpool scrub mypool
# zpool status
```

```
pool: mypool
state: ONLINE
scan: scrub
  in progress since Wed Feb 19 20:52:54 2014
  116G scanned out of 8.60T at 649M/s, 3h48m to go
  0 repaired, 1.32% done
cfg:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>raidz2</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada1p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada2p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada3p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada4p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada5p3</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

details:

errors: No known data errors
```

若發生需要取消清潔作業的事，可以下`zpool scrub -s mypool`。

自我修復校驗碼（Checksum）會隨資料區塊一併儲存，這使得檔案系統可以做到自我修復。這個功能可以在校驗碼與儲存池中的另一個裝置不同時自動修復資料。

例如來說，有兩個磁碟做鏡像（Mirror），其中一個磁碟機開始失常並無法正常儲存資料，甚至是資料放在長期封存的儲存裝置上，已經很久沒有被存取。

傳統的檔案系統需要執行演算法來檢查並修復資料如`fsck(8)`，這些指令耗費時間，且在嚴重時需要系統管理者手動決定要做那種修復操作。當ZFS偵測到資料區塊的校驗碼不對時，它除了把資料交給需要的應用程式外，也會修正在磁碟上錯誤的資料。這件事情不需要與系統管理者作任何互動便會在一般的儲存池操作時完成。

接下來的例子會示範自我修復會如何運作。

建立一個使用磁碟`/dev/ada0`及`/dev/ada1`做鏡像的儲存池。

```
# zpool create healer mirror /dev/ada0 /dev/ada1
# zpool status healer
```

NAME        STATE     READ WRITE CKSUM

healer      ONLINE       0     0     0
mirror-0    ONLINE       0     0     0
ada0        ONLINE       0     0     0
ada1        ONLINE       0     0     0

errors: No known data errors

# zpool list

NAME     SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT

healer   960M  92.5K   960M         -         -     0%    0%  1.00x  ONLINE  -

# cp /some/important/data /healer
# zfs list

NAME     SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT

healer   960M  67.7M   892M     7%  1.00x  ONLINE  -

# sha1 /healer > checksum.txt
# cat checksum.txt
SHA1(/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f

write random data to the first mirror to simulate data loss.
To avoid ZFS detecting the error immediately, zpool is exported,
then imported after simulating data loss.

This is a dangerous operation for important data.
Use it only for demonstration purposes, not in a production
environment, and do not use this example on other file systems.
Be sure to back up the storage pool before performing any
instructions!
status: One or more devices has experienced an unrecoverable error. An attempt was made to correct the error. Applications are unaffected.

action: Determine if the device needs to be replaced, and clear the errors using 'zpool clear' or replace the device with 'zpool replace'.

see: http://illumos.org/msg/ZFS-8000-4J

scan: scrub in progress since Mon Dec 10 12:23:30 2012
10.4M scanned out of 67.0M at 267K/s, 0h3m to go
9.63M repaired, 15.56% done
<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>healer</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mirror-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada1</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0,627</td>
</tr>
</tbody>
</table>

(Repairing)

Errors: No known data errors

清潔作業會從 ada0 讀取資料並重新寫入任何在 ada1 上有錯誤校驗碼的資料。這個操作可以由 zpool status 的輸出中發現修復中 (repairing) 的項目來辨識。這個作業完成後，儲存池的狀態會更改為:

```bash
# zpool status healer
pool: healer
state: ONLINE
```

scan: scrub repaired 66.5M in 0h2m with 0 errors on Mon Dec 10 12:26:25 2012

config:

<table>
<thead>
<tr>
<th>NAME</th>
<th>STATE</th>
<th>READ</th>
<th>WRITE</th>
<th>CKSUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>healer</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mirror-0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada0</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ada1</td>
<td>ONLINE</td>
<td>0</td>
<td>0</td>
<td>0,2.72K</td>
</tr>
</tbody>
</table>

清潔操作完成便同步了 ada0 到 ada1 間的所有資料。執行 zpool clear 可以清除 (Clear) 儲存池狀態的錯誤訊息。

```bash
# zpool clear healer
# zpool status healer
pool: healer
state: ONLINE
```

scan: scrub repaired 66.5M in 0h2m with 0 errors on Mon Dec 10 12:26:25 2012
儲存池現在恢復完整運作的狀態且清除了所有的錯誤了。

19.3.9. 擴增儲存池可用的備援儲存池大小會受到每個vdev中容量最小的裝置限制。最小的裝置可以替換成較大的裝置，在更換（Replace）或修復（Resilver）工作後，儲存池可以成長到該新裝置的可用容量。例如，要作一個1 TB磁碟機與一個2 TB磁碟機的鏡像，可用的空間會是1 TB，當1 TB磁碟機備更換成另一個2 TB的磁碟機時，修復程序會複製既有的資料到新的磁碟機，由於現在兩個裝置都有2 TB的容量，所以鏡像的可用空間便會成長到2 TB。

可以在每個裝置用zpool online -e來觸發擴充的動作，在擴充完所有裝置後，儲存池便可用額外的空間。

19.3.10. 輸入與輸出儲存池儲存池在移動到其他系統之前需要做輸出（Export），會卸載所有的資料集，然後標記每個裝置為已輸出，為了避免被其他磁碟子系統存取，因此仍會鎖定這些裝置。這個動作讓儲存池可以在支援ZFS的其他機器、其他作業系統做輸入（Import），甚至是不同的硬體架構（有一些注意事項，請參考zpool(8)）。當資料集有被開啓的檔案，可使用zpool export -f來強制輸出儲存池，使用這個指令需要小心，資料集是被強制卸載的，因此有可能造成在該資料集開啓檔案的應用程式發生無法預期的結果。

輸出未使用的儲存池:
```
# zpool export mypool
```
輸入儲存池會自動掛載資料集，若不想自動掛載，可以使用zpool import -N。zpool import -o可以設定在輸入時暫時使用的屬性。zpool import altroot=允許輸入時指定基礎掛載點（Base mount point）來替換檔案系統根目錄。若儲存池先前用在不同的系統且不正常輸出，可能會需要使用zpool import -f來強制輸入。zpool import -a會輸入所有沒有被其他系統使用的儲存池。

列出所有可以輸入的儲存池:
```
# zpool import
```

```
pool: mypool
id: 9930174748043525076
state: ONLINE
action: The pool can be imported using its name or numeric identifier.
config:
mypool      ONLINE
ada2p3     ONLINE
```

使用替代的根目錄輸入儲存池:
```
417
```
zpool import -o altroot=/mnt mypool

zfs list

NAME     USED  AVAIL  REFER  MOUNTPOINT
mypool   110K  47.0G    31K  /mnt/mypool

升级储存池在升级 FreeBSD 之後或储存池是其他使用旧版 ZFS 的系统导入的,储存池可以手动升级到最新版本的 ZFS 来支援新的功能。在升级前请评估储存池是否还要在旧的系统做导入,由于升级是一个单向的程序,旧的储存池可以升级,但有新功能的储存池无法降级。

升级一个 v28 的储存池以支援功能旗标 (Feature Flags):

zpool status

pool: mypool
state: ONLINE
status: The pool is formatted using a legacy on-disk format. The pool can still be used, but some features are unavailable.
action: Upgrade the pool using 'zpool upgrade'. Once this is done, the pool will no longer be accessible on software that does not support feature flags.
scan: none requested
config:
NAME        STATE     READ WRITE CKSUM
mypool      ONLINE       0     0     0
mirror-0    ONLINE       0     0     0
ada0        ONLINE       0     0     0
ada1        ONLINE       0     0     0
errors: No known data errors

zpool upgrade

This system supports ZFS pool feature flags. The following pools are formatted with legacy version numbers and can be upgraded to use feature flags. After being upgraded, these pools will no longer be accessible by software that does not support feature flags.

VER  POOL
---  --------
28   mypool

418
Use 'zpool upgrade -v' for a list of available legacy versions.

Every feature flags pool has all supported features enabled.

```
# zpool upgrade mypool
```

This system supports ZFS pool feature flags.

Successfully upgraded 'mypool' from version 28 to feature flags.

Enabled the following features on 'mypool':
- async_destroy
- empty_bpobj
- lz4_compress
- multi_vdev_crash_dump

ZFS的新功能在zpool upgrade尚未完成之前無法使用。可以用zpool upgrade -v來查看升級後有哪些新功能，也同時會列出已經支援哪些功能。

升级存储池支援新版的功能旗标 (Feature flags):

```
# zpool status
pool: mypool
state: ONLINE
status: Some supported features are not enabled on the pool. The pool can still be used, but some features are unavailable.
action: Enable all features using 'zpool upgrade'. Once this is done, the pool may no longer be accessible by software that does not support the features. See zpool-features(7) for details.
scan: none requested
cfgset:
```

NAME        STATE     READ WRITE CKSUM
-------      -------   ------ ------
mypool      ONLINE       0     0     0
mirror-0  ONLINE       0     0     0
ada0    ONLINE       0     0     0
ada1    ONLINE       0     0     0
errors: No known data errors

# zpool upgrade
```
This system supports ZFS pool feature flags.
```
All pools are formatted using feature flags.

Some supported features are not enabled on the following pools. Once a feature is enabled the pool may become incompatible with software.
that does not support the feature. See zpool-features for details.

### POOL FEATURE

- zstore
- multi_vdev_crash_dump
- spacemap_histogram
- enabled_txg
- hole_birth
- extensible_dataset
- bookmarks
- filesystem_limits

```
# zpool upgrade mypool
```

This system supports ZFS pool feature flags. Enabled the following features on 'mypool':
- spacemap_histogram
- enabled_txg
- hole_birth
- extensible_dataset
- bookmarks
- filesystem_limits

In a system using a storage pool for booting, the Boot code must also be updated to support the new storage pool version, which can be done using `gpart bootcode`. Currently, there are two types of Boot code that can be used, depending on the system boot method:

- **GPT** (most common option)
- **EFI** (newer systems)

For traditional boot methods using GPT, execute the following command:

```
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
```

For boot methods using EFI, execute the following command:

```
# gpart bootcode -p /boot/boot1.efifat -i 1 ada1
```

Updating Boot code to all storage pools that can boot. Refer to `gpart(8)` for more information.

#### 19.3.12.

Displaying recorded storage pool history

Modifying storage pool commands will be recorded, as will the creation, modification, or replacement of the dataset. This history record is used to view the storage pool, who performed the action, what action, and when.

History records are not stored in a log file (Log file), but are stored in the storage pool. To view this history record, the command name is `zpool history`:

```
# zpool history
```
Zpool history for 'tank':

2013-02-26 23:02:35 zpool create tank mirror /dev/ada0 /dev/ada1

2013-02-27 18:50:58 zfs set atime=off tank

2013-02-27 18:51:09 zfs set checksum=fletcher4 tank

2013-02-27 18:51:18 zfs create tank/backup

Output: Displays the results of the zpool and zfs commands run on the storage pool along with the times. Only commands that change the storage pool or something similar are recorded, such as zfs list. When no pool name is specified, all pool histories are listed.

zpool history -i

History for 'tank':

2013-02-26 23:02:35 [internal pool create txg:5] pool spa 28; zfs spa 28; zpl 5; uts 9.1-RELEASE 901000 amd64

2013-02-27 18:50:53 [internal property set txg:50] atime=0 dataset=21

2013-02-27 18:50:58 zfs set atime=off tank


2013-02-27 18:51:09 zfs set checksum=fletcher4 tank


2013-02-27 18:51:18 zfs create tank/backup

zpool history -l

History for 'tank':

2013-02-26 23:02:35 zpool create tank mirror /dev/ada0 /dev/ada1

2013-02-27 18:50:58 zfs set atime=off tank

2013-02-27 18:51:09 zfs set checksum=fletcher4 tank

2013-02-27 18:51:18 zfs create tank/backup
### zpool iostat

<table>
<thead>
<tr>
<th></th>
<th>alloc</th>
<th>free</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td></td>
<td></td>
</tr>
<tr>
<td>write</td>
<td></td>
<td></td>
</tr>
<tr>
<td>read</td>
<td></td>
<td></td>
</tr>
<tr>
<td>write</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**data**

- 288G  1.53T  2  12  9.23K  61.5K

**mirror**

- 288G  1.53T  2  12  9.23K  61.5K

**ada1**

- -  -  0  4  5.61K  61.7K

**ada2**

- -  -  1  4  5.04K  61.7K

----------

要持續監視 I/O 的活動可以在最後的參數指定一個數字, 這個數字代表每次更新資訊所間隔的秒數。在每次經過間隔的時間後會列出新一行的統計資訊, 按下 Ctrl+C 可以中止監視。或者在指令列的間隔時間之後再指定一個數字, 代表總共要顯示的統計資訊筆數。

使用 `-v` 可以顯示更詳細的 I/O 統計資訊。

每個在儲存池中的裝置會以一行統計資訊顯示。這可以幫助了解每一個裝置做了多少讀取與寫入的操作, 並可協助確認是否有別的裝置拖慢了整個儲存池的速度。以下範例會顯示有兩個裝置的鏡像儲存池:

```
# zpool iostat -v
```

19.3.14.

分割儲存儲存池由一個或多個鏡像 vdev 所組成的儲存池可以切分開成兩個儲存池。除非有另外指定, 否則每個鏡像的最後一個成員會被分離來建立一個含有相同資料的新儲存池。在做這個操作的第一次應先使用 `-n`, 會顯示預計會做的操作而不會真的執行, 這可以協助確認操作是否與使用者所要的相同。

19.4.

zfs 管理

zfs 工具負責建立、摧毀與管理在一個儲存池中所有的 ZFS 資料集。儲存池使用 zpool 來管理。

19.4.1.

建立與摧毀資料集

不同於傳統的磁碟與磁碟區管理程式 (Volume manager), 在 ZFS 中的空間並不需要預先分配。傳統的檔案系統在分割與分配空間完後, 若沒有增加新的磁碟便無法再增加額外的檔案系統。在 ZFS, 可以隨時建立新的檔案系統, 每個資料集 (Dataset) 都有自己的性質, 包含壓縮 (Compression)、去重 (Deduplication)、快取 (Caching) 與配額 (Quota) 功能以外還有有用的性質如唯讀 (Readonly)、區分大小寫 (Case sensitivity)、網路檔案分享 (Network file sharing) 以及掛載點 (Mount point)。資料集可以存於於其他資料集中, 且子資料集會繼承其父資料集的性質。每個資料集都可以作為一個單位來管理、委託 (Delegate)、備份 (Replicate)、快照 (Snapshot)、監禁 (Jail) 與摧毀 (Destroy), 代替每種不同類型或集合的檔案建立各別的資料集還有多多的優點。唯一的缺點是在當有非常大量資料集時, 像 zfs list 這樣的指令會變的較緩慢, 且掛載上百個或甚至上千個資料集可能會使 FreeBSD 的開機程序變慢。
建立一个新的资料集并启用LZ4压缩：

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
<th>MOUNTPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>781M</td>
<td>93.2G</td>
<td>144K</td>
<td>none</td>
</tr>
<tr>
<td>mypool/ROOT</td>
<td>777M</td>
<td>93.2G</td>
<td>144K</td>
<td>none</td>
</tr>
<tr>
<td>mypool/ROOT/default</td>
<td>777M</td>
<td>93.2G</td>
<td>777M</td>
<td>/</td>
</tr>
<tr>
<td>mypool/tmp</td>
<td>176K</td>
<td>93.2G</td>
<td>176K</td>
<td>/tmp</td>
</tr>
<tr>
<td>mypool/usr</td>
<td>616K</td>
<td>93.2G</td>
<td>144K</td>
<td>/usr</td>
</tr>
<tr>
<td>mypool/usr/home</td>
<td>184K</td>
<td>93.2G</td>
<td>184K</td>
<td>/usr/home</td>
</tr>
<tr>
<td>mypool/usr/ports</td>
<td>144K</td>
<td>93.2G</td>
<td>144K</td>
<td>/usr/ports</td>
</tr>
<tr>
<td>mypool/usr/src</td>
<td>144K</td>
<td>93.2G</td>
<td>144K</td>
<td>/usr/src</td>
</tr>
<tr>
<td>mypool/var</td>
<td>1.20M</td>
<td>93.2G</td>
<td>608K</td>
<td>/var</td>
</tr>
<tr>
<td>mypool/var/crash</td>
<td>148K</td>
<td>93.2G</td>
<td>148K</td>
<td>/var/crash</td>
</tr>
<tr>
<td>mypool/var/log</td>
<td>178K</td>
<td>93.2G</td>
<td>178K</td>
<td>/var/log</td>
</tr>
<tr>
<td>mypool/var/mail</td>
<td>144K</td>
<td>93.2G</td>
<td>144K</td>
<td>/var/mail</td>
</tr>
<tr>
<td>mypool/var/tmp</td>
<td>152K</td>
<td>93.2G</td>
<td>152K</td>
<td>/var/tmp</td>
</tr>
</tbody>
</table>

# zfs create -o compress=lz4 mypool/usr/mydataset

# zfs list

<table>
<thead>
<tr>
<th>NAME</th>
<th>USED</th>
<th>AVAIL</th>
<th>REFER</th>
<th>MOUNTPOINT</th>
</tr>
</thead>
<tbody>
<tr>
<td>mypool</td>
<td>880M</td>
<td>93.1G</td>
<td>144K</td>
<td>none</td>
</tr>
<tr>
<td>mypool/ROOT</td>
<td>777M</td>
<td>93.2G</td>
<td>144K</td>
<td>none</td>
</tr>
<tr>
<td>mypool/ROOT/default</td>
<td>777M</td>
<td>93.2G</td>
<td>777M</td>
<td>/</td>
</tr>
<tr>
<td>mypool/tmp</td>
<td>176K</td>
<td>93.2G</td>
<td>176K</td>
<td>/tmp</td>
</tr>
<tr>
<td>mypool/usr</td>
<td>704K</td>
<td>93.2G</td>
<td>144K</td>
<td>/usr</td>
</tr>
<tr>
<td>mypool/usr/home</td>
<td>184K</td>
<td>93.2G</td>
<td>184K</td>
<td>/usr/home</td>
</tr>
<tr>
<td>mypool/usr/mydataset</td>
<td>87.5K</td>
<td>93.2G</td>
<td>87.5K</td>
<td>/usr/mydataset</td>
</tr>
<tr>
<td>mypool/usr/ports</td>
<td>144K</td>
<td>93.2G</td>
<td>144K</td>
<td>/usr/ports</td>
</tr>
<tr>
<td>mypool/usr/src</td>
<td>144K</td>
<td>93.2G</td>
<td>144K</td>
<td>/usr/src</td>
</tr>
<tr>
<td>mypool/var</td>
<td>1.20M</td>
<td>93.2G</td>
<td>610K</td>
<td>/var</td>
</tr>
<tr>
<td>mypool/var/crash</td>
<td>148K</td>
<td>93.2G</td>
<td>148K</td>
<td>/var/crash</td>
</tr>
<tr>
<td>mypool/var/log</td>
<td>178K</td>
<td>93.2G</td>
<td>178K</td>
<td>/var/log</td>
</tr>
<tr>
<td>mypool/var/mail</td>
<td>144K</td>
<td>93.2G</td>
<td>144K</td>
<td>/var/mail</td>
</tr>
<tr>
<td>mypool/var/tmp</td>
<td>152K</td>
<td>93.2G</td>
<td>152K</td>
<td>/var/tmp</td>
</tr>
</tbody>
</table>
19.4.2. 建立與摧毀磁碟區

磁碟區（Volume）是特殊類型的資料集，不會被掛載成一個檔案系統，而是會被當做儲存區塊裝置出現在 /dev/zvol/poolname/dataset 下。這讓磁碟區可供其他檔案系統使用，拿來備份虛擬機器的磁碟或是使用 iSCSI 或 HAST 通訊協定匯出。

磁碟區可以被格式化成任何檔案系統，或不使用檔案系統來儲存原始資料。對一般使用者，磁碟區就像是一般的磁碟，可以放置一般的檔案系統在這些 zvols 上，並提供一般磁碟或檔案系統一般的所沒有的功能。例如，使用壓縮屬性在一個 250 MB 的磁碟區可建立一個壓縮的 FAT 檔案系統。
# zfs create -V 250m -o compression=on tank/fat32
# zfs list tank
NAME USED AVAIL REFER MOUNTPOINT
tank 258M  670M   31K /tank
# newfs_msdos -F32 /dev/zvol/tank/fat32
# mount -t msdosfs /dev/zvol/tank/fat32 /mnt
# df -h /mnt | grep fat32
Filesystem           Size Used Avail Capacity Mounted on
/dev/zvol/tank/fat32 249M  24k  249M     0%   /mnt
# mount | grep fat32
/dev/zvol/tank/fat32 on /mnt (msdosfs, local)

摧毀一個磁碟區與摧毀一個一般檔案系統資料集差不多。操作上幾乎是即時的, 但在背景會需要花費數分鐘來釋放空間再次可用。

19.4.3. 重新命名資料集
資料集的名稱可以使用 `zfs rename` 變更。父資料集也同樣可以使用這個指令來更改名稱。重新命名一個資料集到另一個父資料集也會更改自父資料集繼承的屬性值。重新命名資料集後, 會被卸載然後重新掛載到新的位置 (依繼承的新父資料集而定), 可使用 `-u` 來避免重新掛載。

重新命名一個資料集並移動該資料集到另一個父資料集:

```bash
# zfs list
NAME USED AVAIL REFER MOUNTPOINT
mypool 780M 93.2G 144K none
mypool/ROOT 777M 93.2G 144K none
mypool/ROOT/default 777M 93.2G 777M /
mypool/tmp 176K 93.2G 176K /tmp
mypool/usr 704K 93.2G 144K /usr
mypool/usr/home 184K 93.2G 184K /usr/home
mypool/usr/mydataset 87.5K 93.2G 87.5K /usr/mydataset
mypool/usr/ports 144K 93.2G 144K /usr/ports
mypool/usr/src 144K 93.2G 144K /usr/src
mypool/var 1.21M 93.2G 614K /var
mypool/var/crash 148K 93.2G 148K /var/crash
mypool/var/log 178K 93.2G 178K /var/log
mypool/var/mail 144K 93.2G 144K /var/mail
mypool/var/tmp 152K 93.2G 152K /var/tmp
# zfs rename mypool/usr/mydataset mypool/var/newname
# zfs list
NAME USED AVAIL REFER MOUNTPOINT
mypool 780M 93.2G 144K none
mypool/ROOT 777M 93.2G 144K none
```
快照也可以像这样重新命名，由于快照的本质使其无法被重新命名到另一个父数据集。要递归重新命名快照可指定 `-r`，然后在子数据集中所有同名的快照也会一并被重新命名。

```
# zfs list -t snapshot
NAME                                USED  AVAIL  REFER  MOUNTPOINT
mypool/var/newname@first_snapshot      0      -  87.5K  -
# zfs rename mypool/var/newname@first_snapshot new_snapshot_name
# zfs list -t snapshot
NAME                                   USED  AVAIL  REFER  MOUNTPOINT
mypool/var/newname@new_snapshot_name      0      -  87.5K  -
```

设定数据集属性

每个ZFS数据集有几个属性可以用来自定义其行为。大部分的属性会自动继承自其父数据集，但可以被自己覆盖。设定数据集上的属性可使用 `zfs set property=value dataset`。大部分属性有限制可用的值，`zfs get`会显示每个可以使用的属性及其可用的值。大部分可以使用 `zfs inherit` 还原成其继承的值。

也可设定使用者自订的属性。这些属性也会成为数据集设定的一部分，且可以用来提供数据集或其内容的额外资讯。要别分自订属性与ZFS提供的属性，会使用冒号(：)建立一个自订命名空间供自订属性使用。

```
# zfs set custom:costcenter=1234 tank
# zfs get custom:costcenter tank
NAME PROPERTY           VALUE              SOURCE
tank custom:costcenter  1234 local
```

移除自订属性，可用 `zfs inherit` 加上 `-r`。若父数据集未定义任何自订属性，将会将该属性完全移除（更改动作仍会记录在存储池的历史记录）。

```
# zfs inherit -r custom:costcenter tank
# zfs get custom:costcenter tank
NAME    PROPERTY           VALUE              SOURCE
```

19.4.4.

设定数据集属性

建立快照

可以使用 `zfs snapshot` 来建立。

加入 `-r` 可以递对所有同名的子资料集建立快照。

建立一个整存池的递快照:

```bash
# zfs list -t all
NAME                                   USED  AVAIL  REFER  MOUNTPOINT
mypool                                 780M  93.2G   144K  none
mypool/ROOT                            777M  93.2G   144K  none
mypool/ROOT/default                    777M  93.2G   777M  /
mypool/tmp                             176K  93.2G   176K  /tmp
mypool/usr                             616K  93.2G   144K  /usr
mypool/usr/home                        184K  93.2G   184K  /usr/home
mypool/usr/ports                       144K  93.2G   144K  /usr/ports
mypool/usr/src                         144K  93.2G   144K  /usr/src
mypool/var                            1.29M  93.2G   616K  /
mypool/var/crash                       148K  93.2G   148K  /var/crash
mypool/var/log                         178K  93.2G   178K  /var/log
mypool/var/mail                        144K  93.2G   144K  /var/mail
mypool/var/newname                    87.5K  93.2G  87.5K  /var/newname
mypool/var/newname@new_snapshot_name      0      -  87.5K  -
mypool/var/tmp                         152K  93.2G   152K  /var/tmp
# zfs snapshot -r mypool@my_recursive_snapshot
# zfs list -t snapshot
NAME                                        USED  AVAIL  REFER  MOUNTPOINT
mypool@my_recursive_snapshot                   0      -   144K  -
mypool/ROOT@my_recursive_snapshot              0      -   144K  -
mypool/ROOT/default@my_recursive_snapshot      0      -   777M  -
mypool/tmp@my_recursive_snapshot               0      -   176K  -
mypool/usr@my_recursive_snapshot               0      -   144K  -
mypool/usr/home@my_recursive_snapshot          0      -   184K  -
mypool/usr/ports@my_recursive_snapshot         0      -   144K  -
mypool/usr/src@my_recursive_snapshot           0      -   144K  -
mypool/var@my_recursive_snapshot               0      -   616K  -
mypool/var/crash@my_recursive_snapshot         0      -   148K  -
mypool/var/log@my_recursive_snapshot           0      -   178K  -
mypool/var/mail@my_recursive_snapshot          0      -   144K  -
mypool/var/newname@new_snapshot_name           0      -  87.5K  -
mypool/var/newname@my_recursive_snapshot       0      -  87.5K  -
mypool/var/tmp@my_recursive_snapshot           0      -   152K  -
```

建立的快照不会显示在一般 `zfs list` 操作结果，要列出快照需在 `zfs list` 后加上 `-t snapshot`，使用 `-t all` 可以同时列出档案系统的内容及快照。
快照並不會直接掛載,因此MOUNTPOINT欄位的路徑如此顯示。在AVAIL欄位不會有可用的磁碟空間,因為快照建立之後便無法再寫入。

比较快照与其原来建立时的资料集:

```bash
# zfs list -rt all mypool/usr/home
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
mypool/usr/home                         184K  93.2G   184K  /usr/home
mypool/usr/home@my_recursive_snapshot      0      -   184K  -
```

同时显示资料集与快照可以了解快照如何使用。

快照只会保存有更动(差异)的资料,并非整个档案系统的内容,这个意思是说,快照只会在有做更动时使用一小部分的空閒空间,複製一个档案到该资料集,可以让空间使用量变得更为明显,然后做第二个快照:

```bash
# cp /etc/passwd /var/tmp
# zfs snapshot mypool/var/tmp@after_cp
# zfs list -rt all mypool/var/tmp
```

第二快照只会包含资料集做了複製动作之后的更动,这样子的机制可以节省大量的空閒空间。注意在複製之后快照mypool/var/tmp@my_recursive_snapshot於USED欄位中的大小也更改了,这说明了这个更动在前次快照与之后快照间的关係。

19.4.5.2. 比对快照

ZFS提供了内建指令可以用来比对两个快照(Snapshot)之间的差异,在使用者想要查看一段时间的档案系统所的变更时非常有用。

例如`zfs diff`可以让使用者在最后一次快照中找到意外删除的档案。

对前面一节所做的两个快照使用这个指令会产生以下结果:

```bash
# zfs list -rt all mypool/var/tmp
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                         206K  93.2G   118K  /var/tmp
mypool/var/tmp@my_recursive_snapshot    88K      -   152K  -
mypool/var/tmp@after_cp                   0      -   118K  -
```

```bash
# zfs diff mypool/var/tmp@my_recursive_snapshot
M       /var/tmp/
+       /var/tmp/passwd
```

指令会列出指定快照(在这个例子中为mypool/var/tmp@my_recursive_snapshot)与目前档案系统间的更改。第一個欄位是更改的类型：

+ 加入了该路径或档案。
- 刪除除了该路径或档案。
M 修改了该路径或档案。
R 重新命名了该路径或档案。
對照這個表格來看輸出的結果，可以明顯的看到 passwd 是在快照 mypool/var/tmp@my_recursive_snapshot 建立之後才加入的，結果也同樣看的到掛載到 /var/tmp 的父目錄已經做過修改。

在使用 ZFS 備份功能來傳輸一個資料集到另一個主機備份時比對兩個快照也同樣很有用。比對兩個快照需要提供兩個資料集的完整資料集名稱與快照名稱:

```bash
# cp /var/tmp/passwd /var/tmp/passwd.copy
# zfs snapshot mypool/var/tmp@diff_snapshot
# zfs diff mypool/var/tmp@my_recursive_snapshot mypool/var/tmp@diff_snapshot
M       /var/tmp/
+       /var/tmp/passwd
+       /var/tmp/passwd.copy
# zfs diff mypool/var/tmp@my_recursive_snapshot mypool/var/tmp@after_cp
M       /var/tmp/
+       /var/tmp/passwd
```

備份管理者可以比對兩個自傳送主機所接收到的兩個快照並查看實際在資料集中的變更。請參考備份一節來取得更多資訊。

19.4.5.3. 使用快照還原只要至少有一個可用的快照便可以隨時還原。大多數在已不需要目前資料集, 想要用較舊版的資料的情況, 例如, 本地開發的測試發生錯誤、不良的系統更新破壞了系統的整體功能或需要還原意外刪除的檔案或目錄,...等，都是非常常見的情形。幸運的，要還原到某個快照只需要簡單輸入 `zfs rollback snapshotname`。會依快照所做的變更數量來決定處理的時間，還原的操作會在一段時間後完成。在這段時間中，資料集會一直保持一致的狀態，類似一個符合 ACID 原則的資料庫在做還原。還原可在資料集處於上線及可存取的情況下完成，不需要停機。還原到快照之後，資料集便回到當初執行快照時相同的狀態，所有沒有在快照中的其他資料便會被丟棄，因此往後若還有可能需要部份資料時，建議在還原到前一個快照之前先對目前的資料集做快照，這樣一來，使用者便可以在快照之際來快換，而不會遺失重要的資料。

在此時，使用者發現到刪除的太多檔案並希望能夠還原。ZFS 提供了簡單的方可以取回檔案，便是在還原 (Rollback)，但這只在有定期對重要的資料使用快照時可用。要拿回檔案並從最後一次快照重新開始，可執行以下指令:

```bash
# zfs list -rt all mypool/var/tmp
NAME                                   USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                         262K  93.2G   120K  /var/tmp
mypool/var/tmp@my_recursive_snapshot    88K      -   152K  -
mypool/var/tmp@after_cp               53.5K      -   118K  -
mypool/var/tmp@diff_snapshot              0      -   120K  -
# ls /var/tmp
passwd          passwd.copy     vi.recover
# rm /var/tmp/passwd*
# ls /var/tmp
vi.recover
```

在此時，使用者發現到刪除的太多檔案並希望能夠還原。ZFS 提供了簡單的方可以取回檔案，便是在還原 (Rollback)，但這只在有定期對重要的資料使用快照時可用。要拿回檔案並從最後一次快照重新開始，可執行以下指令:
還原操作會將資料集還原為最後一次快照的狀態。這也可以還原到更早之前，有其他在其之後建立的快照。

要這麼做時，ZFS會發出這個警告:

```
# zfs list -rt snapshot mypool/var/tmp
AME                                   USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp@my_recursive_snapshot    88K      -   152K  -
mypool/var/tmp@after_cp               53.5K      -   118K  -
mypool/var/tmp@diff_snapshot              0      -   120K  -
```

```
# zfs rollback mypool/var/tmp@my_recursive_snapshot
cannot rollback to 'mypool/var/tmp@my_recursive_snapshot': more recent snapshots exist
```

使用`-r`來力圖删除跟該快照之間有其他快照存在的快照，因為ZFS無法追蹤不同資料集狀態間的變更。在使用者未指定`-r`來確認這個動作前，ZFS不會刪除受影響的快照。若確定了這麼做，那必須要知道會遺失所有在這之間的快照，然後可執行以下命令:

```
# zfs rollback -r mypool/var/tmp@my_recursive_snapshot
# zfs list -rt snapshot mypool/var/tmp
```

```
AME                                   USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp@my_recursive_snapshot     8K      -   152K  -
```

可從`zfs list -t snapshot`的結果來確認`zfs rollback -r`會移除的快照。

19.4.5.4. 從快照還原個別檔案
快照會掛載在父資料集下的隱藏目錄:
```
./zfs/snapshots/snapshotname
```
預設不會顯示這些目錄，即使使用了`ls -a`指令。

雖然該目錄不會顯示，但該目錄實際上存在，而且可以像一般的目錄一樣存取。一個名稱為`snapdir`的屬性可以控制是否在目錄清單中顯示這些隱藏目錄，設定該屬性為可見`visible`可以讓這些目錄出現在外`ls`以及其他處理目錄內容的指令中。

```
# zfs get snapdir mypool/var/tmp
NAME            PROPERTY  VALUE    SOURCE
mypool/var/tmp  snapdir   hidden   default
```

```
# ls -a /var/tmp
.
..              passwd          vi.recover
```

```
431
```
要还原个别文件到先前的状态非常简单，只要从快照中复制文件到父资料集。在.zfs/snapshot目录结构下有一个与先前所做的快照名称相同的目录，可以很容易找到。在下个示例中，我们会示範从隐藏的.zfs目录还原一个文件，透过从包含该文件的最新版快照复制：

```bash
# zfs set snapdir=visible mypool/var/tmp
# ls -a /var/tmp
..
..              .zfs            passwd          vi.recover
# ls /var/tmp/.zfs/snapshot
after_cp                my_recursive_snapshot
# ls /var/tmp/.zfs/snapshot/after_cp
passwd          vi.recover
# cp /var/tmp/.zfs/snapshot/after_cp/passwd /var/tmp
```

执行ls .zfs/snapshot时，虽然snapdir可能已经设为隐藏，但可能仍可以显示该目录中的内容，这取决于管理者是否要显示这些目录，可以只显示特定的资料集，而其他的则不显示。从这个隐藏的.zfs/snapshot复制档案或目录非常简单，除此之外，尝试其他的动作则会出现以下错误:

```bash
# cp /etc/rc.conf /var/tmp/.zfs/snapshot/after_cp/
cp: /var/tmp/.zfs/snapshot/after_cp/rc.conf: Read-only file system
```

这个错误用来提醒使用者快照是唯读的，在建立之后不能更改。无法复制档案进去或从该快照目录中移除，因为这会变更该资料集所代表的状态。快照所消耗的空闲空间是依照自快照之后父档案系统做了多少变更来决定，快照的written属性可以用来追踪有多少空间被快照所使用。

使用zfs destroy dataset@snapshot可以摧毁快照并回收空闲空间。加上-r可以递回移除所有在父资料集下使用同名的快照。加入-n -v来显示将要移除的快照清单以及估计回收的空间，而不实际执行摧毁的操作。

19.4.6. 管理副本（Clone）

副本（Clone）是快照的复制，但更像是一般的资料集，与快照不同的是，副本是非唯读的（可写），且可挂载，可以有自己的属性。使用zfs clone建立副本之后，便无法再摧毁用来建立副本的快照。副本与快照的父/子关系可以使用zfs promote来对换。提升副本之后，快照便会成为副本的子资料集，而不是原来父资料集，这个动作会改变空间计算的方式，但并不会实际改变空间的使用量。副本可以被挂载到ZFS档案系统阶层中的任何一点，而非法只能位于原来快照的位置底下。要示範复制功能会用到这个示例资料集：
會使用到副本一般是要在可以保留快照以便出错时还原指定的资料集做实验，严格的来说这并不是必要，因为副本与资料集可同时存在，不会有任何问题。

```
# zfs clone camino/home/joe@backup camino/home/joenew
# ls /usr/home/joe*
/usr/home/joe: backup.txz plans.txt
/usr/home/joenew: backup.txz plans.txt
# df -h /usr/home
Filesystem          Size    Used   Avail Capacity  Mounted on
usr/home/joe        1.3G    128k    1.3G     0%    /usr/home/joe
usr/home/joenew     1.3G    128k    1.3G     0%    /usr/home/joenew
```

建立完成的副本便有与建立快照时状态相同的资料集，现在副本可以独立于原来的资料集来做更改。剩下唯一与资料集之间的关系便是快照，ZFS会在性质origin记录这个关系，一旦在快照与副本之间的相关关系因为使用zfs promote提升而移除时，副本的origin也会因为成为完全独立的资料集而移除。以下范例会示范这个动作:

```
# zfs get origin camino/home/joenew
NAME                  PROPERTY  VALUE                     SOURCE
camino/home/joenew    origin    camino/home/joe@backup    -
# zfs promote camino/home/joenew
# zfs get origin camino/home/joenew
NAME                  PROPERTY  VALUE   SOURCE
camino/home/joenew    origin    -       -
```

作为部份更改之后，例如复制loader.conf到提升后的副本，这个例子中的旧目录便无须保留，取而代之的是提升后的副本，这个动作可以用两个连续的指令来完成:在旧资料集上执行zfs destroy并在与旧资料相似名称(也可能用完全不同的名称)的副本上执行zfs rename。

```
# cp /boot/defaults/loader.conf /usr/home/joenew
# zfs destroy -f camino/home/joe
# zfs rename camino/home/joenew camino/home/joe
# ls /usr/home/joe
backup.txz     loader.conf     plans.txt
# df -h /usr/home
Filesystem          Size    Used   Avail Capacity  Mounted on
usr/home/joe        1.3G    128k    1.3G     0%    /usr/home/joe
```
複本現在我們可以如同一般的資料集一樣使用，它們的內容包含來自原始快照的資料以及後來加入的檔案，例如 `loader.conf`。

複本可以在許多不同的情境下使用提供 ZFS 的使用者有用的功能，例如，Jail 可以透過含有已安裝了各種應用程式集的快照提供，使用者可以複製這些快照然後加入自己想要試的應用程式，但更改可以滿足需求，便可以提升複本為完整的資料集然後提供給終端使用者，讓終端使用者可以如同實際擁有資料集一樣的使用，這個以節省提供這些 Jail 的時間與管理成本。

19.4.7. 備份（Replication）

將資料保存在單一儲存池上的單一儲存會讓資料暴露出在盜竊、自然或人為的風險之下，定期備份整個储存池非常要，ZFS 提供了內建的序列化（Serialization）功能可以將資料以串流傳送到標準輸出。使用這項技術，不僅可以將資料儲存到另一個已連結到本地系統的儲存池，也可以透過網路將資料傳送到另一個系統，這種備份方式以快照為基礎（請參考章節 ZFS 快照（Snapshot））。

用來備份資料的指令為 `zfs send` 及 `zfs receive`。

以下例子將示範使用兩個儲存池來做 ZFS 備份:

```
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M         -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%     4%  1.00x  ONLINE  -
```

名為 `mypool` 的儲存池為主要的儲存池，資料會定期寫入與讀取的位置。第二個儲存池 `backup` 用來待命（Standby），萬一主要儲存池無法使用時可替換。注意，ZFS 並不會自動做容錯轉移（Fail-over），必須要由系統管理者在需要的時候手動完成。快照會用來提供一個與檔系統一致的版本來做備份，`mypool` 的快照建立之後，便可以複製到 `backup` 儲存池，只有快照可以做備份，最近一次快照之後所做的變更不會包含在內容裡面。

```
# zfs snapshot mypool@backup1
# zfs list -t snapshot
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool@backup1             0      -  43.6M  -
```

快照存在以後，便可以使用 `zfs send` 來建立一個代表快照內容的串流，這個串流可以儲存成檔案或由其他儲存池接收。串流會寫入到標準輸出，但必須要重新導向到一個位於在已掛載到備份儲存池上的檔案。

```
# zfs send mypool@backup1
Error: Stream can not be written to a terminal.
You must redirect standard output.
```

要使用 `zfs send` 備份一個資料集，可重新導向到一個位於在已掛載到備份儲存池上的檔案。確定該儲存池有足夠的空間容納要傳送的快照，這裡是指該快照中內含的所有資料，並非只有上次快照到該快照間的變更。

```
# zfs send mypool@backup1 > /backup/backup1
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M         -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%     4%  1.00x  ONLINE  -
```
zfs send 會傳輸在快照backup1中所有的資料到儲存池backup。可以使用cron(8)排程來自動完成建立與傳送快照的動作。若不想將備份以封存案儲存，ZFS可用實際的檔案系統來接收資料，讓備份的資料可以直接被存取。要取得實際包含在串流中的資料可以用zfs receive將串流轉換回檔案與目錄。以下例子會以管線符號連接zfs send及zfs receive，將資料從一個儲存池複製到另一個，傳輸完成後可以直接使用接收儲存池上資料。一個資料集只可以被複製到另一個空的資料集。

```bash
# zfs snapshot mypool@replica1
# zfs send -v mypool@replica1 | zfs receive backup/mypool
```

```
send from @ to mypool@replica1 estimated size is 50.1M
total estimated size is 50.1M
```

```
TIME   SENT   SNAPSHOT
```

```
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M         -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%     4%  1.00x  ONLINE  -
```

```
19.4.7.1.
漸進式備份zfs send也可以比較兩個快照之間的差異，並且只傳送兩者之間的差異，這樣做可以節省磁碟空間及傳輸時間。例如:

# zfs snapshot mypool@replica2
# zfs list -t snapshot
```

```
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool@replica1         5.72M      -  43.6M  -
mypool@replica2             0      -  44.1M  -
```

```
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT
backup  960M  61.7M   898M         -         -     0%    6%  1.00x  ONLINE  -
mypool  960M  50.2M   910M         -         -     0%    5%  1.00x  ONLINE  -
```

```
會建立一個名為replica2的第二個快照，這個快照只中只會含目前與前次快照replica1之檔案系統所做的變更。使用zfs send -i並指定要用來產生漸進備份串流的快照，串流中只會有做過更改的資料。這個動作只在接收到端已經有初始快照時才可用。

```
# zfs send -v -i mypool@replica1 mypool@replica2 | zfs receive /backup/mypool
```

```
send from @replica1 to mypool@replica2 estimated size is 5.02M
total estimated size is 5.02M
```

```
TIME   SENT   SNAPSHOT
```

```
# zpool list
```

```
435
```
如此一來，便成功傳輸進式的串流，只有做過更改的資料會被備份，不會傳送完整的replica1。由於不會備份完整的儲存池，只傳送差異的部份，所以可以減少傳輸的時間並節省磁碟空間，特別是在網路緩慢或需要考量每位元傳輸成本時非常有用。

從儲存池mypool複製所有檔案與資料的新檔案系統backup/mypool便可以使用。若指定-P，會一併複製資料集的屬性，這包含壓縮(Compression)設定，配額(Quota)及掛載點(Mount point)。若指定-R，會複製所有指定資料集的子資料集，及這些子資料集的所有屬性。可將傳送與接收自動來定期使用第二個儲存池做備份。

19.4.7.2. 透過SSH傳送加密的備份透過網路來傳送串流是一個做遠端備份不錯的方式，但是一些缺點，透過網路連線傳送的資料沒有加密，這會讓任何人都可以在未告知傳送方的情況下攔截並轉換串流回資料，這是我們所不想要看到的情況，特別是在使用網絡傳送串流到遠端的主機時。SSH可用來加密要透過網路連線傳送的資料，在ZFS只需要將串流重新導向到標準輸出，如此一來便可簡單的轉接到SSH。若要讓檔案系統內容在傳送或在遠端系統中也維持在加密的狀態可考慮使用PEFS。有一些設定以及安全性注意事項必須先完成，只有對zfs send操作必要的步驟才會在此說明，要取得更多有關SSH的資訊請參考OpenSSH。

必要的環境設定:
• 使用SSH金鑰設定傳送端與接收端之間無密碼的SSH存取
• 正常會需要root的權限來傳送與接收串流，這需要可以root登入到接收端系統。但預設因安全性考慮會關閉以root登入。ZFS委託(ZFS Delegation)

系統可以用來允許一個非root使用者在每個系統上執行各自的發送與接收操作。

• 在傳送端系統上:
  # zfs allow -u someuser send,snapshot mypool

• 要掛載儲存池，無權限的使用者必須擁有該目錄且必須允許一般使用者掛載檔案系統。在接收端系統上:
# sysctl vfs.usermount=1

vfs.usermount: 0 -> 1

# sysrc -f /etc/sysctl.conf vfs.usermount=1

# zfs create recvpool/backup

# zfs allow -u someuser create,mount,receive recvpool/backup

# chown someuser /recvpool/backup

無權限的使用者現在有權力可以接收並掛載資料集，且home資料集可以被複製到遠端系統:

% zfs snapshot

% zfs send

-reR mypool/home@monday | ssh someuser@backuphost zfs recv

-dvu recvpool/backup

替儲存在儲存池mypool上的檔案系統資料集home製作一個遞迴快照monday，然後使用zfs send -R來傳送包含該資料集及其所有子資料集、快照、複製與設定的串流。輸出會被導向到SSH連線的遠端主機backuphost上等侯輸入的zfs receive，在此建議使用完整網域名稱或IP位置。接收端的機器會寫入資料到recvpool儲存池上的backup資料集，在zfs recv加上-d可覆寫在接收端使用相同名稱的快照，加上-u可讓檔案系統在接收端不會被掛載，當使用-v，會顯示更多有關傳輸的詳細資訊，包含已花費的時間及已傳輸的資料量。
要移除所有配額:

```bash
# zfs set userquota@joe=none
```

使用者配額的屬性不會顯示在 `zfs get all`。

非root的使用者只可以看到自己所屬群組的配額，除非它們有被授予userquota權限，擁有這個權限的使用者可以檢視與設定任何人人的配額。

要設定群組配額的一般格式為:

```
groupquota@group = size
```

要設定群組`firstgroup`的配額為50 GB可使用:

```bash
# zfs set groupquota@firstgroup=50G
```

要移除群組`firstgroup`的配額，或確保該群組未設定配額可使用:

```bash
# zfs set groupquota@firstgroup=none
```

如同使用者配額屬性，非root使用者只可以查看自己所屬群組的配額。而root或擁有groupquota權限的使用者，可以檢視並設定所有群組的任何配額。

要顯示在檔案系統或快照上每位使用者所使用的空間及配額可使用 `zfs userspace`，要取得群組的資訊則可使用 `zfs groupspace`，要取得有關支援的選項資訊或如何只顯示特定選項的資訊請參考 `zfs(1)`。

有足夠權限的使用者及root可以使用以下指令列出 `storage/home/bob`的配額:

```bash
# zfs get quota storage/home/bob
```

保留空間(Reservation)可以確保資料集最少可用的空間，其他任何資料集無法使用保留的空間，這個功能能在要確保有足夠的可用空間來存放重要資料集或日誌檔時特別有用。

reservation屬性的一般格式為 `reservation=size`，所以在 `storage/home/bob`設定保留10 GB的空間可以用:

```bash
# zfs set reservation=10G storage/home/bob
```

要清除任何保留空間:

```bash
# zfs set reservation=none storage/home/bob
```

同樣的原則可以應用在 `refreservation` 屬性來設定參考保留空間(Reference Reservation)，參考保留空間的一般格式為 `refreservation=size`。
複標儲存功能，需在目錄去開池啟要成，而壓縮額能不需要使用體。達記憶外的功能需要使用大量省空間記憶複量改開空來功的大小重，要注意的是功能去壓縮量：啟塊複便連量重區會替的寫入重，這在資料中有大量或資料複代有資料的參考檔來時可以節與現區區來有的資料複的資料重區偵測塊塊，當新的資料重，ZFS會到資料集。被寫入料量足夠於配額功能未壓縮版本況空，可能會有多的資可用。份有但份比備確保是由間更備是的狀壓縮份能的儲存類功能壓縮似備配額功能一限能功題，通常會使用使用時也可能會有量來能在與的資料足夠配額為更改影（性導上。使用者了壓縮達響到的率，因屬致但限一個使用者量增加的資料有然實際沒雖於改變象便的可壓縮資料，奇怪間剩造多或可用的，這可能會是，較麼餘成也會因此而量現更可能有少空的空外的資料。若使用者在之後庫如一個資料新了一個將案，儲存額還，會有更間檔例依但衡是以量的據期的空少製作用。使用者限一會生無一個使用者在一個資料集併上可以使用多配額副間會有更多，使用者在一個資料集上可以使用多配額副間會有更多。
只有要被寫入到儲存池的新資料會做去重複的動作，先前已寫入到儲存池的資料不會因此啟動這個選項而做去重複。

查看已啟用去重複屬性儲存池會如下:

```
# zpool list
NAME  SIZE  ALLOC  FREE   CKPOINT  EXPANDSZ   FRAG  CAP   DEDUP   HEALTH   ALTROOT
pool 2.84G 20.9M 2.82G         -         -     0%   0%   3.00x   ONLINE   -
```

DEDUP欄位會顯示儲存池的實際去重複率，數值為1.00x代表資料尚未去重複。在下一個例子會在前面所建立的去重複儲存池中複製三份Port樹到不同的目錄中。

```
# for d in dir1 dir2 dir3; do
  mkdir $d && cp -R /usr/ports $d
done
```

已經偵測到重複的資料並做去重複:

```
# zpool list
NAME  SIZE  ALLOC  FREE   CKPOINT  EXPANDSZ   FRAG  CAP   DEDUP   HEALTH   ALTROOT
pool 2.84G 20.9M 2.82G         -         -     0%   0%   3.00x   ONLINE   -
```

DEDUP欄位顯示有3.00x的去重複率，這代表已偵測到多份複製的Port樹資料並做了去重複的動作，且只會使用第三份資料所佔的空間。去重複能節省空間的潛力可以非常巨大，但會需要消耗大量的記憶體來持續追蹤去重複的資料區塊。

去重複並非總是有效的，特別是當儲存池中的資料本身並沒有重複時。ZFS可以透過在現有儲存池上模擬開啓去重複功能來顯示可能節省的空間:

```
# zdb -S pool
Simulated DDT histogram:
bucket              allocated                       referenced
______   ______________________________   ______________________________
refcnt   blocks   LSIZE   PSIZE   DSIZE   blocks   LSIZE   PSIZE   DSIZE
------   ------   ------   ------   ------   ------   ------   ------   ------
1    2.58M    289G    264G    264G    2.58M    289G    264G    264G
4    37.6K    692M    276M    276M     170K   3.04G   1.26G   1.26G
8    2.18K   45.2M   19.4M   19.4M    20.0K    425M    176M    176M
16      174   2.83M   1.20M   1.20M    3.33K   48.4M   20.4M   20.4M
32       40   2.17M    222K    222K    1.70K   97.2M   9.91M   9.91M
64        9     56K   10.5K   10.5K      865   4.96M    948K    948K
128        2   9.50K      2K      2K      419   2.11M    438K    438K
256        5   61.5K     12K     12K    1.90K   23.0M   4.47M   4.47M
1K        2      1K      1K      1K    2.98K   1.49M   1.49M   1.49M
```

ZFS 調整 讓調校作都能以最佳項作 狀運個可對各種工的數這裡有,目可以 在面調校19.6.1.

19.6. 主題階有擁權限上擁資料集或其子資料集的使用者有它們權限予在目指定給其他人。若該使用者指派任zfs allow委託權限委託19.5.2.

載位置在檔掛有權限還制避免用案系統。這裡:非掛項另載一個的使用者可以檔一濫設定載,因此需要的 FreeBSD:掛權限予建指定的建到父新資料集會。這裡需要注意子資料集的牽涉可以給立立個使用者的它可以給快照。一個使用中的統計的予份使用者使用功備。可以給快照。一個使用中的統計的予份使用者使用功備。一份使用者可以給目每功錄例家立個使用者的它們摧毀的管理如,若一個資料集,予能。無權限執面全ZFS系統可能行性

19.5.委託管理 jailed的資料集指定到一個可以Jail,而zfs jail19.4.12. ZFS Jail與足夠觀功省效體的間能。去記憶以大大提的空能只有在可以節可複且有高重麼議能,且壓縮開少那,功果可能會功節啟。記憶壓縮能也可體。若資料是可壓縮的,空常間效先透記憶就耗複能。省去空費量功的這個省功上顯複常節儲存成。若在此的能的,因非間為去差達重啟並不能明空比例便動在這。

vfs.zfs.resilver_delay - Number of milliseconds of delay inserted between each I/O during a resilver. To ensure that a resilver does not interfere with the normal operation of the pool, if any other I/O is happening the resilver will delay between each command. This value controls the limit of total IOPS (I/Os Per Second) generated by the resilver. The granularity of the setting is determined by the value of `kern.hz` which defaults to 1000 ticks per second. This setting may be changed, resulting in a different effective IOPS limit. The default value is 2, resulting in a limit of: 1000 ticks/sec / 2 = 500 IOPS. Returning the pool to an Online state may be more important if another device failing could Fault the pool, causing data loss. A value of 0 will give the resilver operation the same priority as other operations, speeding the healing process. The speed of resilver is only limited when there has been other recent activity on the pool, as determined by `vfs.zfs.scan_idle`. This value can be adjusted at any time with `sysctl(8)`.

vfs.zfs.scan_idle - Number of milliseconds since the last operation before the pool is considered idle. When the pool is idle the rate limiting for scrub and resilver are disabled. This value can be adjusted at any time with `sysctl(8)`.

vfs.zfs.txg.timeout - Maximum number of seconds between transaction group s. The current transaction group will be written to the pool and a fresh transaction group started if this amount of time has elapsed since the previous transaction group. A transaction group may be triggered earlier if enough data is written. The default value is 5 seconds. A larger value may improve read performance by delaying asynchronous writes, but this may cause uneven performance when the transaction group is written. This value can be adjusted at any time with `sysctl(8)`.

19.6.2. i386 上的 ZFS ZFS 所提供的部份功能需要使用大量記憶體, 且可能需要對有限 RAM 的系統調校來取得最佳效率。
要取得更多详细关于ZFS相关的建议清单，请参考https://wiki.freebsd.org/ZFSTuningGuide。

19.7. 其他资源
• FreeBSD Wiki - ZFS
• FreeBSD Wiki - ZFS Tuning
• Illumos Wiki - ZFS
• Oracle Solaris ZFS Administration Guide
• Calomel Blog - ZFS Raidz Performance, Capacity and Integrity

19.8. ZFS特色与术语

ZFS是一个从本质上与众不同的文件系统，因为它不仅仅是一个文件系统，ZFS结合了文件系统及磁碟区管理程式，让额外的储存装置可以即时的加入到系统并可让既有的文件系统立即使用这些在储存池中空閒的空间。

透过结合传统的区域分为二的两个角色，ZFS能够克服以往RAID磁碟群组无法扩充的限制。每个在储存池顶层的装置称为vdev，它可以是一个简单的磁碟或是一个RAID如镜像或RAID-Z阵列。

ZFS的文件系统（称为资料集Dataset）每个资料集均可存取整个储存池所共通的可用空间，随着使用储存池来配置空间块，储存池能给每个文件系统使用的可用空间就会减少，这种方法可以避免分割区块会使的可用空间分散分割区块之常见的问题。

储存池(Pool)
储存池(Pool)是建构ZFS最基础的单位。一个储存池可由一个或多个vdev所组成，是用来储存资料的底层装置。储存池会被拿来建立一个或多個文件系统(资料集Dataset)或区块装置(磁碟区Volume)，这些资料集与磁碟区会共用储存池的剩余可用空间。

每一个储存池可由名称与GUID来辨识。可用的功能会依储存池上的ZFS版本而有所不同。
vdev types (vdev Types)

Storage pools are made up of one or more vdevs. vdevs can be a disk or RAID Transform disk group. When using multiple vdevs, ZFS can distribute data across each vdev to improve efficiency and capacity.

- Disk (Disk) - The most basic vdev type is a standard data block device, which can be a whole disk (e.g., /dev/ada0 or /dev/da0) or a partition (/dev/ada0p3). In FreeBSD, using a partition to replace an entire disk will not affect performance, which may differ from Solaris documentation.

- File (File) - Besides disks, ZFS storage pools can use files as a foundation, which is especially useful for testing and experiments. When creating a zpool, use the file's full path as the device path. All vdevs must be at least 128 MB in size.

- Mirror (Mirror) - To create a mirror, use the mirror keyword followed by the list of devices that will be members of the mirror. A mirror needs to be made up of two or more devices, with all data written to all members. Mirror vdevs can recover from a single device failure without losing any data.

- RAID-Z - ZFS implements RAID-Z, modifies standard RAID-5 to improve parity (Parity) for better distribution and removes the "RAID-5 write hole" that occurs outside of the expected period. ZFS supports three levels of RAID-Z, providing different degrees of redundancy to replace or reduce the available space, with the parity device's number and the pool's capacity to tolerate disk failures named, from RAID-Z1 to RAID-Z3.

  In RAID-Z1 configuration with 4 disks, each disk 1 TB, the available storage space is 3 TB, and if one disk fails, it can still be degraded (Degraded) to operate. If the failed disk has not been replaced and resilvered (Resilver) before another disk fails, all data in the storage pool will be lost.

  In RAID-Z3 configuration with 8 1 TB disks, the disk group will be able to provide 5 TB of available space, and it can continue to operate even with 3 disk failures.

  Sun™ recommends not using more than 9 disks per vdev. If the configuration requires more disks, it is recommended to split it into two vdevs, so the pool's data will be distributed to these two vdevs.

  Using two RAID-Z2 vdevs, each containing 8 disks, can create a similar RAID-60 array.

  RAID-Z vdevs pool capacity is close to the smallest disk multiplied by non-parity disks.
資料集
複寫入
製作時
ZIL
L2ARC
委託（及權限的管理）
子資料及父屬
資料集級性可以覆設屬性。
當子資料集的屬父自性繼承父可以的
mypool/home/user
成後的子資料集並且會繼承性。
這可以在繼屬往續一様用像目錄的命採層級，資料集會算池格式。
的根說也儲存部一個資料集,子poolname/path@snapshot
無資料集都有名稱二的
一獨磁碟區案系統、每檔個資複，或資料集（Dataset）做後不需要之外的期預發生關機著謂ZFS入完成的新資料，這也意整內容並不會，只會放寫棄未遺原来檔時，失案的完機案的程中系統當或電過（源）在寫入檔中斷位置新的發生寫入指向完成後才Metadata在資料更寫入會將一個資料寫入新資料會位置塊，而是區到只接ZFS時，不會覆被，當資料要寫直寫舊資料所在覆傳案系統,在統的檔複步的不會用到則製全ZIL像案,完寫入於類同只會有的同步利工這庫作，一般非遲減少延能。增並進效ZIL的空到一儲存寫入量，之後併再間磁碟一般。這可大再寫入到ZIL速度快較)，資料會先寫入寫入到ZIL與增加寫入限即（寫入加速時。

ufs.zfs.l2arc_write_boost
快寫入少控制每有多到可元數位秒
ufs.zfs.l2arc_write_max
限的制速度則個限兩總制會，之後，可以的制為寫入合這（Boost limit）制限與加速在（Write limit）制L2ARC限會限率寫入（到寫入間來騰之前，區空塊）出去除盡快到直取用，速會率制早快多而耗損加入，的資料取裝因寫入次速過過了避免SSD速度要從少為讀取,可以磁碟L2ARC起不比加快，RAM適合放放適合在並不儲存為也同（去加速可以複重樣）檔取的，快從一案讀取L2ARC般磁碟讀取。須速度那SSD選進用的，已是RC些顯態的遲，與速度低較固的延L2ARC的有磁碟快較固碟常傳置片此來為比起SSD）

deduplication
熱階段預可在,

sysctl(8)

Ram

Deduplication

Cache vdev

Cache vdev

SSD

(vdev (Cache vdev))
本複快照（Snapshot）磁碟區檔案系統（File system）間（Quota）與（Reservation）留空保有的量父與子使用的空轉間配響既外的父變為額空由於前的子，這個操作不需要間（被，子。快照為複為父，提本照升本可以複依快除刪快照無法被，因為本會複該為複立區。數則的參考計塊時，在先前資料覆建減會磁碟區會成長，當在被才檔塊案系統或區複的資料本多可以複，而開快照本一案系統可分支讓檔複寫入版本的做，本，複快照本是可製法立照，獨無複只能用或掛製hold。在除，這可快照快作樣才刪hold，而記每個的多個不同名稱可以唯一標錯誤快照，要刪回嘗何該會的任作除動時，便傳試快照hold空會快照區塊被。才回收間也可使用可用這個資料沒的計到任直會數變快照何照，滅有磁碟區區案刪除塊，當檔、資料集或是使用這個資料器，可以用來個參考複記，有多追蹤少快照本持續都會有一儲存做區池後所的變更每。個在塊中的資料指定在之任何來還原，案系統到的（還原原档）先前版本的案，也可以復。可以用讀的方式但掛於快照載這些僅限唯檔隨區案系統而成長，會著在目前停止塊使用的資料這些資料。

(Extent)

ZFS複虛擬化磁區案系統格式時使用層執的檔常有用頂行其他磁碟區非及。要在以驗ZFS與資料集相製複寫入快照時含、功能本複能，包似的、許塊，磁碟區置。裝磁碟區區有（Volume）多立磁碟區也可以建除了一般檔案系統資料集之外，ZFS的擁權限自案與目、有。旗標及檔Metadata的檔某含掛一各且內錄案系統會載在系統目層級處檔案系統，其他的數資料集最常當做檔案系統使用。如同大多ZFS的擁桿權限自案與目、有。旗標及檔Metadata的檔某含掛一各且內錄案系統會載在系統目層級處檔案系統，而其他的數資料集最常當做檔案系統使用。如同大多
配置每个资料区块的同时也会做资料校验，资料校验用的演算法是依资料集属性而有所不同的，请参考 set。

每个资料区块会在读取的过程完成校验，让 ZFS 可以侦测到隐藏的损坏。若资料不符合预期的校验码，ZFS 会尝试从任何可用的备援来还原资料，例如镜像（Mirror）或 RAID-Z。要检验所有资料的校验码可以使用清洁（Scrub），资料校验的演算法有:

* fletcher2
* fletcher4
* sha256

fletcher 演算法最快，而 sha256 虽较消耗效能，但其有强大的密码杂湊与较低的冲突率。也可关闭资料校验，但并不建议。

压缩（Compression）每个资料集都有压缩属性，预设是关闭的，这个属性可以设定使用以下几个压缩演算法的其中一种来压缩写入到资料集的新资料。压缩除了减少空间使用量外，常也会增加读取与写入的吞吐量，因为会减少读取与写入的资料区块。

* LZ4 - ZFS 储存池版本 5000（功能旗标）后所增加，LZ4 现在是建议的压缩演算法，在处理可压缩的资料时 LZ4 压缩比 LZJB 快将近 50%，在处理不可压缩的资料时快将近三倍，LZ4 解压缩也比 LZJB 将近 80%。在现代的 CPU 上，LZ4 经常平均可用 500 MB/s 的速度压缩，而解压缩可到达 1.5 GB/s (每个 CPU 核心)。
* LZJB - 预设的压缩演算法。由 Jeff Bonwick 所开发（ZFS 的创始人之一）。LZJB 与 GZIP 相比，可以较低的 CPU 提供较好的压缩性能。在未来的预设的压缩演算法将会更換为 LZ4。
* GZIP - 在 ZFS 可用的热门串流压缩演算法。使用 GZIP 主要的优点之一便是可设定压缩层级。当设定 compress 属性，管理者可以选择压缩层级范围从最低的压缩层级 gzip1 到最高的压缩层级 gzip9。这能让管理者可以控制要使用多少 CPU 来节省磁碟空间。
* ZLE - 零长度编号是一个特殊的压缩演算法，它只会压缩连续的零。这种压缩演算法只在资料集中含有大量为零的资料区块时有用。
清潔去備份，確保作可以做動至某資料區會在每月底後並不需要作，清潔建動至池可以使用特殊性存別置用裝儲存以避免，如此藏能確保那作可以做，而塊碼個資料區會在不定中校已驗驗，已驗知碼碼中Metadata儲存根據儲存會讀取所有中的資料，檢的致性有ZFS體。記憶使用壓縮也能節近相量額衡功為能，因取代此功慮使用壓縮較體及會影響能有很大的功對響磁碟來讀取實際導實並不致在，足夠記憶的要有於資料需要使用1 TB是驗個，所以會唯儲存每消大塊湊個資料需要使用DDT塊別儲存兩於個資料相同，若資料不同會註與記湊位查兩的資料則會一個一個個資料來相同，若認為碼只要調校來的功碼加密雜全，去區在校清的會計有資算然後既否塊新資料時，參考位置清碼單重偵測記憶空參考區節塊有，完塊讓，使用去寫入重偵測時可以再失損整池法個原儲存損壞中，磁碟但份備情或其他復料的能可以做備援上儲存有沒外的以時可性數上設定塊(每案系統個在備的值時，數於1)vfs.zfs.scrub_delay fsck(8)
上線
修復
参考留空間保
留空資料集保間群
留空資料集保間群
配額使用者配額
參考資料集配額
置於線上個(裝別處態置正狀作。運連結時代表所有該裝員均已成常且vdev一個池或處於線上儲存(稱作這個程序算的資料到新的失寫入計遺訊來磁碟機散在其他遺偶資失磁碟上驗奇的校且當有磁碟更存換必須回磁碟後,新的故障先前所被空保算額用,所以也不會留空間。計間佔裔列入並不會,主資料集的後間空功足夠完成這必須項要有能成間空的之外才
storage/home/bob
留空列入間作一個計保如,若要製。算後間不同的是,資料集所使用的空並不會由裔快照及間空供這個資料集使用。常於正留保,會至10 GB保試留空的空剩使用所有嘗空的時餘storage/home/bob裔,這代表若在空最小間其後後保境多留空不管很有用,如:那一種許在要間後如:快照,但運屬是他作,似的方式類性算。
間快照計間列入所使用的,該的快照保也會被也算。而這個硬性來間制只包限量資料集能使用的快照及集空間資料集後包制裔配額快照會資料集的含、子資料集及限
the dataset quota, the ZFS supports different types of quotas: the user quota (userquota), group quota (groupquota), and the reference quota (refquota). In ZFS, setting quotas on a volume (volsize), because setting quotas on a volume means that the users can only use the quota within that volume. Additionally, the reference quota (refreservation) is used to ensure that references to a dataset are kept within the allocated space, and it is immutable. This quota cannot be set to zero, and it is useful for ensuring that datasets are not accidentally deleted by users.
若有足夠的備援可避免儲存池或vdev進入故障（Faulted）狀態，個別裝置若可由管理者設為離線（Offline）狀態，管理者可以選擇要設定那一個磁碟為離線來準備更換或是讓其更容易辨識。

降級（Degraded）一個儲存池或vdev處於降級（Degraded）狀態代表其有一個或多個磁碟已斷線或故障，此時儲存池仍可以使用,但只要再有其他的裝置故障，儲存池會無法復原原。重新連線缺少的裝置或更換故障的磁碟,並在新裝置完成修復（Resilver）程序可讓儲存池返回線上（Online）狀態。

故障（Faulted）一個儲存池或vdev處於故障（Faulted）狀態代表無法運作,會無法存取在該裝置上的資料。當在vdev中缺少或故障的裝置數超過備援的層級,儲存池或vdev會進入故障（Faulted）狀態。若缺少的裝置可以重新連結上,儲存池便會返回線上（Online）狀態。若沒有足夠的備援可補償故障的磁碟數量就會遺失儲存池中的內容且只能從備份還原。
Chapter 20.

20.1. Overview

File systems are an integral part of any operating system. They allow users to upload and store files, provide access to data, and make hard drives useful. Different operating systems differ in their native file system. Traditionally, the native FreeBSD file system has been the Unix File System UFS which has been modernized as UFS2. Since FreeBSD 7.0, the Z File System (ZFS) is also available as a native file system. See Z档案系统 (ZFS) for more information.

In addition to its native file systems, FreeBSD supports a multitude of other file systems so that data from other operating systems can be accessed locally, such as data stored on locally attached USB storage devices, flash drives, and hard disks. This includes support for the Linux™ Extended File System (EXT).

There are different levels of FreeBSD support for the various file systems. Some require a kernel module to be loaded and others may require a toolset to be installed. Some non-native file system support is full read-write while others are read-only.

In this chapter, you will learn:

- The difference between native and supported file systems.
- Which file systems are supported by FreeBSD.
- How to enable, configure, access, and make use of non-native file systems.

Before starting to read this chapter, you need to:

- Understand UNIX™ and FreeBSD basics.
- Be familiar with the basics of kernel configuration and compilation.
- Feel comfortable installing software in FreeBSD.
- Have some familiarity with disks, storage, and device names in FreeBSD.

20.2. Linux™ 檔案系統

FreeBSD provides built-in support for several Linux™ file systems. This section demonstrates how to load support for and how to mount the supported Linux™ file systems.

20.2.1. ext2

Kernel support for ext2 file systems has been available since FreeBSD 2.2. In FreeBSD 8.x and earlier, the code is licensed under the GPL. Since FreeBSD 9.0, the code has been rewritten and is now BSD licensed.

The ext2fs(5) driver allows the FreeBSD kernel to both read and write to ext2 file systems. This driver can also be used to access ext3 and ext4 file systems. The ext2fs(5) filesystem has full read and write support for ext4 as of FreeBSD 12.0-RELEASE. Additionally, extended attributes and ACLs are also supported, while journalling and encryption are not. Starting with FreeBSD 12.1-RELEASE, a DTrace provider will be available as well. Prior versions of FreeBSD can access ext4 in read and write mode using sysutils/fusefs-ext2.

To access an ext file system, first load the kernel loadable module:

```
# kldload ext2fs
```
Then, mount the ext volume by specifying its FreeBSD partition name and an existing mount point. This example mounts 
\(/\text{dev/ad1s1}\) on 
\(/\text{mnt}\):

```
# mount -t ext2fs /dev/ad1s1 /mnt
```
21.1. 概述

虛擬化軟體可以讓同一台機器得以同時執行多種作業系統。在 PC 上的這類軟體系統通常涉及的角色有執行虛擬化軟體的主端（Host）作業系統及數個安裝在其中的客端（Guest）作業系統。

讀完這章，您將了解：

- 主端作業系統及客端作業系統的差別。
- 如何在 Intel™-based Apple™ Mac™ 電腦安裝 FreeBSD。
- 如何在 Microsoft™ Windows™ 使用 Virtual PC 安裝 FreeBSD。
- 如何以 FreeBSD 為客端安裝在 bhyve。
- 如何調校 FreeBSD 系統來取得虛擬化的最佳效能。

在開始閱讀這章之前，您需要了解：

- UNIX™ 與 FreeBSD 的基礎。
- 知道如何安裝 FreeBSD。
- 知道如何設定網路連線。
- 知道如何安裝其他第三方軟體。

21.2. 在 Mac OS™ X 的 Parallels 安裝 FreeBSD

在 Mac OS™ X 的 Parallels Desktop 是一套商業軟體可在 Intel™ 為基礎的 Apple™ Mac™ 的 Mac OS™ 10.4.6 或更新版本上執行。該軟體完全支援使用 FreeBSD 為客端作業系統。在 Mac OS™ X 裝好 Parallels 後，使用者必須完成虛擬機器的設定後才可安裝想使用的客端作業系統。

21.2.1. 在 Parallels/Mac OS™ X 安裝 FreeBSD

在 Parallels 上安裝 FreeBSD 的第一步是建立供安裝 FreeBSD 使用的新虛擬機器。提示出現後請選擇 Guest OS Type 為 FreeBSD：
根據您對此虛擬 FreeBSD 作業系統的規畫選擇合理的磁碟及記憶體空間，對大多數在 Parallels 下的 FreeBSD 使用來講 4GB 的磁碟空間與 512MB 的 RAM 便足夠。
Click the lock to prevent further changes.
Virtual Machine Configuration

FreeBSD

CPUs: 2
Memory: 256 MB
Disk space: 8 GB

Configure...
在FreeBSD虛擬機器新設後，就可以繼續以其安裝FreeBSD。安裝方面，比較好的作法是使用官方的FreeBSD CD/DVD或者是自官方FTP站下載的ISO映像檔。複製適合的ISO映像檔到Mac™檔案系統本地端或放入CD/DVD到Mac™的CD-ROM磁碟機。在FreeBSD Parallels視窗的右下角點選磁碟圖示後會出現一個視窗，可用來建立虛擬機器內的CD-ROM磁碟機與磁碟上的ISO檔案或實際CD-ROM磁碟機的關聯。

建立與CD-ROM來源的關聯後，點選重新開機圖示重新開啟FreeBSD虛擬機器。
會重開機進入一個特殊的BIOS畫面並檢查是否有CD-ROM。

在此處會找到FreeBSD安裝媒體並開始正常的FreeBSD安裝程序。完成安裝，但不要在此時嘗試設定Xorg。

當安裝完成後，重開機將會進入新安裝的FreeBSD虛擬機器。
21.2.2. In Parallels setting FreeBSD in a successful manner will install FreeBSD into Mac OS™ X. After installation, there are several steps to complete to optimize the system in the virtual machine.

1. Setting Boot Loader

   The most important step is reducing the kern.hz parameter to reduce FreeBSD's CPU usage when running in the Parallels environment. Add the following line to /boot/loader.conf to complete this action:

   ```
kern.hz=100
   ```

   If this setting is not completed, the FreeBSD Parallels client will consume approximately 15% of the CPU on an iMac™. After this change is applied, the usage rate will decrease to nearly 5%.

2. Building a New Core Configuration

   All SCSI, FireWire, and USB devices can be removed from the custom core configuration. Parallels’ virtual network cards use ed(4) driver programs, so besides ed(4) and miibus(4), all network devices can be removed from the core.

3. Setting Network

   The simplest network setting is using DHCP to connect the virtual machine to the same subnet as the host Mac™. This can be done by adding the following line to /etc/rc.conf:

   ```
   ifconfig_ed0="DHCP"
   ```

   More advanced network settings are described in Advanced Network Configuration.
21.3. 在 Windows™ 的 Virtual PC 安装 FreeBSD 为客端

给 Windows™ 使用的 Virtual PC 是一套可免费下载的 Microsoft™ 软体产品，请参考此网站取得系统需 求。

Virtual PC 在 Microsoft™ Windows™ 上安装完成之后，使用者可以设定一台虚拟机器然后安装想要的客端作业系统。

21.3.1. 在 Virtual PC 安装 FreeBSD

安装 FreeBSD 到 Virtual PC 的第一个步骤是建立新的虚拟机器来安装 FreeBSD。当提示画面出现时，请选择 Create a virtual machine:

- 当提示画面出现时，选择 Operating system 为 Other:
  - Add an existing virtual machine
    - You can add a virtual machine to the Virtual PC Console from existing .vme files.
然後，根據您對此虛擬 FreeBSD 作業系統的規畫選擇合理的磁碟及記憶體空間，對大多數在 Virtual PC 下的 FreeBSD 來說 4GB 的磁碟空間與 512MB 的 RAM 即足夠。
Storing and completing the settings:

- Choose FreeBSD virtual machine then select Settings, and then set network type as well as network card:

  - An existing virtual hard disk
  - A new virtual hard disk

New Virtual Machine Wizard

Virtual Hard Disk Options
Before you can install an operating system on this virtual machine, you must add a new or existing virtual hard disk to it.

A virtual hard disk is a .vhd file that is stored on your physical hard disk and is used to contain the guest operating system, applications, and data files.

The first virtual hard disk you create or select for your virtual machine is called Hard Disk 1 in Settings and is the startup disk.

Do you want to use:

- [ ] An existing virtual hard disk
- [ ] A new virtual hard disk

New Virtual Machine Wizard

Virtual Hard Disk Location
This wizard creates a dynamically expanding virtual hard disk with the specified size.

A name for the new virtual hard disk. Unless you specify a different location, the virtual hard disk file will automatically be saved in the same location as the virtual machine configuration file.

Name and location:

```
achines\FreeBSD-CURRENT\FreeBSD-CURRENT Hard Disk.vhd
```

Browse...

Maximum virtual hard disk size: 130,557 MB
Virtual hard disk size: 4,000 MB

To learn more about the different types of virtual hard disks, see Virtual PC Help. For advanced virtual hard disk options, use the Virtual Disk Wizard.
FreeBSD 虛擬機器建立完成之後，便可安裝 FreeBSD 到該虛擬機器。安裝最 好使用官方 FreeBSD CD/DVD 或使用官方 FTP 站下載的 ISO 映像檔。複製適當的 ISO 映像檔到本地 Windows™ 檔案系統或插入 CD/DVD 到 CD 磁碟機，然後雙擊點選 FreeBSD 虛擬機器來開機。接著，點選 CD 並在 Virtual PC 視窗選擇 Capture ISO Image…，這將會顯示一個視窗可以建立虛擬機器中的 CD-ROM 與 ISO 檔或磁碟或實體 CD-ROM 磁碟機之間的關聯。
建立與CD-ROM來源的關聯後，點選Action及Reset重新開機FreeBSD虛擬機器。Virtual PC會重新開始並進入特殊的BIOS來做CD-ROM的第一次檢查。
在这种情况会找到FreeBSD安装媒体然后开始正常的FreeBSD安装。接着继续安装，但此时请不要尝试设置Xorg。
當安裝完成之後，記得退出 CD/DVD 或釋放 ISO 映像檔。最後，重新開機進入新安裝的 FreeBSD 虛擬機器。
21.3.2. In Virtual PC, FreeBSD can be installed successfully on Microsoft™ Windows™. There are several steps to be completed to optimize system operation on the virtual machine. 

1. Set Boot Loader variable. One of the most important steps is to reduce `kern.hz`, thereby reducing the CPU usage of FreeBSD in the Virtual PC environment. This can be achieved by adding the following lines to `/boot/loader.conf`:

```
kern.hz=100
```

If this setting is not completed, the FreeBSD Virtual PC guest OS will consume 40% of the CPU. After this change, the usage rate would drop to approximately 3%.

2. Create new kernel configuration. All SCSI, FireWire, and USB devices can be removed from the custom kernel configuration. The virtual network card provided by Virtual PC is driven by `de(4)`, so除了`de(4)` and `miibus(4)`外, all network devices can be removed from the kernel.

3. Set network. The most basic network setting is to use DHCP to connect the virtual machine to the same network as the main Microsoft™ Windows™ network, which can be done by adding `ifconfig_de0="DHCP"` to `/etc/rc.conf`.

Advanced network settings are described in the advanced network settings.
21.4. 在 Mac OS™ 的 VMware Fusion 安装 FreeBSD 为客户端。VMware Fusion 是一套商业软件可在 Intel™ 为基础的 Apple™ Mac™ 的 Mac OS™ 10.4.9 或更新版本上执行。该软件完全支援使用 FreeBSD 作为客户端作业系统。在 Mac OS™ X 装好 VMware Fusion 后，使用者必须完成虚拟机的设定后才可安装想使用的客户端作业系统。

21.4.1. 在 VMware Fusion 安装 FreeBSD 第一步骤是启动 VMware Fusion 载入 Virtual Machine Library，点选 New 建立虚拟机器：这个动作会载入 New Virtual Machine Assistant，点选 Continue 继续。
Choose Operating System
Select the operating system to be used in this virtual machine.

Select the operating system for this virtual machine:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Windows</td>
<td>eComStation2</td>
</tr>
<tr>
<td>Linux</td>
<td>eComStation</td>
</tr>
<tr>
<td>Apple OS X</td>
<td>FreeBSD 12 64-bit</td>
</tr>
<tr>
<td>VMware ESX</td>
<td>FreeBSD 12</td>
</tr>
<tr>
<td>Other</td>
<td>FreeBSD 11 64-bit</td>
</tr>
<tr>
<td></td>
<td>FreeBSD 11</td>
</tr>
<tr>
<td></td>
<td>FreeBSD version 10 and earlier 64-bit</td>
</tr>
<tr>
<td></td>
<td>FreeBSD version 10 and earlier</td>
</tr>
</tbody>
</table>
Choose Firmware Type

Select the firmware type to be used to boot this virtual machine.

Specify the boot firmware:

- [ ] Legacy BIOS
- [x] UEFI
- [ ] UEFI Secure Boot
Choose a Virtual Disk
Select a virtual disk to be used with this virtual machine.

Choose a virtual disk option:

- [ ] Create a new virtual disk
- [ ] Use an existing virtual disk

Choose virtual disk...

Guest OS: FreeBSD 12 64-bit
Option: New Hard Disk
Capacity: 20 GB
Finish
The configuration of the virtual machine is now complete.

Virtual Machine Summary
Guest Operating System FreeBSD 12 64-bit
New Hard Disk Capacity 20 GB
Memory 256 MB
Networking Share with my Mac (NAT)
Device Summary CD/DVD, USB Controller, Sound Card

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.
Finish
The configuration of the virtual machine is now complete.

Save As: FreeBSD 12 64-bit.vmwarevm
Tags: 
Where: Virtual Machines

Share this virtual machine with other users on this Mac
Some features will be limited when sharing a virtual machine. Sharing is only available when the virtual machine is saved in a shared folder.

Cancel  Save

To change the default virtual machine settings, click Customize Settings. To run the virtual machine now, click Finish.

Customize Settings  Go Back  Finish
安装完成后，可以修改虚拟机的设定，例如记忆体使用量。

虚拟机要使用的CPU数量：479

Connect CD/DVD Drive

这个CD/DVD驱动已配置为使用以下文件：

FreeBSD-13.0-RELEASE-amd64-dvd1.iso

Advanced options
最後一件

事

是

更改

虛擬

機

器

連

線

到

網

路

的

方式,要

允

許

除

了主

端

以外的

機

器

連

線

到

虛擬

機

器,請

選

擇

Connect directly to the physical network (Bridged)。

否

則

會

偏好

使用

Share the host's internet connection (NAT)

來讓虛擬機

器可以

存

取

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際

際
在修
改設定
之後，
開機
進
入
新安裝的
FreeBSD
虛擬機
器。

21.4.2. 在
VMware Fusion
設定
FreeBSD
在成
功
將
FreeBSD
安裝到
Mac OS™ X
的
VMware Fusion
後，有
數
個
設定
步
驟
要完成
來
最
佳化
系統在
虛擬機
器
上
的
運
作。

1. 設定
Boot Loader
變數
最重要的一個步
驟
是
減
少
kern.hz，
來
減
少
FreeBSD
在
VMware Fusion
環境
下
CPU
的使用
量。
這可以
透
過加入
下
列幾
行到
\( /boot/loader.conf \)
來
完成
：

```
kern.hz=100
```

若
沒
有完成此
設定
，
閒
置
的
FreeBSD VMware Fusion
客端
將
會
消
耗掉
單
一
處
理器的
iMac™
將
近
15%
的
CPU。
完成此
更
改
後使用
率
會
減
至接
近
5%。

2. 建
立
新
核心設定檔
所有的
SCSI, FireWire
及
USB
裝
置
可以從自訂的
核心設定檔
中
移
除
。VMware Fusion
提供的
虛擬
網
路
卡
使用
em(4)
驅
動
程式，所
以
除
了
em(4)
外的所有網
路
裝
置
可以自
核心
中
移
除
。

3. 設定
網
路
最基本的網
路設定
是使用
DHCP
來讓虛擬機
器
連
線
到與主
端
Mac™
相同的
區
域
網
路，
這可以
透
過加入
\( \text{ifconfig}_\text{em0} = \text{"DHCP"} \)
到
\( \text{etc/rc.conf} \)
來
完成。

更
進
階
的
網
路設定
在
進
階
網
路設定
中
描述
。
做為客端
系統也可
運作的很
好，虛擬化軟
體可支援
最常見的幾
個作業
系統，這當然也包
含 FreeBSD。

VirtualBox™ guest additions
支援以下功能：
- 剪貼簿共享。
- 滑鼠指標整合。
- 主機時間同步。
- 視窗縮放。
- 無痕模式。

以下指令均是在 FreeBSD 客端中執行。

首先，在 FreeBSD 客端安裝 emulators/virtualbox-ose-additions 套件或 Port，以下指令會安裝 Port：

```
# cd /usr/ports/emulators/virtualbox-ose-additions && make install clean
```

加入下行到 /etc/rc.conf：

```
vboxguest_enable="YES"
vboxservice_enable="YES"
```

若有使用 ntpd(8) 或 ntpdate(8)，便可關閉主機時間同步功能：

```
vboxservice_flags="--disable-timesync"
```

Xorg 會自動辨識 vboxvideo 驅動程式，也可手動在 /etc/X11/xorg.conf 中輸入：

```
Section "Device"
  Identifier "Card0"
  Driver "vboxvideo"
  VendorName "InnoTek Systemberatung GmbH"
  BoardName "VirtualBox Graphics Adapter"
EndSection
```

要使用 vboxmouse 驅動程式，可調整在 /etc/X11/xorg.conf 中與滑鼠相關的一節：

```
Section "InputDevice"
  Identifier "Mouse0"
  Driver "vboxmouse"
EndSection
```

HAL 的使用者應建立以下 /usr/local/etc/hal/fdi/policy/90-vboxguest.fdi 或複製自 /usr/local/shared/hal/fdi/policy/10osvendor/90-vboxguest.fdi：

```
<?xml version="1.0" encoding="utf-8"?>
```
The vboxmouse driver is configured by adding an entry to the VirtualBox configuration file. Here is an example of the vboxmouse configuration:

```xml
<deviceinfo version="0.2">
    <device>
        <match key="info.subsystem" string="pci">
            <match key="info.product" string="VirtualBox guest Service">
                <append key="info.capabilities" type="strlist">input</append>
                <append key="info.capabilities" type="strlist">input.mouse</append>
                <merge key="input.x11_driver" type="string">vboxmouse</merge>
                <merge key="input.device" type="string">/dev/vboxguest</merge>
            </match>
        </match>
    </device>
</deviceinfo>
```

Shared folders for file transfers between host and VM are accessible by mounting them using `mount_vboxvfs`. A shared folder can be created on the host using the VirtualBox GUI or via `vboxmanage`. For example, to create a shared folder called `myshare` under `/mnt/bsdboxshare` for the VM named `BSDBox`, run:

```bash
# vboxmanage sharedfolder add 'BSDBox' --name myshare --hostpath /mnt/bsdboxshare
```

Note that the shared folder name must not contain spaces. Mount the shared folder from within the guest system like this:

```bash
# mount_vboxvfs -w myshare /mnt
```
以 FreeBSD 作为主端使用 VirtualBox™

VirtualBox™ 是一套积极开发、完整的虚拟化套件，适用大多商业系统，包含 Windows™, Mac OS™, Linux™ 与 FreeBSD，它同样能够执行类 Windows™ 或 UNIX™ 的客户端系统。它是以开源软体的方式发布，但开源元件可在扩充包中使用，这些元件包含对 USB 2.0 设备的支持。更多资讯可在 VirtualBox wiki 的 Downloads 页面。目前，这些扩充套件并不支援 FreeBSD。

21.6.1. 安装 VirtualBox™

VirtualBox™ 可於 emulators/virtualbox-ose 以 FreeBSD 套件或 Port 的方式取得。要安装 Port 可使用以下指令:

```bash
# cd /usr/ports/emulators/virtualbox-ose
# make install clean
```

在 Port 的设定选单中 GuestAdditions 相关程式是最有用的选项之一，这些程式可在客户端作業系统提供数个有用的功能，如滑鼠指针整合（允许滑鼠在主端与客户端之间移动，不需要按特殊快速键来切换）与较快的影像绘图速度，特别是在 Windows™ 的客户端系统。

Guest additions 可在客户端系统安装完之后的 Devices 选单中找到。

还有一些设定需要在 VirtualBox™ 第一次启动时做修改，Port 会安装一个核心模组在 /boot/modules，该模组必须在核心中载入:

```bash
# kldload vboxdrv
```

要确保该模组在重新开机后会载入，可加入下行到 /boot/loader.conf:

```bash
vboxdrv_load="YES"
```

要使用可支援桥接或仅限主端（Host-only）的网路，可加入下行到 /etc/rc.conf，然后重新启动电脑:

```bash
vboxnet_enable="YES"
```

在安装 VirtualBox™ 的过程中会建立 vboxusers 群组，所有需要存取 VirtualBox™ 的使用者均需要加入成为此群组的成员，`pw` 可用來加入新的成员:

```bash
# pw groupmod vboxusers -m yourusername
```

`/dev/vboxnetctl` 的预设权限是受限的，需要更改後才可使用桥接网路:

```bash
# chown root:vboxusers /dev/vboxnetctl
# chmod 0660 /dev/vboxnetctl
```

要永久变更权限，可加入下列几行到 /etc/devfs.conf:

```bash
own vboxnetctl root:vboxusers
```

要变更权限，可加入下列几行到 /etc/devfs.conf:

```bash
own vboxnetctl root:vboxusers
```
要执行VirtualBox™，可在Xorg工作阶段输入:

```
% VirtualBox
```

要取得更多有关设定与使用VirtualBox™的资讯，请参考官方网站。供FreeBSD特定的资讯与疑難排解操作指示，可参考FreeBSD wiki中相关页面。

### 21.6.2. VirtualBox™ USB支援

VirtualBox™可以配置通过USB设备到来宾操作系统。主机控制器的OSE版本是有限的，直到扩展包支持USB 2.0和3.0设备在FreeBSD。

为了让VirtualBox™意识到连接到机器的USB设备，用户需要是operator组的成员。

```
# pw groupmod operator -m yourusername
```

然后，将以下内容添加到`/etc/devfs.rules`，或者如果它还未存在，则创建此文件：

```
[system=10]
add path 'usb/*' mode 0660 group operator
```

若服务未执行，请加入下行到`/etc/rc.conf`:

```
devfs_system_ruleset="system"
```

然后重新启动devfs:

```
# service devfs restart
```

重新启动工作阶段与VirtualBox™来让这些变更生效，并建立必要的USB过滤器。

### 21.6.3. VirtualBox™ Host DVD/CD存取

透过共享实体磁碟机可以让客户端系统能够存取主机系统的DVD/CD磁碟机。在VirtualBox™中，这个功能可在虚拟机器设定中的储存(Storage)视窗中设定。若需要，可先建立一个空的IDECD/DVD装置，然后在跳出的选单中选择要用作虚拟CD/DVD磁碟机的主机磁碟机，此时会出现一个标签为Passthrough的核选方块，勾选此核选方块可以让虚拟机器直接使用该硬体，例如，音乐CD或烧录机只会在勾选此选项时能够运作。

VirtualBox™DVD/CD功能要能运作需要执行HAL，因此需在`/etc/rc.conf`中开启，若该服务尚未启动，则启动它:

```
hald_enable="YES"
```

486
為了讓使用者能夠使用 VirtualBox™ DVD/CD 功能，這些使用者需要存取 /dev/xpt0, /dev/cdN 以及 /dev/passN，這通常讓這些使用者成為 operator 的成員來達成。對這些裝置的權限必須加入下行到 /etc/devfs.conf 來修正：

```
perm cd* 0660
perm xpt0 0660
perm pass* 0660
```

```
# service devfs restart
```

### 21.7.

以 FreeBSD 為主端安裝 bhyve

The bhyve BSD-licensed hypervisor became part of the base system with FreeBSD 10.0-RELEASE.

This hypervisor supports a number of guests, including FreeBSD, OpenBSD, and many Linux™ distributions. By default, bhyve provides access to serial console and does not emulate a graphical console. Virtualization offload features of newer CPUs are used to avoid the legacy methods of translating instructions and manually managing memory mappings.

The bhyve design requires a processor that supports Intel™ Extended Page Tables (EPT) or AMD™ Rapid Virtualization Indexing (RVI) or Nested Page Tables (NPT). Hosting Linux™ guests or FreeBSD guests with more than one vCPU requires VMX unrestricted mode support (UG). Most newer processors, specifically the Intel™ Core™ i3/i5/i7 and Intel™ Xeon™ E3/E5/E7, support these features. UG support was introduced with Intel’s Westmere micro-architecture. For a complete list of Intel™ processors that support EPT, refer to [https://ark.intel.com/content/www/us/en/ark/search/featurefilter.html?productType=873&0_ExtendedPageTables=True](https://ark.intel.com/content/www/us/en/ark/search/featurefilter.html?productType=873&0_ExtendedPageTables=True). RVI is found on the third generation and later of the AMD Opteron™ (Barcelona) processors. The easiest way to tell if a processor supports bhyve is to run `dmesg` or look in `/var/run/dmesg.boot` for the `POPCNT` processor feature flag on the `Features2` line for AMD™ processors or `EPT` and `UG` on the `VT-x` line for Intel™ processors.

#### 21.7.1. 備份主端

The first step to creating a virtual machine in bhyve is configuring the host system. First, load the bhyve kernel module:

```
# kldload vmm
```

Then, create a tap interface for the network device in the virtual machine to attach to. In order for the network device to participate in the network, also create a bridge interface containing the tap interface and the physical interface as members. In this example, the physical interface is igb0:

```
# ifconfig tap0 create
# sysctl net.link.tap.up_on_open=1
net.link.tap.up_on_open: 0 -> 1
# ifconfig bridge0 create
# ifconfig bridge0 addm igb0 addm tap0
```
Create a file to use as the virtual disk for the guest machine. Specify the size and name of the virtual disk:

```
# truncate -s 16G guest.img
```

Download an installation image of FreeBSD to install:

```
# fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/10.3/FreeBSD-10.3-RELEASE-amd64-bootonly.iso
```

FreeBSD comes with an example script for running a virtual machine in bhyve. The script will start the virtual machine and run it in a loop, so it will automatically restart if it crashes. The script takes a number of options to control the configuration of the machine:

- `-c` controls the number of virtual CPUs,
- `-m` limits the amount of memory available to the guest,
- `-t` defines which tap device to use,
- `-d` indicates which disk image to use,
- `-i` tells bhyve to boot from the CD image instead of the disk, and
- `-I` defines which CD image to use. The last parameter is the name of the virtual machine, used to track the running machines. This example starts the virtual machine in installation mode:

```
# sh /usr/shared/examples/bhyve/vmrun.sh -c 1 -m 1024M -t tap0 -d guest.img -i -I FreeBSD-10.3-RELEASE-amd64-bootonly.iso guestname
```

The virtual machine will boot and start the installer. After installing a system in the virtual machine, when the system asks about dropping in to a shell at the end of the installation, choose `[Yes]`.

Reboot the virtual machine. While rebooting the virtual machine causes bhyve to exit, the `vmrun.sh` script runs `bhyve` in a loop and will automatically restart it. When this happens, choose the reboot option from the boot loader menu in order to escape the loop. Now the guest can be started from the virtual disk:

```
# sh /usr/shared/examples/bhyve/vmrun.sh -c 4 -m 1024M -t tap0 -d guest.img guestname
```

In order to boot operating systems other than FreeBSD, the `sysutils/grub2-bhyve` port must be first installed. Next, create a file to use as the virtual disk for the guest machine:

```
# truncate -s 16G linux.img
```

Starting a virtual machine with bhyve is a two step process. First a kernel must be loaded, then the guest can be started. The Linux™ kernel is loaded with `sysutils/grub2-bhyve`. Create a `device.map` that grub will use to map the virtual devices to the files on the host system:
Use `sysutils/grub2-bhyve` to load the Linux™ kernel from the ISO image:

```
# grub-bhyve -m device.map -r cd0 -M 1024M linuxguest
```

This will start grub. If the installation CD contains a `grub.cfg`, a menu will be displayed. If not, the `vmlinuz` and `initrd` files must be located and loaded manually:

```
grub>
ls (hd0) (cd0) (cd0,msdos1) (host) (lvm/VolGroup-lv_swap) (lvm/VolGroup-lv_root)
grub>
ls (hd0,msdos1) /
lost+found/ grub/ efi/ System.map-2.6.32-431.el6.x86_64 config-2.6.32-431.el6.x86_64 symvers-2.6.32-431.el6.x86_64.gz vmlinuz-2.6.32-431.el6.x86_64
initramfs-2.6.32-431.el6.x86_64.img
```

```
grub> linux (hd0,msdos1) /vmlinuz-2.6.32-431.el6.x86_64
```

Now that the Linux™ kernel is loaded, the guest can be started:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-blk,./linux.img
```

```
-s
```

The system will boot and start the installer. After installing a system in the virtual machine, reboot the virtual machine. This will cause bhyve to exit. The instance of the virtual machine needs to be destroyed before it can be started again:

```
# bhyvectl --destroy --vm=linuxguest
```

Now the guest can be started directly from the virtual disk. Load the kernel:

```
# grub-bhyve -m device.map -r hd0,msdos1 -M 1024M linuxguest
```

```
grub>
ls (hd0) (hd0,msdos2) (hd0,msdos1) (cd0) (cd0,msdos1) (host) (lvm/VolGroup-lv_swap) (lvm/VolGroup-lv_root)
```

```
```
grub> initrd
(hd0,msdos1)/initramfs-2.6.32-431.el6.x86_64.img
grub> boot

Boot the virtual machine:
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 \\
- s 3:0,virtio-blk,./linux.img
- l com1,stdio
- c 4
- m 1024M linuxguest

Linux™ will now boot in the virtual machine and eventually present you with the login prompt.

Login and use the virtual machine. When you are finished, reboot the virtual machine to exit bhyve.

Destroy the virtual machine instance:
# bhyvectl --destroy --vm=linuxguest

21.7.4.

使用
UEFI
韌
體
開機

bhyve
虛擬機
器

In addition to bhyveload and grub-bhyve, the bhyve hypervisor can also boot virtual machines using the UEFI userspace firmware. This option may support guest operating systems that are not supported by the other loaders.

In order to make use of the UEFI support in bhyve, first obtain the UEFI firmware images. This can be done by installing sysutils/bhyve-firmware port or package.

With the firmware in place, add the flags -l bootrom, /path/to/firmware to your bhyve command line. The actual bhyve command may look like this:
# bhyve -AHP -s 0:0,hostbridge -s 1:0,lpc \\
- s 2:0,virtio-net,tap1 \\
- s 3:0,virtio-blk,./disk.img \\
- s 4:0,ahci-cd,./install.iso \\
- c 4
- m 1024M \\
- l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI.fd

guest sysutils/bhyve-firmware also contains a CSM-enabled firmware, to boot guests with no UEFI support in legacy BIOS mode:
# bhyve -AHP -s 0:0,hostbridge -s 1:0,lpc \\
- s 2:0,virtio-net,tap1 \\
- s 3:0,virtio-blk,./disk.img \\
- s 4:0,ahci-cd,./install.iso \\
- c 4
- m 1024M \\
- l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI_CSM.fd

guest

21.7.5.

供
bhyve
客端
用
的
圖
型
化
UEFI Framebuffer

The UEFI firmware support is particularly useful with predominantly graphical guest operating systems such as Microsoft Windows™.

Support for the UEFI-GOP framebuffer may also be enabled with the -s 29,fbuf,tcp=0.0.0.0:5900 flags. The framebuffer resolution may be configured with w=800 and h=600, and bhyve can be instructed to wait for a VNC connection before booting the guest by adding wait.
may be accessed from the host or over the network via the VNC protocol.

bhyve指令的结果会如下:

```bash
# bhyve -AHP -s 0:0,hostbridge -s 31:0,lpc -s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img -s 4:0,ahci-cd,./install.iso -c 4 -m 1024M -s 29,fbuf,tcp -l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI.fd
```

Note, in BIOS emulation mode, the framebuffer will cease receiving updates once control is passed from firmware to guest operating system.

21.7.6. If ZFS is available on the host machine, using ZFS volumes instead of disk image files can provide significant performance benefits for the guest VMs. A ZFS volume can be created by:

```bash
# zfs create -V16G -o volmode=dev zroot/linuxdisk0
```

When starting the VM, specify the ZFS volume as the disk drive:

```bash
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-blk,/dev/zvol/zroot/linuxdisk0 -l com1,stdio -c 4 -m 1024M linuxguest
```

21.7.7. It is advantageous to wrap the bhyve console in a session management tool such as `sysutils/tmux` or `sysutils/screen` in order to detach and reattach to the console. It is also possible to have the console of bhyve be a null modem device that can be accessed with `cu`. To do this, load the `nmdm` kernel module and replace `-l com1,stdio` with `-l com1,/dev/nmdm0A`. The `/dev/nmdm` devices are created automatically as needed, where each is a pair, corresponding to the two ends of the null modem cable (`/dev/nmdm0A` and `/dev/nmdm0B`). See `nmdm(4)` for more information.

```bash
# kldload nmdm
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-blk,./linux.img -l com1,/dev/nmdm0A -c 4 -m 1024M linuxguest
# cu -l /dev/nmdm0B
```

Connected Ubuntu 13.10 handbook ttyS0 handbook login:
A device node is created in `/dev/vmm` for each virtual machine. This allows the administrator to easily see a list of the running virtual machines:

```
# ls -al /dev/vmm
```

A specified virtual machine can be destroyed using `bhyvectl`:

```
# bhyvectl --destroy --vm=guestname
```

### 21.7.9. Persistent setting

In order to configure the system to start bhyve guests at boot time, the following configurations must be made in the specified files:

1. `/etc/sysctl.conf`
   ```
   net.link.tap.up_on_open=1
   ```
2. `/etc/rc.conf`
   ```
   cloned_interfaces="bridge0 tap0"
   ifconfig_bridge0="addm igb0 addm tap0"
   kld_list="nmdm vmm"
   ```

### 21.8. FreeBSD as the main installation

Xen is a GPLv2-licensed type 1 hypervisor for Intel™ and ARM™ architectures. FreeBSD has included i386™ and AMD™ 64-Bit DomU and Amazon EC2 unprivileged domain (virtual machine) support since FreeBSD 8.0 and includes Dom0 control domain (host) support in FreeBSD 11.0. Support for para-virtualized (PV) domains has been removed from FreeBSD 11 in favor of hardware virtualized (HVM) domains, which provides better performance.

Xen™ is a bare-metal hypervisor, which means that it is the first program loaded after the BIOS. A special privileged guest called the Domain-0 (Dom0 for short) is then started. The Dom0 uses its special privileges to directly access the underlying physical hardware, making it a high-performance solution. It is able to access the disk controllers and network adapters directly. The Xen™ management tools to manage and control the Xen™ hypervisor are also used by the Dom0 to create, list, and destroy VMs. Dom0 provides virtual disks and networking for unprivileged domains, often called DomU. Xen™ Dom0 can be compared to the service console of other hypervisor solutions, while the DomU is where individual guest VMs are run.
Xen™ can migrate VMs between different Xen™ servers. When the two xen hosts share the same underlying storage, the migration can be done without having to shut the VM down first. Instead, the migration is performed live while the DomU is running and there is no need to restart it or plan a downtime. This is useful in maintenance scenarios or upgrade windows to ensure that the services provided by the DomU are still provided. Many more features of Xen™ are listed on the Xen Wiki Overview page. Note that not all features are supported on FreeBSD yet.

21.8.1. Xen™ Dom0

To run the Xen™ hypervisor on a host, certain hardware functionality is required. Hardware virtualized domains require Extended Page Table (EPT) and Input/Output Memory Management Unit (IOMMU) support in the host processor.

In order to run a FreeBSD Xen™ Dom0 the box must be booted using legacy boot (BIOS).

21.8.2. Xen™ Dom0 integration

Users of FreeBSD 11 should install the emulators/xen-kernel47 and sysutils/xen-tools47 packages that are based on Xen version 4.7. Systems running on FreeBSD-12.0 or newer can use Xen 4.11 provided by emulators/xen-kernel411 and sysutils/xen-tools411, respectively.

Configuration files must be edited to prepare the host for the Dom0 integration after the Xen packages are installed. An entry to /etc/sysctl.conf disables the limit on how many pages of memory are allowed to be wired. Otherwise, DomU VMs with higher memory requirements will not run.

```bash
# echo 'vm.max_wired=-1' >> /etc/sysctl.conf
```

Another memory-related setting involves changing /etc/login.conf, setting the memorylocked option to unlimited. Otherwise, creating DomU domains may fail with Cannot allocate memory errors. After making the change to /etc/login.conf, run cap_mkdb to update the capability database. See limited resources for details.

```bash
# sed -i '' -e 's/memorylocked=64K/memorylocked=unlimited/' /etc/login.conf
# cap_mkdb /etc/login.conf
```

Add an entry for the Xen™ console to /etc/ttys:

```bash
# echo 'xc0     "/usr/libexec/getty Pc"         xterm   onifconsole  secure' >> /etc/ttys
```

Selecting a Xen™ kernel in /boot/loader.conf activates the Dom0. Xen™ also requires resources like CPU and memory from the host machine for itself and other DomU domains. How much CPU and memory depends on the individual requirements and hardware capabilities. In this example, 8 GB of memory and 4 virtual CPUs are made available for the Dom0. The serial console is also activated and logging options are defined.

The following command is used for Xen 4.7 packages:

```bash
# sysrc -f /boot/loader.conf hw.pci.mcfg=0
# sysrc -f /boot/loader.conf if_tap_load="YES"
# sysrc -f /boot/loader.conf xen_kernel="/boot/xen"
# sysrc -f /boot/loader.conf xen_cmdline="dom0_mem=8192M dom0_max_vcpus=4"
```
For Xen versions 4.11 and higher, the following command should be used instead:

```
# sysrc -f /boot/loader.conf if_tap_load="YES"
# sysrc -f /boot/loader.conf xen_kernel="/boot/xen"
# sysrc -f /boot/loader.conf xen_cmdline="dom0_mem=8192M dom0_max_vcpus=4 dom0=pvh console=com1,vga com1=115200,8n1 guest_loglvl=all loglvl=all"
```

Log files that Xen™ creates for the DomU VMs are stored in `/var/log/xen`. Please be sure to check the contents of that directory if experiencing issues.

Activate the xencommons service during system startup:

```
# sysrc xencommons_enable=yes
```

These settings are enough to start a Dom0-enabled system. However, it lacks network functionality for the DomU machines. To fix that, define a bridged interface with the main NIC of the system which the DomU VMs can use to connect to the network. Replace `em0` with the host network interface name.

```
# sysrc cloned_interfaces="bridge0"
# sysrc ifconfig_bridge0="addm em0 SYNCDHCP"
# sysrc ifconfig_em0="up"
```

Restart the host to load the Xen™ kernel and start the Dom0.

```
# reboot
```

After successfully booting the Xen™ kernel and logging into the system again, the Xen™ management tool `xl` is used to show information about the domains.

```
# xl list
```

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Mem</th>
<th>VCPUs</th>
<th>State</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain-0</td>
<td>0</td>
<td>8192</td>
<td>4</td>
<td>r-----</td>
<td>962.0</td>
</tr>
</tbody>
</table>

The output confirms that the Dom0 (called `Domain-0`) has the ID `0` and is running. It also has the memory and virtual CPUs that were defined in `/boot/loader.conf` earlier. More information can be found in the Xen Documentation. DomU guest VMs can now be created.

### 21.8.3. Xen™ DomU客端 VM設置

Unprivileged domains consist of a configuration file and virtual or physical hard disks. Virtual disk storage for the DomU can be files created by `truncate(1)` or ZFS volumes as described in 建立與摧毀磁碟區. In this example, a 20 GB volume is used. A VM is created with the ZFS volume, a FreeBSD ISO image, 1 GB of RAM and two virtual CPUs. The ISO installation file is retrieved with `fetch(1)` and saved locally in a file called `freebsd.iso`.
A ZFS volume of 20 GB called `xendisk0` is created to serve as the disk space for the VM.

```bash
# zfs create -V20G -o volmode=dev zroot/xendisk0
```

The new DomU guest VM is defined in a file. Some specific definitions like name, keymap, and VNC connection details are also defined. The following `freebsd.cfg` contains a minimum DomU configuration for this example:

```bash
# cat freebsd.cfg
builder
=  "hvm"

name
=  "freebsd"

memory
=  1024

vcpus
=  2

vif
=  ["mac=00:16:3E:74:34:32,bridge=bridge0"]

disk
=  ["/dev/zvol/tank/xendisk0,raw,hda,rw",
     "/root/freebsd.iso,raw,hdc:cdrom,r"]

vnc
=  1

vnclisten
=  "0.0.0.0"

serial
=  "pty"

usbdevice
=  "tablet"
```

These lines are explained in more detail:

1. This defines what kind of virtualization to use. `hvm` refers to hardware-assisted virtualization or hardware virtual machine. Guest operating systems can run unmodified on CPUs with virtualization extensions, providing nearly the same performance as running on physical hardware.

2. Name of this virtual machine to distinguish it from others running on the same Dom0. Required.

3. Quantity of RAM in megabytes to make available to the VM. This amount is subtracted from the hypervisor's total available memory, not the memory of the Dom0.

4. Number of virtual CPUs available to the guest VM. For best performance, do not create guests with more virtual CPUs than the number of physical CPUs on the host.

5. Virtual network adapter. This is the bridge connected to the network interface of the host. The `mac` parameter is the MAC address set on the virtual network interface. This parameter is optional, if no MAC is provided Xen™ will generate a random one.

6. Full path to the disk, file, or ZFS volume of the disk storage for this VM. Options and multiple disk definitions are separated by commas.

7. Defines the Boot medium from which the initial operating system is installed. In this example, it is the ISO imaged downloaded earlier. Consult the Xen™ documentation for other kinds of devices and options to set.

8. Options controlling VNC connectivity to the serial console of the DomU. In order, these are: active VNC support, define IP address on which to listen, device node for the serial console, and...
the input method for precise positioning of the mouse and other input methods.

The keymap defines which keymap to use, and is English by default.

After the file has been created with all the necessary options, the DomU is created by passing it to `xl create` as a parameter.

```
# xl create freebsd.cfg
```

Each time the Dom0 is restarted, the configuration file must be passed to `xl create` again to re-create the DomU. By default, only the Dom0 is created after a reboot, not the individual VMs. The VMs can continue where they left off as they stored the operating system on the virtual disk. The virtual machine configuration can change over time (for example, when adding more memory). The virtual machine configuration files must be properly backed up and kept available to be able to re-create the guest VM when needed.

The output of `xl list` confirms that the DomU has been created.

```
# xl list
Name                                        ID   Mem VCPUs      State   Time
((s))
Domain-0                                     0  8192     4     r-----  1653.4
freebsd                                      1  1024     1     -b-----  663.9
```

To begin the installation of the base operating system, start the VNC client, directing it to the main network address of the host or to the IP address defined on the `vnclisten` line of `freebsd.cfg`. After the operating system has been installed, shut down the DomU and disconnect the VNC viewer. Edit `freebsd.cfg`, removing the line with the `cdrom` definition or commenting it out by inserting a `#` character at the beginning of the line. To load this new configuration, it is necessary to remove the old DomU with `xl destroy`, passing either the name or the id as the parameter. Afterwards, recreate it using the modified `freebsd.cfg`.

```
# xl destroy freebsd
# xl create freebsd.cfg
```

The machine can then be accessed again using the VNC viewer. This time, it will boot from the virtual disk where the operating system has been installed and can be used as a virtual machine.

21.8.4.1. 主端開機疑難排解

This section contains basic information in order to help troubleshoot issues found when using FreeBSD as a Xen™ host or guest.

21.8.4.1.1. 主端開機疑難排解

Please note that the following troubleshooting tips are intended for Xen™ 4.11 or newer. If you are still using Xen™ 4.7 and having issues consider migrating to a newer version of Xen™.

In order to troubleshoot host boot issues you will likely need a serial cable, or a debug USB cable. Verbose Xen™ boot output can be obtained by adding options to the `xen_cmdline` option found in `loader.conf`. A couple of relevant debug options are:

- `iommu=debug`: can be used to print additional diagnostic information about the iommu.
- `dom0=verbose`: can be used to print additional diagnostic information about the dom0 build.
• sync_console: flag to force synchronous console output. Useful for debugging to avoid losing messages due to rate limiting. Never use this option in production environments since it can allow malicious guests to perform DoS attacks against Xen™ using the console.

FreeBSD should also be booted in verbose mode in order to identify any issues. To activate verbose booting, run this command:

```
# sysrc -f /boot/loader.conf boot_verbose="YES"
```

If none of these options help solving the problem, please send the serial boot log to freebsd-xen@FreeBSD.org and xen-devel@lists.xenproject.org for further analysis.

21.8.4.2. 建立疑難排解

Issues can also arise when creating guests, the following attempts to provide some help for those trying to diagnose guest creation issues.

The most common cause of guest creation failures is the `xl` command spitting some error and exiting with a return code different than 0. If the error provided is not enough to help identify the issue, more verbose output can also be obtained from `xl` by using the `v` option repeatedly.

```
# xl -vvv create freebsd.cfg
```

Parsing config from freebsd.cfg

libxl: debug: libxl_create.c:1693:do_domain_create: Domain 0:ao 0x800d750a0: create: how = 0x0 callback = 0x0 poller = 0x800d6f0f0

libxl: debug: libxl_device.c:397:libxl__device_disk_set_backend: Disk vdev = xvda spec.backend = unknown

libxl: debug: libxl_device.c:432:libxl__device_disk_set_backend: Disk vdev = xvda, using backend phy

libxl: debug: libxl_create.c:1018:initiate_domain_create: Domain 1:running bootloader

libxl: debug: libxl_bootloader.c:328:libxl__bootloader_run: Domain 1:not a PV/PVH domain, skipping bootloader


domainbuilder: detail: xc_dom_malloc_filemap : 326 kB

libxl: debug: libxl_dom.c:988:libxl__load_hvm_firmware_module: Loading BIOS: /usr/local/shared/seabios/bios.bin...

If the verbose output does not help diagnose the issue there are also QEMU and Xen™ toolstack logs in `/var/log/xen`. Note that the name of the domain is appended to the log name, so if the domain is named `freebsd` you should find a `/var/log/xen/xl-freebsd.log` and likely a `/var/log/xen/qemu-dm-freebsd.log`. Both log files can contain useful information for debugging. If none of this helps solve the issue, please send the description of the issue you are facing and as much information as possible to freebsd-xen@FreeBSD.org and xen-devel@lists.xenproject.org in order to get help.
Chapter 22.

22.1. Overview of FreeBSD's users and contributors distributed worldwide, and therefore FreeBSD supports multiple languages, allowing users to use non-native languages to view, input, or process data. Users can choose from the majority of languages, including but not limited to:

- Chinese
- German
- Japanese
- Korean
- French
- Russian
- Vietnamese

Internationalization (Internationalization) can be abbreviated as i18n, which means from the first letter to the last letter. L10n also uses the same naming convention, but is derived from Localization. i18n/L10n's methods, standards, and applications allow users to choose their own language.

In this chapter, we will discuss FreeBSD's internationalization and localization features. After reading this chapter, you will understand:

- How language names are composed.
- How to set up the language for the login shell.
- How to set the console for non-English language users.
- How to set Xorg to use different languages.
- How to find applications that support i18n.
- Where you can find more specific language settings.

Before starting to read this chapter, you need to understand:

- How to install other third-party applications.

22.2. Language settings

Language settings consist of three elements: language code, country code, and encoding. Language code and country code are used to represent cities and specific languages.

Common language and country codes provide several LanguageCode_CountryCode examples:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>en_US</td>
<td>English</td>
<td>USA</td>
<td>UTF-8</td>
</tr>
<tr>
<td>ru_RU</td>
<td>Russian</td>
<td>Russia</td>
<td>CP1251</td>
</tr>
<tr>
<td>zh_TW</td>
<td>Traditional Chinese</td>
<td>Taiwan</td>
<td>GB2312</td>
</tr>
</tbody>
</table>

The full list of available languages is available via the following command:

```
% locale -a | more
```

To query the current language settings:

```
% locale
```

Language-specific character sets such as ISO8859-1, ISO8859-15, KOI8-R, and CP437 are described in the `multibyte(3)` manual. The available character sets can be queried via the IANA Registry.
Some languages, such as Chinese or Japanese, cannot be represented with ASCII characters, so they require the use of Wide (Wide) characters or Multibyte (Multibyte) characters to expand the language code. EUC and Big5 are examples of Wide characters or Multibyte characters. Old applications may misinterpret these characters as control characters, while new applications usually can identify these characters, depending on the needs. Users may need to enable Wide characters or Multibyte character support or set the correct character encoding to compile applications.

 FreeBSD uses Xorg-compatible language codes.

The remaining part of this section will explain various methods to set up languages on the FreeBSD system. The next section will discuss how to find and compile applications with i18n support.

22.2.1. Setting up the login shell's language

Language settings can be set in the user's ~/.login_conf or in the environment startup files: ~/.profile, ~/.bashrc, or ~/.cshrc. There are two environment variables that need to be set:

- LANG is used to set the language
- MM_CHARSET is used to set the application program MIME character set

Apart from the user's shell settings, these variables should also be set for specific application programs as well as Xorg settings.

There are two ways to complete the required variable settings:

- Login class (Login class)
- Startup file (Startup file)

The following two sections will show examples of how to use these two methods.

22.2.1.1. Login class (Login class)

The first way, and the recommended way, can set the language and MIME character set needed by any possible shell settings. This setting can also be manually set by each user or by the superuser for all users.

The following example shows a setting in a user's ~/.login_conf file for the Latin-1 encoding:

```bash
me:
charset=ISO-8859-1:
lang=de_DE.ISO8859-1:
```

The following example sets the language for traditional Chinese in Big-5 encoding for a user's ~/.login_conf file. There are some parts of the application program that cannot correctly handle Chinese, Japanese, and Korean language variables, so additional settings are required:

```
# Users who do not wish to use monetary units or time formats
me:
lang=zh_TW.Big5:
setenv=LC_ALL=zh_TW.Big5,LC_COLLATE=zh_TW.Big5,LC_CTYPE=zh_TW.Big5,LC_MESSAGES=zh_TW.Big5,LC_MONETARY=zh_TW.Big5,LC_NUMERIC=zh_TW.Big5,LC_TIME=zh_TW.Big5:
charset=big5:
xmodifiers="@im=gcin": #Set gcin as the XIM Input Server
```
若套用之前的Latin-1編碼範例如下:

German Users Accounts:
- charset=ISO-8859-1
- lang=de_DE.ISO8859-1
- tc=default

請參考login.conf(5)以取得更多有關這些變數的詳細資訊。請注意，它已經有預先定義的russian class。

每編輯/etc/login.conf之後，請記得要執行以下指令來更新登入類別的能力資料庫:
```
# cap_mkdb /etc/login.conf
```

變更登入類別的工具除除了手動編輯/etc/login.conf之外，尚有多工具可用來為新建立的使用者設定關係。

當使用vipw來新增使用者時，可指定language來設定語言:
```
user:password:1111:11:language:0:0:User Name:/home/user:/bin/sh
```

當使用adduser來新增使用者時，可對所有使用者或指定的使用者事先設定預設的語

若所有新的使用者都使用同樣的語言，可在/etc/adduser.conf設定defaultclass=language。

要在建立使用者時覆蓋預設的設定，可在出現此提示時輸入需要的語言:
```
Enter login class: default
```

或執行adduser時指定語言:
```
# adduser -class language
```

若使用pw來新增使用者，則可指定語言如下:
```
# pw useradd user_name -L language
```

To change the login class of an existing user, chpass can be used. Invoke it as superuser and provide the username to edit as the argument.
啟動檔（Startup File）

第二種方法，較不建議使用，因為每種使用到的Shell都需要手動設定，而每種Shell都有不同的設定檔及語法。

例如將一位使用者的sh shell設定為德語，需要將下列行加入到~/.profile，若要設定給使用該Shell的所有使用者則必須將下列行加入到/etc/profile或/usr/shared/skel/dot.profile:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

然而，在csh shell所使用的設定檔名稱及語法不同。同樣的設定需加入下列行至~/.csh.login，/etc/csh.login或/usr/shared/skel/dot.login:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

更複雜一點的情況，Xorg的 ~/.xinitrc 系統設定會依使用的Shell而有所不同。第一個例子是針對sh shell而第二個則是針對csh shell:

```
LANG=de_DE.ISO8859-1; export LANG
```

22.2.2. Console 設定已有許多字型可在Console使用，要看可用的字型清单，可輸入ls /usr/shared/syscons/fonts。要設定Console的字型，可在/etc/rc.conf指定去掉了.fnt字尾的字型名稱font_name:

```
font8x16=font_name
font8x14=font_name
font8x8=font_name
```

鍵盤對應表（Keymap）及螢幕對應表（Screenmap）用可加入下行到/etc/rc.conf來設定:

```
scrnmap=screenmap_name
keymap=keymap_name
keychange="fkey_number sequence"
```

要看可用的螢幕對應表，可輸入ls /usr/shared/syscons/scrnmaps。在設定螢幕對應表screenmap_name時請去掉了.scm字尾。在VGA Adapter的字型字元矩陣擴充位元8到位元9時會需要使用螢幕對應表與相關的字型對應來解決，因此若螢幕字型使用位元8的欄位，字母會移出虛擬繪圖區（Pseudographics area）。
要查看可用的鍵盤對應表，可輸入`ls /usr/shared/syscons/keymaps`。在設定鍵盤對應表`keymap_name`時請去掉`.kbd`字尾。若要不重開機測試鍵盤對應用可使用`kbdmap(1)`。

`keychange`項目用在當功能鍵序列無法定義在鍵盤對應表時，可設定對應選擇終對機類型的功能鍵。

接下來，在`/etc/ttys`為所有虛擬終端機項目設定正確的Console終端機類型。已定義供特定字元集使用的終端機類型摘要了可用的終端機類型:

<table>
<thead>
<tr>
<th>字元集終端機類型</th>
<th>詞彙</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO8859-1 or ISO8859-15</td>
<td>cons25l1</td>
</tr>
<tr>
<td>ISO8859-2</td>
<td>cons25l2</td>
</tr>
<tr>
<td>ISO8859-7</td>
<td>cons25l7</td>
</tr>
<tr>
<td>KOI8-R</td>
<td>cons25r</td>
</tr>
<tr>
<td>KOI8-U</td>
<td>cons25u</td>
</tr>
<tr>
<td>CP437 (VGA預設值)</td>
<td>cons25</td>
</tr>
<tr>
<td>US-ASCII</td>
<td>cons25w</td>
</tr>
</tbody>
</table>

對於使用寬字元或多位元組字元的語言，需從Port套件集安裝支援該語言的Console。可用的Port摘要在Port套件集中可用的Console。

完整的Port位置如下:

<table>
<thead>
<tr>
<th>語言及位置</th>
<th>執行目錄</th>
</tr>
</thead>
<tbody>
<tr>
<td>繁體中文 (BIG-5)</td>
<td>chinese/big5con</td>
</tr>
<tr>
<td>中文/日文/韓文</td>
<td>chinese/cce</td>
</tr>
<tr>
<td>中文/日文/韓文</td>
<td>chinese/zhcon</td>
</tr>
<tr>
<td>日文</td>
<td>chinese/kon2</td>
</tr>
<tr>
<td>日文</td>
<td>japanese/kon2-14dot</td>
</tr>
<tr>
<td>日文</td>
<td>japanese/kon2-16dot</td>
</tr>
</tbody>
</table>

若在`/etc/rc.conf`有開啟`moused`，可能會需要額外的設定。預設`syscons(4)`驅動程式的滑鼠標會占用字元集0xd0-0xd3的範圍，若語言有使用到此範圍，可加入以下行到`/etc/rc.conf`來移動滑鼠的範圍:

```
mousechar_start=3
```

22.2.3. Xorg

設定X Window系統會說明如何安裝並設定Xorg。當要設定Xorg在地化時，可從FreeBSD Port套件集中取得其他可用的字型及輸入法。應用程式特定的i18n設定像是字型與選單，可以在`~/.Xresources`中調校且可允許使用者在圖形化應用程式選單檢視其所選的語言。

X輸入法(X Input Method, XIM)協定是Xorg針對輸入非英文字元的標準。可用的輸入法摘要了在FreeBSD套件集中可用的輸入法應用程式。也可使用其他如Fcitx及Uim應用程式。

<table>
<thead>
<tr>
<th>語言及輸入法</th>
<th>執行目錄</th>
</tr>
</thead>
<tbody>
<tr>
<td>中文</td>
<td>chinese/gcin</td>
</tr>
<tr>
<td>中文</td>
<td>chinese/ibus-chewing</td>
</tr>
</tbody>
</table>
FreeBSD Port套件集中含有許多內建支援寬字元或多位元字元的應用程式可支援各種語言。該類型的應用程式在名稱上會註明i18n以易於辨識。雖然如此，但不一定支援您所需要的語言。有一部份應用程式可以使用指定的字元集來編譯。通常會在Port的Makefile中設定，或者傳送參數給configure。請參考各FreeBSD Port原始碼中的i18n說明文件以取得更多有關需要的設定資訊或Port的Makefile來了解在編譯時有哪些可以使用的編譯選項。
This section provides configuration examples for localizing a FreeBSD system for the Russian language. It then provides some additional resources for localizing other languages.

### 22.4.1. 俄語 (KOI8-R 編碼)

This section shows the specific settings needed to localize a FreeBSD system for the Russian language. Refer to [Using Localization](#) for a more complete description of each type of setting.

To set this locale for the login shell, add the following lines to each user's `~/.login_conf`:

```
me:My Account:
:charset=KOI8-R:
:lang=ru_RU.KOI8-R:
```

To configure the console, add the following lines to `/etc/rc.conf`:

```
keymap="ru.utf-8"
scrnmap="utf-82cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
mousechar_start=3
```

For each `ttyv` entry in `/etc/ttys`, use `cons25r` as the terminal type.

To configure printing, a special output filter is needed to convert from KOI8-R to CP866 since most printers with Russian characters come with hardware code page CP866. FreeBSD includes a default filter for this purpose, `/usr/libexec/lpr/ru/koi2alt`. To use this filter, add this entry to `/etc/printcap`:

```
lp|Russian local line printer:
:sh:of=/usr/libexec/lpr/ru/koi2alt:
:lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Refer to [printcap(5)](#) for a more detailed explanation.

To configure support for Russian filenames in mounted MS-DOS™ file systems, include `-L` and the locale name when adding an entry to `/etc/fstab`:

```
/dev/ad0s2      /dos/c  msdos   rw,-Lru_RU.KOI8-R 0 0
```

Refer to [mount_msdosfs(8)](#) for more details.

To configure Russian fonts for Xorg, install the `x11-fonts/xorg-fonts-cyrillic` package. Then, check the "Files" section in `/etc/X11/xorg.conf`. The following line must be added before any other `FontPath` entries:

```
```

504
To activate a Russian keyboard, add the following to the "Keyboard" section of /etc/xorg.conf:

```
Option "XkbLayout"   "us,ru"
Option "XkbOptions"  "grp:toggle"
```

Make sure that XkbDisable is commented out in that file.

For `grp:toggle` use Ctrl+Alt, for `grp:ctrl_shift_toggle` use Ctrl+Shift. For `grp:caps_toggle` use CapsLock. The old CapsLock function is still available in LAT mode only using Shift+CapsLock.

`grp:caps_toggle` does not work in Xorg for some unknown reason.

If the keyboard has "Windows™" keys, and some non-alphabetical keys are mapped incorrectly, add the following line to /etc/xorg.conf:

```
Option "XkbVariant"   ",winkeys"
```

The Russian XKB keyboard may not work with non-localized applications. Minimally localized applications should call a XtSetLanguageProc (NULL, NULL, NULL); function early in the program.


### 22.4.2. 其他特定语言资源

This section lists some additional resources for configuring other locales.

#### Traditional Chinese for Taiwan

The FreeBSD-Taiwan Project has a Chinese HOWTO for FreeBSD at http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/.

#### Greek Language Localization

A complete article on Greek support in FreeBSD is available here, in Greek only, as part of the official FreeBSD Greek documentation.

#### Japanese and Korean Language Localization


#### Non-English FreeBSD Documentation

Some FreeBSD contributors have translated parts of the FreeBSD documentation to other languages. They are available through links on the FreeBSD web site or in /usr/shared/doc.
Chapter 23.

FreeBSD

23.1.概述

FreeBSD 在每次的發布之間持續在開發。有些人偏好正式發布的版本,也有另一群人喜歡使用最新的開發版本。然而,即使在正式發布的版本也時常會有安全性與其他緊急修復的更新,因此,無論使用哪種版本,FreeBSD 都提供所有必要的工具來讓系統能維持最新的版本,且讓各種版本都能簡單的升級。本章將說明如何追蹤開發版本的系統及讓 FreeBSD 系統維持最新版本的基本工具。

讀完這章,您將了解:

• 如何使用 freebsd-update, Subversion 來維持 FreeBSD 系統為最新版。
• 如何比對已安裝系統與已知原始複本間的狀態。
• 如何使用 Subversion 或說明文件 Port 來維持已安裝的文件為新版。
• 兩種開發分支間的差異: FreeBSD-STABLE 與 FreeBSD-CURRENT。
• 如何重新編譯及重新安裝整個基礎系統 (Base system)。

在開始閱讀這章之前,您需要:

• 正確的設定網路連線 (進階網路設定)。
• 了解如何安裝其他第三方軟體 (安裝應用程式: 套件與 Port)。

本章會經常使用 svnlite 來取得與更新 FreeBSD 原始碼。您也可以使用 devel/subversion Port 或套件。
要保持最新的部分。预设是更新整个基础系统（Base system）和核心。

可指定个别元件，例如：src/base 或 src/sys。

虽然如此，最好的选项是维持预设值，因为更改指定特定项目时需要列出每一个需要的项目。时间一久可能会因为原始代码和Binary档案没有更新而造成严重的后果。

# Paths which start with anything matching an entry in an IgnorePaths
# statement will be ignored.

IgnorePaths /boot/kernel/linker.hints

要保持特定目录在更新过程不被变动，例如/bin 或 /sbin，可以将它们的路径加到此叙述中。

这个选项可以防止freebsd-update覆盖本地的修正。

# Paths which start with anything matching an entry in an UpdateIfUnmodified
# statement will only be updated if the contents of the file have not been
# modified by the user (unless changes are merged; see below).

UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile

这个选项只会更新特定目录中未修改的设定档。任何使用者修改的档案都不会自动更新。有一个选项KeepModifiedMetadata可以让freebsd-update在合并时储存使用者所做的变更。

# When upgrading to a new FreeBSD release, files which match MergeChanges
# will have any local changes merged into the version from the new release.

MergeChanges /etc/ /var/named/etc/ /boot/device.hints

列出freebsd-update应尝试合并的设定档目录。档案合并程序是类似mergemaster(8)做的diff(1)修改，但是选项比较少。合并的动作为含接受、开启编辑器，或让freebsd-update中止。

如果有疑虑，请先备份/etc，然后再接受合并。更多信息关于mergemaster的资讯，请参见mergemaster(8)。

# Directory in which to store downloaded updates and temporary
# files used by FreeBSD Update.

Directory /var/db/freebsd-update

这个目录是所有修补档和暂存档的存放处。当使用者进行版本升级时，这个位置应至少有1GB的可用磁碟空间。

# When upgrading between releases, should the list of Components be
# read strictly (StrictComponents yes) or merely as a list of components
# which *might* be installed of which FreeBSD Update should figure out
# which actually are installed and upgrade those (StrictComponents no)?
# StrictComponents no

当这个选项设定为yes时，freebsd-update将会假设Components清单已完成，将不会对清单之外的项目做变更。实际上freebsd-update会尝试更新属于Components清单中的档案。
FreeBSD 9.X

升級從一個主要版本（Major version）另 FreeBSD 9.0 从的次要版本（Minor version）到 例23.2.3. 及 号評估護確多個系統時這會回維讓有用,因其可別檔,做是數編譯可變更任其他重新即自訂 /usr/src/sys/conf/newvers.sh 核心重新自訂的會安裝。更重行更新安裝核心重餘與核心原始非安裝的會安裝。指令來編譯更新設的會安裝核心任應重影,若啟動,發生任錯誤收該修到電子郵件通知補但補動新會需要如果任開機執補補新,系統將何行中的過重。第一個將後會單。第二個清,如果有,偵測若變更會執列出 FreeBSD 可以使用以下全性安 FreeBSD 保多有訊更可以參考補員可使用,被簡化 23.2.2. 套全性安 GENERIC /etc/freebsd-update.conf 可使用,請在一份執核心確留前,有後版本自訂於題問核心將及行版本時在隨 GENERIC freebsd-update install done

Uninstalling updates...# freebsd-update rollback

@daily                                  root    freebsd-update cron

# freebsd-update fetch

freebsd-update
核心的複本在 /boot/GENERIC。請参考在 FreeBSD 9.X 及之後版本自訂核心關於如何取得 GENERIC 核心複本的說明。

在 FreeBSD 9.0 系統執行以下指令，將會把系統升級至 FreeBSD 9.1:

```
# freebsd-update -r 9.1-RELEASE upgrade
```

收到這個指令後，freebsd-update 會開始評估設定檔和目前的系統來收集升級所需的資訊。螢幕會顯示偵測到或沒偵測到的元件清單。例如:

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 9.0-RELEASE from update1.FreeBSD.org... done.
Fetching metadata index... done.
Inspecting system... done.
The following components of FreeBSD seem to be installed:
  kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
  src/gnu src/include src/krb5 src/lib src/libexec src/release src/rescue
  src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
  world/base world/info world/lib32 world/manpages
The following components of FreeBSD do not seem to be installed:
  kernel/generic world/catpages world/dict world/doc world/games
  world/proflibs
Does this look reasonable (y/n)? y
```

此時，freebsd-update 將會嘗試下載所有升級需要的檔案。在某些情況，會詢問使用者一些關於要安裝什麼或要如何繼續。使用自訂核心，上述的步驟將會產生如下的警告:

```
WARNING: This system is running a "MYKERNEL" kernel, which is not a kernel configuration distributed as part of FreeBSD 9.0-RELEASE. This kernel will not be updated: you MUST update the kernel manually before running "/usr/sbin/freebsd-update install"
```

這時的警告可以安全地忽略，升級過程將會使用更新過的 GENERIC 核心來進行。

所有的修補與合併動作會在另一個目錄進行，並不會直接修改。當成功套用所有修補，所有設定檔已合併且過程順利，使用者可使用以下指令將變更安裝到磁碟:

```
509
```
核心與核心模組會先進行修補，若系統正在執行自訂的核心，使用 nextboot(8) 來設定下次開機使用更新過的 /boot/GENERIC:

```
# nextboot -k GENERIC
```

若機器在遠端進行更新，請在使用 GENERIC 核心重新開機前，確定該核心包含所有系統所需的驅動程式以正常開機並連線至網路。特別是在執行的自訂核心有使用到由核心模組提供的內建功能時，請確定將這些模組已暫時使用 /boot/loader.conf 設定檔載入 GENERIC 核心。

建議關閉非必須的服務和磁碟與網路掛載直到升級程序完成。

機器現在應該使用更新過的核心重新開機:

```
# shutdown -r now
```

一旦系統重新上線，使用以下指令繼續 freebsd-update。

由於程序的狀態已被儲存，freebsd-update 不會重新開始，但會進行下一個階段並移除所有舊的共用程式庫和目標檔。

```
# freebsd-update install
```

取決於是否有任何程式庫版本編號衝突，也可能只有兩個而不是三個安裝階段。

升級程序現在完成了。如果所做的是主要的版本升級，則需依照主要版本升級後的套件升級的說明重新安裝所有的 Port 和套件。

23.2.3.1. 在 FreeBSD 9.X 及之後版本自訂核心在使用 freebsd-update 前，請確定已有 GENERIC 核心的複本於 /boot/GENERIC。若只編譯過一次自訂核心，那麼 /boot/kernel.old 就是 GENERIC 核心，只需要將該目錄重新命名為 /boot/kernel。

若有編譯自訂核心過超過一次，或已經不曉得編譯自訂核心的次數，則需取得與目前作業系統版本相符的 GENERIC 核心複本。若可直接操作實體系統，則可以從安裝媒體取得 GENERIC 核心複本:

```
# mount /cdrom
# cd /cdrom/usr/freebsd-dist
# tar -C/ -xvf kernel.txz boot/kernel/kernel
```

或者，可以從原始碼重新編譯 GENERIC 核心:

```
# cd /usr/src
# make kernel __MAKE_CONF=/dev/null SRCCONF=/dev/null
```

這個核心要被 freebsd-update 認做 GENERIC 核心，GENERIC 設定檔必須不能做任何修改，也建議在編譯核心時不要使用其他特殊選項。
23.2.3.2. 主要版本號升級後的套件升級

一般來說，已安裝的應用程式在次要版本升級仍可沒問題的正常執行。但主要版本升級會採用不同的應用程式 Binary 介面 (Application Binary Interfaces, ABIs) 會導致大部份第三方應用程式無法正常執行。因此在主要版本升級後，需要升級及所有已安裝的套件和 Port，套件可以使用 `pkg upgrade` 來升級，而 Port 則需使用 `ports-mgmt/portmaster` 工具。

強制升級所有已安裝的套件會使用檔案庫中新版本的套件來取得目前套件，即使該版本沒增加。

因為在升級 FreeBSD 主要版本時會變更 ABI 版本，因此這是必要動作。強制升級可以執行以下指令來完成:

```
# pkg-static upgrade -f
```

重新編譯所有已安裝的應用程式可以執行以下指令來完成:

```
# portmaster -af
```

這個指令會在安裝每個應用程式有可設定選項時顯示設定畫面，並會等待使用者操作該畫面，要避免這種情況並使用預設的設定選項，可在上述指令加上 `-G` 參數。

完成軟體升級後，最後需執行 `freebsd-update` 來完成最後的升級動作:

```
# freebsd-update install
```

若有使用臨時 GENERIC 核心，便應在此時依據設定 FreeBSD 核心的說明編譯並安裝新的自訂核心。

重新開機使用新的 FreeBSD 版本後，升級程序便正式完成。

23.2.4. 比對系統狀態

已安裝的 FreeBSD 版本狀態可以使用 `freebsd-update IDS` 與另一個已知良好複本來做比對測試。

這個指令會評估目前版本的系統工具，程式庫和設定檔，可做為內建的入侵偵測系統 (Intrusion Detection System, IDS)。因 `freebsd-update` 儲存於磁碟上，被竄改的可能性是顯而易見的，雖然這個可能性會因使用 `kern.securelevel` 以及將 `freebsd-update` 在不使用時以唯讀儲存而降低，最好的解決方案是能夠與安全的磁碟，如 DVD 或儲存於外部的 USB 磁碟裝置比對系統。替代的方式是使用內建工具的 IDS 功能，在 Binary 檢驗有詳細說明。

要開始比對，需指定輸出的檔案來儲存結果:

```
# freebsd-update IDS >> outfile.ids
```

系統將會開始檢查並且會產生相當長的檔案清單，內容包含發佈版本已知的與目前安裝版本的 SHA256 灣倉值會儲存到指定的輸出檔。

清單中的項目會相當的多，但輸出的格式可以很簡單的用來分析。例如，要取得與發佈版本不同的檔案清單，可使用以下指令:

```
# freebsd-update IDS >> outfile.ids
```

系統將會開始檢查並且會產生相當長的檔案清單，內容包含發佈版本已知的與目前安裝版本的 SHA256 灣倉值會儲存到指定的輸出檔。
實際的檔設計會更多，此範例的輸出已精簡。部份檔案可能本身就會被修改。例如 `/etc/passwd` 在新增使用者到系統時會被修改，核心模組也有可能因使用 `freebsd-update` 更新而有所不同。要排除特定的檔案或目錄可將這些檔案或目錄加入到 `/etc/freebsd-update.conf` 中的 `IDSIgnorePaths` 選項。

23.3. 新文件集說明文件是 FreeBSD 作業系統不可或缺的一部份。最新版本的 FreeBSD 文件除了可在 FreeBSD 網站 (https://www.freebsd.org/doc/) 取得，也可很簡單的取得本地的 FreeBSD 網站、使用手冊、FAQ 及文章副本。

本節將說明如何使用原始碼與 FreeBSD Port 套件來取得最新版本 FreeBSD 文件副本。

有關於編輯與提出修正說明文件的資訊，請參考『FreeBSD 文件計畫入門』(FreeBSD Documentation Project Primer)。

23.3.1. 自原始碼更新說明文件

從原始碼重新編譯 FreeBSD 文件需要一些不属于 FreeBSD 基礎系統的工具。需要的工具可安裝由 FreeBSD 文件計畫所開發的 `textproc/docproj` 套件或 Port。

安装完成之後，可使用 `svnlite` 來取得乾淨的文件原始碼副本：

```
# svnlite checkout https://svn.FreeBSD.org/doc/head /usr/doc
```

第一次下載文件原始碼需要一些時間，請耐心等候執行完畢。

往後更新文件原始碼可執行：

```
# svnlite update /usr/doc
```

下载最新的文件原始碼到 `/usr/doc` 之後，便完成要更新已安裝文件的準備動作。

完整更新所有可用的語言可以執行：

```
# cd /usr/doc
# make install clean
```

若只想要更新特定語言，可對 `/usr/doc` 中特定語言的子目錄執行 `make`:

```
# cd /usr/doc/en_US.ISO8859-1
# make install clean
```
另

一個

更

新文件的方式是在

/usr/doc

或

特定

的

语言

子目

录

下

执行此

指令:

```
# make update
```

要

指定

安装的

输出

格式可使用

FORMATS

来设定:

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

有

数

个

选项

可

更新部份文件或只

编译

特定

翻译来简化更

新程序。这些

选项

可在

/etc/make.conf

设为

系统

全域

的预

设选项

,或是

透

过指令传

送

给

make

。选

项

有

:

DOC_LANG

要

编译

与安装的

语言

及编

码

清单

,例

如

en_US.ISO8859-1

代表

英

语

文件。

FORMATS

要

编译

的

输出

格式

清单

,目前

支援

html,

html-split,

txt,

ps

及

pdf

。

DOCDIR

要安装文件的

位置

,预

设为

/usr/shared/doc

。

要取得

更多可

做为

FreeBSD

系统

全域

选项

的

make

变量

,请参考

make.conf(5)

。

23.3.2.

自

Port

更

新说明文件

前一节介绍了解释原始

码

更

新

FreeBSD

文件的方

法

,本节

将

说明使用

Port

套

件集的

替

代方

法

,可

由

以下方式

达

成

:

•

安装

事

先

编译

好

的文件

套

件,

无

须

在本

地

编译

任

何

东西或安</p>
Port 提供設定選單來指定要編譯與安裝的格式,預設會選擇分頁的 HTML (類似 http://www.FreeBSD.org 使用的格式) 以及 PDF。

此外，編譯文件 Port 時也可指定數個 make 選項，包括:

- WITH_HTML: 編譯一份文件使用一個 HTML 檔的 HTML 格式。格式化後的文件會儲存至名稱為 article.html 或 book.html 的檔案。
- WITH_PDF: 格式化後的文件會儲存至名稱為 article.pdf 或 book.pdf 的檔案。
- DOCBASE: 指定要安裝文件的位置，預設為 /usr/local/shared/doc/freebsd。

以下範例使用變數來安裝 PDF 的匈牙利語文件到特定目錄:

```
# cd /usr/ports/misc/freebsd-doc-hu
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

文件套件或 Port 可以依安裝應用程式: 套件與 Port 的說明更新。例如以下指令會使用 ports-mgmt/portmaster 更新已安裝的匈牙利語文件:

```
# portmaster -PP hu-freebsd-doc
```

23.4. 追蹤開發分支

FreeBSD 有兩個開發分支: FreeBSD-CURRENT 及 FreeBSD-STABLE。

本節將說明每個分支及其的特性，也會說明如何在各別分支維持系統為最新版。

23.4.1. 使用 FreeBSD-CURRENT

FreeBSD-CURRENT 是 FreeBSD 開發的“最前线”，FreeBSD-CURRENT 的使用者需具備較強的技術能力。技術能力較弱的使用者應改追蹤 FreeBSD-STABLE 開發分支。

FreeBSD-CURRENT 是 FreeBSD 最新的原始碼，其中包括正進行的開發工作、實驗性的變更以及不定會在下一個官方發行版出現的過渡機制。雖然 FreeBSD 開發者每幾天編譯 FreeBSD-CURRENT 原始碼，但仍可能有短暫時間原始碼是無法編譯的。這些問題會儘快被解決，但無論 FreeBSD-CURRENT 帶來災難或是新功能，同步原始碼時都要考量這個問題。

FreeBSD-CURRENT 主要給以下三種族群:

1. 致力於開發某一部份原始碼樹的 FreeBSD 社群成員。
2. FreeBSD 社群成員中活耀的測試人員。他們願意花時間解決問題，對 FreeBSD 的變更及向提出專業建議並送交修補。
3. 隨時關注的使用者，使用目前原始碼做為參考用途，或是偶爾提供意見或貢獻原始碼。

不應將 FreeBSD-CURRENT 514
若要 FreeBSD-STABLE，特别的是问题中未在其中，因此不可的，有时候仍在 FreeBSD-STABLE 的范围内。当尚未快速，很可能会有途径。使用已知的，当问题出现时，很可能有一些可能相的。递归的，指示要订阅的递归的，会追踪 SVN 的加入。这些对加强 FreeBSD-CURRENT 活跃！我们非下一次升级解要资料行版的。操作。阅读 FreeBSD-CURRENT 的代码的部份到替换的，选择档。依 FreeBSD-STABLE 中的，他们可用的，是有限的。是有关的。问题在递归的，该递归的会加入。您们对加强 FreeBSD-CURRENT 同步变更，不照要订阅的递归的，原始操作。要个指示只有点骤码的网页。这也是一次修补，以关录每的提供提副。解来随 FreeBSD-STABLE 解要了问题的方式，因此有收评的目前人解。加入。邮递论坛。这是镜像站以依 FreeBSD-STABLE 的，是有限的，是有关的。
1. 更新並編譯

   # svnlite update /usr/src

①  取得最新版本的原始碼，請參考更新原始碼來了解更多取得與更新原始碼的資訊。

②  檢查 /usr/src/UPDATING 看是否有後在原始碼編譯之前或之後需要手動操作的步驟。

③  前往原始碼目錄。

④  編譯世界（World），即除了核心（Kernel）外的所有東西。

⑤  編譯並安裝核心，此動作等同於 make buildkernel installkernel。

⑥  重新啟動系統以使用新的核心。

⑦  前往原始碼目錄。

⑧  安裝世界。

⑨  更新與合併在 /etc/ 中的設定檔。

⑩  重新啟動系統以使用新編譯好的世界與核心。

23.5.2. 備原始碼更新

閱讀 /usr/src/UPDATING，從原始碼編譯之前與之後任何需要手動操作步驟會在此檔案中說明。
更 新 原始 碼 FreeBSD 的 原始 碼 位於 /usr/src/, 視 建 議 透 過 Subversion 版本 控制 系統 來更 新這份 原始 碼, 要 確 認原始 碼 已 在版 本 控制 系統 的 管 控 下可 用:

```
# svnlite info /usr/src
Path: /usr/src
Working Copy Root Path: /usr/src
```

此 結 果 代表 /usr/src/ 已 在版 本 控制 系統 的 管 控 下並且 可 用 svnlite(1) 來更 新:

```
# svnlite update /usr/src
```

若 該目 錄 最 近 沒 有 更 新 過, 可 能 會 需 要 一些 時 間 來 完 成 更 新 作。 在 更 新 完 毕 之 後, 原始 碼 便 為 最新 版 本, 並 可 開 始 依 下 一 章節 的 說 明 來 編 譯 程序。

若輸 出 結 果 顯 示 '/usr/src' is not a working copy 代表 有 錯 誤 档 案 或 原始 碼 是 采 用 其他 方 式 安 裝, 若是 如 此, 便 需 重 新 取 出 (checkout) 原始 碼。

### 表 18. FreeBSD 版本與 筆 證列庫路 徑

<table>
<thead>
<tr>
<th>版 本</th>
<th>路 徑</th>
<th>說 明</th>
</tr>
</thead>
<tbody>
<tr>
<td>X.Y - RELEASE</td>
<td>base/releng/X.Y</td>
<td>發佈 版 本 加 上 關 鍵 的 安 全 性 與 錯誤 修 正, 視 建 議 多 半 使用 該 分 支。</td>
</tr>
<tr>
<td>X.Y - STABLE</td>
<td>base/stable/X</td>
<td>發佈 版 本 加 上 所 有 在 該 分 支 上 其 他 發 開 中 的 程 式, STABLE 代 講 不 會 變 更 應 用 程 式 Binary 介 面 (Applications Binary Interface, ABI), 所 以 在 先 前 版 本 所 編 譯 的 軟 體 仍 可 以 正 常 運 作, 舉 例 講, 被 編 譯 在 FreeBSD 10.1 可 行 的 軟 體 在 編 譯 完 FreeBSD 10-STABLE 之 後 仍 可 行。</td>
</tr>
<tr>
<td>X - CURRENT</td>
<td>base/head/</td>
<td>最 新 未 發 佈 的 FreeBSD 發 開 版 本, CURRENT 分 支 可 能 會 有 重 大 錯誤 或 不 相 容 的 問 題 影 響 用 者, 說 明 這 些 問 題 通 常 會 非 常 快 速 地 被 修 正。</td>
</tr>
</tbody>
</table>

查 看 FreeBSD 目 前 使用 的 版 本 可 使 用 uname(1):
### 23.5.4. From Source Code Build World (world)

To provide the latest tools for compiling the kernel, run `make` to compile the world:

```bash
# cd /usr/src
# make buildworld
# make buildkernel
```

Compiled programs are written to `/usr/obj`.

These are basic steps; other options for controlling compilation will be discussed in later chapters.

### 23.5.4.1. Execute Clean World (cleanworld)

The FreeBSD build system leaves the previously compiled programs in the temporary `/usr/obj` location to avoid re-compiling unchanged source code, which can speed up subsequent compilation. To force re-compilation of all programs, use `cleanworld` before starting compilation:

```bash
# make cleanworld
```

### 23.5.4.2. Setting the Number of Jobs (jobs)

Increasing the number of jobs on a multiprocessor can increase compilation speed, which can be achieved using `sysctl hw.ncpu` to check the number of processes. Different processors use different versions of FreeBSD, so the only way to determine the effect of differing job numbers on compilation speed is by testing. At the start of the compilation, consider choosing a value between 1/2 and 2 times the number of cores, with the number of jobs specified using `-j`:

Example 44. Use four jobs to compile the world and kernel:

```bash
# make -j4 buildworld buildkernel
```

518
只編譯核心若原始碼有更動，便須執行 buildworld，完成之後，便可隨時執行 buildkernel 来編譯核心，若要只編譯核心可:
```bash
# cd /usr/src
# make buildkernel
```

23.5.4.4. 編譯自訂核心標準的 FreeBSD 核心是以一個名為 GENERIC 的核心設定檔(Kernel config file)為基礎, GENERIC 核心中內含了所有最常用的裝置驅動程式與選項，有時這個檔案對編譯自訂核心也非常有用, 可根據其來加入或移除裝置驅動程式或選項來滿足特定需求。

例如, 要開發一個 RAM 受到嚴重限制的小型嵌入式電腦，便可移除不需要的裝置驅動程式或選項來縮小核心。

核心設定檔位於 /usr/src/sys/arch/conf/，其中使用的 arch 即為 `uname -m` 輸出的結果，大部份的電腦為 amd64，那其設定檔目錄則為 /usr/src/sys/amd64/conf/。

```

```

```
/usr/src 可以被刪除或重建，所以建議將自訂核心設定檔放在另一個目錄，如 /root，並將核心設定檔以連結放至 conf 目錄，若該目錄被刪除或覆寫，便可重新建立一个新的核心設定的連結。
```

自訂設定檔可由複製 GENERIC 設定檔來建立, 在此範例中，新的自訂核心要用在儲存伺服器, 所以將其命名為 STORAGESERVER:
```bash
# cp /usr/src/sys/amd64/conf/GENERIC /root/STORAGESERVER
# cd /usr/src/sys/amd64/conf
# ln -s /root/STORAGESERVER .
```

接著編譯/root/STORAGESERVER，要加入或移除裝置或選項可見 config(5)。

自訂核心要在指令列設定 KERNCONF 为核心設定檔來編譯:
```bash
# make buildkernel KERNCONF=STORAGESERVER
```

23.5.5. 安裝編譯好的程式的在完成 buildworld 與 buildkernel 兩個步驟之後，便可安裝新的核心與世界:
```bash
# cd /usr/src
# make installkernel
# shutdown -r now
# cd /usr/src
# make installworld
# shutdown -r now
```

若使用自訂核心，則同樣須設定 KERNCONF 來使用新的自訂核心:
# cd /usr/src
# make installkernel KERNCONF=STORAGESERVER
# shutdown -r now
# cd /usr/src
# make installworld
# shutdown -r now

23.5.6.
完成
更新
還有最後一些的工作要做來完成更新，任何修改過的設定檔要與新版本的設定檔合併、移除找到的過時程式庫，然後重新啟動系統。

23.5.6.1.
使用`mergemaster(8)`合併設定檔

`mergemaster(8)`可簡單的將修改過的系統設定檔與新版設定檔合併。

使用`-Ui`，`mergemaster(8)`會自動更新那些未被使用者修改過的設定檔並安裝尚不在的檔案:

```
# mergemaster -Ui
```

若檔案需要手動合併，會有互動式介面可讓使用者選擇要保留那一邊的檔案，请参考`mergemaster(8)`取得更多資訊。

23.5.6.2.
檢查過時的檔案與程式的檔案或目錄可以在更新之後保留，可使用以下指令找出這些檔案:

```
# make check-old
```

並用以下指令刪除:

```
# make delete-old
```

部份廢棄的程式庫也可以保留下來，可使用以下指令來偵測這些程式庫:

```
# make check-old-libs
```

並使用以下指令刪除:

```
# make delete-old-libs
```

那些仍使用舊程式庫的程式將在刪除程式庫之後無法正常运作，而這些程式須要在刪除舊程式庫之後重新編譯或更換。

當確認所有舊檔案或目錄可安全的刪除時，要避免刪除每一個檔案時均需按下`y`與Enter鍵可在指令設定`BATCH_DELETE_OLD_FILES`，例如:

```
520
```
到一個建置要在本地置享寫入錄可的通用機應設定目
途的方也可用在同法機
置建正將機上果安裝到在皆好作常執
並如同掛NFS試將好核心不要在測機機上試但測透主
在World機上如
KERNCONF核心置名稱建,且機應列出置在機應建置在
確保集中的所有主基系統機致一主置建置在NFS集中的主所有在此
建也可是同時是不是一置定機
將運做測機正要在,這挑選再機做為buildkernel下要的足夠執
速度集中需binary。一部
先,要先Binary認機確群又的一部主
 binary
又的一部主
binary

(NFS)
事多有訊何做關餘的主工問，要算完大部份的作，而其運機上做
過間與磁碟所有的東西會
編譯新原始一部主當有多部主需要重樹多部
啟變更生效:腦所有的便

23.5.6.3.
更啟

DISTDIR

# make BATCH_DELETE_OLD_FILES=yes delete-old-libs

WRKDIRPREFIX

mergemaster
使用 DTrace 檢階低不安全的查還存。因此全 (low level)，只有全因安差異行，下可能可以或不可以在腳之執修反然。的亦上經情況 FreeBSD 他。因此，在是不相容的，因為有不同的參 providers 上們具型的版本中兩不業則 FreeBSD 未中個作出現 FreeBSD。此外，在。這些可能會在核心 FreeBSD (type) 追蹤中相，Solaris™ 與 FreeBSD 存 providers 在一些不同的 DWARF `ctfconvert` and 資料通過類於裝了一種形，簡化訊除錯封它資式的 Compact C Type Format，似使用 FreeBSD 會在 FreeBSD 10.0-RELEASE 執行從模組模組: 要載所有需要的實是作無 (kernel modules) 法入模組的，並且在載組核心模組中，為 DTrace 之前做一 FreeBSD 之然是雖然的是有還似，但 DTrace 要的 Solaris™ FreeBSD 差異存和，。最實差異作 24.2.
24.3. 开启 DTrace 支援

在 FreeBSD 9.2 和 10.0 中, DTrace 内建於 GENERIC 核心裡。FreeBSD 早期版本的使用者或喜欢在 DTrace 支援下静态编译的使用者应加入下列几行到客製化核心配置文件,并根据配置 FreeBSD Kernel 中的说明重新编译核心:

```
options KDTRACE_HOOKS
options DDB_CTF
makeoptions DEBUG=-g
makeoptions WITH_CTF=1
```

AMD64 架构的使用者应加入下列几行:

```
options KDTRACE_FRAME
```

此选项提供对 FBT 的支援,虽然 DTrace 可以在没有此选项的情况下运作,但对程式边界追踪的支援有限。

一旦 FreeBSD 系统使用新的核心重新启动,或者使用 kldload dtraceall 载入 DTrace 核心模组后,系统需要支援 Korn shell,因为 DTrace 工具箱有好几个用 ksh 拓写工具。确保已经安装 shells/ksh93 套件或者 port, 也可以在 shells/pdksh 或者 shells/mksh 下执行这些工具。

最后, 安装目前的 DTrace 工具箱, 这是一组用於收集系统资讯的程序, 有一些程序可以检查打开的文件、记忆体、CPU 使用情况等等。FreeBSD 10 将其中一些程序安装在 /usr/share/dtrace 中。在其他 FreeBSD 的版本中, 要安装 DTrace 工具箱, 请使用 sysutils/dtrace-toolkit 套件或者 port。

```
523
```
# cd /usr/local/share/dtrace-toolkit
# ./hotkernel
Sampling... Hit Ctrl-C to end.

按照說明，使用Ctrl+C組合鍵停止行程，中止後，腳本將顯示一整列核心函式和時間資訊，按照時間遞增排序:

<table>
<thead>
<tr>
<th>MODULE</th>
<th>COUNT</th>
<th>PCNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xc1097063</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`_thread_lock_flags</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>0xc10e6aa4</td>
<td>1</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`sched_userret</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`kern_select</td>
<td>2</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`generic_copyin</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`_mtx_assert</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`vm_fault</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`sopoll_generic</td>
<td>3</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`fixup_filename</td>
<td>4</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`_isitmyx</td>
<td>4</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`find_instance</td>
<td>4</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`_mtx_unlock_flags</td>
<td>5</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`syscall</td>
<td>5</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`DELAY</td>
<td>5</td>
<td>0.0%</td>
</tr>
<tr>
<td>0xc108a253</td>
<td>6</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`witness_lock</td>
<td>7</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`read_aux_data_no_wait</td>
<td>7</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`Xint0x80_syscall</td>
<td>7</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`witness_checkorder</td>
<td>7</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`sse2_pagezero</td>
<td>8</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`strncmp</td>
<td>9</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`spinlock_exit</td>
<td>10</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`_mtx_lock_flags</td>
<td>11</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`witness_unlock</td>
<td>15</td>
<td>0.0%</td>
</tr>
<tr>
<td>kernel`sched_idletd</td>
<td>137</td>
<td>0.3%</td>
</tr>
<tr>
<td>0xc10981a5</td>
<td>42139</td>
<td>99.3%</td>
</tr>
</tbody>
</table>

此腳本也是用於核心模組，要使用此功能，請使用-m執行腳本:

# ./hotkernel -m
Sampling... Hit Ctrl-C to end. 

^C
抓取和输出系统调用时间，给设定行程ID (PID) 或行程名称的行程。在以下的例子中，生成了/bin/csh 新物件，然后，procsystime 被执行并一直等待，同时在 csh 的另一个化身上输入一些指令，以下是本次测试的结果:

```
# ./procsystime -n csh

Tracing... Hit Ctrl-C to end...

^C

Elapsed Times for processes csh, SYSCALL          TIME
          (ns)                        
g getpid               6131
   sigreturn               8121
    close              19127
   fcntl              19959
    dup              26955
    setpgid              28070
    stat               31899
    setitimer              40938
    wait4              62717
   sigaction              67372
   sigprocmask             119091
    gettimeofday             183710
    write             263242
   execve             492547
    ioctl             770073
   vfork            3258923
   sigsuspend            6985124
    read                             3988049784
```

如图所示，read() 系统调用使用的时最多 (以奈秒为单位)，而 getpid() 系统调用使用的时最少。
Chapter 25. USB Device Mode / USB OTG

25.1. 概述
This chapter covers the use of USB Device Mode and USB On The Go (USB OTG) in FreeBSD. This includes virtual serial consoles, virtual network interfaces, and virtual USB drives.

When running on hardware that supports USB device mode or USB OTG, like that built into many embedded boards, the FreeBSD USB stack can run in **device mode**. Device mode makes it possible for the computer to present itself as different kinds of USB device classes, including serial ports, network adapters, and mass storage, or a combination thereof. A USB host like a laptop or desktop computer is able to access them just like physical USB devices. Device mode is sometimes called the "USB gadget mode".

There are two basic ways the hardware can provide the device mode functionality: with a separate "client port", which only supports the device mode, and with a USB OTG port, which can provide both device and host mode. For USB OTG ports, the USB stack switches between host-side and device-side automatically, depending on what is connected to the port. Connecting a USB device like a memory stick to the port causes FreeBSD to switch to host mode. Connecting a USB host like a computer causes FreeBSD to switch to device mode. Single purpose "client ports" always work in device mode.

What FreeBSD presents to the USB host depends on the **hw.usb.template** sysctl. Some templates provide a single device, such as a serial terminal; others provide multiple ones, which can all be used at the same time. An example is the template 10, which provides a mass storage device, a serial console, and a network interface. See **usb_template(4)** for the list of available values.

Note that in some cases, depending on the hardware and the hosts operating system, for the host to notice the configuration change, it must be either physically disconnected and reconnected, or forced to rescan the USB bus in a system-specific way. When FreeBSD is running on the host, **usbconfig(8)** reset can be used. This also must be done after loading **usb_template.ko** if the USB host was already connected to the USBOTG socket.

讀完這章, 您將了解:
• How to set up USB Device Mode functionality on FreeBSD.
• How to configure the virtual serial port on FreeBSD.
• How to connect to the virtual serial port from various operating systems.
• How to configure FreeBSD to provide a virtual USB network interface.
• How to configure FreeBSD to provide a virtual USB storage device.

25.2. USB 虛擬序列埠

25.2.1. 設置 USB 裝置模式序列埠
Virtual serial port support is provided by templates number 3, 8, and 10. Note that template 3 works with Microsoft Windows 10 without the need for special drivers and INF files. Other host operating systems work with all three templates. Both **usb_template(4)** and **umodem(4)** kernel modules must be loaded.

To enable USB device mode serial ports, add those lines to **/etc/ttys**:

```
ttyU0  
   /usr/libexec/getty 3wire
   vt100   onifconsole secure

ttyU1  
   /usr/libexec/getty 3wire
   vt100   onifconsole secure
```

然後加入這些行到 **/etc/devd.conf**:
25.2.2. FreeBSD

To connect to a board configured to provide USB device mode serial ports, connect the USB host, such as a laptop, to the board's USB OTG or USB client port. Use `pstat -t` on the host to list the terminal lines. Near the end of the list you should see a USB serial port, eg “ttyU0”. To open the connection, use:

```
# cu -l /dev/ttyU0
```

After pressing the Enter key a few times you will see a login prompt.

25.2.3. macOS

To connect to a board configured to provide USB device mode serial ports, connect the USB host, such as a laptop, to the board's USB OTG or USB client port. To open the connection, use:

```
# cu -l /dev/cu.usbmodemFreeBSD1
```

25.2.4. Linux

To connect to a board configured to provide USB device mode serial ports, connect the USB host, such as a laptop, to the board's USB OTG or USB client port. To open the connection, use:

```
# cu -l /dev/ttyU0
```

After pressing the Enter key a few times you will see a login prompt.
To connect to a board configured to provide USB device mode serial ports, connect the USB host, such as a laptop, to the board's USB OTG or USB client port. To open a connection, you will need a serial terminal program, such as PuTTY. To check the COM port name used by Windows, run Device Manager, expand "Ports (COM & LPT)". You will see a name similar to "USB Serial Device (COM4)". Run a serial terminal program of your choice, for example PuTTY. In the PuTTY dialog, set "Connection type" to "Serial", type the COMx obtained from Device Manager in the "Serial line" dialog box and click Open.

25.3. USB装置模式网络接口

Virtual network interfaces support is provided by templates number 1, 8, and 10. Note that none of them works with Microsoft Windows. Other host operating systems work with all three templates. Both `usb_template(4)` and `if_cdce(4)` kernel modules must be loaded.

Make sure the necessary modules are loaded and the correct template is set at boot by adding those lines to `/boot/loader.conf`, creating it if it does not already exist:

```
if_cdce_load = "YES"
hw.usb.template = 1
```

To load the module and set the template without rebooting, use:

```
# kldload if_cdce
# sysctl hw.usb.template=1
```

25.4. USB虚拟存储装置

Mass Storage target is provided by templates 0 and 10. Both `usb_template(4)` and `cfumass(4)` kernel modules must be loaded. `cfumass(4)` interfaces to the CTL subsystem, the same one that is used for iSCSI or Fibre Channel targets. On the host side, USB Mass Storage initiators can only access a single LUN, LUN 0.

25.4.1. 使用`cfumass`启动

The simplest way to set up a read-only USB storage target is to use the `cfumass` rc script. To configure it this way, copy the files to be presented to the USB host machine into the `/var/cfumass` directory, and add this line to `/etc/rc.conf`:

```
cfumass_enable="YES"
```

To configure the target without restarting, run this command:

```
# service cfumass start
```

528
Differently from serial and network functionality, the template should not be set to 0 or 10 in /boot/loader.conf. This is because the LUN must be set up before setting the template. The cfumass startup script sets the correct template number automatically when started.

25.4.2. 使用其他方式設定USB大容量存儲目標

The rest of this chapter provides detailed description of setting the target without using the cfumass rc file. This is necessary if eg one wants to provide a writeable LUN.

USB Mass Storage does not require the ctld(8) daemon to be running, although it can be used if desired. This is different from iSCSI. Thus, there are two ways to configure the target: ctladm(8), or ctld(8). Both require the cfumass.ko kernel module to be loaded. The module can be loaded manually:

```
# kldload cfumass
```

If cfumass.ko has not been built into the kernel, /boot/loader.conf can be set to load the module at boot:

```
cfumass_load="YES"
```

A LUN can be created without the ctld(8) daemon:

```
# ctladm create -b block -o file=/data/target0
```

This presents the contents of the image file /data/target0 as a LUN to the USB host. The file must exist before executing the command. To configure the LUN at system startup, add the command to /etc/rc.local.

ctld(8) can also be used to manage LUNs. Create /etc/ctl.conf, add a line to /etc/rc.conf to make sure ctld(8) is automatically started at boot, and then start the daemon.

This is an example of a simple /etc/ctl.conf configuration file. Refer to ctl.conf(5) for a more complete description of the options.

```
target naa.50015178f369f092 {
  lun 0 {
    path /data/target0
    size 4G
  }
}
```

The example creates a single target with a single LUN. The naa.50015178f369f092 is a device identifier composed of 32 random hexadecimal digits. The path line defines the full path to a file or zvol backing the LUN. That file must exist before starting ctld(8). The second line is optional and specifies the size of the LUN.

To make sure the ctld(8) daemon is started at boot, add this line to /etc/rc.conf:

```
ctld_enable="YES"
```
To start ctld(8) now, run this command:

```
# service ctld start
```

When ctld(8) Daemon starts, it will read /etc/ctl.conf, so if this file is modified after the Daemon starts, you need to reload the changes in order to see them:

```
# service ctld reload
```
Part IV: 網路通訊

FreeBSD 是一種廣泛的被使用在高效率的網路伺服器中的作業系統，這些篇章包括了:

• 序列通訊
• PPP 和在乙太網路上使用 PPP
• 電子郵件
• 萬行網路伺服器
• 防火牆
• 其他的進階網路主題

這些篇章是讓您在需要查資料的時候翻閱用的。您不需要依照特定的順序來讀，也不需要將這些篇章全部讀過之後才將 FreeBSD 用在網路環境下。
Chapter 26.

序列

26.1. 概述

UNIX™ 从最早的第一台 UNIX™ 仰赖序列线路来让使用者输入与输出，虽与每秒 10 个字元的序列印表机及键盘组合的终端机时代比起来已改变很多。本章将说明几种可在 FreeBSD 使用的序列通讯方式。

读完这章，您将了解：
- 如何连线终端机到 FreeBSD 系统。
- 如何使用数据机拨号给远端主机。
- 如何允许远端使用者透过数据机登录 FreeBSD 系统。
- 如何从序列 Console 启动 FreeBSD 系统。

在开始阅读这章之前，您需要：
- 了解如何设定并安装自订核心。
- 了解 FreeBSD 的权限与程序。
- 能够取得要在 FreeBSD 使用的序列硬体的技术手册。

26.2. 序列术语与硬体

The following terms are often used in serial communications:

**bps**
Bits per Second (bps) is the rate at which data is transmitted.

**DTE**
Data Terminal Equipment (DTE) is one of two endpoints in a serial communication. An example would be a computer.

**DCE**
Data Communications Equipment (DCE) is the other endpoint in a serial communication. Typically, it is a modem or serial terminal.

**RS-232**
The original standard which defined hardware serial communications. It has since been renamed to TIA-232.

When referring to communication data rates, this section does not use the term *baud*. Baud refers to the number of electrical state transitions made in a period of time, while bps is the correct term to use.

To connect a serial terminal to a FreeBSD system, a serial port on the computer and the proper cable to connect to the serial device are needed. Users who are already familiar with serial hardware and cabling can safely skip this section.

26.2.1. 序列线与埠

There are several different kinds of serial cables. The two most common types are null-modem cables and standard RS-232 cables. The documentation for the hardware should describe the type of cable required.

These two types of cables differ in how the wires are connected to the connector. Each wire represents a signal, with the defined signals summarized in [RS-232C 信号名称]. A standard serial 532...
A null-modem cable switches the "Transmitted Data" pin of the connector on one end with the "Received Data" pin on the other end. The connector can be either a DB-25 or a DB-9.

A null-modem cable can be constructed using the pin connections summarized in:

<table>
<thead>
<tr>
<th>DB-25</th>
<th>DB-25</th>
<th>Null-Modem Cable</th>
</tr>
</thead>
<tbody>
<tr>
<td>SG 7</td>
<td>SG 7</td>
<td>Connects to SG 7</td>
</tr>
<tr>
<td>TD 2</td>
<td>TD 3</td>
<td>Connects to TD 3</td>
</tr>
<tr>
<td>RD 3</td>
<td>RD 2</td>
<td>Connects to RD 2</td>
</tr>
<tr>
<td>RTS 4</td>
<td>RTS 5</td>
<td>Connects to RTS 5</td>
</tr>
<tr>
<td>CTS 5</td>
<td>CTS 4</td>
<td>Connects to CTS 4</td>
</tr>
<tr>
<td>DTR 20</td>
<td>DTR 6</td>
<td>Connects to DTR 6</td>
</tr>
<tr>
<td>DSR 6</td>
<td>DSR 20</td>
<td>Connects to DSR 20</td>
</tr>
<tr>
<td>DCD 8</td>
<td>DCD 1</td>
<td>Connects to DCD 1</td>
</tr>
<tr>
<td>SG 5</td>
<td>SG 5</td>
<td>Connects to SG 5</td>
</tr>
<tr>
<td>RTS 7</td>
<td>RTS 8</td>
<td>Connects to RTS 8</td>
</tr>
<tr>
<td>CTS 8</td>
<td>CTS 7</td>
<td>Connects to CTS 7</td>
</tr>
</tbody>
</table>

Table 19. RS-232C Signal Names

<table>
<thead>
<tr>
<th>Signal</th>
<th>Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>RD</td>
<td>2</td>
</tr>
<tr>
<td>TD</td>
<td>3</td>
</tr>
<tr>
<td>DTR</td>
<td>4</td>
</tr>
<tr>
<td>DSR</td>
<td>6</td>
</tr>
<tr>
<td>DCD</td>
<td>8</td>
</tr>
<tr>
<td>SG</td>
<td>5</td>
</tr>
<tr>
<td>RTS</td>
<td>7</td>
</tr>
<tr>
<td>CTS</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 20. DB-25 to DB-25 Null-Modem Line

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Pin #</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>7</td>
<td>SG</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>TD</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>RD</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>RTS</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>CTS</td>
</tr>
<tr>
<td>20</td>
<td>6</td>
<td>DTR</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>DSR</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>DCD</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>SG</td>
</tr>
</tbody>
</table>

Table 21. DB-9 to DB-9 Null-Modem Line

<table>
<thead>
<tr>
<th>Pin #</th>
<th>Pin #</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>3</td>
<td>RD</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>TD</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>DTR</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>DSR</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>SG</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>DTR</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>DCD</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>RTS</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>CTS</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>CTS</td>
</tr>
</tbody>
</table>
Serial ports are the devices through which data is transferred between the FreeBSD host computer and the terminal. Several kinds of serial ports exist. Before purchasing or constructing a cable, make sure it will fit the ports on the terminal and on the FreeBSD system.

Most terminals have DB-25 ports. Personal computers may have DB-25 or DB-9 ports. A multiport serial card may have RJ-12 or RJ-45/ ports. See the documentation that accompanied the hardware for specifications on the kind of port or visually verify the type of port.

In FreeBSD, each serial port is accessed through an entry in `/dev`. There are two different kinds of entries:

- Call-in ports are named `/dev/ttyuN` where `N` is the port number, starting from zero. If a terminal is connected to the first serial port (COM1), use `/dev/ttyu0` to refer to the terminal. If the terminal is on the second serial port (COM2), use `/dev/ttyu1`, and so forth. Generally, the call-in port is used for terminals. Call-in ports require that the serial line assert the “Data Carrier Detect” signal to work correctly.

- Call-out ports are named `/dev/cuauN` on FreeBSD versions 8.X and higher and `/dev/cuadN` on FreeBSD versions 7.X and lower. Call-out ports are usually not used for terminals, but are used for modems. The call-out port can be used if the serial cable or the terminal does not support the “Data Carrier Detect” signal.

FreeBSD also provides initialization devices (`/dev/ttyuN.init` and `/dev/cuauN.init` or `/dev/cuadN.init`) and locking devices (`/dev/ttyuN.lock` and `/dev/cuauN.lock` or `/dev/cuadN.lock`). The initialization devices are used to initialize communications port parameters each time a port is opened, such as `crtscts` for modems which use `RTS/CTS` signaling for flow control. The locking devices are used to lock flags on ports to prevent users or programs changing certain parameters. Refer to `termios(4)`, `sio(4)`, and `stty(1)` for information on terminal settings, locking and initializing devices, and setting terminal options, respectively.

By default, FreeBSD supports four serial ports which are commonly known as COM1, COM2, COM3, and COM4. FreeBSD also supports dumb multi-port serial interface cards, such as the BocaBoard 1008 and 2016, as well as more intelligent multi-port cards such as those made by Digiboard. However, the default kernel only looks for the standard COM ports.

To see if the system recognizes the serial ports, look for system boot messages that start with `uart`.
If the system does not recognize all of the needed serial ports, additional entries can be added to /boot/device.hints. This file already contains hint.uart.0.* entries for COM1 and hint.uart.1.* entries for COM2. When adding a port entry for COM3 use 0x3E8, and for COM4 use 0x2E8. Common IRQ addresses are 5 for COM3 and 9 for COM4.

To determine the default set of terminal I/O settings used by the port, specify its device name. This example determines the settings for the call-in port on COM2:

```
# stty -a -f /dev/ttyu1
```

System-wide initialization of serial devices is controlled by /etc/rc.d/serial. This file affects the default settings of serial devices. To change the settings for a device, use stty. By default, the changed settings are in effect until the device is closed and when the device is reopened, it goes back to the default set. To permanently change the default set, open and adjust the settings of the initialization device. For example, to turn on CLOCAL mode, 8 bit communication, and XON/XOFF flow control for ttyu5, type:

```
# stty -f /dev/ttyu5.init clocal cs8 ixon ixoff
```

To prevent certain settings from being changed by an application, make adjustments to the locking device. For example, to lock the speed of ttyu5 to 57600 bps, type:

```
# stty -f /dev/ttyu5.lock 57600
```

Now, any application that opens ttyu5 and tries to change the speed of the port will be stuck with 57600 bps.

26.3. 终端

Terminals provide a convenient and low-cost way to access a FreeBSD system when not at the computer's console or on a connected network. This section describes how to use terminals with FreeBSD.

The original UNIX™ systems did not have consoles. Instead, users logged in and ran programs through terminals that were connected to the computer's serial ports.

The ability to establish a login session on a serial port still exists in nearly every UNIX™-like operating system today, including FreeBSD. By using a terminal attached to an unused serial port, a user can log in and run any text program that can normally be run on the console or in an xterm window.

Many terminals can be attached to a FreeBSD system. An older spare computer can be used as a terminal wired into a more powerful computer running FreeBSD. This can turn what might otherwise be a single-user computer into a powerful multiple-user system.

FreeBSD supports three types of terminals:

- **Dumb terminals**

  Dumb terminals are specialized hardware that connect to computers over serial lines. They are called "dumb" because they have only enough computational power to display, send, and receive text. No programs can be run on these devices. Instead, dumb terminals connect to a computer that runs the needed programs.
There are hundreds of kinds of dumb terminals made by many manufacturers, and just about any kind will work with FreeBSD. Some high-end terminals can even display graphics, but only certain software packages can take advantage of these advanced features.

Dumb terminals are popular in work environments where workers do not need access to graphical applications.

Computers Acting as Terminals

Since a dumb terminal has just enough ability to display, send, and receive text, any spare computer can be a dumb terminal. All that is needed is the proper cable and some terminal emulation software to run on the computer.

This configuration can be useful. For example, if one user is busy working at the FreeBSD system's console, another user can do some text-only work at the same time from a less powerful personal computer hooked up as a terminal to the FreeBSD system.

There are at least two utilities in the base-system of FreeBSD that can be used to work through a serial connection: `cu(1)` and `tip(1)`.

For example, to connect from a client system that runs FreeBSD to the serial connection of another system:

```
# cu -l /dev/cuauN
```

Ports are numbered starting from zero. This means that `COM1` is `/dev/cuau0`.

Additional programs are available through the Ports Collection, such as `comms/minicom`.

X Terminals

X terminals are the most sophisticated kind of terminal available. Instead of connecting to a serial port, they usually connect to a network like Ethernet. Instead of being relegated to text-only applications, they can display any Xorg application.

This chapter does not cover the setup, configuration, or use of X terminals.

26.3.1. 終端機設定

This section describes how to configure a FreeBSD system to enable a login session on a serial terminal. It assumes that the system recognizes the serial port to which the terminal is connected and that the terminal is connected with the correct cable.

In FreeBSD, `init` reads `/etc/ttys` and starts a `getty` process on the available terminals. The `getty` process is responsible for reading a login name and starting the `login` program. The ports on the FreeBSD system which allow logins are listed in `/etc/ttys`. For example, the first virtual console, `ttyv0`, has an entry in this file, allowing logins on the console. This file also contains entries for the other virtual consoles, serial ports, and pseudo-ttys. For a hardwired terminal, the serial port's `/dev` entry is listed without the `/dev` part. For example, `/dev/ttyv0` is listed as `ttyv0`.

The default `/etc/ttys` configures support for the first four serial ports, `ttyu0` through `ttyu3`:

<table>
<thead>
<tr>
<th>Port</th>
<th>Command</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ttyu0</code></td>
<td><code>/usr/libexec/getty std.9600</code></td>
<td>dialup off secure</td>
</tr>
<tr>
<td><code>ttyu1</code></td>
<td><code>/usr/libexec/getty std.9600</code></td>
<td>dialup off secure</td>
</tr>
<tr>
<td><code>ttyu2</code></td>
<td><code>/usr/libexec/getty std.9600</code></td>
<td>dialup off secure</td>
</tr>
<tr>
<td><code>ttyu3</code></td>
<td><code>/usr/libexec/getty std.9600</code></td>
<td>dialup off secure</td>
</tr>
</tbody>
</table>

When attaching a terminal to one of those ports, modify the default entry to set the required speed and terminal type, to turn the device on and, if needed, to change the port's secure setting. If the...
Even with the most meticulous attention to detail, something could still go wrong while setting up a terminal. Here is a list of common symptoms and some suggested fixes.

If no login prompt appears, make sure the terminal is plugged in and powered up. If it is a personal computer acting as a terminal, make sure it is running terminal emulation software on the correct serial port.
Make sure the cable is connected firmly to both the terminal and the FreeBSD computer. Make sure it is the right kind of cable. Make sure the terminal and FreeBSD agree on the bps rate and parity settings. For a video display terminal, make sure the contrast and brightness controls are turned up. If it is a printing terminal, make sure paper and ink are in good supply.

Use `ps` to make sure that a `getty` process is running and serving the terminal. For example, the following listing shows that a `getty` is running on the second serial port, `ttyu1`, and is using the `std.38400` entry in `/etc/gettytab`:

```
# ps -axww|grep ttyu
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyu1
```

If no `getty` process is running, make sure the port is enabled in `/etc/ttys`. Remember to run `kill -HUP 1` after modifying `/etc/ttys`.

If the `getty` process is running but the terminal still does not display a login prompt, or if it displays a prompt but will not accept typed input, the terminal or cable may not support hardware handshaking. Try changing the entry in `/etc/ttys` from `std.38400` to `3wire.38400`, then run `kill -HUP 1` after modifying `/etc/ttys`. The `3wire` entry is similar to `std`, but ignores hardware handshaking. The baud rate may need to be reduced or software flow control enabled when using `3wire` to prevent buffer overflows.

If garbage appears instead of a login prompt, make sure the terminal and FreeBSD agree on the bps rate and parity settings. Check the `getty` processes to make sure the correct `getty` type is in use. If not, edit `/etc/ttys` and run `kill -HUP 1`.

If characters appear doubled and the password appears when typed, switch the terminal, or the terminal emulation software, from "half duplex" or "local echo" to "full duplex."

26.4. 撥入服務

Configuring a FreeBSD system for dial-in service is similar to configuring terminals, except that modems are used instead of terminal devices. FreeBSD supports both external and internal modems.

External modems are more convenient because they often can be configured via parameters stored in non-volatile RAM and they usually provide lighted indicators that display the state of important RS-232 signals, indicating whether the modem is operating properly.

Internal modems usually lack non-volatile RAM, so their configuration may be limited to setting DIP switches. If the internal modem has any signal indicator lights, they are difficult to view when the system's cover is in place.

When using an external modem, a proper cable is needed. A standard RS-232C serial cable should suffice. FreeBSD needs the RTS and CTS signals for flow control at speeds above 2400 bps, the CD signal to detect when a call has been answered or the line has been hung up, and the DTR signal to reset the modem after a session is complete. Some cables are wired without all of the needed signals, so if a login session does not go away when the line hangs up, there may be a problem with the cable. Refer to 序列線與埠 for more information about these signals.

Like other UNIX™-like operating systems, FreeBSD uses the hardware signals to find out when a call has been answered or a line has been hung up and to hangup and reset the modem after a call. FreeBSD avoids sending commands to the modem or watching for status reports from the modem. FreeBSD supports the NS8250, NS16450, NS16550, and NS16550A-based RS-232C (CCITT V.24) communications interfaces. The 8250 and 16450 devices have single-character buffers. The 16550 devices provide multi-character buffers.
device provides a 16-character buffer, which allows for better system performance. Bugs in plain 16550 devices prevent the use of the 16-character buffer, so use 16550A devices if possible. Because single-character-buffer devices require more work by the operating system than the 16-character-buffer devices, 16550A-based serial interface cards are preferred. If the system has many active serial ports or will have a heavy load, 16550A-based cards are better for low-error-rate communications.

The rest of this section demonstrates how to configure a modem to receive incoming connections, how to communicate with the modem, and offers some troubleshooting tips.

26.4.1. **數據機設定**

As with terminals, `init` spawns a `getty` process for each configured serial port used for dial-in connections. When a user dials the modem's line and the modems connect, the "Carrier Detect" signal is reported by the modem. The kernel notices that the carrier has been detected and instructs `getty` to open the port and display a `login:` prompt at the specified initial line speed. In a typical configuration, if garbage characters are received, usually due to the modem's connection speed being different than the configured speed, `getty` tries adjusting the line speeds until it receives reasonable characters. After the user enters their login name, `getty` executes `login`, which completes the login process by asking for the user's password and then starting the user's shell.

There are two schools of thought regarding dial-up modems. One configuration method is to set the modems and systems so that no matter at what speed a remote user dials in, the dial-in RS-232 interface runs at a locked speed. The benefit of this configuration is that the remote user always sees a system login prompt immediately. The downside is that the system does not know what a user's true data rate is, so full-screen programs like Emacs will not adjust their screen-painting methods to make their response better for slower connections.

The second method is to configure the RS-232 interface to vary its speed based on the remote user's connection speed. Because `getty` does not understand any particular modem's connection speed reporting, it gives a `login:` message at an initial speed and watches the characters that come back in response. If the user sees junk, they should press `Enter` until they see a recognizable prompt. If the data rates do not match, `getty` sees anything the user types as junk, tries the next speed, and gives the `login:` prompt again. This procedure normally only takes a keystroke or two before the user sees a good prompt. This login sequence does not look as clean as the locked-speed method, but a user on a low-speed connection should receive better interactive response from full-screen programs.

When locking a modem's data communications rate at a particular speed, no changes to `/etc/gettytab` should be needed. However, for a matching-speed configuration, additional entries may be required in order to define the speeds to use for the modem. This example configures a 14.4 Kbps modem with a top interface speed of 19.2 Kbps using 8-bit, no parity connections. It configures `getty` to start the communications rate for a V.32bis connection at 19.2 Kbps, then cycles through 9600 bps, 2400 bps, 1200 bps, 300 bps, and back to 19.2 Kbps. Communications rate cycling is implemented with the `nx=` capability. Each line uses a `tc=` (table continuation) entry to pick up the rest of the settings for a particular data rate.

```
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:
:nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:
:nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:
:nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:
```
For a 28.8 Kbps modem, or to take advantage of compression on a 14.4 Kbps modem, use a higher communications rate, as seen in this example:

```
# Additions for a V.32bis or V.34 Modem
# Starting at 57.6 Kbps
```
High-speed modems, like V.32, V.32bis, and V.34 modems, use hardware (RTS/CTS) flow control. Use `stty` to set the hardware flow control flag for the modem port. This example sets the `crtscts` flag on COM2’s dial-in and dial-out initialization devices:

```
# stty -f /dev/ttyu1.init crtscts
# stty -f /dev/cuau1.init crtscts
```

26.4.2. **Troubleshooting a Dial-up Modem**

This section provides a few tips for troubleshooting a dial-up modem that will not connect to a FreeBSD system.

Hook up the modem to the FreeBSD system and boot the system. If the modem has status indication lights, watch to see whether the modem’s DTR indicator lights when the login: prompt appears on the system’s console. If it lights up, that should mean that FreeBSD has started a `getty` process on the appropriate communications port and is waiting for the modem to accept a call. If the DTR indicator does not light, login to the FreeBSD system through the console and type `ps ax` to see if FreeBSD is running a `getty` process on the correct port:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu0
```

If the second column contains a `d0` instead of a `??` and the modem has not accepted a call yet, this means that `getty` has completed its open on the communications port. This could indicate a problem with the cabling or a misconfigured modem because `getty` should not be able to open the communications port until the carrier detect signal has been asserted by the modem.

If no `getty` processes are waiting to open the port, double-check that the entry for the port is correct in `/etc/ttys`. Also, check `/var/log/messages` to see if there are any log messages from `init` or `getty`.

Next, try dialing into the system. Be sure to use 8 bits, no parity, and 1 stop bit on the remote system. If a prompt does not appear right away, or the prompt shows garbage, try pressing `Enter` about once per second. If there is still no login: prompt, try sending a `BREAK`. When using a high-speed modem, try dialing again after locking the dialing modem’s interface speed. If there is still no login: prompt, check `/etc/gettytab` again and double-check that:

- The initial capability name specified in the entry in `/etc/ttys` matches the name of a capability in `/etc/gettytab`.
- Each `nx=` entry matches another `gettytab` capability name.
- Each `tc=` entry matches another `gettytab` capability name.

If the modem on the FreeBSD system will not answer, make sure that the modem is configured to answer the phone when DTR is asserted. If the modem seems to be configured correctly, verify that the DTR line is asserted by checking the modem’s indicator lights. If it still does not work, try sending an email to the FreeBSD general questions mailing list describing the modem and the problem.

26.5. **Dial-out Service**

The following are tips for getting the host to connect over the modem to another computer. This is
appropriate for establishing a terminal session with a remote host. This kind of connection can be helpful to get a file on the Internet if there are problems using PPP. If PPP is not working, use the terminal session to FTP the needed file. Then use zmodem to transfer it to the machine.

26.5.1. 使用 Stock Hayes 数据机

A generic Hayes dialer is built into `tip`. Use `at=hayes` in `/etc/remote`.

The Hayes driver is not smart enough to recognize some of the advanced features of newer modems messages like `BUSY`, `NO DIALTONE`, or `CONNECT 115200`. Turn those messages off when using `tip` with `ATX0&W`.

The dial timeout for `tip` is 60 seconds. The modem should use something less, or else `tip` will think there is a communication problem. Try `ATS7=45&W`.

26.5.2. 使用 AT 指令

Create a “direct” entry in `/etc/remote`. For example, if the modem is hooked up to the first serial port, `/dev/cuau0`, use the following line:

```
cuau0:dv=/dev/cuau0:br#19200:pa=none
```

Use the highest bps rate the modem supports in the `br` capability. Then, type `tip cuau0` to connect to the modem.

Or, use `cu` as root with the following command:

```
# cu -lline -sspeed
```

`line` is the serial port, such as `/dev/cuau0`, and `speed` is the speed, such as `57600`. When finished entering the AT commands, type `~.` to exit.

26.5.3. @ 符号无法运作

The `@` sign in the phone number capability tells `tip` to look in `/etc/phones` for a phone number. But, the `@` sign is also a special character in capability files like `/etc/remote`, so it needs to be escaped with a backslash:

```
pn=@
```

26.5.4. 从指令列拨号

Put a “generic” entry in `/etc/remote`. For example:

```
tip115200| Dial any phone number at 115200 bps:  
:dv=/dev/cuau0:br#115200:at=hayes:pa=none:du:
tip57600| Dial any phone number at 57600 bps:  
:dv=/dev/cuau0:br#57600:at=hayes:pa=none:du:
```

This should now work:

```
542
```
Users who prefer *cu* over *tip*, can use a generic *cu* entry:

```
# cu115200|Use cu to dial any number at 115200bps:
   :dv=/dev/cuau1:br#57600:at=hayes:pa=none:du:
   and type:
   # cu 5551234 -s 115200
```

### 26.5.5. 設定 bps 率

Put in an entry for *tip1200* or *cu1200*, but go ahead and use whatever bps rate is appropriate with the *br* capability. *tip* thinks a good default is 1200 bps which is why it looks for a *tip1200* entry. 1200 bps does not have to be used, though.

### 26.5.6. 透過終端伺服器存取多個主機

Rather than waiting until connected and typing `CONNECT host` each time, use *tip*’s *cm* capability. For example, these entries in `/etc/remote` will let you type `tip pain` or `tip muffin` to connect to the hosts `pain` or `muffin`, and `tip deep13` to connect to the terminal server.

```
pain|pain.deep13.com|Forrester's machine:
   :cm=CONNECT pain
   :tc=deep13:
muffin|muffin.deep13.com|Frank's machine:
   :cm=CONNECT muffin
   :tc=deep13:
depth13:Gizmonics Institute terminal server:
   :dv=/dev/cuau2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

### 26.5.7. 在 *tip* 使用超過一行

This is often a problem where a university has several modem lines and several thousand students trying to use them. Make an entry in `/etc/remote` and use `@` for the *pn* capability:

```
big-university: pn=@:tc=dialout
dialout: dv=/dev/cuau3:br#9600:at=courier:du:pa=none:
```

Then, list the phone numbers in `/etc/phones`:

```
big-university 5551111
big-university 5551112
```
tip will try each number in the listed order, then give up. To keep retrying, run tip in a while loop.

26.5.8. 使用強制字元 Ctrl+P is the default "force" character, used to tell tip that the next character is literal data. The force character can be set to any other character with the ~s escape, which means "set a variable.

Type ~sforce= single-char followed by a newline. single-char is any single character. If single-char is left out, then the force character is the null character, which is accessed by typing Ctrl+2 or Ctrl+Space. A pretty good value for single-char is Shift+Ctrl+6, which is only used on some terminal servers.

To change the force character, specify the following in ~/.tiprc:

```
force=single-char
```

26.5.9. 大寫字元 This happens when Ctrl+A is pressed, which is tip's "raise character", specially designed for people with broken caps-lock keys. Use ~s to set raisechar to something reasonable. It can be set to be the same as the force character, if neither feature is used.

Here is a sample ~/.tiprc for Emacs users who need to type Ctrl+2 and Ctrl+A:

```
force=^^
raisechar=^^
```
The ^^ is Shift+Ctrl+6.

26.5.10. 使用tip傳輸檔 When talking to another UNIX™-like operating system, files can be sent and received using ~p (put) and ~t (take). These commands run cat and echo on the remote system to accept and send files.

The syntax is:

```
~p local-file [ remote-file ]
~t remote-file [ local-file ]
```

There is no error checking, so another protocol, like zmodem, should probably be used.

26.5.11. 在zmodem使用tip?

To receive files, start the sending program on the remote end. Then, type ~C rz to begin receiving them locally.

To send files, start the receiving program on the remote end. Then, type ~C sz files to send them to the remote system.
FreeBSD has the ability to boot a system with a dumb terminal on a serial port as a console. This configuration is useful for system administrators who wish to install FreeBSD on machines that have no keyboard or monitor attached, and developers who want to debug the kernel or device drivers.

As described in FreeBSD開機程序, FreeBSD employs a three stage bootstrap. The first two stages are in the boot block code which is stored at the beginning of the FreeBSD slice on the boot disk. The boot block then loads and runs the boot loader as the third stage code.

In order to set up booting from a serial console, the boot block code, the boot loader code, and the kernel need to be configured.

26.6.1. 快速序列 Console 設定

This section provides a fast overview of setting up the serial console. This procedure can be used when the dumb terminal is connected to COM1.

Procedure: Configuring a Serial Console on COM1

1. Connect the serial cable to COM1 and the controlling terminal.
2. To configure boot messages to display on the serial console, issue the following command as the superuser:
   ```
   sysrc -f /boot/loader.conf console=comconsole
   ```
3. Edit /etc/ttys and change off to on and dialup to vt100 for the ttyu0 entry. Otherwise, a password will not be required to connect via the serial console, resulting in a potential security hole.
4. Reboot the system to see if the changes took effect.

If a different configuration is required, see the next section for a more in-depth configuration explanation.

26.6.2. 深入序列 Console 設定

This section provides a more detailed explanation of the steps needed to setup a serial console in FreeBSD.

Procedure: Configuring a Serial Console

1. Prepare a serial cable.
   Use either a null-modem cable or a standard serial cable and a null-modem adapter. See 繞線與埠 for a discussion on serial cables.
2. Unplug the keyboard.
   Many systems probe for the keyboard during the Power-On Self-Test (POST) and will generate an error if the keyboard is not detected. Some machines will refuse to boot until the keyboard is plugged in.
   If the computer complains about the error, but boots anyway, no further configuration is needed.
If the computer refuses to boot without a keyboard attached, configure the BIOS so that it ignores this error. Consult the motherboard's manual for details on how to do this.

Try setting the keyboard to “Not installed” in the BIOS. This setting tells the BIOS not to probe for a keyboard at power-on so it should not complain if the keyboard is absent. If that option is not present in the BIOS, look for an “Halt on Error” option instead. Setting this to “All but Keyboard” or to “No Errors” will have the same effect.

If the system has a PS/2™ mouse, unplug it as well. PS/2™ mice share some hardware with the keyboard and leaving the mouse plugged in can fool the keyboard probe into thinking the keyboard is still there.

While most systems will boot without a keyboard, quite a few will not boot without a graphics adapter. Some systems can be configured to boot with no graphics adapter by changing the “graphics adapter” setting in the BIOS configuration to “Not installed”. Other systems do not support this option and will refuse to boot if there is no display hardware in the system. With these machines, leave some kind of graphics card plugged in, even if it is just a junky mono board. A monitor does not need to be attached.

3. Plug a dumb terminal, an old computer with a modem program, or the serial port on another UNIX™ box into the serial port.

4. Add the appropriate `hint.sio.*` entries to `/boot/device.hints` for the serial port. Some multi-port cards also require kernel configuration options. Refer to `sio(4)` for the required options and device hints for each supported serial port.

5. Create `boot.config` in the root directory of the a partition on the boot drive. This file instructs the boot block code how to boot the system. In order to activate the serial console, one or more of the following options are needed. When using multiple options, include them all on the same line:

   - `-h` Toggles between the internal and serial consoles. Use this to switch console devices. For instance, to boot from the internal (video) console, use `-h` to direct the boot loader and the kernel to use the serial port as its console device. Alternatively, to boot from the serial port, use `-h` to tell the boot loader and the kernel to use the video display as the console instead.

   - `-D` Toggles between the single and dual console configurations. In the single configuration, the console will be either the internal console (video display) or the serial port, depending on the state of `-h`. In the dual console configuration, both the video display and the serial port will become the console at the same time, regardless of the state of `-h`. However, the dual console configuration takes effect only while the boot block is running. Once the boot loader gets control, the console specified by `-h` becomes the only console.

   - `-P` Makes the boot block probe the keyboard. If no keyboard is found, the `-D` and `-h` options are automatically set. Due to space constraints in the current version of the boot blocks, `-P` is capable of detecting extended keyboards only. Keyboards with less than 101 keys and without F11 and F12 keys may not be detected. Keyboards on some laptops may not be properly found because of this limitation. If this is the case, do not use `-P`.
Use either -P to select the console automatically or -h to activate the serial console. Refer to boot(8) and boot.config(5) for more details.

The options, except for -P, are passed to the boot loader. The boot loader will determine whether the internal video or the serial port should become the console by examining -h. This means that if -D is specified but -h is not specified in /boot.config, the serial port can be used as the console only during the boot block as the boot loader will use the internal video display as the console.

6. Boot the machine. When FreeBSD starts, the boot blocks echo the contents of /boot.config to the console. For example:

```
/boot.config:
-P
```

The second line appears only if -P is in /boot.config and indicates the presence or absence of the keyboard. These messages go to either the serial or internal console, or both, depending on the option in /boot.config:

Options Message goes to
none internal console
-h serial console
-D serial and internal consoles
-Dh serial and internal consoles
-P, keyboard present internal console
-P, keyboard absent serial console

After the message, there will be a small pause before the boot blocks continue loading the boot loader and before any further messages are printed to the console. Under normal circumstances, there is no need to interrupt the boot blocks, but one can do so in order to make sure things are set up correctly.

Press any key, other than Enter, at the console to interrupt the boot process. The boot blocks will then prompt for further action:

```
>>> FreeBSD/i386 BOOT
Default: 0:ad (0,a) /boot/loader
```

Verify that the above message appears on either the serial or internal console, or both, according to the options in /boot.config. If the message appears in the correct console, press Enter to continue the boot process.

If there is no prompt on the serial terminal, something is wrong with the settings. Enter -h then Enter or Return to tell the boot block (and then the boot loader and the kernel) to choose the serial port for the console. Once the system is up, go back and check what went wrong.

During the third stage of the boot process, one can still switch between the internal console and the serial console by setting appropriate environment variables in the boot loader. See loader(8) for more information.
This line in `/boot/loader.conf` or `/boot/loader.conf.local` configures the boot loader and the kernel to send their boot messages to the serial console, regardless of the options in `/boot.config`:

```
console="comconsole"
```

That line should be the first line of `/boot/loader.conf` so that boot messages are displayed on the serial console as early as possible.

If that line does not exist, or if it is set to `console="vidconsole"`, the boot loader and the kernel will use whichever console is indicated by `-h` in the boot block. See `loader.conf(5)` for more information.

At the moment, the boot loader has no option equivalent to `-P` in the boot block, and there is no provision to automatically select the internal console and the serial console based on the presence of the keyboard.

While it is not required, it is possible to provide a login prompt over the serial line. To configure this, edit the entry for the serial port in `/etc/ttys` using the instructions in `終端機設定`. If the speed of the serial port has been changed, change `std.9600` to match the new setting.

### 26.6.3. 設定使用更快的序列埠速度

By default, the serial port settings are 9600 baud, 8 bits, no parity, and 1 stop bit. To change the default console speed, use one of the following options:

- Edit `/etc/make.conf` and set `BOOT_COMCONSOLE_SPEED` to the new console speed. Then, recompile and install the boot blocks and the boot loader:

  ```
  # cd /sys/boot
  # make clean
  # make
  # make install
  ```

  If the serial console is configured in some other way than by booting with `-h`, or if the serial console used by the kernel is different from the one used by the boot blocks, add the following option, with the desired speed, to a custom kernel configuration file and compile a new kernel:

  ```
  options CONSPEED=19200
  ```

- Add the `-S_19200_` boot option to `/boot.config`, replacing `19200` with the speed to use.

- Add the following options to `/boot/loader.conf`. Replace `115200` with the speed to use.

  ```
  boot_multicons="YES"
  boot_serial="YES"
  comconsole_speed="115200"
  console="comconsole,vidconsole"
  ```
To configure the ability to drop into the kernel debugger from the serial console, add the following options to a custom kernel configuration file and compile the kernel using the instructions in FreeBSD core. Note that while this is useful for remote diagnostics, it is also dangerous if a spurious BREAK is generated on the serial port. Refer to `ddb(4)` and `ddb(8)` for more information about the kernel debugger.

```
options BREAK_TO_DEBUGGER
options DDB
```
27. PPP

27.1. 概述

FreeBSD 支援点对点(Point-to-Point, PPP)通讯协定，可透过拨号数据机用来建立网路或网际网路连线。本章将说明如何设定在 FreeBSD 中以数据机为基础的通讯服务。

读完这章，您将了解：
• 如何设定、使用 PPP 连线及排除问题。
• 如何设定在乙太网路(Ethernet)上的 PPP (PPPoE)。
• 如何设定在 ATM 上的 PPP (PPPoA)。

在开始阅读这章之前，您需要：
• 熟悉基本网路术语。
• 了解拨号连线及 PPP 的基础及目的。

27.2. 設定 PPP

FreeBSD provides built-in support for managing dial-up PPP connections using ppp(8). The default FreeBSD kernel provides support for tun which is used to interact with a modem hardware.

Configuration is performed by editing at least one configuration file, and configuration files containing examples are provided. Finally, ppp is used to start and manage connections.

In order to use a PPP connection, the following items are needed:
• A dial-up account with an Internet Service Provider (ISP).
• A dial-up modem.
• The dial-up number for the ISP.
• The login name and password assigned by the ISP.
• The IP address of one or more DNS servers. Normally, the ISP provides these addresses. If it did not, FreeBSD can be configured to use DNS negotiation.

If any of the required information is missing, contact the ISP.

The following information may be supplied by the ISP, but is not necessary:
• The IP address of the default gateway. If this information is unknown, the ISP will automatically provide the correct value during connection setup. When configuring PPP on FreeBSD, this address is referred to as HISADDR.
• The subnet mask. If the ISP has not provided one, 255.255.255.255 will be used in the ppp(8) configuration file. * If the ISP has assigned a static IP address and hostname, it should be input into the configuration file. Otherwise, this information will be automatically provided during connection setup.

The rest of this section demonstrates how to configure FreeBSD for common PPP connection scenarios. The required configuration file is /etc/ppp/ppp.conf and additional files and examples are available in /usr/shared/examples/ppp/.

Throughout this section, many of the file examples display line numbers. These line numbers have been added to make it easier to follow the discussion and are not meant to be placed in the actual file.
When editing a configuration file, proper indentation is important. Lines that end in a colon (`:`) should start in the first column (beginning of the line) while all other lines should be indented as shown using spaces or tabs.

### 27.2.1 基礎設定

In order to configure a PPP connection, first edit `/etc/ppp/ppp.conf` with the dial-in information for the ISP. This file is described as follows:

```plaintext
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION
4       set device /dev/cuau0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\sCARRIER TIMEOUT 5 \n                 "" AT OK-AT-OK ATE1Q0 OK \dATDT\T TIMEOUT 40 CONNECT"
7       set timeout 180
8       enable dns
9
10    provider:
11      set phone "(123) 456 7890"
12      set authname foo
13      set authkey bar
14      set timeout 300
15      set ifaddr x.x.x.x/0 y.y.y.y/0 255.255.255.255 0.0.0.0
16      add default HISADDR
```

- **Line 1**: Identifies the default entry. Commands in this entry (lines 2 through 9) are executed automatically when `ppp` is run.
- **Line 2**: Enables verbose logging parameters for testing the connection. Once the configuration is working satisfactorily, this line should be reduced to:
  ```plaintext
  set log phase tun
  ```
- **Line 3**: Displays the version of `ppp(8)` to the PPP software running on the other side of the connection.
- **Line 4**: Identifies the device to which the modem is connected, where `COM1` is `/dev/cuau0` and `COM2` is `/dev/cuau1`.
- **Line 5**: Sets the connection speed. If `115200` does not work on an older modem, try `38400` instead.
- **Lines 6 & 7**: The dial string written as an expect-send syntax. Refer to `chat(8)` for more information.
Note that this command continues onto the next line for readability. Any command in `ppp.conf` may do this if the last character on the line is \
.

Line 8
Sets the idle timeout for the link in seconds.

Line 9
Instructs the peer to confirm the DNS settings. If the local network is running its own DNS server, this line should be commented out, by adding a # at the beginning of the line, or removed.

Line 10
A blank line for readability. Blank lines are ignored by `ppp(8)`.

Line 11
Identifies an entry called `provider`. This could be changed to the name of the ISP so that `load ISP` can be used to start the connection.

Line 12
Use the phone number for the ISP. Multiple phone numbers may be specified using the colon (:) or pipe character (|) as a separator. To rotate through the numbers, use a colon. To always attempt to dial the first number first and only use the other numbers if the first number fails, use the pipe character. Always enclose the entire set of phone numbers between quotation marks ("") to prevent dialing failures.

Lines 13 & 14
Use the user name and password for the ISP.

Line 15
Sets the default idle timeout in seconds for the connection. In this example, the connection will be closed automatically after 300 seconds of inactivity. To prevent a timeout, set this value to zero.

Line 16
Sets the interface addresses. The values used depend upon whether a static IP address has been obtained from the ISP or if it instead negotiates a dynamic IP address during connection. If the ISP has allocated a static IP address and default gateway, replace x.x.x.x with the static IP address and replace y.y.y.y with the IP address of the default gateway. If the ISP has only provided a static IP address without a gateway address, replace y.y.y.y with 10.0.0.2/0.

If the IP address changes whenever a connection is made, change this line to the following value. This tells `ppp(8)` to use the IP Configuration Protocol (IPCP) to negotiate a dynamic IP address:

```
set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255 0.0.0.0
```

Line 17
Keep this line as-is as it adds a default route to the gateway. The HISADDR will automatically be replaced with the gateway address specified on line 16. It is important that this line appears after line 16.

Depending upon whether `ppp(8)` is started manually or automatically, a `/etc/ppp/ppp.linkup` may also need to be created which contains the following lines. This file is required when running `ppp` in -auto mode. This file is used after the connection has been established. At this point, the IP address will have been assigned and it is now be possible to add the routing table entries. When creating this file, make sure that provider matches the value demonstrated in line 11 of `ppp.conf`.

```
provider: 552
```
add default HISADDR

This file is also needed when the default gateway address is "guessed" in a static IP address configuration. In this case, remove line 17 from /etc/ppp/ppp.conf and create /etc/ppp/ppp.linkup with the above two lines. More examples for this file can be found in /usr/shared/examples/ppp/.

By default, ppp must be run as root. To change this default, add the account of the user who should run ppp to the network group in /etc/group. Then, give the user access to one or more entries in /etc/ppp/ppp.conf with allow. For example, to give fred and mary permission to only the provider: entry, add this line to the provider: section:

```
allow users fred mary
```

To give the specified users access to all entries, put that line in the default section instead.

27.2.2. 階段設定

It is possible to configure PPP to supply DNS and NetBIOS nameserver addresses on demand. To enable these extensions with PPP version 1.x, the following lines might be added to the relevant section of /etc/ppp/ppp.conf.

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

And for PPP version 2 and above:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

This will tell the clients the primary and secondary name server addresses, and a NetBIOS nameserver host.

In version 2 and above, if the set dns line is omitted, PPP will use the values found in /etc/resolv.conf.

27.2.2.1. PAP と CHAP 認証

Some ISPs set their system up so that the authentication part of the connection is done using either of the PAP or CHAP authentication mechanisms. If this is the case, the ISP will not give a login: prompt at connection, but will start talking PPP immediately.

PAP is less secure than CHAP, but security is not normally an issue here as passwords, although being sent as plain text with PAP, are being transmitted down a serial line only. There is not much room for crackers to "eavesdrop".

The following alterations must be made:

```
13      set authname MyUserName
```

553
This line specifies the PAP/CHAP user name. Insert the correct value for MyUserName.

This line specifies the PAP/CHAP password. Insert the correct value for MyPassword. You may want to add an additional line, such as:

```
accept PAP
```

or

```
accept CHAP
```

This functionality may be enabled by the following line in `/etc/ppp/ppp.conf`:

```
nat enable yes
```

Alternatively, NAT may be enabled by command-line option `-nat`. There is also `/etc/rc.conf` knob `ppp_nat`, which is enabled by default.

When using this feature, it may be useful to include the following `/etc/ppp/ppp.conf` options to enable incoming connections forwarding:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

or do not trust the outside at all

```
nat deny_incoming yes
```

While ppp is now configured, some edits still need to be made to `/etc/rc.conf`. Working from the top down in this file, make sure the `hostname=` line is set:
If the ISP has supplied a static IP address and name, use this name as the host name. Look for the `network_interfaces` variable. To configure the system to dial the ISP on demand, make sure the `tun0` device is added to the list, otherwise remove it.

```
network_interfaces="lo0 tun0"
```

The `ifconfig_tun0` variable should be empty, and a file called `/etc/start_if.tun0` should be created. This file should contain the line:

```
ppp -auto mysystem
```

This script is executed at network configuration time, starting the ppp daemon in automatic mode. If this machine acts as a gateway, consider including `-alias`. Refer to the manual page for further details.

Make sure that the router program is set to `NO` with the following line in `/etc/rc.conf`:

```
router_enable="NO"
```

It is important that the `routed` daemon is not started, as `routed` tends to delete the default routing table entries created by `ppp`.

It is probably a good idea to ensure that the `sendmail_flags` line does not include the `-q` option, otherwise `sendmail` will attempt to do a network lookup every now and then, possibly causing your machine to dial out. You may try:

```
sendmail_flags="-bd"
```

The downside is that `sendmail` is forced to re-examine the mail queue whenever the ppp link. To automate this, include `!bg` in `ppp.linkup`:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

An alternative is to set up a “dfilter” to block SMTP traffic. Refer to the sample files for further details.

27.2.4.

All that is left is to reboot the machine. After rebooting, either type:

```
555
```
and then dial provider to start the PPP session, or, to configure ppp to establish sessions automatically when there is outbound traffic and start_if.tun0 does not exist, type:

```
# ppp -auto provider
```

It is possible to talk to the ppp program while it is running in the background, but only if a suitable diagnostic port has been set up. To do this, add the following line to the configuration:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

This will tell PPP to listen to the specified UNIX™ domain socket, asking clients for the specified password before allowing access. The %d in the name is replaced with the tun device number that is in use.

Once a socket has been set up, the pppctl(8) program may be used in scripts that wish to manipulate the running program.

27.2.5. 設定撥入服務 provides a good description on enabling dial-up services using getty(8).

An alternative to getty is comms/mgetty+sendfax port, a smarter version of getty designed with dial-up lines in mind.

The advantages of using mgetty is that it actively talks to modems, meaning if port is turned off in /etc/ttys then the modem will not answer the phone.

Later versions of mgetty (from 0.99beta onwards) also support the automatic detection of PPP streams, allowing clients scriptless access to the server.

Refer to http://mgetty.greenie.net/doc/mgetty_toc.html for more information on mgetty.

By default the comms/mgetty+sendfax port comes with the AUTO_PPP option enabled allowing mgetty to detect the LCP phase of PPP connections and automatically spawn off a ppp shell. However, since the default login/password sequence does not occur it is necessary to authenticate users using either PAP or CHAP.

This section assumes the user has successfully compiled, and installed the comms/mgetty+sendfax port on his system.

Ensure that /usr/local/etc/mgetty+sendfax/login.config has the following:

```
/AutoPPP/ -     - /etc/ppp/ppp-pap-dialup
```

This tells mgetty to run ppp-pap-dialup for detected PPP connections.

Create an executable file called /etc/ppp/ppp-pap-dialup containing the following:

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

556
For each dial-up line enabled in /etc/ttys, create a corresponding entry in /etc/ppp/ppp.conf. This will happily co-exist with the definitions we created above.

pap:
enable pap
set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
enable proxy

Each user logging in with this method will need to have a username/password in /etc/ppp/ppp.secret, or alternatively add the following option to authenticate users via PAP from /etc/passwd.

enable passwdauth

To assign some users a static IP number, specify the number as the third argument in /etc/ppp/ppp.secret. See /usr/shared/examples/ppp/ppp.secret.sample for examples.

27.3. PPP 障難排除

This section covers a few issues which may arise when using PPP over a modem connection. Some ISPs present the “ssword” prompt while others present “password”. If the ppp script is not written accordingly, the login attempt will fail. The most common way to debug ppp connections is by connecting manually as described in this section.

27.3.1. 檢查裝置節點

When using a custom kernel, make sure to include the following line in the kernel configuration file:

device   uart

The “uart” device is already included in the GENERIC kernel, so no additional steps are necessary in this case. Just check the dmesg output for the modem device with:

# dmesg | grep uart

This should display some pertinent output about the “uart” devices. These are the COM ports we need. If the modem acts like a standard serial port, it should be listed on “uart1”, or “COM2”. If so, a kernel rebuild is not required. When matching up, if the modem is on “uart1”, the modem device would be /dev/cuau1.

27.3.2. 手動連線

Connecting to the Internet by manually controlling ppp is quick, easy, and a great way to debug a connection or just get information on how the ISP treats ppp client connections. Let's start PPP from the command line. Note that in all of our examples we will use example as the hostname of the machine running PPP. To start ppp:

# ppp
This second command sets the modem device to /dev/cuau1.

```
set speed 115200
```

This sets the connection speed to 115,200 kbps.

```
enable dns
```

This tells ppp to configure the resolver and add the nameserver lines to /etc/resolv.conf. If ppp cannot determine the hostname, it can manually be set later.

```
term
```

This switches to "terminal" mode in order to manually control the modem.

```
deflink: Entering terminal mode on /dev/cuau1
type '~h' for help
```

```
at
```

Use `at` to initialize the modem, then use `atdt` and the number for the ISP to begin the dial in process.

```
CONNECT
```

Confirmation of the connection, if we are going to have any connection problems, unrelated to hardware, here is where we will attempt to resolve them.

```
ISP Login:myusername
```

At this prompt, return the prompt with the username that was provided by the ISP.

```
ISP Pass:mypassword
```

At this prompt, reply with the password that was provided by the ISP. Just like logging into FreeBSD, the password will not echo.

Shell or PPP: ppp
Depending on the ISP, this prompt might not appear. If it does, it is asking whether to use a shell on the provider or to start ppp. In this example, ppp was selected in order to establish an Internet connection.

Notice that in this example the first p has been capitalized. This shows that we have successfully connected to the ISP.

We have successfully authenticated with our ISP and are waiting for the assigned IP address.

We have made an agreement on an IP address and successfully completed our connection.

add default HISADDR

Here we add our default route, we need to do this before we can talk to the outside world as currently the only established connection is with the peer. If this fails due to existing routes, put a bang character ! in front of the add. Alternatively, set this before making the actual connection and it will negotiate a new route accordingly.

If everything went good we should now have an active connection to the Internet, which could be thrown into the background using CTRL+z. If PPP returns to ppp then the connection has been lost. This is good to know because it shows the connection status. Capital P's represent a connection to the ISP and lowercase p's show that the connection has been lost.

27.3.3. 除錯

If a connection cannot be established, turn hardware flow CTS/RTS to off using set ctsrts off. This is mainly the case when connected to some PPP-capable terminal servers, where PPP hangs when it tries to write data to the communication link, and waits for a Clear To Send (CTS) signal which may never come. When using this option, include set accmap as it may be required to defeat hardware dependent on passing certain characters from end to end, most of the time XON/XOFF. Refer to ppp(8) for more information on this option and how it is used.

An older modem may need set parity even. Parity is set at none by default, but is used for error checking with a large increase in traffic, on older modems.

PPP may not return to the command mode, which is usually a negotiation error where the ISP is waiting for negotiating to begin. At this point, using ~p will force ppp to start sending the configuration information.

If a login prompt never appears, PAP or CHAP authentication is most likely required. To use PAP or CHAP, add the following options to PPP before going into terminal mode:

set authname myusername

Where myusername should be replaced with the username that was assigned by the ISP.
Where mypassword should be replaced with the password that was assigned by the ISP.

If a connection is established, but cannot seem to find any domain name, try to ping(8) an IP address. If there is 100 percent (100%) packet loss, it is likely that a default route was not assigned. Double check that add default HISADDR was set during the connection. If a connection can be made to a remote IP address, it is possible that a resolver address has not been added to /etc/resolv.conf. This file should look like:

domain example.com
nameserver x.x.x.x
nameserver y.y.y.y

Where x.x.x.x and y.y.y.y should be replaced with the IP address of the ISP's DNS servers.

To configure syslog(3) to provide logging for the PPP connection, make sure this line exists in /etc/syslog.conf:

!ppp
*
/var/log/ppp.log

27.4.

在乙太网络使用PPP (PPPoE)

本节介绍如何设定在乙太网络使用PPP (PPPoE)。

以下有一个可用的 ppp.conf 例子:

default:
set log Phase tun command # you can add more detailed logging if you wish
set ifaddr 10.0.0.1/0 10.0.0.2/0
name_of_service_provider:
set device PPPoE:xl1 # replace xl1 with your Ethernet device
set authname YOURLOGINNAME
set authkey YOURPASSWORD
set dial
set login
add default HISADDR

以root身份执行:

# ppp -ddial name_of_service_provider

将以下参数加到 /etc/rc.conf:

560
27.4.1. 使用 PPPoE 服务标签有時需要使用服务标签(Service Tag)才能建立连线，服务标签用來區別不同網路要各自连线的 PPPoE 伺服器。所需的服務標籤資訊應該會在 ISP 所提供的文件中說明。

最後的手段是嘗試安裝 net/rr-pppoe 套件或 Port。但請注意，這可能會解除安装數據機中的程式並使其無法運作，所以請三思而為。只需要安裝數據機所提供的程式，然後由該程式進入 System 選單，基本資料(Profile name)的名稱應該會列出來，通常是 ISP 的名稱。

基本資料名稱(Profile Name)即服務標籤，會被用在 ppp.conf 中的 PPPoE 設定項目 set device 的提供商(Provider)部份。請參考 ppp(8) 以取得詳細說明，結果應如下:

```
set device PPPoE:xl1:ISP
```

別忘記更改 xl1 為乙太網路卡的裝置名稱。別忘記更改 ISP 為基本資料名稱。

要取得更進一步資訊，請參考 Renaud Waldura 所著的 Cheaper Broadband with FreeBSD on DSL。

27.4.2. 在 3Com™HomeConnect™ ADSL Modem Dual Link 使用 PPPoE

這台數據機並不採用 RFC 2516 所定義的規格。為了要讓 FreeBSD 能夠與這台裝置通訊，必須設定 sysctl，這可以透過更新 /etc/sysctl.conf 來讓開機時自動設定。

```
net.graph.nonstandard_pppoe=1
```
或可以執行以下指令立即使更改:

```
# sysctl net.graph.nonstandard_pppoe=1
```

不幸的是，由於這是一個全系統的設定，這可能導致一般 PPPoE 客戶端或伺服器無法與 3Com™HomeConnect™ ADSL 資料機同時使用。

27.5. 在 ATM 使用 PPP (PPPoA)

The following describes how to set up PPP over ATM (PPPoA). PPPoA is a popular choice among European DSL providers.
27.5.1. 使用 mpd

The mpd application can be used to connect to a variety of services, in particular PPTP services. It can be installed using the net/mpd5 package or port. Many ADSL modems require that a PPTP tunnel is created between the modem and computer.

Once installed, configure mpd to suit the provider's settings. The port places a set of sample configuration files which are well documented in /usr/local/etc/mpd/. A complete guide to configure mpd is available in HTML format in /usr/ports/shared/doc/mpd/.

Here is a sample configuration for connecting to an ADSL service with mpd. The configuration is spread over two files, first the mpd.conf:

```
This example mpd.conf only works with mpd 4.x.

default:
load adsl

adsl:
new -i ng0 adsl adsl
set bundle authname username ①
set bundle password password ②
set bundle disable multilink
set link no pap acfcomp protocomp
set link disable chap
set link accept chap
set link keep-alive 30 10
set ipcp no vjcomp
set ipcp ranges 0.0.0.0/0 0.0.0.0/0
set iface route default
set iface disable on-demand
set iface enable proxy-arp
set iface idle 0
open
```

① The username used to authenticate with your ISP.
② The password used to authenticate with your ISP.

Information about the link, or links, to establish is found in mpd.links. An example mpd.links to accompany the above example is given beneath:

```
adsl:
set link type pptp
set pptp mode active
```

562
To initialize the connection easily, issue the following command as root:

```bash
# mpd -b adsl
```

To view the status of the connection:

```
% ifconfig ng0
ng0: flags = 88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
inet 216.136.204.117 -->
204.152.186.171 netmask 0xffffffff
```

Using `mpd` is the recommended way to connect to an ADSL service with FreeBSD.

It is also possible to use FreeBSD to connect to other PPPoA services using `net/pptpclient`.

To use `net/pptpclient` to connect to a DSL service, install the port or package, then edit `/etc/ppp/ppp.conf`. An example section of `ppp.conf` is given below. For further information on `ppp.conf` options consult `ppp(8)`.

```
adsl:
    set log phase chat lcp ipcp ccp tun command
    set timeout 0
    enable dns
    set authname username
    set authkey password
    set ifaddr 0 0
    add default HISADDR
```

- The username for the DSL provider.
- The password for your account.

Since the account's password is added to `ppp.conf` in plain text form, make sure nobody can read the contents of this file:

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

This will open a tunnel for a PPP session to the DSL router. Ethernet DSL modems have a preconfigured LAN IP address to connect to. In the case of the Alcatel SpeedTouch™ Home, this...
The router's documentation should list the address the device uses. To open the tunnel and start a PPP session:

```
# pptp address adsl
```

If an ampersand ("&") is added to the end of this command, pptp will return the prompt. A tun virtual tunnel device will be created for interaction between the pptp and ppp processes. Once the prompt is returned, or the pptp process has confirmed a connection, examine the tunnel:

```
% ifconfig tun0
```

```
tun0:  flags = 8051<UP,POINTOPOINT,RUNNING,MULTICAST>  mtu 1500
inet 216.136.204.21 --
                  198.152.186.171 netmask 0xffffff00
Opened by PID 918
```

If the connection fails, check the configuration of the router, which is usually accessible using a web browser. Also, examine the output of `pptp` and the contents of the log file, `/var/log/ppp.log` for clues.
28.1. 概述

"電子郵件"或稱 email, 是現今使用最廣泛的溝通方式之一。本章主要介紹如何在 FreeBSD 上執行郵件伺服器, 以及如何使用 FreeBSD 收發信件。若欲瞭解細節請參閱參考書目內的參考書籍。

讀完這章, 您將了 解:

- 哪些軟體元件與收發電子郵件有關。
- FreeBSD 內的 Sendmail 設定檔在哪。
- 遠端信箱 (Mailbox) 與本機信箱的差異。
- 如何阻擋垃圾郵件寄件者 (Spammer) 非法使用郵件伺服器作為中繼站。
- 如何安裝與設定其他的郵件傳輸代理程式 (Mail Transfer Agent) 來取代 Sendmail。
- 如何排除常見的郵件伺服器問題。
- 如何設定系統只能寄送郵件。
- 如何在撥號連線上使用郵件。
- 如何設定 SMTP 認證來增加安全性。
- 如何安裝並使用郵件使用者代理程式 (Mail User Agent) 如 mutt 來寄發與接收入件。
- 如何從遠端的 POP 或 IMAP 伺服器下載郵件。
- 如何自動套用過濾器及規則在收到的電子郵件上。

在開始閱讀這章之前, 您需要:

- 正確的設定網路 (進階網路設定)。
- 正確的設定郵件主機的 DNS 資訊 (網路伺服器)。
- 了 解如何安裝其他第三方軟體 (安裝應用程式: 套件與 Port)。

28.2. 郵件組成

There are five major parts involved in an email exchange: the Mail User Agent (MUA), the Mail Transfer Agent (MTA), a mail host, a remote or local mailbox, and DNS. This section provides an overview of these components.

郵件使用者代理程式 (Mail User Agent, MUA)

The Mail User Agent (MUA) is an application which is used to compose, send, and receive emails. This application can be a command line program, such as the built-in `mail` utility or a third-party application from the Ports Collection, such as mutt, alpine, or elm. Dozens of graphical programs are also available in the Ports Collection, including Claws Mail, Evolution, and Thunderbird. Some organizations provide a web mail program which can be accessed through a web browser. More information about installing and using a MUA on FreeBSD can be found in 郵件使用者代理程式.

郵件傳輸代理程式 (Mail Transfer Agent, MTA)

The Mail Transfer Agent (MTA) is responsible for receiving incoming mail and delivering outgoing mail. FreeBSD ships with Sendmail as the default MTA, but it also supports numerous other mail server daemons, including Exim, Postfix, and qmail. Sendmail configuration is described in Sendmail設定檔. If another MTA is installed using the Ports Collection, refer to its post-installation message for FreeBSD-specific configuration details and the application's website for more general configuration instructions.

郵件主機 (Mail Host) 與郵件信箱 (Mailbox)

The mail host is a server that is responsible for delivering and receiving mail for a host or a 565.
The mail host collects all mail sent to the domain and stores it either in the default \texttt{mbox} or the alternative Maildir format, depending on the configuration. Once mail has been stored, it may either be read locally using a MUA or remotely accessed and collected using protocols such as POP or IMAP. If mail is read locally, a POP or IMAP server does not need to be installed.

To access mailboxes remotely, a POP or IMAP server is required as these protocols allow users to connect to their mailboxes from remote locations. IMAP offers several advantages over POP. These include the ability to store a copy of messages on a remote server after they are downloaded and concurrent updates. IMAP can be useful over low-speed links as it allows users to fetch the structure of messages without downloading them. It can also perform tasks such as searching on the server in order to minimize data transfer between clients and servers.

Several POP and IMAP servers are available in the Ports Collection. These include \texttt{mail/qpopper}, \texttt{mail/imap-uw}, \texttt{mail/courier-imap}, and \texttt{mail/dovecot2}.

It should be noted that both POP and IMAP transmit information, including username and password credentials, in clear-text. To secure the transmission of information across these protocols, consider tunneling sessions over \texttt{ssh(1)} (SSH通道) or using SSL (OpenSSL).

The Domain Name System (DNS) and its daemon \texttt{named} play a large role in the delivery of email. In order to deliver mail from one site to another, the MTA will look up the remote site in DNS to determine which host will receive mail for the destination. This process also occurs when mail is sent from a remote host to the MTA.

In addition to mapping hostnames to IP addresses, DNS is responsible for storing information specific to mail delivery, known as Mail eXchanger MX records. The MX record specifies which hosts will receive mail for a particular domain.

To view the MX records for a domain, specify the type of record. Refer to \texttt{host(1)}, for more details:

\begin{verbatim}
% host -t mx FreeBSD.org
FreeBSD.org mail is handled by 10 mx1.FreeBSD.org
\end{verbatim}

Refer to \texttt{網域名稱系統(DNS)} for more information about DNS and its configuration.

Sendmail is the default MTA installed with FreeBSD. It accepts mail from MUAs and delivers it to the appropriate mail host, as defined by its configuration. Sendmail can also accept network connections and deliver mail to local mailboxes or to another program.

The configuration files for Sendmail are located in \texttt{/etc/mail}. This section describes these files in more detail.

\texttt{/etc/mail/access} This access database file defines which hosts or IP addresses have access to the local mail server and what kind of access they have. Hosts listed as \texttt{OK}, which is the default option, are allowed to send mail to this host as long as the mail's final destination is the local machine. Hosts listed as \texttt{REJECT} are rejected for all mail connections. Hosts listed as \texttt{RELAY} are allowed to send mail for any destination using this mail server. Hosts listed as \texttt{ERROR} will have their mail returned with the specified mail error. If a host is listed as \texttt{SKIP}, Sendmail will abort the current search for this entry without accepting or rejecting the mail. Hosts listed as \texttt{QUARANTINE} will have their messages held and will receive the specified text as the reason for the hold.
Examples of using these options for both IPv4 and IPv6 addresses can be found in the FreeBSD sample configuration, /etc/mail/access.sample:

```plaintext
# $FreeBSD: head/zh_TW.UTF-8/books/handbook/book.xml 53653 2019-12-03 17:05:41Z rcury $
#
# Mail relay access control list. Default is to reject mail unless the
destination is local, or listed in /etc/mail/local-host-names

## Examples (commented out for safety)
#From:cyberspammer.com          ERROR:"550 We don't accept mail from spammers"
#From:okay.cyberspammer.com     OK
#Connect:sendmail.org           RELAY
#To:sendmail.org                RELAY
#Connect:128.32                 RELAY
#Connect:128.32.2               SKIP
#Connect:suspicious.example.com QUARANTINE:Mail from suspicious host
#Connect:[127.0.0.3]            OK
```

To configure the access database, use the format shown in the sample to make entries in /etc/mail/access, but do not put a comment symbol (`#`) in front of the entries. Create an entry for each host or network whose access should be configured. Mail senders that match the left side of the table are affected by the action on the right side of the table.

Whenever this file is updated, update its database and restart Sendmail:

```
# makemap hash /etc/mail/access < /etc/mail/access
# service sendmail restart
```

/etc/mail/aliases

This database file contains a list of virtual mailboxes that are expanded to users, files, programs, or other aliases. Here are a few entries to illustrate the file format:

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

The mailbox name on the left side of the colon is expanded to the target(s) on the right. The first entry expands the root mailbox to the localuser mailbox, which is then looked up in the /etc/mail/aliases database. If no match is found, the message is delivered to localuser. The second entry shows a mail list. Mail to ftp-bugs is expanded to the three local mailboxes joe, eric, and paul. A remote mailbox could be specified as user@example.com. The third entry shows how to write mail to a file, in this case /dev/null. The last entry demonstrates how to send mail to a program, /usr/local/bin/procmail, through a UNIX™ pipe. Refer to aliases(5) for more information about the format of this file.
Whenever this file is updated, run `newaliases` to update and initialize the aliases database.

`/etc/mail/sendmail.cf`

This is the master configuration file for Sendmail. It controls the overall behavior of Sendmail, including everything from rewriting email addresses to printing rejection messages to remote mail servers. Accordingly, this configuration file is quite complex. Fortunately, this file rarely needs to be changed for standard mail servers.

The master Sendmail configuration file can be built from `m4(1)` macros that define the features and behavior of Sendmail. Refer to `/usr/src/contrib/sendmail/cf/README` for some of the details.

Whenever changes to this file are made, Sendmail needs to be restarted for the changes to take effect.

`/etc/mail/virtusertable`

This database file maps mail addresses for virtual domains and users to real mailboxes. These mailboxes can be local, remote, aliases defined in `/etc/mail/aliases`, or files. This allows multiple virtual domains to be hosted on one machine.

FreeBSD provides a sample configuration file in `/etc/mail/virtusertable.sample` to further demonstrate its format. The following example demonstrates how to create custom entries using that format:

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

This file is processed in a first match order. When an email address matches the address on the left, it is mapped to the local mailbox listed on the right. The format of the first entry in this example maps a specific email address to a local mailbox, whereas the format of the second entry maps a specific email address to a remote mailbox. Finally, any email address from `example.com` which has not matched any of the previous entries will match the last mapping and be sent to the local mailbox `joe`. When creating custom entries, use this format and add them to `/etc/mail/virtusertable`. Whenever this file is edited, update its database and restart Sendmail:

```
# makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
# service sendmail restart
```

`/etc/mail/relay-domains`

In a default FreeBSD installation, Sendmail is configured to only send mail from the host it is running on. For example, if a POP server is available, users will be able to check mail from remote locations but they will not be able to send outgoing emails from outside locations. Typically, a few moments after the attempt, an email will be sent from `MAILER-DAEMON` with a `5.7 Relaying Denied` message.

The most straightforward solution is to add the ISP's FQDN to `/etc/mail/relay-domains`. If multiple addresses are needed, add them one per line:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

568
After creating or editing this file, restart Sendmail with

```
service sendmail restart
```

Now any mail sent through the system by any host in this list, provided the user has an account
on the system, will succeed. This allows users to send mail from the system remotely without
opening the system up to relaying SPAM from the Internet.

FreeBSD comes with Sendmail already installed as the MTA which is in charge of outgoing and
incoming mail. However, the system administrator can change the system's MTA. A wide choice of
alternative MTAs is available from the `mail` category of the FreeBSD Ports Collection.

Once a new MTA is installed, configure and test the new software before replacing Sendmail. Refer
to the documentation of the new MTA for information on how to configure the software.

Once the new MTA is working, use the instructions in this section to disable Sendmail and configure
FreeBSD to use the replacement MTA.

### 28.4.1. Disable Sendmail

If Sendmail's outgoing mail service is disabled, it is important that it is replaced
with an alternative mail delivery system. Otherwise, system functions such as
`periodic(8)` will be unable to deliver their results by email. Many parts of the
system expect a functional MTA. If applications continue to use Sendmail's
binaries to try to send email after they are disabled, mail could go into an inactive
Sendmail queue and never be delivered.

In order to completely disable Sendmail, add or edit the following lines in
`/etc/rc.conf`:

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

To only disable Sendmail's incoming mail service, use only this entry in
`/etc/rc.conf`:

```
sendmail_enable="NO"
```

More information on Sendmail's startup options is available in
`rc.sendmail(8)`.

### 28.4.2. Replace the Default MTA

When a new MTA is installed using the Ports Collection, its startup script is also installed and
startup instructions are mentioned in its package message. Before starting the new MTA, stop the
running Sendmail processes. This example stops all of these services, then starts the Postfix
service:

```
# service sendmail stop
# service postfix start
```

To start the replacement MTA at system boot, add its configuration line to
`/etc/rc.conf`. This entry
enables the Postfix MTA:
Some extra configuration is needed as Sendmail is so ubiquitous that some software assumes it is already installed and configured. Check `/etc/periodic.conf` and make sure that these values are set to `NO`. If this file does not exist, create it with these entries:

```
daily_clean_hoststat_enable="NO"
daily_status_mail_rejects_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"
```

Some alternative MTAs provide their own compatible implementations of the Sendmail command-line interface in order to facilitate using them as drop-in replacements for Sendmail. However, some MUAs may try to execute standard Sendmail binaries instead of the new MTA's binaries.

FreeBSD uses `/etc/mail/mailer.conf` to map the expected Sendmail binaries to the location of the new binaries. More information about this mapping can be found in `mailwrapper(8)`.

The default `/etc/mail/mailer.conf` looks like this:

```
# $FreeBSD: head/zh_TW.UTF-8/books/handbook/book.xml 53653 2019-12-03 17:05:41Z
rcyu $
#
# Execute the "real" sendmail program, named /usr/libexec/sendmail/sendmail
#
sendmail        /usr/libexec/sendmail/sendmail
send-mail       /usr/libexec/sendmail/sendmail
mailq           /usr/libexec/sendmail/sendmail
newaliases      /usr/libexec/sendmail/sendmail
hoststat        /usr/libexec/sendmail/sendmail
purgestat       /usr/libexec/sendmail/sendmail
```

When any of the commands listed on the left are run, the system actually executes the associated command shown on the right. This system makes it easy to change what binaries are executed when these default binaries are invoked.

Some MTAs, when installed using the Ports Collection, will prompt to update this file for the new binaries. For example, Postfix will update the file like this:

```
# Execute the Postfix sendmail program, named /usr/local/sbin/sendmail
#
sendmail        /usr/local/sbin/sendmail
send-mail       /usr/local/sbin/sendmail
mailq           /usr/local/sbin/sendmail
newaliases      /usr/local/sbin/sendmail
```
If the installation of the MTA does not automatically update `/etc/mail/mailer.conf`, edit this file in a text editor so that it points to the new binaries. This example points to the binaries installed by `mail/ssmtp`:

- `sendmail`        /usr/local/sbin/ssmtp
- `send-mail`       /usr/local/sbin/ssmtp
- `mailq`           /usr/local/sbin/ssmtp
- `newaliases`      /usr/local/sbin/ssmtp
- `hoststat`        /usr/bin/true
- `purgestat`       /usr/bin/true

Once everything is configured, it is recommended to reboot the system. Rebooting provides the opportunity to ensure that the system is correctly configured to start the new MTA automatically on boot.

28.5. 疑難排解

28.5.1. Why do I have to use the FQDN for hosts on my site?
The host may actually be in a different domain. For example, in order for a host in `foo.bar.edu` to reach a host called `mumble` in the `bar.edu` domain, refer to it by the Fully-Qualified Domain Name (FQDN), `mumble.bar.edu`, instead of just `mumble`. This is because the version of BIND which ships with FreeBSD no longer provides default abbreviations for non-FQDNs other than the local domain. An unqualified host such as `mumble` must either be found as `mumble.foo.bar.edu`, or it will be searched for in the root domain. In older versions of BIND, the search continued across `mumble.bar.edu` and `mumble.edu`. RFC 1535 details why this is considered bad practice or even a security hole.

As a good workaround, place the line:

```
search foo.bar.edu bar.edu
```

Instead of the previous:

```
domain foo.bar.edu
```

into `/etc/resolv.conf`. However, make sure that the search order does not go beyond the "boundary between local and public administration", as RFC 1535 calls it.

28.5.2. How can I run a mail server on a dial-up PPP host?
Connect to a FreeBSD mail gateway on the LAN. The PPP connection is non-dedicated. One way to do this is to get a full-time Internet server to provide secondary MX services for the domain. In this example, the domain is `example.com` and the ISP has configured `example.net` to provide secondary MX services to the domain:

```
example.com.          MX        10      example.com.
example.com.          MX        20      example.net.
```
Only one host should be specified as the final recipient. For Sendmail, add `Cw example.com` in `/etc/mail/sendmail.cf` on `example.com`. When the sending MTA attempts to deliver mail, it will try to connect to the system, `example.com`, over the PPP link. This will time out if the destination is offline. The MTA will automatically deliver it to the secondary MX site at the Internet Service Provider (ISP), `example.net`. The secondary MX site will periodically try to connect to the primary MX host, `example.com`.

Use something like this as a login script:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
(sleep 60; /usr/sbin/sendmail -q) &
/usr/sbin/ppp -direct pppmyisp
```

When creating a separate login script for users, instead use `sendmail -qRexample.com` in the script above. This will force all mail in the queue for `example.com` to be processed immediately.

A further refinement of the situation can be seen from this example from the FreeBSD Internet service provider's mailing list:

```
> we provide the secondary MX for a customer. The customer connects to our services several times a day automatically to get the mails to his primary MX (We do not call his site when a mail for his domains arrived). Our sendmail sends the mailqueue every 30 minutes. At the moment he has to stay 30 minutes online to be sure that all mail is gone to the primary MX.
>
> Is there a command that would initiate sendmail to send all the mails now? The user has not root-privileges on our machine of course.
```

In the privacy flags section of sendmail.cf, there is a definition `Opgoaway,restrictqrun`.

Remove `restrictqrun` to allow non-root users to start the queue processing.

You might also like to rearrange the MXs. We are the 1st MX for our customers like this, and we have defined:

```
# If we are the best MX for a host, try directly instead of generating local config error.
OwTrue
```

That way a remote site will deliver straight to you, without trying the customer connection. You then send to your customer. Only works for hosts, so you need to get your customer to name their mail machine `customer.com` as well as...
hostname.customer.com in the DNS. Just put an A record in the DNS for customer.com.

28.6.

進階主題

This section covers more involved topics such as mail configuration and setting up mail for an entire domain.

28.6.1.

基礎設定

Out of the box, one can send email to external hosts as long as /etc/resolv.conf is configured or the network has access to a configured DNS server. To have email delivered to the MTA on the FreeBSD host, do one of the following:

• Run a DNS server for the domain.
• Get mail delivered directly to the FQDN for the machine.

In order to have mail delivered directly to a host, it must have a permanent static IP address, not a dynamic IP address. If the system is behind a firewall, it must be configured to allow SMTP traffic.

To receive mail directly at a host, one of these two must be configured:

• Make sure that the lowest-numbered MX record in DNS points to the host's static IP address.
• Make sure there is no MX entry in the DNS for the host.

Either of the above will allow mail to be received directly at the host.

Try this:

# hostname example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX

In this example, mail sent directly to yourlogin@example.FreeBSD.org should work without problems, assuming Sendmail is running correctly on example.FreeBSD.org.

For this example:

# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by nevdull.FreeBSD.org

All mail sent to example.FreeBSD.org will be collected on hub under the same username instead of being sent directly to your host.

The above information is handled by the DNS server. The DNS record that carries mail routing information is the MX entry. If no MX record exists, mail will be delivered directly to the host by way of its IP address.

The MX entry for freefall.FreeBSD.org at one time looked like this:

freefall    MX    30    mail.crl.net
freefall        MX  40  agora.rdrop.com
freefall        MX  10  freefall.FreeBSD.org
freefall        MX  20  who.cdrom.com

had many MX entries. The lowest MX number is the host that receives mail directly, if available. If it is not accessible for some reason, the next lower-numbered host will accept messages temporarily, and pass it along when a lower-numbered host becomes available.

Alternate MX sites should have separate Internet connections in order to be most useful. Your ISP can provide this service.

When configuring a MTA for a network, any mail sent to hosts in its domain should be diverted to the MTA so that users can receive their mail on the master mail server.

To make life easiest, a user account with the same username should exist on both the MTA and the system with the MUA. Use adduser(8) to create the user accounts.

The MTA must be the designated mail exchanger for each workstation on the network. This is done in the DNS configuration with an MX record:

example.FreeBSD.org A   204.216.27.XX       ; Workstation
MX  10 nevdull.FreeBSD.org  ; Mailhost

This will redirect mail for the workstation to the MTA no matter where the A record points. The mail is sent to the MX host.

This must be configured on a DNS server. If the network does not run its own DNS server, talk to the ISP or DNS provider.

The following is an example of virtual email hosting. Consider a customer with the domain customer1.org, where all the mail for customer1.org should be sent to mail.myhost.com. The DNS entry should look like this:

customer1.org       MX  10  mail.myhost.com

An A record is not needed for customer1.org in order to only handle email for that domain. However, running ping against customer1.org will not work unless an A record exists for it.

Tell the MTA which domains and/or hostnames it should accept mail for. Either of the following will work for Sendmail:

• Add the hosts to /etc/mail/local-host-names when using the FEATURE(use_cw_file).
• Add a Cwyour.host.com line to /etc/sendmail.cf.

There are many instances where one may only want to send mail through a relay. Some examples are:

• The computer is a desktop machine that needs to use programs such as mail(1), using the ISP's mail relay.
• The computer is a server that does not handle mail locally, but needs to pass off all mail to a
While any MTA is capable of filling this particular niche, it can be difficult to properly configure a full-featured MTA just to handle offloading mail. Programs such as Sendmail and Postfix are overkill for this use. Additionally, a typical Internet access service agreement may forbid one from running a "mail server".

The easiest way to fulfill those needs is to install the `mail/ssmtp` port:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

Once installed, `mail/ssmtp` can be configured with `/usr/local/etc/ssmtp/ssmtp.conf`:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Use the real email address for `root`. Enter the ISP's outgoing mail relay in place of `mail.example.com`. Some ISPs call this the "outgoing mail server" or "SMTP server".

Make sure to disable Sendmail, including the outgoing mail service. See 締束Sendmail for details.

`mail/ssmtp` has some other options available. Refer to the examples in `/usr/local/etc/ssmtp` or the manual page of ssmtp for more information.

Setting up ssmtp in this manner allows any software on the computer that needs to send mail to function properly, while not violating the ISP's usage policy or allowing the computer to be hijacked for spamming.

When using a static IP address, one should not need to adjust the default configuration. Set the hostname to the assigned Internet name and Sendmail will do the rest.

When using a dynamically assigned IP address and a dialup PPP connection to the Internet, one usually has a mailbox on the ISP's mail server. In this example, the ISP's domain is example.net, the user name is user, the hostname is bsd.home, and the ISP has allowed relay.example.net as a mail relay.

In order to retrieve mail from the ISP's mailbox, install a retrieval agent from the Ports Collection. `mail/fetchmail` is a good choice as it supports many different protocols. Usually, the ISP will provide POP. When using user PPP, email can be automatically fetched when an Internet connection is established with the following entry in `/etc/ppp/ppp.linkup`:

```
MYADDR:!bg su user -c fetchmail
```

When using Sendmail to deliver mail to non-local accounts, configure Sendmail to process the mail queue as soon as the Internet connection is established. To do this, add this line after the above fetchmail entry in `/etc/ppp/ppp.linkup`:

```
575
```
In this example, there is an account for `user` on `bsd.home`. In the home directory of `user` on `bsd.home`, create a `.fetchmailrc` which contains this line:

```
poll example.net protocol pop3 fetchall pass MySecret
```

This file should not be readable by anyone except `user` as it contains the password `MySecret`.

In order to send mail with the correct `from:` header, configure Sendmail to use `user@example.net` rather than `user@bsd.home` and to send all mail via `relay.example.net`, allowing quicker mail transmission.

The following `.mc` should suffice:

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)
FEATURE(nouucp)
MAILER(local)
MAILER(smtp)
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(`example.net')
FEATURE(allmasquerade)
FEATURE(masquerade_envelope)
FEATURE(nocanonify)
FEATURE(nodns)
define(`SMART_HOST', `relay.example.net')
dmbbsd.home
define(`confDOMAIN_NAME',`bsd.home')
define(`confDELIVERY_MODE',`deferred')
```

Refer to the previous section for details of how to convert this file into the `sendmail.cf` format. Do not forget to restart Sendmail after updating `sendmail.cf`.

28.9. SMTP 認証

Configuring SMTP authentication on the MTA provides a number of benefits. SMTP authentication adds a layer of security to Sendmail, and provides mobile users who switch hosts the ability to use the same MTA without the need to reconfigure their mail client's settings each time.

1. Install `security/cyrus-sasl2` from the Ports Collection. This port supports a number of compile-time options. For the SMTP authentication method demonstrated in this example, make sure that `LOGIN` is not disabled.

2. After installing `security/cyrus-sasl2`, edit `/usr/local/lib/sasl2/Sendmail.conf`, or create it if it does not exist, and add the following line:

```
576
```
3. Next, install `security/cyrus-sasl2-saslauthd` and add the following line to `/etc/rc.conf`:

```
saslauthd_enable="YES"
```

Finally, start the saslauthd daemon:

```
# service saslauthd start
```

This daemon serves as a broker for Sendmail to authenticate against the FreeBSD `passwd(5)` database. This saves the trouble of creating a new set of usernames and passwords for each user that needs to use SMTP authentication, and keeps the login and mail password the same.

4. Next, edit `/etc/make.conf` and add the following lines:

```
SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
SENDMAIL_LDFLAGS=-L/usr/local/lib
SENDMAIL_LDADD=-lsasl2
```

These lines provide Sendmail the proper configuration options for linking to `cyrus-sasl2` at compile time. Make sure that `cyrus-sasl2` has been installed before recompiling Sendmail.

5. Recompile Sendmail by executing the following commands:

```
# cd /usr/src/lib/libsmutil
# make cleandir && make obj && make
# cd /usr/src/lib/libsm
# make cleandir && make obj && make
# cd /usr/src/usr.sbin/sendmail
# make cleandir && make obj && make && make install
```

This compile should not have any problems if `/usr/src` has not changed extensively and the shared libraries it needs are available.

6. After Sendmail has been compiled and reinstalled, edit `/etc/mail/freebsd.mc` or the local `.mc`. Many administrators choose to use the output from `hostname(1)` as the name of `.mc` for uniqueness. Add these lines:

```
dnl set SASL options
TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
```

These options configure the different methods available to Sendmail for authenticating users. To use a method other than `pwcheck`, refer to the Sendmail documentation.
Finally, run `make(1)` while in `/etc/mail`. That will run the new `.mc` and create a `.cf` named `freebsd.cf` or the name used for the local `.mc`. Then, run `make install restart`, which will copy the file to `sendmail.cf`, and properly restart Sendmail. For more information about this process, refer to `/etc/mail/Makefile`.

To test the configuration, use a MUA to send a test message. For further investigation, set the `LogLevel` of Sendmail to 13 and watch `/var/log/maillog` for any errors. For more information, refer to `SMTP authentication`.

28.10. 邮件使用者代理程式

A MUA is an application that is used to send and receive email. As email “evolves” and becomes more complex, MUAs are becoming increasingly powerful and provide users increased functionality and flexibility. The `mail` category of the FreeBSD Ports Collection contains numerous MUAs. These include graphical email clients such as Evolution or Balsa and console based clients such as mutt or alpine.

28.10.1. mail

`mail(1)` is the default MUA installed with FreeBSD. It is a console based MUA that offers the basic functionality required to send and receive text-based email. It provides limited attachment support and can only access local mailboxes.

Although `mail` does not natively support interaction with POP or IMAP servers, these mailboxes may be downloaded to a local `mbox` using an application such as fetchmail.

In order to send and receive email, run `mail`:

```
% mail
```

The contents of the user's mailbox in `/var/mail` are automatically read by `mail`. Should the mailbox be empty, the utility exits with a message indicating that no mail could be found. If mail exists, the application interface starts, and a list of messages will be displayed. Messages are automatically numbered, as can be seen in the following example:

```
/var/mail/marcs": 3 messages 3 new
> N  1 root@localhost  Mon Mar  8 14:05  14/510
"test"
N  2 root@localhost  Mon Mar  8 14:05  14/509
"user account"
N  3 root@localhost  Mon Mar  8 14:05  14/509
"sample"
```

Messages can now be read by typing `t` followed by the message number. This example reads the first email:

```
& t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
```
This is a test message, please reply if you receive it.

As seen in this example, the message will be displayed with full headers. To display the list of messages again, press h.

If the email requires a reply, press either R or r mail keys. R instructs mail to reply only to the sender of the email, while r replies to all other recipients of the message. These commands can be suffixed with the mail number of the message to reply to. After typing the response, the end of the message should be marked by a single . on its own line. An example can be seen below:

```
& R 1
To: root@localhost
Subject: Re: test
Thank you, I did get your email.
```

EOT

In order to send a new email, press m, followed by the recipient email address. Multiple recipients may be specified by separating each address with the , delimiter. The subject of the message may then be entered, followed by the message contents. The end of the message should be specified by putting a single . on its own line.

```
& mail root@localhost
Subject: I mastered mail
Now I can send and receive email using mail ...
```

EOT

While using mail, press ? to display help at any time. Refer to mail(1) for more help on how to use mail.

mail(1) was not designed to handle attachments and thus deals with them poorly. Newer MUAs handle attachments in a more intelligent way. Users who prefer to use mail may find the converters/mpack port to be of considerable use.

28.10.2. mutt

mutt is a powerful MUA, with many features, including:

- The ability to thread messages.
- PGP support for digital signing and encryption of email.
- MIME support.
Maildir support.

Highly customizable. Refer to http://www.mutt.org for more information on mutt.

mutt may be installed using the mail/mutt port. After the port has been installed, mutt can be started by issuing the following command:

```
% mutt
```

mutt will automatically read and display the contents of the user mailbox in /var/mail. If no mails are found, mutt will wait for commands from the user. The example below shows mutt displaying a list of messages:

```
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
1 N Mar 09 Super-User ( 1) test
2 N Mar 09 Super-User ( 1) user account
3 N Mar 09 Super-User ( 1) sample
```

To read an email, select it using the cursor keys and press Enter. An example of mutt displaying email can be seen below:
Similar to `mail(1)`, mutt can be used to reply only to the sender of the message as well as to all recipients. To reply only to the sender of the email, press `r`. To send a group reply to the original sender as well as all the message recipients, press `g`.

By default, mutt uses the `vi(1)` editor for creating and replying to emails. Each user can customize this by creating or editing the `.muttrc` in their home directory and setting the `editor` variable or by setting the `EDITOR` environment variable. Refer to [http://www.mutt.org/](http://www.mutt.org/) for more information about configuring mutt.

To compose a new mail message, press `m`. After a valid subject has been given, mutt will start `vi(1)` so the email can be written. Once the contents of the email are complete, save and quit from `vi`. mutt will resume, displaying a summary screen of the mail that is to be delivered. In order to send the mail, press `y`. An example of the summary screen can be seen below:

```
This is a test message, please reply if you receive it.
```
mutt contains extensive help which can be accessed from most of the menus by pressing ?. The top line also displays the keyboard shortcuts where appropriate.

28.10.3. alpine

alpine is aimed at a beginner user, but also includes some advanced features. alpine has had several remote vulnerabilities discovered in the past, which allowed remote attackers to execute arbitrary code as users on the local system, by the action of sending a specially-prepared email. While known problems have been fixed, alpine code is written in an insecure style and the FreeBSD Security Officer believes there are likely to be other undiscovered vulnerabilities. Users install alpine at their own risk.

The current version of alpine may be installed using the mail/alpine port. Once the port has installed, alpine can be started by issuing the following command:

```
% alpine
```

The first time alpine runs, it displays a greeting page with a brief introduction, as well as a request from the alpine development team to send an anonymous email message allowing them to judge how many users are using their client. To send this anonymous message, press Enter. Alternatively, press E to exit the greeting without sending an anonymous message. An example of the greeting page is shown below:

```
-- Attachments
- I 1 /tmp/mutt-bsd-c0hobscQ [text/plain, 7bit, us-ascii, 1.1K]

-- Mutt: Compose [Approx. msg size: 1.1K Atts: 1]-----------------------------
```
The main menu is then presented, which can be navigated using the cursor keys. This main menu provides shortcuts for the composing new mails, browsing mail directories, and administering address book entries. Below the main menu, relevant keyboard shortcuts to perform functions specific to the task at hand are shown.

The default directory opened by alpine is **inbox**. To view the message index, press **I**, or select the **MESSAGE INDEX** option shown below:

The message index shows messages in the current directory and can be navigated by using the cursor keys. Highlighted messages can be read by pressing **Enter**.

Pine is a trademark of the University of Washington.

Copyright 1989-2003. PINE is a trademark of the University of Washington.
In the screenshot below, a sample message is displayed by alpine. Contextual keyboard shortcuts are displayed at the bottom of the screen. An example of one of a shortcut is `r`, which tells the MUA to reply to the current message being displayed.

Replying to an email in alpine is done using the pico editor, which is installed by default with alpine. pico makes it easy to navigate the message and is easier for novice users to use than `vi(1)` or `mail(1)`. Once the reply is complete, the message can be sent by pressing `Ctrl + X`. alpine will ask for confirmation before sending the message.

---

Date: Tue, 9 Mar 2004 10:28:36 +0200 (CST)
From: Super-User <root@localhost>
To: marcs@localhost
Subject: test

This is a test message, please reply if you receive it.
alpine can be customized using the SETUP option from the main menu. Consult http://www.washington.edu/alpine/ for more information.

28.11. 使用 fetchmail

fetchmail is a full-featured IMAP and POP client. It allows users to automatically download mail from remote IMAP and POP servers and save it into local mailboxes where it can be accessed more easily. fetchmail can be installed using the mail/fetchmail port, and offers various features, including:

• Support for the POP3, APOP, KPOP, IMAP, ETRN and ODMR protocols.
• Ability to forward mail using SMTP, which allows filtering, forwarding, and aliasing to function normally.
• May be run in daemon mode to check periodically for new messages.
• Can retrieve multiple mailboxes and forward them, based on configuration, to different local users.

This section explains some of the basic features of fetchmail. This utility requires a .fetchmailrc configuration in the user's home directory in order to run correctly. This file includes server information as well as login credentials. Due to the sensitive nature of the contents of this file, it is advisable to make it readable only by the user, with the following command:

```bash
chmod 600 .fetchmailrc
```

The following .fetchmailrc serves as an example for downloading a single user mailbox using POP. It tells fetchmail to connect to example.com using a username of joesoap and a password of XXX.

This example assumes that the user joesoap exists on the local system.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

I did receive your message...
The next example connects to multiple POP and IMAP servers and redirects to different local usernames where applicable:

```
poll example.com proto pop3:
  user "joesoap", with password "XXX", is "jsoap" here;
  user "andrea", with password "XXXX";
```

```
poll example2.net proto imap:
  user "john", with password "XXXXX", is "myth" here;
```

fetchmail can be run in daemon mode by running it with `-d`, followed by the interval (in seconds) that fetchmail should poll servers listed in `.fetchmailrc`. The following example configures fetchmail to poll every 600 seconds:

```
% fetchmail -d 600
```

More information on fetchmail can be found at [http://www.fetchmail.info/](http://www.fetchmail.info/).

The next section displays some basic procmail rules, as well as brief descriptions of what they do. Rules must be inserted into a `.procmailrc`, which must reside in the user's home directory. The majority of these rules can be found in `procmailex(5)`.

To forward all mail from `user@example.com` to an external address of `goodmail@example2.com`:

```
:0
* ^From.*user@example.com ! goodmail@example2.com
```

To forward all mails shorter than 1000 bytes to an external address of `goodmail@example2.com`:

```
:0
* < 1000 ! goodmail@example2.com
```

To send all mail sent to `alternate@example.com` to a mailbox called `alternate`:

```
586
```

使用 procmail procmail 是一个强大的应用程序，用于过滤 incoming mail。它允许用户定义“规则”可以匹配 incoming mails 来执行特定功能或将其重定向到 alternative mailboxes 或电子邮件地址。procmail 可以通过`mail/procmail`port 安装。

一旦安装，它可以直接集成到大多数 MTAs。参阅 MTA 文档获取更多信息。或者，procmail 可以通过添加以下行到用户 home directory 的`.forward`来集成。

```
"|exec /usr/local/bin/procmail || exit 75"
```

The following section displays some basic procmail rules, as well as brief descriptions of what they do. Rules must be inserted into a `.procmailrc`, which must reside in the user's home directory. The majority of these rules can be found in `procmailex(5)`.

To forward all mail from `user@example.com` to an external address of `goodmail@example2.com`:

```
:0
* ^From.*user@example.com ! goodmail@example2.com
```

To forward all mails shorter than 1000 bytes to an external address of `goodmail@example2.com`:

```
:0
* < 1000 ! goodmail@example2.com
```

To send all mail sent to `alternate@example.com` to a mailbox called `alternate`:

```
586
```
To send all mail with a subject of "Spam" to /dev/null:

```
^Subject:.*Spam
/dev/null
```

A useful recipe that parses incoming FreeBSD.org mailing lists and places each list in its own mailbox:

```
^Sender:.owner-freebsd-[^@]+@FreeBSD.ORG
LISTNAME=${MATCH}
LISTNAME??^/[^@]+FreeBSD-${MATCH}
```
Chapter 29.

29.1. 概述

本章節涵蓋一些在UNIX™系統常用的網路服務，包含安裝、設定、測試及維護各種不同類型的網路服務。本章會提供範例設定檔以供參考。

讀完本章，您將了解:

• 如何管理inetd Daemon。

• 如何設定網路檔案系統（Network File System, NFS）。

• 如何設定網路資訊伺服器（Network Information Server, NIS）來集中管理及共用使用者帳號。

• 如何設定FreeBSD成為LDAP伺服器或客戶端。

• 如何設定使用DHCP自動網路設定。

• 如何設定網域名稱伺服器（Domain Name Server, DNS）。

• 如何設定ApacheHTTP伺服器。

• 如何設定檔案傳輸協定（File Transfer Protocol, FTP）伺服器。

• 如何設定Samba檔案與列印伺服器供Windows™客戶端使用。

• 如何同步時間與日期，並使用網路時間協定（Network Time Protocol, NTP）設定時間伺服器。

• 如何設定iSCSI。

本章假設您有以下基礎知識:

• /etc/rc Script。

• 網路術語。

• 安裝其他第三方軟體（安裝應用程式：套件與Port）。

29.2. inetd 超級伺服器

The inetd(8) daemon is sometimes referred to as a Super-Server because it manages connections for many services. Instead of starting multiple applications, only the inetd service needs to be started. When a connection is received for a service that is managed by inetd, it determines which program the connection is destined for, spawns a process for that program, and delegates the program a socket. Using inetd for services that are not heavily used can reduce system load, when compared to running each daemon individually in stand-alone mode.

Primarily, inetd is used to spawn other daemons, but several trivial protocols are handled internally, such as chargen, auth, time, echo, discard, and daytime.

This section covers the basics of configuring inetd.

29.2.1. 設定檔

Configuration of inetd is done by editing /etc/inetd.conf. Each line of this configuration file represents an application which can be started by inetd. By default, every line starts with a comment ( ), meaning that inetd is not listening for any applications. To configure inetd to listen for an application's connections, remove the at the beginning of the line for that application.

After saving your edits, configure inetd to start at system boot by editing /etc/rc.conf:

```
inetd_enable="YES"
```

To start inetd now, so that it listens for the service you configured, type:

```
# service inetd start
```

Once inetd is started, it needs to be notified whenever a modification is made to

```
/etc/inetd.conf
```

Example:

```
# service inetd reload
```

Typically, the default entry for an application does not need to be edited beyond removing the

```
#
```

In some situations, it may be appropriate to edit the default entry.

As an example, this is the default entry for

```
ftpd(8)
```

over IPv4:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

The seven columns in an entry are as follows:

- **service-name**
- **socket-type**
- **protocol**
- **{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]**
- **user[:group][/login-class]**
- **server-program**
- **server-program-arguments**

where:

- **service-name**
  The service name of the daemon to start. It must correspond to a service listed in
  `/etc/services`.
  This determines which port inetd listens on for incoming connections to that service. When
  using a custom service, it must first be added to
  `/etc/services`.

- **socket-type**
  Either `stream`, `dgram`, `raw`, or `seqpacket`. Use `stream` for TCP connections and `dgram` for UDP
  services.

- **protocol**
  Use one of the following protocol names:

<table>
<thead>
<tr>
<th>Protocol Name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp or tcp4</td>
<td>TCP IPv4</td>
</tr>
<tr>
<td>udp or udp4</td>
<td>UDP IPv4</td>
</tr>
<tr>
<td>tcp6</td>
<td>TCP IPv6</td>
</tr>
<tr>
<td>udp6</td>
<td>UDP IPv6</td>
</tr>
<tr>
<td>tcp46</td>
<td>Both TCP IPv4 and IPv6</td>
</tr>
</tbody>
</table>

- **{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]**
- **user[:group][/login-class]**
- **server-program**
- **server-program-arguments**
Protocol Name
Explanation
udp46
Both UDP IPv4 and IPv6
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]}

In this field, wait or nowait must be specified. max-child, max-connections-per-ip-per-minute and max-child-per-ip are optional.

wait|nowait indicates whether or not the service is able to handle its own socket. dgram socket types must use wait while stream daemons, which are usually multi-threaded, should use nowait. wait usually hands off multiple sockets to a single daemon, while nowait spawns a child daemon for each new socket.

The maximum number of child daemons inetd may spawn is set by max-child. For example, to limit ten instances of the daemon, place a /10 after nowait. Specifying /0 allows an unlimited number of children.

max-connections-per-ip-per-minute limits the number of connections from any particular IP address per minute. Once the limit is reached, further connections from this IP address will be dropped until the end of the minute. For example, a value of /10 would limit any particular IP address to ten connection attempts per minute.

max-child-per-ip limits the number of child processes that can be started on behalf on any single IP address at any moment. These options can limit excessive resource consumption and help to prevent Denial of Service attacks.

An example can be seen in the default settings for fingerd(8):

finger stream tcp nowait/3/10 nobody /usr/libexec/fingerd fingerd -k -s

The username the daemon will run as. Daemons typically run as root, daemon, or nobody.

server-program
The full path to the daemon. If the daemon is a service provided by inetd internally, use internal.

server-program-arguments
Used to specify any command arguments to be passed to the daemon on invocation. If the daemon is an internal service, use internal.

Like most server daemons, inetd has a number of options that can be used to modify its behavior. By default, inetd is started with -wW -C 60. These options enable TCP wrappers for all services, including internal services, and prevent any IP address from requesting any service more than 60 times per minute.

To change the default options which are passed to inetd, add an entry for inetd_flags in /etc/rc.conf. If inetd is already running, restart it with service inetd restart.

The available rate limiting options are:
-c maximum
Specify the default maximum number of simultaneous invocations of each service, where the default is unlimited. May be overridden on a per-service basis by using max-child in /etc/inetd.conf.

-C rate
Specify the default maximum number of times a service can be invoked from a single IP address per minute. May be overridden on a per-service basis by using max-connections-per-ip-per-minute in /etc/inetd.conf.
Specify the maximum number of times a service can be invoked in one minute, where the default is 256. A rate of 0 allows an unlimited number.

Specify the maximum number of times a service can be invoked from a single IP address at any one time, where the default is unlimited. May be overridden on a per-service basis by using max-child-per-ip in /etc/inetd.conf.

Additional options are available. Refer to inetd(8) for the full list of options.

29.2.3. 安全注意事項
Many of the daemons which can be managed by inetd are not security-conscious. Some daemons, such as fingerd, can provide information that may be useful to an attacker. Only enable the services which are needed and monitor the system for excessive connection attempts.

max-connections-per-ip, max-child and max-child-per-ip can be used to limit such attacks.

By default, TCP wrappers is enabled. Consult hosts_access(5) for more information on placing TCP restrictions on various inetd invoked daemons.

29.3. 網路檔案系統 (NFS)
FreeBSD supports the Network File System (NFS), which allows a server to share directories and files with clients over a network. With NFS, users and programs can access files on remote systems as if they were stored locally.

NFS has many practical uses. Some of the more common uses include:

• Data that would otherwise be duplicated on each client can be kept in a single location and accessed by clients on the network.
• Several clients may need access to the /usr/ports/distfiles directory. Sharing that directory allows for quick access to the source files without having to download them to each client.
• On large networks, it is often more convenient to configure a central NFS server on which all user home directories are stored. Users can log into a client anywhere on the network and have access to their home directories.
• Administration of NFS exports is simplified. For example, there is only one file system where security or backup policies must be set.
• Removable media storage devices can be used by other machines on the network. This reduces the number of devices throughout the network and provides a centralized location to manage their security. It is often more convenient to install software on multiple machines from a centralized installation media.

NFS consists of a server and one or more clients. The client remotely accesses the data that is stored on the server machine. In order for this to function properly, a few processes have to be configured and running.

These daemons must be running on the server:

<table>
<thead>
<tr>
<th>Daemon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nfsd</td>
<td>The NFS daemon which services requests from NFS clients.</td>
</tr>
<tr>
<td>mountd</td>
<td>The NFS mount daemon which carries out requests received from nfsd.</td>
</tr>
<tr>
<td>rpcbind</td>
<td>This daemon allows NFS clients to discover which port the NFS server is using.</td>
</tr>
</tbody>
</table>
Running `nfsiod(8)` on the client can improve performance, but is not required.

The file systems which the NFS server will share are specified in `/etc/exports`. Each line in this file specifies a file system to be exported, which clients have access to that file system, and any access options. When adding entries to this file, each exported file system, its properties, and allowed hosts must occur on a single line. If no clients are listed in the entry, then any client on the network can mount that file system.

The following `/etc/exports` entries demonstrate how to export file systems. The examples can be modified to match the file systems and client names on the reader's network. There are many options that can be used in this file, but only a few will be mentioned here. See `exports(5)` for the full list of options.

This example shows how to export `/cdrom` to three hosts named `alpha`, `bravo`, and `charlie`:

```
/cdrom -ro alpha bravo charlie
```

The `-ro` flag makes the file system read-only, preventing clients from making any changes to the exported file system. This example assumes that the host names are either in DNS or in `/etc/hosts`. Refer to `hosts(5)` if the network does not have a DNS server.

The next example exports `/home` to three clients by IP address. This can be useful for networks without DNS or `/etc/hosts` entries. The `-alldirs` flag allows subdirectories to be mount points. In other words, it will not automatically mount the subdirectories, but will permit the client to mount the directories that are required as needed.

```
/usr/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

This next example exports `/a` so that two clients from different domains may access that file system. The `-maproot=root` allows `root` on the remote system to write data on the exported file system as `root`. If `-maproot=root` is not specified, the client's `root` user will be mapped to the server's `nobody` account and will be subject to the access limitations defined for `nobody`.

```
/a  -maproot=root  host.example.com box.example.org
```

A client can only be specified once per file system. For example, if `/usr` is a single file system, these entries would be invalid as both entries specify the same host:

```
# Invalid when /usr is one file system
/usr/src   client
/usr/ports client
```

The correct format for this situation is to use one entry:

```
/usr/src /usr/ports  client
```

The following is an example of a valid export list, where `/usr` and `/exports` are local file systems:
Export src and ports to client01 and client02, but only client01 has root privileges on it

```
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports               client02
```

The client machines have root and can mount anywhere on /exports. Anyone in the world can mount /exports/obj read-only

```
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

To enable the processes required by the NFS server at boot time, add these options to /etc/rc.conf:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_flags="-r"
```

The server can be started now by running this command:

```
# service nfsd start
```

Whenever the NFS server is started, mountd also starts automatically. However, mountd only reads /etc/exports when it is started. To make subsequent /etc/exports edits take effect immediately, force mountd to reread it:

```
# service mountd reload
```

29.3.2. 設定客戶端

To enable NFS clients, set this option in each client's /etc/rc.conf:

```
nfs_client_enable="YES"
```

Then, run this command on each NFS client:

```
# service nfsclient start
```

The client now has everything it needs to mount a remote file system. In these examples, the server's name is server and the client's name is client. To mount /home on server to the /mnt mount point on client:

```
# mount server:/home /mnt
```

The files and directories in /home will now be available on client, in the /mnt directory.

To mount a remote file system each time the client boots, add it to /etc/fstab:
29.3.3. Locking

Some applications require file locking to operate correctly. To enable locking, add these lines to /etc/rc.conf on both the client and server:

```
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Then start the applications:

```
# service lockd start
# service statd start
```

If locking is not required on the server, the NFS client can be configured to lock locally by including `-L` when running `mount`. Refer to `mount_nfs(8)` for further details.

29.3.4. Automatic Mounting

The automatic mounter daemon, amd, automatically mounts a remote file system whenever a file or directory within that file system is accessed. File systems that are inactive for a period of time will be automatically unmounted by amd.

This daemon provides an alternative to modifying `/etc/fstab` to list every client. It operates by attaching itself as an NFS server to the `/host` and `/net` directories. When a file is accessed within one of these directories, amd looks up the corresponding remote mount and automatically mounts it. `/net` is used to mount an exported file system from an IP address while `/host` is used to mount an export from a remote hostname. For instance, an attempt to access a file within `/host/foobar/usr` would tell amd to mount the `/usr` export on the host `foobar`.

Example:

Export

In this example, `showmount -e` shows the exported file systems that can be mounted from the NFS server, `foobar`:

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /host/foobar/usr
```

The output from `showmount` shows `/usr` as an export. When changing directories to `/host/foobar/usr`, amd intercepts the request and attempts to resolve the hostname `foobar`. If successful, amd automatically mounts the desired export.

To enable amd at boot time, add this line to `/etc/rc.conf`:

```
594
```
To start amd now:

```
# service amd start
```

Custom flags can be passed to amd from the `amd_flags` environment variable. By default, `amd_flags` is set to:

```
amd_flags=-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map
```

The default options with which exports are mounted are defined in `/etc/amd.map`. Some of the more advanced features of amd are defined in `/etc/amd.conf`. Consult `amd(8)` and `amd.conf(5)` for more information.

**29.3.5. 使用autofs(5)自动挂载**

The `autofs(5)` automount facility is supported starting with FreeBSD 10.1-RELEASE. To use the automounter functionality in older versions of FreeBSD, use `amd(8)` instead. This chapter only describes the `autofs(5)` automounter.

The `autofs(5)` facility is a common name for several components that, together, allow for automatic mounting of remote and local filesystems whenever a file or directory within that filesystem is accessed. It consists of the kernel component, `autofs(5)`, and several userspace applications: `automount(8)`, `automountd(8)` and `autounmountd(8)`. It serves as an alternative for `amd(8)` from previous FreeBSD releases. Amd is still provided for backward compatibility purposes, as the two use different map format; the one used by autofs is the same as with other SVR4 automounters, such as the ones in Solaris, MacOS X, and Linux.

The `autofs(5)` virtual filesystem is mounted on specified mountpoints by `automount(8)`, usually invoked during boot. Whenever a process attempts to access file within the `autofs(5)` mountpoint, the kernel will notify `automountd(8)` daemon and pause the triggering process. The `automountd(8)` daemon will handle kernel requests by finding the proper map and mounting the filesystem according to it, then signal the kernel to release blocked process. The `autounmountd(8)` daemon automatically unmounts automounted filesystems after some time, unless they are still being used.

The primary `autofs` configuration file is `/etc/auto_master`. It assigns individual maps to top-level mounts. For an explanation of `auto_master` and the map syntax, refer to `auto_master(5)`. There is a special automounter map mounted on `/net`. When a file is accessed within this directory, `autofs(5)` looks up the corresponding remote mount and automatically mounts it. For instance, an attempt to access a file within `/net/foobar/usr` would tell `automountd(8)` to mount the `/usr` export from the host `foobar`.

Example 48. 使用`autofs(5)`挂载Export

In this example, `showmount -e` shows the exported file systems that can be mounted from the NFS server, `foobar`:

```
% showmount -e foobar
Exports list on foobar:
...595
```
The output from `showmount` shows /usr as an export. When changing directories to /host/foobar/usr, `automountd(8)` intercepts the request and attempts to resolve the hostname foobar. If successful, `automountd(8)` automatically mounts the source export.

To enable `autofs(5)` at boot time, add this line to `/etc/rc.conf`:

```
autofs_enable="YES"
```

Then `autofs(5)` can be started by running:

```
# service automount start
# service automountd start
# service autounmountd start
```

The `autofs(5)` map format is the same as in other operating systems. Information about this format from other sources can be useful, like the Mac OS X document.

Consult the `automount(8)`, `automountd(8)`, `autounmountd(8)` and `auto_master(5)` manual pages for more information.

29.4. 網路資訊系統 (NIS)

Network Information System (NIS) is designed to centralize administration of UNIX™-like systems such as Solaris™, HP-UX, AIX™, Linux, NetBSD, OpenBSD, and FreeBSD. NIS was originally known as Yellow Pages but the name was changed due to trademark issues. This is the reason why NIS commands begin with `yp`.

NIS is a Remote Procedure Call (RPC)-based client/server system that allows a group of machines within an NIS domain to share a common set of configuration files. This permits a system administrator to set up NIS client systems with only minimal configuration data and to add, remove, or modify configuration data from a single location.

FreeBSD uses version 2 of the NIS protocol.

29.4.1. NIS 詞彙與程序

Table 28.1 summarizes the terms and important processes used by NIS:

<table>
<thead>
<tr>
<th>詞彙</th>
<th>說明</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIS domain name</td>
<td>NIS servers and clients share an NIS domain name. Typically, this name does not have anything to do with DNS.</td>
</tr>
<tr>
<td><code>rpcbind(8)</code></td>
<td>This service enables RPC and must be running in order to run an NIS server or act as an NIS client.</td>
</tr>
</tbody>
</table>
This service binds an NIS client to its NIS server. It will take the NIS domain name and use RPC to connect to the server. It is the core of client/server communication in an NIS environment. If this service is not running on a client machine, it will not be able to access the NIS server.

This is the process for the NIS server. If this service stops running, the server will no longer be able to respond to NIS requests so hopefully, there is a slave server to take over. Some non-FreeBSD clients will not try to reconnect using a slave server and the ypbind process may need to be restarted on these clients.

This process only runs on NIS master servers. This daemon allows NIS clients to change their NIS passwords. If this daemon is not running, users will have to login to the NIS master server and change their passwords there.

There are three types of hosts in an NIS environment:

• **NIS master server**
  This server acts as a central repository for host configuration information and maintains the authoritative copy of the files used by all of the NIS clients. The `passwd`, `group`, and other various files used by NIS clients are stored on the master server. While it is possible for one machine to be an NIS master server for more than one NIS domain, this type of configuration will not be covered in this chapter as it assumes a relatively small-scale NIS environment.

• **NIS slave servers**
  NIS slave servers maintain copies of the NIS master's data files in order to provide redundancy. Slave servers also help to balance the load of the master server as NIS clients always attach to the NIS server which responds first.

• **NIS clients**
  NIS clients authenticate against the NIS server during log on.

Information in many files can be shared using NIS. The `master.passwd`, `group`, and `hosts` files are commonly shared via NIS. Whenever a process on a client needs information that would normally be found in these files locally, it makes a query to the NIS server that it is bound to instead.

This section describes a sample NIS environment which consists of 15 FreeBSD machines with no centralized point of administration. Each machine has its own `/etc/passwd` and `/etc/master.passwd`. These files are kept in sync with each other only through manual intervention. Currently, when a user is added to the lab, the process must be repeated on all 15 machines.

The configuration of the lab will be as follows:

<table>
<thead>
<tr>
<th>Machine name</th>
<th>IP Address</th>
<th>Machine role</th>
</tr>
</thead>
<tbody>
<tr>
<td>ellington</td>
<td>10.0.0.2</td>
<td>NIS master</td>
</tr>
</tbody>
</table>
If this is the first time an NIS scheme is being developed, it should be thoroughly planned ahead of time. Regardless of network size, several decisions need to be made as part of the planning process.

29.4.3.1. Choosing NIS Domain Name

When a client broadcasts its requests for info, it includes the name of the NIS domain that it is part of. This is how multiple servers on one network can tell which server should answer which request. Think of the NIS domain name as the name for a group of hosts.

Some organizations choose to use their Internet domain name for their NIS domain name. This is not recommended as it can cause confusion when trying to debug network problems. The NIS domain name should be unique within the network and it is helpful if it describes the group of machines it represents. For example, the Art department at Acme Inc. might be in the "acme-art"NIS domain. This example will use the domain name test-domain.

However, some non-FreeBSD operating systems require the NIS domain name to be the same as the Internet domain name. If one or more machines on the network have this restriction, the Internet domain name must be used as the NIS domain name.

29.4.3.2. Choosing a NIS Server

There are several things to keep in mind when choosing a machine to use as a NIS server. Since NIS clients depend upon the availability of the server, choose a machine that is not rebooted frequently. The NIS server should ideally be a stand alone machine whose sole purpose is to be an NIS server. If the network is not heavily used, it is acceptable to put the NIS server on a machine running other services. However, if the NIS server becomes unavailable, it will adversely affect all NIS clients.

29.4.4. Setting NIS Master Server

The canonical copies of all NIS files are stored on the master server. The databases used to store the information are called NIS maps. In FreeBSD, these maps are stored in /var/yp/[domainname] where [domainname] is the name of the NIS domain. Since multiple domains are supported, it is possible to have several directories, one for each domain. Each domain will have its own independent set of maps.

NIS master and slave servers handle all NIS requests through ypser(8). This daemon is responsible for receiving incoming requests from NIS clients, translating the requested domain and map name to a path to the corresponding database file, and transmitting data from the database back to the client.

Setting up a master NIS server can be relatively straight forward, depending on environmental needs. Since FreeBSD provides built-in NIS support, it only needs to be enabled by adding the following lines to /etc/rc.conf:

```bash
nisdomainname="test-domain" ①
nis_server_enable="YES" ②
nis_yppasswdd_enable="YES" ③
```

598
This line sets the NIS domain name to *test-domain*.

This automates the start up of the NIS server processes when the system boots.

This enables the *rpc.yppasswdd(8)* daemon so that users can change their NIS password from a client machine.

Care must be taken in a multi-server domain where the server machines are also NIS clients. It is generally a good idea to force the servers to bind to themselves rather than allowing them to broadcast bind requests and possibly become bound to each other. Strange failure modes can result if one server goes down and others are dependent upon it. Eventually, all the clients will time out and attempt to bind to other servers, but the delay involved can be considerable and the failure mode is still present since the servers might bind to each other all over again.

A server that is also a client can be forced to bind to a particular server by adding these additional lines to `/etc/rc.conf`:

```
nis_client_enable="YES" # run client stuff as well
nis_client_flags="-S NIS domain,server"
```

After saving the edits, type `/etc/netstart` to restart the network and apply the values defined in `/etc/rc.conf`.

Before initializing the NIS maps, start `ypserv(8)`:

```bash
# service ypserv start
```

NIS maps are generated from the configuration files in `/etc` on the NIS master, with one exception: `/etc/master.passwd`. This is to prevent the propagation of passwords to all the servers in the NIS domain. Therefore, before the NIS maps are initialized, configure the primary password files:

```bash
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

It is advisable to remove all entries for system accounts as well as any user accounts that do not need to be propagated to the NIS clients, such as the root and any other administrative accounts.

Ensure that the `/var/yp/master.passwd` is neither group or world readable by setting its permissions to `600`.

After completing this task, initialize the NIS maps. FreeBSD includes the `ypinit(8)` script to do this.

When generating maps for the master server, include `-m` and specify the NIS domain name:

```bash
ellington# ypinit -m test-domain
```

Server Type: MASTER Domain: test-domain

Creating an YP server will require that you answer a few questions. Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n] **n**

Ok, please remember to go back and redo manually whatever fails. If not, something might not work.
At this point, we have to construct a list of this domain's YP servers. rod.darktech.org is already known as the master server. Please continue to add any slave servers, one per line. When you are done with the list, type a <control D>.

Master server: ellington
Next host to add: coltrane
Next host to add: <control D>

The current list of NIS servers looks like this:

ellington
coltrane

Is this correct? [y/n]: y

..output from map generation..

NIS Map update completed.

ellington has been setup as an YP master server without any errors.

This will create /var/yp/Makefile from /var/yp/Makefile.dist. By default, this file assumes that the environment has a single NIS server with only FreeBSD clients. Since test-domain has a slave server, edit this line in /var/yp/Makefile so that it begins with a comment (#):

NOPUSH = "True"

29.4.4.2. 新增使用者

Every time a new user is created, the user account must be added to the master NIS server and the NIS maps rebuilt. Until this occurs, the new user will not be able to login anywhere except on the NIS master. For example, to add the new user jsmith to the test-domain domain, run these commands on the master server:

# pw useradd jsmith
# cd /var/yp
# make test-domain

The user could also be added using adduser jsmith instead of pw useradd smith.

29.4.5. 設定 NIS Slave伺服器

To set up an NIS slave server, log on to the slave server and edit /etc/rc.conf as for the master server. Do not generate any NIS maps, as these already exist on the master server. When running ypinit on the slave server, use -s (for slave) instead of -m (for master). This option requires the name of the NIS master in addition to the domain name, as seen in this example:

coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington
Creating an YP server will require that you answer a few questions. Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors?

y/n: n

Ok, please remember to go back and redo manually whatever fails. If not, something might not work.

There will be no further questions. The remainder of the procedure should take a few minutes, to copy the databases from ellington.

Transferring netgroup...
ypxfr: Exiting: Map successfully transferred

Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred

Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred

Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred

Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred

Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred

Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred

Transferring group.byname...
ypxfr: Exiting: Map successfully transferred

Transferring services.byname...
ypxfr: Exiting: Map successfully transferred

Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred

Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred

Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred

Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred

Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred

Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred

Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred

Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...

coltrane has been setup as an YP slave server without any errors.
Remember to update map ypservers on ellington.

This will generate a directory on the slave server called
/var/yp/test-domain
which contains copies
of the NIS master server's maps. Adding these
/etc/crontab
entries on each slave server will force
the slaves to sync their maps with the maps on the master server:
20      *       *       *       *       root   /usr/libexec/ypxfr passwd.byname
21      *       *       *       *       root   /usr/libexec/ypxfr passwd.byuid

These entries are not mandatory because the master server automatically attempts to push any
map changes to its slaves. However, since clients may depend upon the slave server to provide
correct password information, it is recommended to force frequent password map updates. This is
especially important on busy networks where map updates might not always complete.

To finish the configuration, run
/var/yp/test-domain
on the slave server in order to start the NIS services.

29.4.6. 設定
NIS 客戶端
An NIS client binds to an NIS server using
ypbind(8). This daemon broadcasts RPC requests on the
local network. These requests specify the domain name configured on the client. If an NIS server in
the same domain receives one of the broadcasts, it will respond to ypbind, which will record the
server's address. If there are several servers available, the client will use the address of the first
server to respond and will direct all of its NIS requests to that server. The client will automatically
ping the server on a regular basis to make sure it is still available. If it fails to receive a reply within a
reasonable amount of time, ypbind will mark the domain as unbound and begin broadcasting again
in the hopes of locating another server.

To configure a FreeBSD machine to be an NIS client:
1. Edit
/etc/rc.conf
and add the following lines in order to set the NIS domain name and start
ypbind(8):
nisdomainname="test-domain"
nis_client_enable="YES"

2. To import all possible password entries from the NIS server, use
vipw to remove all user
accounts except one from
/etc/master.passwd. When removing the accounts, keep in mind
that at least one local account should remain and this account should be a member of
wheel. If there is a problem with NIS, this local account can be used to log in remotely,
become the superuser, and fix the problem. Before saving the edits, add the following line
to the end of the file:
This line configures the client to provide anyone with a valid account in the NIS server's password maps an account on the client. There are many ways to configure the NIS client by modifying this line. One method is described in 使用 Netgroups. For more detailed reading, refer to the book Managing NFS and NIS, published by O'Reilly Media.

To import all possible group entries from the NIS server, add this line to /etc/group:

```plaintext
+:*::
```

To start the NIS client immediately, execute the following commands as the superuser:

```
# /etc/netstart
# service ypbind start
```

After completing these steps, running `ypcat passwd` on the client should show the server's passwd map.

### 29.4.7. NIS 安全性

Since RPC is a broadcast-based service, any system running ypbind within the same domain can retrieve the contents of the NIS maps. To prevent unauthorized transactions, `ypserv(8)` supports a feature called "securenets" which can be used to restrict access to a given set of hosts. By default, this information is stored in `/var/yp/securenets`, unless `ypserv(8)` is started with `-p` and an alternate path. This file contains entries that consist of a network specification and a network mask separated by white space. Lines starting with `#` are considered to be comments. A sample securenets might look like this:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0      255.255.240.0
```

If `ypserv(8)` receives a request from an address that matches one of these rules, it will process the request normally. If the address fails to match a rule, the request will be ignored and a warning message will be logged. If the securenets does not exist, `ypserv` will allow connections from any host.

TCP Wrapper is an alternate mechanism for providing access control instead of securenets. While either access control mechanism adds some security, they are both vulnerable to "IP spoofing" attacks. All NIS-related traffic should be blocked at the firewall.

Servers using securenets may fail to serve legitimate NIS clients with archaic TCP/IP...
implementations. Some of these implementations set all host bits to zero when doing broadcasts or fail to observe the subnet mask when calculating the broadcast address. While some of these problems can be fixed by changing the client configuration, other problems may force the retirement of these client systems or the abandonment of securenets.

The use of TCP Wrapper increases the latency of the NIS server. The additional delay may be long enough to cause timeouts in client programs, especially in busy networks with slow NIS servers. If one or more clients suffer from latency, convert those clients into NIS slave servers and force them to bind to themselves.

29.4.7.1. Barring

In this example, the basic system is a faculty workstation within the NIS domain. The passwd map on the master NIS server contains accounts for both faculty and students. This section demonstrates how to allow faculty logins on this system while refusing student logins.

To prevent specified users from logging on to a system, even if they are present in the NIS database, use `vipw` to add `-username` with the correct number of colons towards the end of `/etc/master.passwd` on the client, where `username` is the username of a user to bar from logging in.

The line with the blocked user must be before the `+` line that allows NIS users. In this example, bill is barred from logging on to basic:

```
basic# cat /etc/master.passwd
root:[password]:0:0::0:0:The super-user:/root:/bin/csh
tooral:[password]:0:0::0:0:The other super-user:/root:/bin/sh
developer:*:1:1::0:0:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5::0:0:System &://:/usr/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/usr/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/usr/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/usr/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/usr/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/shared/man:/usr/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/usr/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/usr/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/usr/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/usr/sbin/nologin
-bill::::::::+
```

29.4.8. Netgroups

Barring specified users from logging on to individual systems becomes unscaleable on larger networks and quickly loses the main benefit of NIS: centralized administration.

Netgroups were developed to handle large, complex networks with hundreds of users and machines. Their use is comparable to UNIX™ groups, where the main difference is the lack of a numeric ID and the ability to define a netgroup by including both user accounts and other...
To expand on the example used in this chapter, the NIS domain will be extended to add the users and systems shown in Tables 28.2 and 28.3:

### Table 24. Other Users

<table>
<thead>
<tr>
<th>User</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha, beta</td>
<td>IT department employees</td>
</tr>
<tr>
<td>charlie, delta</td>
<td>IT department apprentices</td>
</tr>
<tr>
<td>echo, foxtrott, golf</td>
<td>employees</td>
</tr>
<tr>
<td>able, baker</td>
<td>interns</td>
</tr>
</tbody>
</table>

### Table 25. Other Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>war, death, famine, pollution</td>
<td>Only IT employees are allowed to log onto these servers.</td>
</tr>
<tr>
<td>pride, greed, envy, wrath, lust, sloth</td>
<td>All members of the IT department are allowed to login onto these servers.</td>
</tr>
<tr>
<td>one, two, three, four</td>
<td>Ordinary workstations used by employees.</td>
</tr>
<tr>
<td>trashcan</td>
<td>A very old machine without any critical data. Even interns are allowed to use this system.</td>
</tr>
</tbody>
</table>

When using netgroups to configure this scenario, each user is assigned to one or more netgroups and logins are then allowed or forbidden for all members of the netgroup. When adding a new machine, login restrictions must be defined for all netgroups. When a new user is added, the account must be added to one or more netgroups. If the NIS setup is planned carefully, only one central configuration file needs modification to grant or deny access to machines.

The first step is the initialization of the NIS netgroup map. In FreeBSD, this map is not created by default. On the NIS master server, use an editor to create a map named `/var/yp/netgroup`.

This example creates four netgroups to represent IT employees, IT apprentices, employees, and interns:

IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) 
INTERNS (,able,test-domain)     (,baker,test-domain)

Each entry configures a netgroup. The first column in an entry is the name of the netgroup. Each set of brackets represents either a group of one or more users or the name of another netgroup. When specifying a user, the three comma-delimited fields inside each group represent:

1. The name of the host(s) where the other fields representing the user are valid. If a hostname is not specified, the entry is valid on all hosts.
2. The name of the account that belongs to this netgroup.
3. The NIS domain for the account. Accounts may be imported from other NIS domains into a netgroup.

If a group contains multiple users, separate each user with whitespace. Additionally, each field may...
contain wildcards. See netgroup(5) for details. Netgroup names longer than 8 characters should not be used. The names are case sensitive and using capital letters for netgroup names is an easy way to distinguish between user, machine and netgroup names.

Some non-FreeBSD NIS clients cannot handle netgroups containing more than 15 entries. This limit may be circumvented by creating several sub-netgroups with 15 users or fewer and a real netgroup consisting of the sub-netgroups, as seen in this example:

```
BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) 
  [...]
BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
BIGGRP3  (,joe31,domain)  (,joe32,domain)
```

BIGGROUP  BIGGRP1  BIGGRP2  BIGGRP3

Repeat this process if more than 225 (15 times 15) users exist within a single netgroup.

To activate and distribute the new NIS map:

```
ellington# cd /var/yp
nellington# make
```

This will generate the three NIS maps `netgroup`, `netgroup.byhost` and `netgroup.byuser`. Use the `map key` option of `ypcat(1)` to check if the new NIS maps are available:

```
ellington% ypcat -k netgroup
nellington% ypcat -k netgroup.byhost
nellington% ypcat -k netgroup.byuser
```

The output of the first command should resemble the contents of `/var/yp/netgroup`. The second command only produces output if host-specific netgroups were created. The third command is used to get the list of netgroups for a user.

To configure a client, use `vipw(8)` to specify the name of the netgroup. For example, on the server named `war`, replace this line:

```
+:::::::::
```

with

```
+@IT_EMP:::::::::
```

This specifies that only the users defined in the netgroup `IT_EMP` will be imported into this system's password database and only those users are allowed to login to this system. This configuration also applies to the `~` function of the shell and all routines which convert between user names and numerical user IDs. In other words, `cd ~user` will not work, `ls -l` will show the numerical ID instead of the username, and `find . -user joe -print` will fail with the message `No such user`. To fix this, import all user entries without allowing them to login into the servers. This can be achieved by adding an extra line:

```
606
```
This line configures the client to import all entries but to replace the shell in those entries with /usr/sbin/nologin. Make sure that extra line is placed after +@IT_EMP:::::::::. Otherwise, all user accounts imported from NIS will have /usr/sbin/nologin as their login shell and no one will be able to login to the system.

To configure the less important servers, replace the old +::::::::: on the servers with these lines:

+@IT_EMP:::::::::
+@IT_APP:::::::::
+:::::::::/usr/sbin/nologin

The corresponding lines for the workstations would be:

+@IT_EMP:::::::::
+@USERS:::::::::
+:::::::::/usr/sbin/nologin

NIS supports the creation of netgroups from other netgroups which can be useful if the policy regarding user access changes. One possibility is the creation of role-based netgroups. For example, one might create a netgroup called BIGSRV to define the login restrictions for the important servers, another netgroup called SMALLSRV for the less important servers, and a third netgroup called USERBOX for the workstations. Each of these netgroups contains the netgroups that are allowed to login onto these machines. The new entries for the NIS netgroup map would look like this:

BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
USERBOX   IT_EMP  ITINTERN USERS

This method of defining login restrictions works reasonably well when it is possible to define groups of machines with identical restrictions. Unfortunately, this is the exception and not the rule. Most of the time, the ability to define login restrictions on a per-machine basis is required.

Machine-specific netgroup definitions are another possibility to deal with the policy changes. In this scenario, the /etc/master.passwd of each system contains two lines starting with "+". The first line adds a netgroup with the accounts allowed to login onto this machine and the second line adds all other accounts with /usr/sbin/nologin as shell. It is recommended to use the "ALL-CAPS" version of the hostname as the name of the netgroup:

+@BOXNAME:::::::::
+:::::::::/usr/sbin/nologin

Once this task is completed on all the machines, there is no longer a need to modify the local versions of /etc/master.passwd ever again. All further changes can be handled by modifying the NIS map. Here is an example of a possible netgroup map for this scenario:
# Define groups of users first

IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)

# Now, define some groups based on roles

USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS

# And a groups for a special tasks

# Allow echo and golf to access our anti-virus-machine
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)

# machine-based netgroups
# Our main servers
WAR       BIGSRV
FAMINE    BIGSRV
# User india needs access to this server
POLLUTION  BIGSRV  (,india,test-domain)

# This one is really important and needs more access restrictions
DEATH     IT_EMP

# The anti-virus-machine mentioned above
ONE       SECURITY

# Restrict a machine to a single user
TWO       (,hotel,test-domain)
# [...more groups to follow]

It may not always be advisable to use machine-based netgroups. When deploying a couple of dozen or hundreds of systems, role-based netgroups instead of machine-based netgroups may be used to keep the size of the NIS map within reasonable limits.

29.4.9.

密

密码格式

NIS requires that all hosts within an NIS domain use the same format for encrypting passwords. If users have trouble authenticating on an NIS client, it may be due to a differing password format. In

608
In this example, the system is using the DES format. Other possible values are **blf** for Blowfish and **md5** for MD5 encrypted passwords.

If the format on a host needs to be edited to match the one being used in the NIS domain, the login capability database must be rebuilt after saving the change:

```
# cap_mkdb /etc/login.conf
```

The format of passwords for existing user accounts will not be updated until each user changes their password after the login capability database is rebuilt.

---

29.5.

**Lightweight Directory Access Protocol (LDAP)**

LDAP is a protocol that uses distributed directory data services to do retrieval, modification, and authentication of objects. It can be seen as a telephone or address book that can store multiple levels of similar data, such as email authentication, employee contact information, and internal website authentication.

This chapter will introduce how to quickly set up an LDAP server on the FreeBSD system. This chapter assumes the administrator has done the planning, which includes:

- Storing what kind of information
- What these pieces of information are used for
- Which users have access to these pieces of information
- How to ensure these pieces of information are not accessed by unauthorized individuals.

29.5.1. LDAP Terminology and Structure

LDAP uses several terms that must be understood before setting it up. All directory entries are composed of a group of properties (attributes), each of which has a unique identifier called a distinguished name (DN) that is made up of other properties, such as common or relative distinguished name (RDN) as absolute path and relative path, DN can be considered as absolute path, RDN as relative path.

LDAP examples:

```
% ldapsearch -xb "uid=trhodes,ou=users,o=example.com"
```

# extended LDIF

# LDAPv3
# base <uid=trhodes,ou=users,o=example.com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL

---

---
# trhodes, users, example.com

dn: 
uid = trhodes, ou = users, o = example.com
mail: trhodes@example.com
cn: Tom Rhodes
uid: trhodes
telephoneNumber: (123) 456-7890

# search result

search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1

29.5.2. 設定 LDAP

FreeBSD 並未提供內建的 LDAP 伺服器，要開始設定前請先安裝 net/openldap-server 套件或 Port:

```
# pkg install openldap-server
```

在套件中已啟用了許多預設選項，可以透過執行 `pkg info openldap-server` 來查看已啟用的選項，若有不足的地方（例如需要啟用 SQL 的支援），請考慮使用適當的方式重新編譯該 Port。

安裝程序會建立目錄 `/var/db/openldap-data` 來儲存資料，同時需要建立儲存憑證的目錄:

```
# mkdir /usr/local/etc/openldap/private
```

接下來是設定憑證機構 (Certificate authority)。以下指令必須在 `/usr/local/etc/openldap/private` 下執行，這很重要是因為檔案權限須要被限制且其他使用者不應有這些檔案的存取權限，更多關於憑證的詳細資訊及相關的參數可在 OpenSSL 中找到。要建立憑證授權，需先輸入這個指令並依提示操作:

```
# openssl req -days 365 -nodes -new -x509 -keyout ca.key -out ../ca.crt
```

提示輸入的項目除了通用名稱 (Common Name) 外其他是可以一樣的，這個項目必須使用跟系統主機名稱不同的名稱。若這是一個自行簽署的憑證 (Self signed certificate)，則在憑證機構 CA 的前面加上主機名稱。

接下來的工作是建立一個伺服器的憑證簽署請求與一個私鑰。請輸入以下指令然後依提示操作:

```
# openssl req -days 365 -nodes -new -keyout server.key -out server.csr
```
在憑證產生程序的過程中請確認
Common Name屬性設定正確。
憑證簽署請求（Certificate Signing Request）必須經過憑證機構簽署後才會成為有效的憑證：

```
# openssl x509 -req -days 365 -in server.csr -out ../server.crt -CA ../ca.crt -CAkey ca.key
```

在憑證產生程序的最後一步是產生並簽署客戶端憑證：

```
# openssl req -days 365 -nodes -new -keyout client.key -out client.csr
# openssl x509 -req -days 3650 -in client.csr -out ../client.crt -CA ../ca.crt -CAkey ca.key
```

記得當提示時要使用同樣的 Common Name屬性。完成之後，請確認為執行的指令產生了8個新檔案。

OpenLDAP伺服器所執行的Daemon為slapd，OpenLDAP是透過slapd.ldif來做設定，OpenLDAP官方已停止採用舊的slapd.conf格式。這裡有些slapd.ldif的設定檔範例可以使用，同時您也可以在/usr/local/etc/openldap/slapd.ldif.sample找到範例資訊。相關可用的選項在slapd-config(5)文件會有說明。

slapd.ldif的每個段落，如同其他LDAP屬性設定一樣會透過唯一DN來辨識，並請確保dn:描述與其相關屬性之間沒有空行。以下的範例中會實際作一個使用TLS的安全通道，首先是全域的設定：

```
# See slapd-config(5) for details on configuration options.
# This file should NOT be world readable.

# This file should NOT be world readable.

# Define global ACLs to disable default read access.

olcArgsFile: /var/run/openldap/slapd.args
olcPidFile: /var/run/openldap/slapd.pid
olcTLSCertificateFile: /usr/local/etc/openldap/server.crt
olcTLSCertificateKeyFile: /usr/local/etc/openldap/private/server.key
olcTLSCACertificateFile: /usr/local/etc/openldap/ca.crt
#olcTLSCipherSuite: HIGH
olcTLSProtocolMin: 3.1
olcTLSVerifyClient: never
```

這個檔案中必須指定憑證機構（Certificate Authority）、伺服器憑證（Server Certificate）與伺服器私鑰（Server Private Key），建議可讓客戶端決定使用的安全密碼（Security Cipher），略過olcTLSCipherSuite選項（此選項不相容openssl以外的TLS客戶端）。

選項olcTLSProtocolMin讓伺服器可求一個安全等級的最低限度，建議使用。伺服器有進行驗証的必要，但客戶端並不需要，因此可設定olcTLSVerifyClient: never。
第二個部份是設定後端要採用的模組有那些，可使用以下方式設定:

# Load dynamic backend modules:

dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/local/libexec/openldap
olcModuleload: back_mdb.la
#olcModuleload: back_bdb.la
#olcModuleload: back_hdb.la
#olcModuleload: back_ldap.la
#olcModuleload: back_passwd.la

第三個部份要載入資料庫所需的ldif綱要(Schema)，這個動作是必要的。

dn: cn=schema,cn=config
objectClass: olcSchemaConfig
cn: schema
include: file:///usr/local/etc/openldap/schema/core.ldif
include: file:///usr/local/etc/openldap/schema/cosine.ldif
include: file:///usr/local/etc/openldap/schema/inetorgperson.ldif
include: file:///usr/local/etc/openldap/schema/nis.ldif

接下來是前端設定的部份:

# Frontend settings

# Sample global access control policy:
#   Root DSE: allow anyone to read it
#   Subschema (sub)entry DSE: allow anyone to read it
#   Other DSEs:
#       Allow self write access
Allow authenticated users read access
Allow anonymous users to authenticate

olcAccess: to dn.base="" by * read
olcAccess: to dn.base="cn=Subschema" by * read

olcAccess: to * by self write
by users read
by anonymous auth

if no access controls are present, the default policy
allows anyone and everyone to read anything but restricts
updates to rootdn. (e.g., "access to * by * read")

rootdn can always read and write EVERYTHING!

olcPasswordHash: {SSHA}

再來是設定後端的部份,之後唯一能夠存取OpenLDAP伺服器設定的方式是使用全域超級使用者。

dn: olcDatabase={0}config,cn=config
objectClass: olcDatabaseConfig
olcDatabase: {0}config
olcAccess: to * by * none
olcRootPW: {SSHA}iae+lrQZILpiUdf16Z9KmDmSwT77Dj4U

預設的管理者使用者名稱是cn=config，可在Shell中輸入slappasswd，決定要使用的密碼並將其產生的編碼放到olcRootPW欄位中。若這個選項在這時沒設定好，在匯入slapd.ldif之後將沒有任何人有辦法修改全域的設定。

最後一個部份是有關資料庫後端的設定:

#######################################################################
# LMDB database definitions
#######################################################################

dn: olcDatabase=mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: mdb
olcDbMaxSize: 1073741824
olcSuffix: dc=domain,dc=example
olcRootDN: cn=mdbadmin,dc=domain,dc=example
Cleartext passwords, especially for the rootdn, should be avoided. See slappasswd(8) and slapd-config(5) for details. Use of strong authentication encouraged.

olcRootPW: {SSHA}X2wHvIWDk6G76CQyCMS1vDCvtICWgn0+

The database directory MUST exist prior to running slapd AND should only be accessible by the slapd and slap tools. Mode 700 recommended.

olcDbDirectory: /var/db/openldap-data

Indices to maintain:

olcDbIndex: objectClass eq

Here, the database directory contains four slapd.ldif files. To convert the current slapd.conf into slapd.ldif format, refer to this page (note that this section also introduces some uncommon options).

Once the settings are completed, place the slapd.ldif in an empty directory. It is recommended to build it as follows:

# mkdir /usr/local/etc/openldap/slapd.d/

Enter the configuration database:

# /usr/local/sbin/slapadd -n0 -F /usr/local/etc/openldap/slapd.d/ -l /usr/local/etc/openldap/slapd.ldif

Start the slapd daemon:

# /usr/local/libexec/slapd -F /usr/local/etc/openldap/slapd.d/

The option -d can be used for debugging, like slapd(8) describes. To check if the server is running correctly:

# ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts

Extended LDIF

# LDAPv3
# base <> with scope baseObject
# filter: (objectclass=*)
# requesting: namingContexts

614
伺服器端仍必須受到信任，若在此之前未做過這個動作，請依照以下指示操作。安裝OpenSSL套件或Port:

```
# pkg install openssl
```

進入ca.crt所在的目錄（以這邊使用的例子來說則是/usr/local/etc/openldap），執行:

```
# c_rehash .
```

現在CA與伺服器憑證可以依其用途被辨識，可進入server.crt所在的目錄執行以下指令來檢查:

```
# openssl verify -verbose -CApath . server.crt
```

若slapd已正執行，就重新啟動它。如同/usr/local/etc/rc.d/slapd所述，要讓slapd開機時可正常執行，須要加入以下行到/etc/rc.conf:

```
lapd_enable="YES"
slapd_flags='-h "ldapi://%2fvar%2frun%2fopenldap%2fldapi/
ldap://0.0.0.0/"'
slapd_sockets="/var/run/openldap/ldapi"
```

開機啟動slapd並不會提供除錯的功能，您可以檢查/var/log/debug.log，dmesg -a及/var/log/messages檢確認為否有正常運作。

以下範例會新增群組team及使用者john到domain.example LDAP資料庫，而該資料庫目前是空的。首先要先建立domain.ldif檔:

```
# cat domain.ldif
dn: dc=domain,dc=example
objectClass: dcObject
objectClass: organization
o: domain.example
dc: domain
```
請查看OpenLDAP說明文件取得更詳細的資訊，使用slappasswd來將純文字的密碼secret更改為已編碼的型式來填寫userPassword欄位。在loginShell所指定的路徑，必須在所有可讓john登入的系統中存在。最後是使用mdb管理者修資料庫:

```bash
# ldapadd -W -D "cn=mdbadmin,dc=domain,dc=example" -f domain.ldif
```

要修改全域設定只能使用全域的超及使用者。例如，假設一開始採用了olcTLSCipherSuite:HIGH:MEDIUM:SSLv3選項，但最後想要把它移除，可以建立一個有以下內容的檔案:

```bash
# cat global_mod

dn: cn=config
changetype: modify
delete: olcTLSCipherSuite
```

然後套用修改內容:
介面\n實際為\n設定態\n動名稱要,如\n的網路\n使用子中的\n若系統\n設定替,這行可能會\n在安裝時\n例已存\n已經\n在\n問\n同步式會\n到\n已設置\n暫\n直\n停啟這個\n完成。避免\n題,因為\nDHCP\n模動可\n訊尚未分\n失敗網\n前\n行配\n資。使用同步\n則執\n路位置\(會\能完成,若網\才\長的時\路\些系統可能需要\在\間\試\n在\DHCP\較\某\嘗\n伺服器可以\作的很\背景\好\求\n在\DHCP\時可\運\快速\的\DHCP\DHCP\回\請\應客戶端\方\。執行\啟\動\DHCP\程序的同時其他\續\啟\會\Script\来加速\行,\。執行\啟\動\DHCP\同步\稱\執\會在背景\時,其\FreeBSD\預\行或\當\客戶端\設\(開機\\)
dhcp-options(5)\。如\的\位\DNS\告\遮罩\關,\子網\設\道\單\與其他相\詳細\可在\路\路\清、預\資\閘\址\訴\訊\將\會在\埠\回\67\UDP\IP\號\68\覆\則\來\。而伺服器,\當\更簡單\路位置\的\資\訊變\網\收\。請参考\FreeBSD\在新安裝的系統\從\內\安裝程式當中,這\自\上設定\客戶端\DHCP\DHCP\動\讓\在\含\要\的\DHCP\設定\。將\介紹如\何\本節,接著會介紹如\安裝並\設定\建\DHCP\isc.org/downloads/dhcp/\整\協\套\件集中有\說明\定\多可用的伺服器。有\Port\關\訊\位於\DHCP\許\通\的完\DHCP\回\請\應\broker\司\台\機\的\部份,在同一\中也可以同時有安裝\端\主\設定\LDAP\針對\說明只\本章節的\伺服器\改\可以修\法\刪\無\問\重\並\常\使用者\取\後\行,或是\域\超級\全\個\後\建\存\若有\端\整\端設定:\法正\的\設定\題\除\的\,可以\執\電\\newpage\n# rm -rf /usr/local/etc/openldap/slapd.d/
# ldapmodify -f global_mod -x -D "cn=config" -W

dhclient
# ifconfig_fxp0="DHCP"

Asynchronously\DHCP\執\模")

Synchronous\DHCP\執\模")

RFC 2131
要改設定系統採用同步模式，在啟動時暫停等候DHCP完成，使用“SYNCDHCP”:

```
ifconfig_fxp0="SYNCDHCP"
```

尚有其他可用的客戶端選項，請在`rc.conf(5)`搜尋`dhclient`來取得詳細資訊。

DHCP客戶端會使用到以下檔案:

- `/etc/dhclient.conf` dhclient用到的設定檔。通常這個檔案只會有註解，因為預設便適用大多數客戶端。這個設定檔在`dhclient.conf(5)`中有說明。
- `/sbin/dhclient`有關指令本身的更多資訊可於`dhclient(8)`找到。
- `/sbin/dhclient-script` FreeBSD特定的DHCP客戶端設定Script。在`dhclient-script(8)`中有說明，但應不須做任何修改便可正常運作。
- `/var/db/dhclient.leases.interface` DHCP客戶端會在這個檔案中儲存有效的約約的資料，寫入的格式類似日誌，在`dhclient.leases(5)`有說明。

29.6.2. 安裝並設定DHCP伺服器

本節將示範如何設定FreeBSD系統成為DHCP伺服器，使用Internet Systems Consortium (ISC)所實作的DHCP伺服器，這個伺服器及其文件可使用`net/isc-dhcp44-server`套件或Port安裝。

`net/isc-dhcp44-server`的安裝程式會安裝一份範例設定檔，複製 `/usr/local/etc/dhcpd.conf.example` 到 `/usr/local/etc/dhcpd.conf` 並在這個新檔案做編輯。

這個設定檔內容包括了子網路及主機的宣告，用來定義要提供給DHCP客戶端的資訊。如以下設定:

```
option domain-name "example.org";
option domain-name-servers ns1.example.org;
option subnet-mask 255.255.255.0;
default-lease-time 600;
max-lease-time 72400;
ddns-update-style none;
```

```
subnet 10.254.239.0 netmask 255.255.255.224 {
range 10.254.239.10 10.254.239.20;
option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}
```

```
host fantasia {
hardware ethernet 08:00:07:26:c0:a5;
}
```
這個選項指定要提供給客戶端的預設搜尋網域。請參考 resolv.conf(5)取得更多資訊。

這個選項指定了客戶端應使用的DNS伺服器清單（以逗號分隔）。如範例中所示，可使用伺服器的完整網域名稱（Fully Qualified Domain Names, FQDN）或伺服器的IP位址。

要提供給客戶端的子網路遮罩。

預設租約到期時間（秒）。客戶端可以自行設定覆蓋這個數值。

一個租約最多允許的時間長度（秒）。若客戶端請求更長的租約，仍會發出租約，但最多只會在 max-lease-time內有效。

預設的none會關閉動態DNS更新。更改此值為interim可讓DHCP伺服器每當發出一個租約便通知DNS伺服器更新，如此一來DNS伺服器便知道網路中該電腦的IP位址。不要更改此預設值，除非DNS伺服器已設定為支援動態DNS。

此行會建立一個可用IP位址的儲存池來保留這些要分配給DHCP客戶端的位址。位址範圍必須在前一行所指定的網路或子網路中有效。

宣告在開始的{括號之前所指定的網路或子網路中有效的預設通訊閘。

指定客戶端的硬體MAC位址，好讓DHCP伺服器在客戶端發出請求時可以辨識客戶端。

指定這個主機應分配相同的IP位址。在此處用主機名稱是正確的，由於DHCP伺服器會在回傳租約資料前先解析主機名稱。此設定檔還支援其他選項，請參考隨伺服器一併安裝的dhcpd.conf(5)來取得詳細資訊與範例。

完成dhcpd.conf的設定之後，在/etc/rc.conf啟動DHCP伺服器：

dhcpd_enable="YES"
dhcpd_ifaces="dc0"

替換dc0為DHCP伺服器要傾聽DHCP客戶端請求的網路介面（多個介面可以空白分隔）。執行以下指令來啟動伺服器：

# service isc-dhcpd start

往後任何對伺服器設定的變更會需要使用service(8)中止dhcpd服務然後啟動。

DHCP伺服器會使用到以下檔案。注意，操作手冊會與伺服器軟體一同安裝。

• /usr/local/sbin/dhcpd 更多有關dhcpd伺服器的資訊可在dhcpd(8)找到。

• /usr/local/etc/dhcpd.conf 伺服器設定檔需要含有所有要提供給客戶端的資訊以及有關伺服器運作的資訊。在dhcpd.conf(5)有此設定檔的說明。

• /var/db/dhcpd.leases DHCP伺服器會儲存一份已發出租約的資料於這個檔案，寫入的格式類似日誌。參考dhcpd.leases(5)會有更完整的說明。

• /usr/local/sbin/dhcrelay
這些資詢被詢問到網,其他的查一次則為再伺服器不需要訊已經對之外取了向區域在本做查,因其快查,使用本伺服器然後接回,地覆取DNS快DNS有收快名稱取其的伺服器,只需要當查詢伺服器名稱情況取台以下快會需要一:

29.7.1. 有的以下名稱情況伺服器權稱:

(Authoritative)

名稱(有名稱快解析伺服器通常有兩伺服器。式權取種或的伺服器與名稱稱:

(Origin)

29.7. 服務器的架設因名稱要根架構機區更具很像:

每案系統檔名稱要主份選一部份的主例會機更詳細,名稱在方,加左如您所見轄範例:

區

術語表26. DNS有以下表格會說明一些與關的術語:

DNS查名稱一個伺服器。架設並不需要詢域來結上做。要在系統名稱取個自的網及其他伺服器小模規快資果,而這些伺服器可管理與DNS訊取得根網層、最網調際網路上節域會協權(Top Level Domain, TLD)的點上址(Domain Name System, DNS)位轉換網,反為域名稱是一種協然。之DNS定用網亦來IP域名稱系統網亦來IP域名稱系統網

1.168.192.in-addr.arpa

org.

example.org.

net/isc-dhcp44-relay

dhcrelay(8)

www.FreeBSD.org
Unbound 是由 FreeBSD 基础系统提供的 DNS 服务器，预设只会提供本机的 DNS 解析。虽然基础系统的套件可以被设定提供本机以外的解析服务，但要解决这样的需求仍建议安装 FreeBSD Port 套件中的 Unbound。

要开启 Unbound 可加入以下到 /etc/rc.conf:

local_unbound_enable="YES"

任何已在 /etc/resolv.conf 中的名称服务器会在新的 Unbound 设置中被设为追随者（Forwarder）。

若任何一个清单中的名称服务器不支援 DNSSEC，则本地的 DNS 解析便会失败，请确认有测试每一台名称服务器并移除所有测试失败的项目。以下指令会显示出信任树或在 192.168.1.1 上执行失败的名称服务器:

```bash
% drill -S FreeBSD.org @192.168.1.1
```

确认每一台名称服务器都支援 DNSSEC 后启动 Unbound:

```
# service local_unbound onestart
```

这将会更新 /etc/resolv.conf 来让查询已用 DNSSEC 确保安全的域现在可以运作，例如，执行以下指令来检验 FreeBSD.org DNSSEC 信任树:

```
% drill -S FreeBSD.org
```

Number of trusted keys: 1

Chasing: freebsd.org. A

DNSSEC Trust tree:
freebsd.org. (A)
|---freebsd.org. (DNSKEY keytag: 36786 alg: 8 flags: 256)
|---freebsd.org. (DNSKEY keytag: 32659 alg: 8 flags: 257)
|---freebsd.org. (DS keytag: 32659 digest type: 2)
|---org. (DNSKEY keytag: 49587 alg: 7 flags: 256)
|---org. (DNSKEY keytag: 9795 alg: 7 flags: 257)
|---org. (DNSKEY keytag: 21366 alg: 7 flags: 257)
|---org. (DS keytag: 21366 digest type: 1)
|   |---. (DNSKEY keytag: 40926 alg: 8 flags: 256)
|       |---. (DNSKEY keytag: 19036 alg: 8 flags: 257)
|---org. (DS keytag: 21366 digest type: 2)
|---. (DNSKEY keytag: 40926 alg: 8 flags: 256)
|---. (DNSKEY keytag: 19036 alg: 8 flags: 257)

Chase successful
Apache HTTP 伺服器是目前最广泛的网页服务器。FreeBSD 默认不会安装这个网页服务器，但可以通过 www/apache24 套件或 Port 安装。

本节将摘要如何在 FreeBSD 上设置和启动 2.x 版的 Apache HTTP Server。要取得有关 Apache 的更详细资讯及其设定项目，请参考 httpd.apache.org。

29.8.1. 设置并启动 Apache 在 FreeBSD 中，主 Apache HTTP Server 设定档会安装于 /usr/local/etc/apache2x/httpd.conf，其中 x 代表版本号。这份 ASCII 文字档中以 # 做为行首的是注解，而最常需修改的项目有：

- ServerRoot “/usr/local” 指定该 Apache 的预设安装路径，Binary 档会储存在服务器根目录（Server root）下的 bin 与 sbin 子目录，而设定档会储存在 etc/apache2x 子目录。
- ServerAdmin you@example.com 更改此项目为你要接收问题回报的电子邮件地址，这个地址也会显示在一些服务器产生的页面上，如：错误页面。
- ServerName www.example.com:80 让管理者可以设定伺服器要回传给客户端的主要名称（Hostname），例如，www 可以更改为其实际的主要名称，若系统并未有注册的 DNS 名称，则可改输入其 IP 位址，若伺服器需要倾听其他埠号，可更改 80 为其他埠号。
- DocumentRoot “/usr/local/www/apache2x/data” 提供文件目录，预设所有的请求均会到此目录，但可以使用符号链接与别名来指向其他地方。

在对 Apache 设定档做变更之前，建议先做备份，在 Apache 设定完成之后，储存案档并使用 apachectl 检验设定，执行 apachectl configtest 的结果应回传 Syntax OK。

要在系统启动时执行 Apache，可加入下行到 /etc/rc.conf：

```
apache24_enable="YES"
```
若 Apache 要使用非预设的选项启动，可加入下行到 /etc/rc.conf 来指定所需的标签参数：

```
apache24_flags=""
```
若 apachectl 未回报设定错，則可启动 httpd：

```
# service apache24 start
```
httpd 服务可以透过在网页浏览器中输入 http://localhost 来测试，将 localhost 更改为执行 httpd 那台主机的完整域名称（Fully-qualified domain name）。预设会显示的网页为 /usr/local/www/apache24/data/index.html。后续若有在 httpd 执行中时修改 Apache 设定档，可使用以下指令来测试是否有关误：

```
# service apache24 configtest
```
注意，configtest並非採用rc(8)標準，不應預期其可在所有的啟動Script中正常運作。

29.8.2. 虛擬主機

虛擬主機允許在一個Apache伺服器執行多個網站，虛擬主機可以是以IP為主(IP-based)或以名稱為主(name-based)。以IP為主的虛擬主機中的每一個網站要使用不同的IP位址。以名稱為主的虛擬主機會使用客戶端的HTTP/1.1標頭來判斷主機名稱，這可讓不同的網站共用相同的IP位址。

要設定Apache使用以名稱為主的虛擬主機可在每一個網站加入VirtualHost區塊，例如，有一個名稱為www.domain.tld的主機擁有一個www.someotherdomain.tld的虛擬域，可加入以下項目到httpd.conf:

```
<VirtualHost *>
  ServerName www.domain.tld
  DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
  ServerName www.someotherdomain.tld
  DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

每一個虛擬主機均需更改其ServerName與DocumentRoot的值為實際要使用的值。

更多有關設定虛擬主機的資訊，可參考Apache官方說明文件於:

http://httpd.apache.org/docs/vhosts/。

29.8.3. Apache模組

Apache使用模組(Module)來擴充伺服器所提供的功能。請參考http://httpd.apache.org/docs/current/mod/取得可用模組的完整清單與設定詳細資訊。

在FreeBSD中有些模組可以隨著www/apache24 Port編譯，只要在/usr/ports/www/apache24輸入make config便可查看有那一些模組是預設開啟的，若模組未與Port一併編譯，FreeBSD Port套件集也提供了一個簡單的方式可安裝各種模組，本節將介紹最常使用的三個模組。

29.8.3.1. mod_ssl

mod_ssl模組利用了OpenSSL通過Secure Sockets Layer (SSLv3)與Transport Layer Security (TLSv1)通訊協定來提供強大的加密，這個模組提供了向受信認的憑證簽署機構申請簽章憑証所需的任何東西，讓FreeBSD上能夠執行安全的網頁伺服器。

在FreeBSD中mod_ssl模組預設在套件與Port均是開啟的，可用的設定項目在http://httpd.apache.org/docs/current/mod/mod_ssl.html會說明。

29.8.3.2. mod_perl

mod_perl模組讓您可以使用Perl撰寫Apache模組，除此之外，嵌入到伺服器的直譯器可避免啟動外部直譯器的額外開銷與Perl耗費的啟動時間。

mod_perl可以使用www/mod_perl2套件或Port安裝，有關使用此模組的說明文件可在http://perl.apache.org/docs/2.0/index.html中找到。
29.8.3.3.

mod_php

PHP: Hypertext Preprocessor (PHP) 是一般用途的脚本语言，特别适用于网站开发，能够嵌入在 HTML 当中。它的语法参考自 C, Java™ 及 Perl，目的在让网页开发人员能够快速的写出动态网页。

要在 Apache 网页伺服器上加入对 PHP5 的支援，可安装 www/mod_php56 套件或 Port，这会安装并设定支援动态 PHP 应用程式所需的模组。安装过程会自动加入下行到 /usr/local/etc/apache24/httpd.conf:

LoadModule php5_module libexec/apache24/libphp5.so

接着，执行 graceful 重新启动来载入 PHP 模组:

# apachectl graceful

由 www/mod_php56 所提供的 PHP 支援是有限的，若需要额外的支援可以使用 lang/php56-extensions Port 来安装，该 Port 提供了选单介面来选择可用的 PHP 扩充套件。

或者，可以找到适当的 Port 来安装其他的扩充套件，例如，要增加 PHP 对 MySQL 资料库服务器的支援可安装 databases/php56-mysql。

在安装完扩充套件之后，必须重新载入 Apache 伺服器来使用新的设定值:

# apachectl graceful

29.8.4.

动网站除除了 mod_perl 与 mod_php 外，也有其他语言可用来建立动态网页内容，这包含 Django 与 Ruby on Rails。

29.8.4.1. Django

Django 是以 BSD 授权的框架 (Framework)，指在让开发人员能快速的写出高效、优雅的网页应用程式。它提供了物件关连对应器 (Object-relational mapper)，所以各种资料型态可当作 Python 的物件来开发，且提供了丰富的动态资料库存取 API 给这些物件，让开发人员不需再写 SQL。

它也同时提供了可扩充的样板系统，来让应用程式的逻辑与 HTML 呈现能够被拆开。

Django 需要 mod_python，以及一个 SQL 资料库引擎才能运作。在 FreeBSD 中的 www/py-django Port 会自动安装 mod_python 以及对 PostgreSQL, MySQL 或 SQLite 资料库的支援，预设为 SQLite，要更改资料库引擎可在 /usr/ports/www/py-django 输入 make config 然后再安装该 Port。

Django 安装完成之后，应用程式会需要一个专案目録并搭配 Apache 能使用内嵌的 Python 直译器，此直译器会用来呼叫网站上指定 URL 的应用程式。

要设定 Apache 递求到网站应用程式，请加入下行到 httpd.conf 来指定专案目録的完整路径:

<Location “/”>
  SetHandler python-program
  PythonPath “[/dir/to/the/django/packages/] + sys.path”
  PythonHandler django.core.handlers.modpython
  SetEnv DJANGO_SETTINGS_MODULE mysite.settings
</Location>

PythonAutoReload On
PythonDebug On

請參考
https://docs.djangoproject.com
來取得如何使用Django的更多信息。

29.8.4.2. Ruby on Rails

Ruby on Rails是另外一套開放源碼的網站框架(Framework)，提供了完整的開發堆疊，這使得網頁開發人員可以更有生產力且能夠快速的寫出強大的應用程式，在FreeBSD它可以使用www/rubygem-rails套件或Port安裝。

請參考
http://guides.rubyonrails.org
來取得更多信息。

29.9. 檔案傳輸協定(FTP)

檔案傳輸協定(File Transfer Protocol, FTP)提供了使用一個簡單的方式能夠將檔案傳輸到與接收自FTP伺服器,FreeBSD內建了FTP伺服器軟體ftpd在基礎系統(Base system)中。

FreeBSD提供了多個設定檔來控制對FTP伺服器的存取,本節將摘要這些檔案的設定方式,請參考ftpd(8)來取得更多信息。

29.9.1. 設定

最重要的設定步驟便是決定那些帳號能夠存取FTP伺服器,FreeBSD系統有數個系統帳號,這些帳號不應該擁有FTP存取權,不允許存取FTP的使用者清单可在/etc/ftpusers找到,預設該檔案內會有所有的系統帳號,其他不應允許存取FTP的使用者也可在此加入。

在某些情況可能會希望限制某些使用者的存取,而不是完全避免這些使用者使用FTP,這可以透過建立/etc/ftpchroot來完成,詳如ftpchroot(5)所述,這個檔案會列出受到FTP存取限制的使用者與群組。

要在伺服器上啟用匿名FTP存取權,可在FreeBSD系統上建立一個名稱為ftp使用者,使用者將能夠使用ftp或anonymous使用者名稱來登入FTP伺服器,當提示輸入密碼時,輸入任何值都會被接受,但慣例上應使用電子郵件位址當做密碼。當匿名使用者登入時FTP伺服器會呼叫chroot(2)來限制使用者只能存取ftp使用者的家目錄。

要設定顯示給FTP客戶端的歡迎信息有兩個文字檔案可以建立,./etc/ftpwelcome的內容會在收到登入提示前顯示給使用者看,登入成功後,則會顯示./etc/ftpmotd的內容。注意,這個檔案的路徑是相對於登入環境的,所以~ftp/etc/ftpmotd的內容只會對匿名使用者顯示。

設定完FTP伺服器之後,在/etc/rc.conf設定適當的變數在開機時啟動該服務:

```
ftpd_enable="YES"
```

要立即啟動服務可:

```
# service ftpd start
```

要測試到FTP伺服器的連線可輸入:

```
% ftp localhost
```

ftpd daemon會使用syslog(3)來記錄訊息,預設,系統記錄Daemon會寫入有關FTP的訊息到625
FTP 日志的位置可以通过更改 /etc/syslog.conf 来修改:

```plaintext
ftp.info      /var/log/xferlog
```

要注意启动匿名 FTP 服务器可能的潜在问题，尤其是在匿名使用者上传文件时需要再次确认，因为这可能会使该 FTP 服务器变成用来交换未经授权软件的交流平台或者更糟的情况。若真的需要匿名 FTP 上传，请检查权限设定，让这些档案在未被管理者审查前不能被其他匿名使用者读取。

29.10. Microsoft™Windows™ 用户端档案与列印服务 (Samba)

Samba 是热门的开源软件套件，使用 SMB/CIFS 通讯协议提供档案与列印服务，此通讯协议内建于 Microsoft™ Windows™ 系统，在非 Microsoft™ Windows™ 的系统可通过安装 Samba 客户端程式库来支援此协定。此通讯协定让客户端可以存取共享的资料与印表机，这些共享的资源可挂载到一个本机的磁碟机，而共享的印表机则可以当本机的印表机使用。

在 FreeBSD 上，可以使用 net/samba48 Port 或套件来安装 Samba 客户端程式库，这个客户端提供了让 FreeBSD 系统能存取 SMB/CIFS 在 Microsoft™ Windows™ 网路上共享的资源。

FreeBSD 系统也可以透过安装 net/samba48 Port 或套件来设定成 Samba 服务器，这让管理者可以在 FreeBSD 系统上建立 SMB/CIFS 的共享资源，让执行 Microsoft™ Windows™ 或 Samba 客户端程式库的客户端能够存取。

29.10.1. 伺服器设定

Samba 的设定位在 /usr/local/etc/smb4.conf，必须先设定这个档案才可使用 Samba。

要共享目录与印表机给在工作群组中的 Windows™ 客户端的简易 smb4.conf 简例如下。对于涉及 LDAP 或 Active Directory 的复杂安装，可使用 `samba-tool(8)` 来建立初始的 smb4.conf。

```
[global]
workgroup = WORKGROUP
server string = Samba Server Version %v
netbios name = ExampleMachine
wins support = Yes
security = user
passdb backend = tdbsam
# Example: share /usr/src accessible only to 'developer' user
[src]
path = /usr/src
valid users = developer
writable = yes
browsable = yes
read only = no
guest ok = no
public = no
```
create mask = 0666
directory mask = 0755

29.10.1.1.
全域設定
在/usr/local/etc/smb4.conf中加入用来描述网络环境的设定有:

workgroup
要提供的工作群组名称。

netbios name
Samba服务器已知的NetBIOS名称,预设为主机的DNS名称。

server string
会显示于net view输出结果及其他会寻找服务器描述文字并显示的网络工具的文字。

wins support
不不论Samba是否要作为WINS服务器,请不要在网络路上开启超越一台服务器的WINS功能。

29.10.1.2.
安全设定
在/usr/local/etc/smb4.conf中最重要的设定便是安全性模式及后端密码格式,以下项目管控的选项有:

security
最常见的设定为security = share及security = user,若客户端使用的使用者名称与在FreeBSD主机上使用的使用者名称相同,则应该使用使用者(user)层级的安全性,这是预设的安全性原则且它会要求客户端在存取共享资源前先登入。

安全性为共享(share)层级时,客户端存取共享资源不需要先使用有效的使用者名称与密码登入服务器,是在旧版Samba所采用的预设安全性模式。

passdb backend
Samba支援数种不同的后端认证模式,客户端可以使用LDAP, NIS+, SQL资料库或修改过的密码档来认证,建议的认证方式是tdbsam,适用於简易的网络环境且在此说明,对于较大或更复杂的网络则较建议使用ldapsam,而smbpasswd是旧版的预设值,现在已废弃不使用。

29.10.2. Samba使用者
FreeBSD使用者帐号必须对应SambaSAMAccount资料库,才能让Windows™客户端存取共享资源,要对应既有的FreeBSD使用者帐号可使用pdbedit(8):

# pdbedit -a username

本节只提到一些最常用的设定,请参考官方Samba HOWTO来取得有关可用设定选项的额外资讯。
# service samba_server start

Performing sanity check on Samba configuration: OK

Starting nmbd.
Starting smbd.

Samba由三个独立的Daemon所组成，nmbd与smbd daemon可通过samba_enable来启动，若同时也需要winbind名称解析服务则需额外设定：

```
winbindd_enable="YES"
```

Samba可以随时停止，要停止可输入:

```
# service samba_server stop
```

Samba是一套拥有整合Microsoft™ Windows™网络功能的复杂软件套件，除了在此处说明的基础设定，要取得更多功能资讯，请参考 http://www.samba.org。

29.11. NTP

随着时间使用，电脑的时钟会渐渐偏移，这对于需要网络上电脑有相同准确度时间的许多网络服务来说是一个大问题。准确的时间同样能确保文件时间戳记的一致性。网络时间协定（Network Time Protocol, NTP）是在网络上可以确保时间准确的方式。

FreeBSD内含ntpd(8)可设定来查询其他NTP服务器来同步电脑的时间或提供时间服务给其他在网络上的电脑。

本节将会介绍如何设定FreeBSD上的ntpd，更进一步的说明文件可於/usr/shared/doc/ntp/找到HTML格式的版本。

29.11.1. NTP设定在FreeBSD，内建的ntpd可用来自同步系统的时钟，Ntpd要使用rc.conf(5)中的变数及下一节会详细说明的/etc/ntp.conf来设定。Ntpd与网络中各点的通讯采用UDP封包，在伺服器与NTP各点间的防火墙必须设定成可允许进/出埠123的UDP封包。

29.11.1.1. /etc/ntp.conf档Ntpd会读取/etc/ntp.conf来得知要从那些NTP服务器查询时，建议可设定多个NTP服务器，来避免万一其中有一个服务器无法连线或是时间不可靠的问题，当ntpd收到回应，它会偏好先采用较可信赖的服务器。查询的服务器可以是来自本地网络的ISP所提供，也可从线上可公开存取的NTP服务器清单中挑选，您可以选择一个离您地理位置较近的服务器并阅读它的使用规则。也有可公开存取的NTP池线上清单可用，由一个地理区域所组织，除此之外FreeBSD提供了赞助的服务器池，0.freebsd.pool.ntp.org。

例49.
/etc/ntp.conf范例这分简单的ntp.conf范例档可以放心的使用，其中包含建议的restrict选项可避免服务器被公开存取。
Disallow ntpq control/query access. Allow peers to be added only based on pool and server statements in this file.

restrict default limited kod nomodify notrap noquery nopeer
restrict source limited kod nomodify notrap noquery

# Allow unrestricted access from localhost for queries and control.
restrict 127.0.0.1
restrict ::1

# Add a specific server.
server ntplocal.example.com iburst

# Add FreeBSD pool servers until 3-6 good servers are available.
tos minclock 3 maxclock 6
pool 0.freebsd.pool.ntp.org iburst

# Use a local leap-seconds file.
leapfile "/var/db/ntpd.leap-seconds.list"

這個檔案的格式在 ntp.conf(5) 有詳細說明，以下的說明僅快速帶過以上範例檔有用到的一些關鍵字。

預設 NTP 伺服器是可以被任何網路主機所存取，restrict 關鍵字可以控制有哪些系統可以存取伺服器。restrict 支援設定多項，每一項可再進一步調整前面所做設定。範例中的設定授權本地系統有完整的查詢及控制權限，而遠端系統只有查詢時間的權限。要了理解詳細的資訊請參考 ntp.conf(5) 中的 Access Control 支援一節。

server 關鍵字可指定要查詢的伺服器，設定檔中可以使用多個 server 關鍵字，一個伺服器列一行。

pool 關鍵字可指定伺服器池，Ntpd 會加入該伺服器池中的一或多台伺服器，直到數量滿足 tos minclock 的設定。

iburst 關鍵字會指示 ntpd 在建立連線時執行 8 连發快速封包交換，可以更快的同步系統時間。

leapfile 關鍵字用來指定含有閏秒 (Leap second) 資訊的檔案位置，該檔案是由 periodic(8) 自動更新。這個關鍵字指定的檔案位置必須與 /etc/rc.conf 中設定的 ntp_db_leapfile 相同。

29.11.1.2 在 /etc/rc.conf 中的 NTP 設定項目設定 ntpd_enable="YES" 可讓開機時會啟動 ntpd。將 ntpd_enable=YES 加到 /etc/rc.conf 之後，可輸入以下指令讓 ntpd 不需重新開機立即使啟動:

# service ntpd start

要使用 ntpd 必須設定 ntpd_enable，以下所列的 rc.conf 變數可視所需請況設定。

設定 ntpd_sync_on_start=YES 可讓 ntpd 可以在系統啟動時一次同步任何差距的時間，正常情況若時鐘的差距超過 1000 秒便會記錄錯誤並且中止。這個設定項目在沒有電池備援的時鐘上特別有用。

設定 ntpd_oomprotect=YES 可保護 ntpd daemon 被系統中止並嘗試從記憶體不夠 (Out Of Memory, OOM) 的情況恢復運作。
設定

ntpd_config=
可更改ntp.conf檔的位置。

ntpd_flags=
可設定使用任其他所需ntpd參數，但要避免使用由/etc/rc.d/ntpd內部管理的參數如下：

• -p (pid檔案位置)
• -c (改用ntpd_config=設定)

29.11.1.3.
使用無特權的ntpd使用者執行Ntpd在FreeBSD上Ntpd現在可以使用無特權的使用者啟動並執行，要達到這個功能需要mac_ntpd(4)規則模組。

/etc/rc.d/ntpd啟動Script會先檢查NTP的設定，若可以的話它會載入mac_ntpd模組，然後以無特權的使用者ntpd(user id 123)來啟動ntpd。

為了解決檔案與目錄存取權限的問題，當設定中有任何檔案相關的選項時，啟動Script不會自動以ntpd身分啟動ntpd。

在ntpd_flags若出現以下任何參數則需要以最下面的方式手動設定才能以ntpd使用者的身分執行：

• -f 或 --driftfile
• -i 或 --jaildir
• -k 或 --keyfile
• -l 或 --logfile
• -s 或 --statsdir

在ntp.conf若出現以下任何關鍵字則需要以最下面的方式手動設定才能以ntpd使用者的身分執行：

• crypto
• driftfile
• key
• logdir
• statsdir

要手動設定以使用者ntpd身分執行ntpd你必須：

• 確保ntpd使用者有權限存取所有在設定檔中指定的檔案與目錄。
• 讓mac_ntpd模組載入或編譯至核心，請參考mac_ntpd(4)取得詳細資訊。
• 在/etc/rc.conf中設定ntpd_user="ntpd"

29.11.2.
在PPP連線使用NTPntpd並不需要永久的際際網路連線才能正常運作，若有一個PPP連線是設定成需要時拨號，那麼便需要避免NTP的流量觸發撥號或是保持連線不中斷，這可在/etc/ppp/ppp.conf使用filter項目設定，例如：

set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
要取得更詳細的資訊，請參考於ppp(8)的PACKET FILTERING小節以及在/usr/shared/examples/ppp/中的範例。

部份網路存取提供商會封鎖較小編號的埠，這會讓NTP無法運作，因為回應永遠無法送到該主機。

29.12. iSCSI Initiator

与Target设定

iSCSI is a way to share storage over a network. Unlike NFS, which works at the file system level, iSCSI works at the block device level.

In iSCSI terminology, the system that shares the storage is known as the **target**. The storage can be a physical disk, or an area representing multiple disks or a portion of a physical disk. For example, if the disk(s) are formatted with ZFS, a zvol can be created to use as the iSCSI storage.

The clients which access the iSCSI storage are called **initiators**. To initiators, the storage available through iSCSI appears as a raw, unformatted disk known as a LUN. Device nodes for the disk appear in /dev/ and the device must be separately formatted and mounted.

FreeBSD provides a native, kernel-based iSCSI target and initiator. This section describes how to configure a FreeBSD system as a target or an initiator.

29.12.1. 設定iSCSI Target

To configure an iSCSI target, create the /etc/ctl.conf configuration file, add a line to /etc/rc.conf to make sure the ctld(8) daemon is automatically started at boot, and then start the daemon.

The following is an example of a simple /etc/ctl.conf configuration file. Refer to ctl.conf(5) for a more complete description of this file's available options.

```plaintext
portal-group pg0 {
  discovery-auth-group no-authentication
  listen 0.0.0.0
  listen [::]
}
target iqn.2012-06.com.example:target0 {
  auth-group no-authentication
  portal-group pg0
  lun 0 {
    path /data/target0-0
    size 4G
  }
}
```

The first entry defines the pg0 portal group. Portal groups define which network addresses the ctld(8) daemon will listen on. The discovery-auth-group no-authentication entry indicates that any initiator is allowed to perform iSCSI target discovery without authentication. Lines three and four configure ctld(8) to listen on all IPv4 (listen 0.0.0.0) and IPv6 (listen [::]) addresses on the default port of 3260.
It is not necessary to define a portal group as there is a built-in portal group called default. In this case, the difference between default and pg0 is that with default, target discovery is always denied, while with pg0, it is always allowed.

The second entry defines a single target. Target has two possible meanings: a machine serving iSCSI or a named group of LUNs. This example uses the latter meaning, where iqn.2012-06.com.example:target0 is the target name. This target name is suitable for testing purposes. For actual use, change com.example to the real domain name, reversed. The 2012-06 represents the year and month of acquiring control of that domain name, and target0 can be any value. Any number of targets can be defined in this configuration file.

The auth-group no-authentication line allows all initiators to connect to the specified target and portal-group pg0 makes the target reachable through the pg0 portal group.

The next section defines the LUN. To the initiator, each LUN will be visible as a separate disk device. Multiple LUNs can be defined for each target. Each LUN is identified by a number, where LUN 0 is mandatory. The path /data/target0-0 line defines the full path to a file or zvol backing the LUN. That path must exist before starting ctld(8). The second line is optional and specifies the size of the LUN.

Next, to make sure the ctld(8) daemon is started at boot, add this line to /etc/rc.conf:

```
ctld_enable="YES"
```

To start ctld(8) now, run this command:
```
# service ctld start
```

As the ctld(8) daemon is started, it reads /etc/ctl.conf. If this file is edited after the daemon starts, use this command so that the changes take effect immediately:
```
# service ctld reload
```

The previous example is inherently insecure as it uses no authentication, granting anyone full access to all targets. To require a username and password to access targets, modify the configuration as follows:

```
auth-group ag0 {
    chap username1 secretsecret
    chap username2 anothersecret
}
```

```
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}
```

```
target iqn.2012-06.com.example:target0 {
}
```

29.12.1.1.
The auth-group section defines username and password pairs. An initiator trying to connect to iqn.2012-06.com.example:target0 must first specify a defined username and secret. However, target discovery is still permitted without authentication. To require target discovery authentication, set discovery-auth-group to a defined auth-group name instead of no-authentication.

It is common to define a single exported target for every initiator. As a shorthand for the syntax above, the username and password can be specified directly in the target entry:

```
target iqn.2012-06.com.example:target0 {
  portal-group pg0
  chap username1 secretsecret
  lun 0 {
    path /data/target0-0
    size 4G
  }
}
```

The iSCSI initiator described in this section is supported starting with FreeBSD 10.0-RELEASE. To use the iSCSI initiator available in older versions, refer to iscontrol(8).

The iSCSI initiator requires that the iscsid(8) daemon is running. This daemon does not use a configuration file. To start it automatically at boot, add this line to /etc/rc.conf:

```
iscsid_enable="YES"
```

To start iscsid(8) now, run this command:

```
# service iscsid start
```

Connecting to a target can be done with or without an /etc/iscsi.conf configuration file. This section demonstrates both types of connections.

29.12.2.1. 不使用設定檔連線到Target

To connect an initiator to a single target, specify the IP address of the portal and the name of the
To verify if the connection succeeded, run `iscsictl` without any arguments. The output should look similar to this:

<table>
<thead>
<tr>
<th>Target name</th>
<th>Target portal</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>iqn.2012-06.com.example:target0</td>
<td>10.10.10.10</td>
<td>Connected: da0</td>
</tr>
</tbody>
</table>

In this example, the iSCSI session was successfully established, with `/dev/da0` representing the attached LUN. If the `iqn.2012-06.com.example:target0` target exports more than one LUN, multiple device nodes will be shown in that section of the output: Connected: da0 da1 da2.

Any errors will be reported in the output, as well as the system logs. For example, this message usually means that the `iscsid(8)` daemon is not running:

Target name | Target portal | State           |
-------------|---------------|-----------------|
| iqn.2012-06.com.example:target0 | 10.10.10.10 | Waiting for iscsid(8) |

The following message suggests a networking problem, such as a wrong IP address or port:

Target name | Target portal | State              |
-------------|---------------|--------------------|
| iqn.2012-06.com.example:target0 | 10.10.10.11 | Connection refused |

This message means that the specified target name is wrong:

Target name | Target portal | State   |
-------------|---------------|---------|
| iqn.2012-06.com.example:target0 | 10.10.10.10 | Not found |

This message means that the target requires authentication:

Target name | Target portal | State      |
-------------|---------------|------------|
| iqn.2012-06.com.example:target0 | 10.10.10.10 | Authentication failed |

To specify a CHAP username and secret, use this syntax:

```
# iscsictl -A -p 10.10.10.10 -t iqn.2012-06.com.example:target0 -u user -s secretsecret
```

To connect using a configuration file, create `/etc/iscsi.conf` with contents like this:
The `t0` specifies a nickname for the configuration file section. It will be used by the initiator to specify which configuration to use. The other lines specify the parameters to use during connection. The `TargetAddress` and `TargetName` are mandatory, whereas the other options are optional. In this example, the CHAP username and secret are shown.

To connect to the defined target, specify the nickname:

```
# iscsictl -An t0
```

Alternately, to connect to all targets defined in the configuration file, use:

```
# iscsictl -Aa
```

To make the initiator automatically connect to all targets in `/etc/iscsi.conf`, add the following to `/etc/rc.conf`:

```
iscsictl_enable="YES"
iscsictl_flags="-Aa"
```
有被看這個有查詢, 可參考的詢查許務。埠定服做數均篩傳的送可用立規選標位上述過條址埠以定的送往封其通服埠指定訊來源位聽定協在自網處網TCP/IP的包,由回來包內的資料斷通封, 主的線階段機封或判封包則集30.2.概念開始在閱讀這章之前, 您需要了讀完這章, 您求評估與各種使用者的滿了為是滿偏好最能使用者FreeBSD管包, 尽運作且有不同的封FreeBSD PF自與般配dummynet 也會使用合配兩也提供了寬流量量: 頻限與建即: [30.1.述 Chapter 30. 防火牆 FreeBSD 使用與] IPFILTER 如防火牆IPFW 設定使用與防火牆。建防火牆FreeBSD 的過則封。包。檢能的透過系統內 送 (Incoming) 概30.2. 透記 (Rules) 連允 (Network packets) 阻擋或的網封來。檢能能的透過系統內送(Intcoming)概30.2.透記(Rules)連允(Network packets)阻擋或的網封來。
According to the network address, NAT preserves the network translation of one IP address to another IP address during the movement of packets. If new packets are received and a stage is reached, the line stage is allowed to pass, and the complete packet will be transferred in the dynamic state table. For example, NAT will automatically generate a rule in the middle of the line stage, which can further strengthen the tracking of packets or the security condition.

The state firewall, because it does not use the destination address to determine the routing, can simultaneously track the external system traffic and choose other paths. Some examples in this chapter also show how to get FTP-style references in the manner.

192.168.0.0/16 is far from the Console, do not set the firewall administrator in the server as well, otherwise it will set the firewall settings inside. This also explains why there is a method of exclusive in this way.
This section of the Handbook focuses on PF as it pertains to FreeBSD. It demonstrates how to enable PF and ALTQ. It also provides several examples for creating rulesets on a FreeBSD system.

30.3.1. 开啟 PF

To use PF, its kernel module must be first loaded. This section describes the entries that can be added to /etc/rc.conf to enable PF.

Start by adding `pf_enable=yes` to /etc/rc.conf:

```
# sysrc pf_enable=yes
```

Additional options, described in `pfctl(8)`, can be passed to PF when it is started. Add or change this entry in /etc/rc.conf and specify any required flags between the two quotes (`"`):

```
pf_flags=""                     # additional flags for pfctl startup
```

PF will not start if it cannot find its ruleset configuration file. By default, FreeBSD does not ship with a ruleset and there is no `/etc/pf.conf`. Example rulesets can be found in `/usr/shared/examples/pf/`. If a custom ruleset has been saved somewhere else, add a line to /etc/rc.conf which specifies the full path to the file:

```
pf_rules="/path/to/pf.conf"
```

Logging support for PF is provided by `pflog(4)`. To enable logging support, add `pflog_enable=yes` to /etc/rc.conf:

```
# sysrc pflog_enable=yes
```

The following lines can also be added to change the default location of the log file or to specify any additional flags to pass to `pflog(4)` when it is started:

```
pflog_logfile="/var/log/pflog"  # where pflogd should store the logfile
pflog_flags=""                  # additional flags for pflogd startup
```

Finally, if there is a LAN behind the firewall and packets need to be forwarded for the computers on the LAN, or NAT is required, enable the following option:

```
gateway_enable="YES"            # Enable as LAN gateway
```

After saving the needed edits, PF can be started with logging support by typing:

```
# service pf start
```
By default, PF reads its configuration rules from `/etc/pf.conf` and modifies, drops, or passes packets according to the rules or definitions specified in this file. The FreeBSD installation includes several sample files located in `/usr/shared/examples/pf/`. Refer to the PF FAQ for complete coverage of PF rulesets.

To control PF, use `pfctl`. Useful `pfctl` options summarize some useful options to this command. Refer to `pfctl(8)` for a description of all available options:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pfctl -e</code></td>
<td>Enable PF.</td>
</tr>
<tr>
<td><code>pfctl -d</code></td>
<td>Disable PF.</td>
</tr>
<tr>
<td><code>pfctl -F all -f /etc/pf.conf</code></td>
<td>Flush all NAT, filter, state, and table rules and reload <code>/etc/pf.conf</code>.</td>
</tr>
<tr>
<td><code>pfctl -vnf /etc/pf.conf</code></td>
<td>Check <code>/etc/pf.conf</code> for errors, but do not load ruleset.</td>
</tr>
</tbody>
</table>

Security/sudo is useful for running commands like `pfctl` that require elevated privileges. It can be installed from the Ports Collection.

To keep an eye on the traffic that passes through the PF firewall, consider installing the `sysutils/pftop` package or port. Once installed, `pftop` can be run to view a running snapshot of traffic in a format which is similar to `top(1)`.

30.3.2. PF ruleset

This section demonstrates how to create a customized ruleset. It starts with the simplest of rulesets and builds upon its concepts using several examples to demonstrate real-world usage of PF's many features.

The simplest possible ruleset is for a single machine that does not run any services and which needs access to one network, which may be the Internet. To create this minimal ruleset, edit `/etc/pf.conf` so it looks like this:

```
block in all
pass out all
keep state
```

The first rule denies all incoming traffic by default. The second rule allows connections created by this system to pass out, while retaining state information on those connections. This state information allows return traffic for those connections to pass back and should only be used on machines that can be trusted. The ruleset can be loaded with:

```
# pfctl -e ; pfctl -f /etc/pf.conf
```

In addition to keeping state, PF provides lists and macros which can be defined for use when creating rules. Macros can include lists and need to be defined before use. As an example, insert these lines at the very top of the ruleset:
TCP services = \{ ssh, smtp, domain, www, pop3, auth, pop3s \}

UDP services = \{ domain \}

PF understands port names as well as port numbers, as long as the names are listed in /etc/services. This example creates two macros. The first is a list of seven TCP port names and the second is one UDP port name. Once defined, macros can be used in rules. In this example, all traffic is blocked except for the connections initiated by this system for the seven specified TCP services and the one specified UDP service:

```
tcp_services = \{ ssh, smtp, domain, www, pop3, auth, pop3s \}
udp_services = \{ domain \}
block all
pass out proto tcp to any port $tcp_services keep state
pass proto udp to any port $udp_services keep state
```

Even though UDP is considered to be a stateless protocol, PF is able to track some state information. For example, when a UDP request is passed which asks a name server about a domain name, PF will watch for the response to pass it back.

Whenever an edit is made to a ruleset, the new rules must be loaded so they can be used:
```
# pfctl -f /etc/pf.conf
```
If there are no syntax errors, pfctl will not output any messages during the rule load. Rules can also be tested before attempting to load them:
```
# pfctl -nf /etc/pf.conf
```
Including -n causes the rules to be interpreted only, but not loaded. This provides an opportunity to correct any errors. At all times, the last valid ruleset loaded will be enforced until either PF is disabled or a new ruleset is loaded.

Adding -v to a pfctl ruleset verify or load will display the fully parsed rules exactly the way they will be loaded. This is extremely useful when debugging rules.

30.3.2.1. 使用 NAT 的简单通讯

This section demonstrates how to configure a FreeBSD system running PF to act as a gateway for at least one other machine. The gateway needs at least two network interfaces, each connected to a separate network. In this example, xl1 is connected to the Internet and xl0 is connected to the internal network.

First, enable the gateway to let the machine forward the network traffic it receives on one interface to another interface. This sysctl setting will forward IPv4 packets:
```
# sysctl net.inet.ip.forwarding=1
```
To forward IPv6 traffic, use:
```
640
```
To enable these settings at system boot, use `sysrc(8)` to add them to `/etc/rc.conf`:

```
# sysrc gateway_enable=yes
# sysrc ipv6_gateway_enable=yes
```

Verify with `ifconfig` that both of the interfaces are up and running.

Next, create the PF rules to allow the gateway to pass traffic. While the following rule allows stateful traffic to pass from the Internet to hosts on the network, the `to` keyword does not guarantee passage all the way from source to destination:

```
pass in on xl1 from xl1:network to xl0:network port $ports keep state
```

That rule only lets the traffic pass in to the gateway on the internal interface. To let the packets go further, a matching rule is needed:

```
pass out on xl0 from xl1:network to xl0:network port $ports keep state
```

While these two rules will work, rules this specific are rarely needed. For a busy network admin, a readable ruleset is a safer ruleset. The remainder of this section demonstrates how to keep the rules as simple as possible for readability. For example, those two rules could be replaced with one rule:

```
pass from xl1:network to any port $ports keep state
```

The `interface:network` notation can be replaced with a macro to make the ruleset even more readable. For example, a `$localnet` macro could be defined as the network directly attached to the internal interface (`$xl1:network`). Alternatively, the definition of `$localnet` could be changed to an IP address/netmask notation to denote a network, such as `192.168.100.1/24` for a subnet of private addresses.

If required, `$localnet` could even be defined as a list of networks. Whatever the specific needs, a sensible `$localnet` definition could be used in a typical pass rule as follows:

```
pass from $localnet to any port $ports keep state
```

The following sample ruleset allows all traffic initiated by machines on the internal network. It first defines two macros to represent the external and internal 3COM interfaces of the gateway.

```
ext_if = "xl0"  # macro for external interface - use tun0 for PPPoE
int_if = "xl1"  # macro for internal interface

localnet = $int_if:network
```

For dialup users, the external interface will use `tun0`. For an ADSL connection, specifically those using PPP over Ethernet (PPPoE), the correct external interface is `tun0`, not the physical Ethernet interface.

```
ext_if = "xl0"  # macro for external interface - use tun0 for PPPoE
int_if = "xl1"  # macro for internal interface

localnet = $int_if:network
```

The following sample ruleset allows all traffic initiated by machines on the internal network. It first defines two macros to represent the external and internal 3COM interfaces of the gateway.

```
ext_if = "xl0"  # macro for external interface - use tun0 for PPPoE
int_if = "xl1"  # macro for internal interface

localnet = $int_if:network
```

For dialup users, the external interface will use `tun0`. For an ADSL connection, specifically those using PPP over Ethernet (PPPoE), the correct external interface is `tun0`, not the physical Ethernet interface.
nat on $ext_if from $localnet to any -> ($ext_if)
block all
pass from { lo0, $localnet } to any keep state

This ruleset introduces the nat rule which is used to handle the network address translation from the non-routable addresses inside the internal network to the IP address assigned to the external interface. The parentheses surrounding the last part of the nat rule ($ext_if) is included when the IP address of the external interface is dynamically assigned. It ensures that network traffic runs without serious interruptions even if the external IP address changes.

Note that this ruleset probably allows more traffic to pass out of the network than is needed. One reasonable setup could create this macro:

```plaintext
client_out = "{ ftp-data, ftp, ssh, domain, pop3, auth, nntp, http, https, cvspserver, 2628, 5999, 8000, 8080 }"
```

to use in the main pass rule:

```plaintext
pass inet proto tcp from $localnet to any port $client_out 
flags S/SA keep state
```

A few other pass rules may be needed. This one enables SSH on the external interface:

```plaintext
pass in inet proto tcp to $ext_if port ssh
```

This macro definition and rule allows DNS and NTP for internal clients:

```plaintext
udp_services = "{ domain, ntp }"
```

pass quick inet proto { tcp, udp } to any port $udp_services keep state

Note the quick keyword in this rule. Since the ruleset consists of several rules, it is important to understand the relationships between the rules in a ruleset. Rules are evaluated from top to bottom, in the sequence they are written. For each packet or connection evaluated by PF, the last matching rule in the ruleset is the one which is applied. However, when a packet matches a rule which contains the quick keyword, the rule processing stops and the packet is treated according to that rule. This is very useful when an exception to the general rules is needed.

30.3.2.2. 建立 FTP Proxy

Configuring working FTP rules can be problematic due to the nature of the FTP protocol. FTP pre-dates firewalls by several decades and is insecure in its design. The most common points against using FTP include:

• Passwords are transferred in the clear.
• The protocol demands the use of at least two TCP connections (control and data) on separate ports.
• When a session is established, data is communicated using randomly selected ports.

All of these points present security challenges, even before considering any potential security...
More secure alternatives for file transfer exist, such as `sftp(1)` or `scp(1)`, which both feature authentication and data transfer over encrypted connections.

For those situations when FTP is required, PF provides redirection of FTP traffic to a small proxy program called `ftp-proxy(8)`, which is included in the base system of FreeBSD. The role of the proxy is to dynamically insert and delete rules in the ruleset, using a set of anchors, to correctly handle FTP traffic.

To enable the FTP proxy, add this line to `/etc/rc.conf`:

```
ftpproxy_enable="YES"
```

Then start the proxy by running `service ftp-proxy start`.

For a basic configuration, three elements need to be added to `/etc/pf.conf`. First, the anchors which the proxy will use to insert the rules it generates for the FTP sessions:

```
nat-anchor "ftpproxy/*"
rdr-anchor "ftpproxy/*"
```

Second, a pass rule is needed to allow FTP traffic in to the proxy.

Third, redirection and NAT rules need to be defined before the filtering rules. Insert this `rdr` rule immediately after the `nat` rule:

```
rdr pass on $int_if proto tcp from any to any port ftp -> 127.0.0.1 port 8021
```

Finally, allow the redirected traffic to pass:

```
pass out proto tcp from $proxy to any port ftp
```

where `$proxy` expands to the address the proxy daemon is bound to.

Save `/etc/pf.conf`, load the new rules, and verify from a client that FTP connections are working:

```
# pfctl -f /etc/pf.conf
```

This example covers a basic setup where the clients in the local network need to contact FTP servers elsewhere. This basic configuration should work well with most combinations of FTP clients and servers. As shown in `ftp-proxy(8)`, the proxy's behavior can be changed in various ways by adding options to the `ftpproxy_flags=` line. Some clients or servers may have specific quirks that must be compensated for in the configuration, or there may be a need to integrate the proxy in specific ways such as assigning FTP traffic to a specific queue.

For ways to run an FTP server protected by PF and `ftp-proxy(8)`, configure a separate `ftp-proxy` in reverse mode, using `-R`, on a separate port with its own redirecting pass rule.
The ICMP protocol sends and receives control messages between hosts and gateways, mainly to provide feedback to a sender about any unusual or difficult conditions enroute to the target host. Routers use ICMP to negotiate packet sizes and other transmission parameters in a process often referred to as path MTU discovery.

From a firewall perspective, some ICMP control messages are vulnerable to known attack vectors. Also, letting all diagnostic traffic pass unconditionally makes debugging easier, but it also makes it easier for others to extract information about the network. For these reasons, the following rule may not be optimal:

```
pass inet proto icmp from any to any
```

One solution is to let all ICMP traffic from the local network through while stopping all probes from outside the network:

```
pass inet proto icmp from $localnet to any keep state
pass inet proto icmp from any to $ext_if keep state
```

Additional options are available which demonstrate some of PF's flexibility. For example, rather than allowing all ICMP messages, one can specify the messages used by `ping(8)` and `traceroute(8)`.

Start by defining a macro for that type of message:

```
icmp_types = "echoreq"
```

and a rule which uses the macro:

```
pass inet proto icmp all icmp-type $icmp_types keep state
```

If other types of ICMP packets are needed, expand `icmp_types` to a list of those packet types. Type `more /usr/src/sbin/pfctl/pfctl_parser.c` to see the list of ICMP message types supported by PF. Refer to `http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml` for an explanation of each message type.

Since Unix `traceroute` uses UDP by default, another rule is needed to allow Unix `traceroute`:

```
# allow out the default range for traceroute(8):
pass out on $ext_if inet proto udp from any to any port 33433 >< 33626 keep state
```

Since `TRACERT.EXE` on Microsoft Windows systems uses ICMP echo request messages, only the first rule is needed to allow network traces from those systems. Unix `traceroute` can be instructed to use other protocols as well, and will use ICMP echo request messages if `-I` is used. Check the `traceroute(8)` man page for details.

### 30.3.2.3.1. Path MTU Discovery

Internet protocols are designed to be device independent, and one consequence of device independence is that the optimal packet size for a given connection cannot always be predicted reliably. The main constraint on packet size is the Maximum Transmission Unit (MTU) which sets the upper limit on the packet size for an interface. Type `ifconfig` to view the MTUs for a system's network interfaces.

TCP/IP uses a process known as path MTU discovery to determine the right packet size for a connection.
This process sends packets of varying sizes with the "Do not fragment" flag set, expecting an ICMP return packet of "type 3, code 4" when the upper limit has been reached. Type 3 means "destination unreachable", and code 4 is short for "fragmentation needed, but the do-not-fragment flag is set". To allow path MTU discovery in order to support connections to other MTUs, add the `destination unreachable` type to the `icmp_types` macro:

```
icmp_types = "{ echoreq, unreach }"
```

Since the pass rule already uses that macro, it does not need to be modified to support the new ICMP type:

```
pass inet proto icmp all icmp-type $icmp_types keep state
```

PF allows filtering on all variations of ICMP types and codes. The list of possible types and codes are documented in `icmp(4)` and `icmp6(4)`.

30.3.2.4. 使用

Tables

Some types of data are relevant to filtering and redirection at a given time, but their definition is too long to be included in the ruleset file. PF supports the use of tables, which are defined lists that can be manipulated without needing to reload the entire ruleset, and which can provide fast lookups. Table names are always enclosed within `< >`, like this:

```
table <clients> { 192.168.2.0/24, !192.168.2.5 }
```

In this example, the `192.168.2.0/24` network is part of the table, except for the address `192.168.2.5`, which is excluded using the `!` operator. It is also possible to load tables from files where each item is on a separate line, as seen in this example `/etc/clients`:

```
192.168.2.0/24
!192.168.2.5
```

To refer to the file, define the table like this:

```
table <clients> persist file="/etc/clients"
```

Once the table is defined, it can be referenced by a rule:

```
pass inet proto tcp from <clients> to any port $client_out flags S/SA keep state
```

A table's contents can be manipulated live, using `pfctl`. This example adds another network to the table:

```
# pfctl -t clients -T add 192.168.1.0/16
```

Note that any changes made this way will take affect now, making them ideal for testing, but will not survive a power failure or reboot. To make the changes permanent, modify the definition of the table in the ruleset or edit the file that the table refers to. One can maintain the on-disk copy of the 645
table using a cron(8) job which dumps the table's contents to disk at regular intervals, using a command such as `pfctl -t clients -T show >/etc/clients`. Alternatively, `/etc/clients` can be updated with the in-memory table contents:

```
# pfctl -t clients -T replace -f /etc/clients
```

30.3.2.5.

使用 Overload Tables 保護 SSH

Those who run SSH on an external interface have probably seen something like this in the authentication logs:

```
Sep 26 03:12:34 skapet sshd[25771]: Failed password for root from 200.72.41.31 port 40992
ssh2
Sep 26 03:12:34 skapet sshd[5279]: Failed password for root from 200.72.41.31 port 40992
ssh2
Sep 26 03:12:35 skapet sshd[5279]: Received disconnect from 200.72.41.31: 11: Bye Bye
Sep 26 03:12:44 skapet sshd[29635]: Invalid user admin from 200.72.41.31
Sep 26 03:12:44 skapet sshd[24703]: input_userauth_request: invalid user admin
Sep 26 03:12:44 skapet sshd[24703]: Failed password for invalid user admin from 200.72.41.31 port 41484 ssh2
```

This is indicative of a brute force attack where somebody or some program is trying to discover the user name and password which will let them into the system.

If external SSH access is needed for legitimate users, changing the default port used by SSH can offer some protection. However, PF provides a more elegant solution. Pass rules can contain limits on what connecting hosts can do and violators can be banished to a table of addresses which are denied some or all access. It is even possible to drop all existing connections from machines which overreach the limits.

To configure this, create this table in the tables section of the ruleset:

```
table <bruteforce> persist
```

Then, somewhere early in the ruleset, add rules to block brute access while allowing legitimate access:

```
block quick from <bruteforce>
pass inet proto tcp from any to $localnet port $tcp_services 
flags S/SA keep state 
(max-src-conn 100, max-src-conn-rate 15/5, 
overload <bruteforce> flush global)
```

The part in parentheses defines the limits and the numbers should be changed to meet local requirements. It can be read as follows:

- `max-src-conn` is the number of simultaneous connections allowed from one host.
- `max-src-conn-rate` is the rate of new connections allowed from any single host (15) per number of seconds (5).
overload means that any host which exceeds these limits gets its address added to the bruteforce table. The ruleset blocks all traffic from addresses in the bruteforce table.

Finally, flush global says that when a host reaches the limit, that all (global) of that host's connections will be terminated (flush).

These rules will not block slow bruteforcers, as described in http://home.nuug.no/~peter/hailmary2013/.

This example ruleset is intended mainly as an illustration. For example, if a generous number of connections in general are wanted, but the desire is to be more restrictive when it comes to ssh, supplement the rule above with something like the one below, early on in the rule set:

pass quick proto { tcp, udp } from any to any port ssh 
flags S/SA keep state 
(max-src-conn 15, max-src-conn-rate 5/3, overload bruteforce flush global)

It May Not be Necessary to Block All Overloaders

It is worth noting that the overload mechanism is a general technique which does not apply exclusively to SSH, and it is not always optimal to entirely block all traffic from offenders.

For example, an overload rule could be used to protect a mail service or a web service, and the overload table could be used in a rule to assign offenders to a queue with a minimal bandwidth allocation or to redirect to a specific web page.

Over time, tables will be filled by overload rules and their size will grow incrementally, taking up more memory. Sometimes an IP address that is blocked is a dynamically assigned one, which has since been assigned to a host who has a legitimate reason to communicate with hosts in the local network.

For situations like these, pfctl provides the ability to expire table entries. For example, this command will remove<bruteforce> table entries which have not been referenced for 86400 seconds:

# pfctl -t bruteforce -T expire 86400

Similar functionality is provided by security/expiretable, which removes table entries which have not been accessed for a specified period of time.

Once installed, expiretable can be run to remove<bruteforce> table entries older than a specified age. This example removes all entries older than 24 hours:

/usr/local/sbin/expiretable -v -d -t 24h bruteforce

30.3.2.6. SPAM 防護

Not to be confused with the spamd daemon which comes bundled with spamassassin, mail/spamd can be configured with PF to provide an outer defense against SPAM. This spamd hooks into the PF configuration using a set of redirections.

Spammers tend to send a large number of messages, and SPAM is mainly sent from a few spammer friendly networks and a large number of hijacked machines, both of which are reported to 647.
blacklists fairly quickly. When an SMTP connection from an address in a blacklist is received, spamd presents its banner and immediately switches to a mode where it answers SMTP traffic one byte at a time. This technique, which is intended to waste as much time as possible on the spammer's end, is called tarpitting. The specific implementation which uses one byte SMTP replies is often referred to as stuttering.

This example demonstrates the basic procedure for setting up spamd with automatically updated blacklists. Refer to the man pages which are installed with mail/spamd for more information.

Procedure: Configuring spamd

1. Install the mail/spamd package or port. To use spamd's greylisting features, fdescfs(5) must be mounted at /dev/fd. Add the following line to /etc/fstab:

   
   fdescfs /dev/fd fdescfs rw 0 0

   Then, mount the filesystem:

   
   # mount fdescfs

2. Next, edit the PF ruleset to include:

   table <spamd> persist
   table <spamd-white> persist
   rdr pass on $ext_if inet proto tcp from <spamd> to
   { $ext_if, $localnet } port smtp -> 127.0.0.1 port 8025
   rdr pass on $ext_if inet proto tcp from !<spamd-white> to
   { $ext_if, $localnet } port smtp -> 127.0.0.1 port 8025

   The two tables <spamd> and <spamd-white> are essential. SMTP traffic from an address listed in <spamd> but not in <spamd-white> is redirected to the spamd daemon listening at port 8025.

3. The next step is to configure spamd in /usr/local/etc/spamd.conf and to add some rc.conf parameters. The installation of mail/spamd includes a sample configuration file (/usr/local/etc/spamd.conf.sample) and a man page for spamd.conf. Refer to these for additional configuration options beyond those shown in this example.

   One of the first lines in the configuration file that does not begin with a # comment sign contains the block which defines the all list, which specifies the lists to use:

   all:
   :traplist:
   :whitelist:

   This entry adds the desired blacklists, separated by colons (:). To use a whitelist to subtract addresses from a blacklist, add the name of the whitelist immediately after the name of that blacklist. For example:

   :blacklist:whitelist:
This is followed by the specified blacklist's definition:

```
traplist:
  black:
    msg="SPAM. Your address %A has sent spam within the last 24 hours":
    method=http:
    file=www.openbsd.org/spamd/traplist.gz
```

where the first line is the name of the blacklist and the second line specifies the list type.

The `msg` field contains the message to display to blacklisted senders during the SMTP dialogue. The `method` field specifies how spamd-setup fetches the list data; supported methods are `http`, `ftp`, from a `file` in a mounted file system, and via `exec` of an external program. Finally, the `file` field specifies the name of the file spamd expects to receive.

The definition of the specified whitelist is similar, but omits the `msg` field since a message is not needed:

```
whitelist:
  white:
    method=file:
    file=/var/mail/whitelist.txt
```

Choose Data Sources with Care

Using all the blacklists in the sample `spamd.conf` will blacklist large blocks of the Internet. Administrators need to edit the file to create an optimal configuration which uses applicable data sources and, when necessary, uses custom lists.

Next, add this entry to `/etc/rc.conf`. Additional flags are described in the man page specified by the comment:

```
spamd_flags="-v" # use "" and see spamd-setup(8) for flags
```

When finished, reload the ruleset, start spamd by typing `service obspamd start`, and complete the configuration using `spamd-setup`. Finally, create a `cron(8)` job which calls `spamd-setup` to update the tables at reasonable intervals.

On a typical gateway in front of a mail server, hosts will soon start getting trapped within a few seconds to several minutes.

PF also supports greylisting, which temporarily rejects messages from unknown hosts with 45n codes. Messages from greylisted hosts which try again within a reasonable time are let through. Traffic from senders which are set up to behave within the limits set by RFC 1123 and RFC 2821 are immediately let through.

More information about greylisting as a technique can be found at the greylisting.org web site. The most amazing thing about greylisting, apart from its simplicity, is that it still works. Spammers and malware writers have been very slow to adapt to bypass this technique.
Procedure: Configuring Greylisting

1. Make sure that `fdescfs(5)` is mounted as described in Step 1 of the previous Procedure.

2. To run spamd in greylisting mode, add this line to `/etc/rc.conf`:
   ```
   spamd_grey="YES"  # use spamd greylisting if YES
   ```
   Refer to the spamd man page for descriptions of additional related parameters.

3. To complete the greylisting setup:
   ```
   # service obspamd restart
   # service obspamlogd start
   ```

Behind the scenes, the spamdb database tool and the spamlogd whitelist updater perform essential functions for the greylisting feature. spamdb is the administrator's main interface to managing the black, grey, and white lists via the contents of the `/var/db/spamdb` database.

30.3.2.7. This section describes how `block-policy`, `scrub`, and `antispoof` can be used to make the ruleset behave sanely.

The `block-policy` is an option which can be set in the `options` part of the ruleset, which precedes the redirection and filtering rules. This option determines which feedback, if any, PF sends to hosts that are blocked by a rule. The option has two possible values:

- `drop` drops blocked packets with no feedback,
- `return` returns a status code such as `Connection refused`.

If not set, the default policy is `drop`. To change the `block-policy`, specify the desired value:

```
set block-policy return
```

In PF, `scrub` is a keyword which enables network packet normalization. This process reassembles fragmented packets and drops TCP packets that have invalid flag combinations. Enabling `scrub` provides a measure of protection against certain kinds of attacks based on incorrect handling of packet fragments. A number of options are available, but the simplest form is suitable for most configurations:

```
scrub in all
```

Some services, such as NFS, require specific fragment handling options. Refer to [https://home.nuug.no/~peter/pf/en/scrub.html](https://home.nuug.no/~peter/pf/en/scrub.html) for more information.

This example reassembles fragments, clears the "do not fragment" bit, and sets the maximum segment size to 1440 bytes:

```
scrub in all fragment reassemble no-df max-mss 1440
```

The `antispoof` mechanism protects against activity from spoofed or forged IP addresses, mainly by blocking packets appearing on interfaces and in directions which are logically not possible.
These rules weed out spoofed traffic coming in from the rest of the world as well as any spoofed packets which originate in the local network:

antispoof for $ext_if
antispoof for $int_if

30.3.2.8. 处理不可路由（Non-Routable）的地址

Even with a properly configured gateway to handle network address translation, one may have to compensate for other people's misconfigurations. A common misconfiguration is to let traffic with non-routable addresses out to the Internet. Since traffic from non-routeable addresses can play a part in several DoS attack techniques, consider explicitly blocking traffic from non-routeable addresses from entering the network through the external interface.

In this example, a macro containing non-routable addresses is defined, then used in blocking rules. Traffic to and from these addresses is quietly dropped on the gateway's external interface.

martians = "{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8, 169.254.0.0/16, 192.0.2.0/24, 0.0.0.0/8, 240.0.0.0/4 }"

block drop in quick on $ext_if from $martians to any
block drop out quick on $ext_if from any to $martians

30.3.3. 开启ALTQ

On FreeBSD, ALTQ can be used with PF to provide Quality of Service (QOS). Once ALTQ is enabled, queues can be defined in the ruleset which determine the processing priority of outbound packets.

Before enabling ALTQ, refer to altq(4) to determine if the drivers for the network cards installed on the system support it.

ALTQ is not available as a loadable kernel module. If the system's interfaces support ALTQ, create a custom kernel using the instructions in設定FreeBSD核心. The following kernel options are available. The first is needed to enable ALTQ. At least one of the other options is necessary to specify the queueing scheduler algorithm:

<table>
<thead>
<tr>
<th>options</th>
<th>ALTQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>options</td>
<td>ALTQ_CBQ</td>
</tr>
<tr>
<td>options</td>
<td>ALTQ_RED</td>
</tr>
<tr>
<td>options</td>
<td>ALTQ_RIO</td>
</tr>
<tr>
<td>options</td>
<td>ALTQ_HFSC</td>
</tr>
<tr>
<td>options</td>
<td>ALTQ_PRIQ</td>
</tr>
</tbody>
</table>

The following scheduler algorithms are available:

CBQ

Class Based Queuing (CBQ) is used to divide a connection's bandwidth into different classes or queues to prioritize traffic based on filter rules.
30.4. IPFW

IPFW is a stateful firewall, written for FreeBSD, which supports both IPv4 and IPv6. It consists of several components: core firewall processing, stateful firewall rules processor and its integration accounting mechanisms, NAT, dummynet(4) traffic shaping, and iptable(4) audit tools.

FreeBSD provides a template firewall policy set in /etc/rc.firewall, which defines several common firewall types to assist new users in writing suitable policies.

IPFWW provides a powerful language for advanced users to define firewalls to security requirements.

This section will introduce how to enable IPFW, firewall policy overview, and a few common firewall types.

30.4.1. Enabling IPFW

IPFW is included in the basic FreeBSD installation as a kernel loadable module, so a custom kernel is not required to enable IPFW.

For users who wish to statically compile IPFW support into a custom kernel, see IPFW core options.

To configure the system to enable IPFW at boot time, add firewall_enable="YES" to /etc/rc.conf:

```bash
# sysrc firewall_enable="YES"
```

To use one of the default firewall types provided by FreeBSD, add another line which specifies the type:

```bash
# sysrc firewall_type="open"
```

The available types are:

- **open**: passes all traffic.
- **client**: protects only this machine.
- **simple**: protects the whole network.
- **closed**: entirely disables IP traffic except for the loopback interface.
workstation: protects only this machine using stateful rules.

UNKNOWN: disables the loading of firewall rules.

filename: full path of the file containing the firewall ruleset.

If firewall_type is set to either client or simple, modify the default rules found in /etc/rc.firewall to fit the configuration of the system.

Note that the filename type is used to load a custom ruleset. An alternate way to load a custom ruleset is to set the firewall_script variable to the absolute path of an executable script that includes IPFW commands. The examples used in this section assume that the firewall_script is set to /etc/ipfw.rules:

```bash
# sysrc firewall_script="/etc/ipfw.rules"
```

To enable logging through syslogd(8), include this line:

```bash
# sysrc firewall_logging="YES"
```

Only firewall rules with the log option will be logged. The default rules do not include this option and it must be manually added. Therefore it is advisable that the default ruleset is edited for logging. In addition, log rotation may be desired if the logs are stored in a separate file.

There is no /etc/rc.conf variable to set logging limits. To limit the number of times a rule is logged per connection attempt, specify the number using this line in /etc/sysctl.conf:

```bash
# echo "net.inet.ip.fw.verbose_limit=5" >> /etc/sysctl.conf
```

To enable logging through a dedicated interface named ipfw0, add this line to /etc/rc.conf instead:

```bash
# sysrc firewall_logif="YES"
```

Then use tcpdump to see what is being logged:

```bash
# tcpdump -t -n -i ipfw0
```

There is no overhead due to logging unless tcpdump is attached.

After saving the needed edits, start the firewall. To enable logging limits now, also set the sysctl value specified above:

```bash
# service ipfw start
# sysctl net.inet.ip.fw.verbose_limit=5
```

```bash
653
```
When a packet enters the IPFW firewall, it is compared against the first rule in the ruleset and progresses one rule at a time, moving from top to bottom in sequence. When the packet matches the selection parameters of a rule, the rule's action is executed and the search of the ruleset terminates for that packet. This is referred to as "first match wins". If the packet does not match any of the rules, it gets caught by the mandatory IPFW default rule number 65535, which denies all packets and silently discards them. However, if the packet matches a rule that contains the `count`, `skipto`, or `tee` keywords, the search continues. Refer to `ipfw(8)` for details on how these keywords affect rule processing.

When creating an IPFW rule, keywords must be written in the following order. Some keywords are mandatory while other keywords are optional. The words shown in uppercase represent a variable and the words shown in lowercase must precede the variable that follows it. The `#` symbol is used to mark the start of a comment and may appear at the end of a rule or on its own line. Blank lines are ignored.

```
CMD RULE_NUMBER set SET_NUMBER ACTION log LOG_AMOUNT PROTO from SRC SRC_PORT to DST DST_PORT OPTIONS
```

This section provides an overview of these keywords and their options. It is not an exhaustive list of every possible option. Refer to `ipfw(8)` for a complete description of the rule syntax that can be used when creating IPFW rules.

**CMD**

Every rule must start with `ipfw add`.

**RULE_NUMBER**

Each rule is associated with a number from 1 to 65534. The number is used to indicate the order of rule processing. Multiple rules can have the same number, in which case they are applied according to the order in which they have been added.

**SET_NUMBER**

Each rule is associated with a set number from 0 to 31. Sets can be individually disabled or enabled, making it possible to quickly add or delete a set of rules. If a SET_NUMBER is not specified, the rule will be added to set 0.

**ACTION**

A rule can be associated with one of the following actions. The specified action will be executed when the packet matches the selection criterion of the rule.

- `allow` | `accept` | `pass` | `permit`: these keywords are equivalent and allow packets that match the rule.
- `check-state`: checks the packet against the dynamic state table. If a match is found, execute the action associated with the rule which generated this dynamic rule, otherwise move to the next rule. A `check-state` rule does not have selection criterion. If no `check-state` rule is present in the ruleset, the dynamic rules table is checked at the first `keep-state` or `limit` rule.
- `count`: updates counters for all packets that match the rule. The search continues with the next rule.
- `deny` | `drop`: either word silently discards packets that match this rule. Additional actions are available. Refer to `ipfw(8)` for details.

**LOG_AMOUNT**

When a packet matches a rule with the `log` keyword, a message will be logged to `syslogd(8)` with a facility name of `SECURITY`. Logging only occurs if the number of packets logged for that particular rule does not exceed a specified LOG_AMOUNT. If no LOG_AMOUNT is specified, the limit is taken from the value of `net.inet.ip.fw.verbose_limit`. A value of zero removes the logging.
Once the limit is reached, logging can be re-enabled by clearing the logging counter or the packet counter for that rule, using `ipfw resetlog`.

Logging is done after all other packet matching conditions have been met, and before performing the final action on the packet. The administrator decides which rules to enable logging on.

This optional value can be used to specify any protocol name or number found in `/etc/protocols`.

The `from` keyword must be followed by the source address or a keyword that represents the source address. An address can be represented by `any`, `me` (any address configured on an interface on this system), `me6`, (any IPv6 address configured on an interface on this system), or `table` followed by the number of a lookup table which contains a list of addresses. When specifying an IP address, it can be optionally followed by its CIDR mask or subnet mask. For example, `1.2.3.4/25` or `1.2.3.4:255.255.255.128`.

An optional source port can be specified using the port number or name from `/etc/services`.

The `to` keyword must be followed by the destination address or a keyword that represents the destination address. The same keywords and addresses described in the SRC section can be used to describe the destination.

An optional destination port can be specified using the port number or name from `/etc/services`.

Several keywords can follow the source and destination. As the name suggests, OPTIONS are optional. Commonly used options include `in` or `out`, which specify the direction of packet flow, `icmptypes` followed by the type of ICMP message, and `keep-state`. When a `keep-state` rule is matched, the firewall will create a dynamic rule which matches bidirectional traffic between the source and destination addresses and ports using the same protocol.

The dynamic rules facility is vulnerable to resource depletion from a SYN-flood attack which would open a huge number of dynamic rules. To counter this type of attack with IPFW, use `limit`. This option limits the number of simultaneous sessions by checking the open dynamic rules, counting the number of times this rule and IP address combination occurred. If this count is greater than the value specified by `limit`, the packet is discarded.

Dozens of OPTIONS are available. Refer to `ipfw(8)` for a description of each available option.

This section demonstrates how to create an example stateful firewall ruleset script named `/etc/ipfw.rules`. In this example, all connection rules use `in` or `out` to clarify the direction. They also use `via` `interface-name` to specify the interface the packet is traveling over.

When first creating or testing a firewall ruleset, consider temporarily setting this tunable:

```
net.inet.ip.fw.default_to_accept="1"
```

This sets the default policy of `ipfw(8)` to be more permissive than the default deny.
Making it slightly more difficult to get locked out of the system right after a reboot.

The firewall script begins by indicating that it is a Bourne shell script and flushes any existing rules. It then creates the `cmd` variable so that `ipfw add` does not have to be typed at the beginning of every rule. It also defines the `pif` variable which represents the name of the interface that is attached to the Internet.

```bash
#!/bin/sh
# Flush out the list before we begin.
ipfw -q -f flush
# Set rules command prefix
cmd="ipfw -q add"
pif="dc0"     # interface name of NIC attached to Internet
```

The first two rules allow all traffic on the trusted internal interface and on the loopback interface:

```bash
$cmd 00005 allow all from any to any via xl0
# No restrictions on Loopback Interface
$cmd 00010 allow all from any to any via lo0
```

The next rule allows the packet through if it matches an existing entry in the dynamic rules table:

`$cmd 00101 check-state`

The next set of rules defines which stateful connections internal systems can create to hosts on the Internet:

```bash
# Allow access to public DNS
# Replace x.x.x.x with the IP address of a public DNS server
# and repeat for each DNS server in /etc/resolv.conf
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state
```

```bash
# Allow access to ISP's DHCP server for cable/DSL configurations.
# Use the first rule and check log for IP address.
# Then, uncomment the second rule, input the IP address, and delete the first rule
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state
```

# Allow outbound HTTP and HTTPS connections

656
The next set of rules controls connections from Internet hosts to the internal network. It starts by denying packets typically associated with attacks and then explicitly allows specific types of connections. All the authorized services that originate from the Internet use limit to prevent flooding.

# Deny all inbound traffic from non-routable reserved address spaces
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif     #RFC 1918 private IP
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif      #RFC 1918 private IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif         #RFC 1918 private IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif        #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif          #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif     #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif       #reserved for docs
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif    #Sun cluster interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif        #Class D & E multicast

# Deny public pings
$cmd 00310 deny icmp from any to any in via $pif

# Deny ident
$cmd 00315 deny tcp from any to any 113 in via $pif

# Deny all Netbios services.
$cmd 00320 deny tcp from any to any 137 in via $pif
# Deny fragments
$cmd 00330 deny all from any to any frag in via $pif

# Deny ACK packets that did not match the dynamic rule table
$cmd 00332 deny tcp from any to any established in via $pif

# Allow traffic from ISP's DHCP server.
# Replace x.x.x.x with the same IP address used in rule 00120.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Allow HTTP connections to internal web server
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow inbound SSH connections
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Reject and log all other incoming connections
$cmd 00499 deny log all from any to any in via $pif

The last rule logs all packets that do not match any of the rules in the ruleset:
# Everything else is denied and logged
$cmd 00999 deny log all from any to any

30.4.4.

**核心**

内

NAT

FreeBSD's IPFW firewall has two implementations of NAT: one being the userland natd(8) daemon, and the more recent IPFW's built-in NAT facility also known as in-kernel NAT. Both work in conjunction with IPFW to provide network address translation. This can be used to provide an Internet Connection Sharing solution so that several internal computers can connect to the Internet using a single public IP address.

To do this, the FreeBSD machine connected to the Internet must act as a gateway. This system must have two NICs, where one is connected to the Internet and the other is connected to the internal LAN. Each machine connected to the LAN should be assigned an IP address in the private network space, as defined by [RFC 1918](https://tools.ietf.org/html/rfc1918).

Some additional configuration is needed in order to enable the in-kernel NAT function of IPFW. To enable in-kernel NAT support at boot time, the following must be set in `/etc/rc.conf`:

```
gateway_enable="YES"
firwall_enable="YES"
```

658
When `firewall_enable` is not set, but `firewall_nat_enable` is, it will have no effect and do nothing, because the in-kernel NAT implementation is only compatible with IPFW.

When the ruleset contains stateful rules, the positioning of the NAT rule is critical and the `skipto` action is used. The `skipto` action requires a rule number so that it knows which rule to jump to. Furthermore, because of the architecture of `libalias(3)`, a library implemented as a kernel module used for the in-kernel NAT facility of IPFW, it is necessary to disable TCP segmentation offloading, or in short TSO. TSO can be disabled on a per network interface basis by using `ifconfig(8)` or on a system wide basis using `sysctl(8)`. To disable TSO system wide, the following must be set in `/etc/sysctl.conf`:

```plaintext
net.inet.tcp.tso="0"
```

The example below builds upon the firewall ruleset shown in the previous section. It adds some additional entries and modifies some existing rules in order to configure the firewall for in-kernel NAT. It starts by adding some additional variables which represent the rule number to skip to, the `keep-state` option, and a list of TCP ports which will be used to reduce the number of rules.

```bash
#!/bin/sh
ipfw -q -f flush

# Create rules
cmd="ipfw -q add"
skip="skipto 1000"
pif=dc0
ks="keep-state"
good_tcpo="22,25,37,53,80,443,110"
```

A NAT instance will also be configured. With in-kernel NAT it is possible to have multiple NAT instances each with their own configuration. Although, for this example only one NAT instance is needed; NAT instance number 1. The configuration takes a few arguments and flags such as:

- `if` which indicates the public interface,
- `same_ports` which takes care that all aliased ports and local port numbers are mapped the same,
- `unreg_only` will result in only unregistered (private) address spaces to be processed by the NAT instance, and
- `reset` which will help to keep a functioning NAT instance even when the public IP address of the IPFW machine changes. For all possible options that can be passed to a single NAT instance configuration consult `ipfw(8)`.

Furthermore, because of the nature of a stateful NATing firewall, it is necessary to allow translated packets to be reinjected in the firewall for further processing, this can be achieved by disabling `one_pass` behavior at the start of the firewall script.

```bash
ipfw disable one_pass

ipfw -q nat 1 config if $pif same_ports unreg_only reset
```

The inbound NAT rule is inserted after the two rules which allow all traffic on the trusted and loopback interfaces and after the reassemble rule but before the `check-state` rule. It is important that the rule number selected for this NAT rule, in this example `100`, is higher than the first three rules and lower than the `check-state` rule. Furthermore, because of the behavior of in-kernel NAT it is advised to place a reassemble rule just before the first NAT rule and after the rules that allow traffic on trusted interface. Normally, IP fragmentation should not happen, but when dealing with IPSEC/ESP/GRE tunneling traffic it might and the reassembling of fragments is necessary before handing the complete packet over to the in-kernel NAT engine.
The reassemble rule was not needed with userland natd(8) because the internal workings of the IPFW divert action already takes care of this automatically as also stated in ipfw(8).

The current NAT instance number and NAT rule number does not match with the default NAT instance number and rule number created by rc.firewall which is a script to set up the baked-in default firewall rulesets present in FreeBSD.

$cmd 005 allow all from any to any via xl0  # exclude LAN traffic
$cmd 010 allow all from any to any via lo0  # exclude loopback traffic
$cmd 099 reass all from any to any in       # reassamble inbound packets
$cmd 100 nat 1 ip from any to any in via $pif # NAT any inbound packets

# Authorized outbound packets
$cmd 120 $skip udp from any to x.x.x.x 53 out via $pif $ks
$cmd 121 $skip udp from any to x.x.x.x 67 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks

The inbound rules remain the same, except for the very last rule which removes the via $pif in order to catch both inbound and outbound rules. The NAT rule must follow this last outbound rule, must have a higher number than that last rule, and the rule number must be referenced by the skipto action. In this ruleset, rule number 1000 handles passing all packets to our configured instance for NAT processing. The next rule allows any packet which has undergone NAT processing to pass.

$cmd 999 deny log all from any to any
$cmd 1000 nat 1 ip from any to any out via $pif # skipto location for outbound stateful rules
$cmd 1001 allow ip from any to any

In this example, rules 100, 101, 125, 1000, and 1001 control the address translation of the outbound and inbound packets so that the entries in the dynamic state table always register the private LANIP address.

Consider an internal web browser which initializes a new outbound HTTP session over port 80. When the first outbound packet enters the firewall, it does not match rule 100 because it is headed out rather than in. It passes rule 101 because this is the first packet and it has not been posted to the dynamic state table yet. The packet finally matches rule 125 as it is outbound on an allowed port and has a source IP address from the internal LAN. On matching this rule, two actions take place. First, the keep-state action adds an entry to the dynamic state table and the specified action, skipto rule 1000, is executed. Next, the packet undergoes NAT and is sent out to the Internet. This packet makes its way to the destination web server, where a response packet is generated and sent.
This new packet enters the top of the ruleset. It matches rule 100 and has its destination IP address mapped back to the original internal address. It then is processed by the check-state rule, is found in the table as an existing session, and is released to the LAN.

On the inbound side, the ruleset has to deny bad packets and allow only authorized services. A packet which matches an inbound rule is posted to the dynamic state table and the packet is released to the LAN. The packet generated as a response is recognized by the check-state rule as belonging to an existing session. It is then sent to rule 1000 to undergo NAT before being released to the outbound interface.

Transition from userland natd(8) to in-kernel NAT might seem seamless at first but there is small catch. When using the GENERIC kernel, IPFW will load the libalias.ko kernel module, when firewall_nat_enable is enabled in rc.conf. Although, the loaded module only provides basic NAT functionality, whereas the userland implementation natd(8) has all functionality available without any extra configuration from its userland library. All functionality refers to the following kernel modules that can additionally be loaded when needed besides the standard libalias.ko kernel module:

- alias_cuseeme.ko
- alias_ftp.ko
- alias_bbt.ko
- skinny.ko
- irc.ko
- alias_pptp.ko
- alias_smedia.ko

Using the kld_list directive in rc.conf to mimic the full functionality of the userland implementation. If a custom kernel is used, the full functionality of the userland library can be compiled in, in the kernel, using the option LIBALIAS.

30.4.4.1. Port new direction

The drawback with NAT in general is that the LAN clients are not accessible from the Internet. Clients on the LAN can make outgoing connections to the world but cannot receive incoming ones. This presents a problem if trying to run Internet services on one of the LAN client machines. A simple way around this is to redirect selected Internet ports on the NAT providing machine to a LAN client.

For example, an IRC server runs on client A and a web server runs on client B. For this to work properly, connections received on ports 6667 (IRC) and 80 (HTTP) must be redirected to the respective machines.

With in-kernel NAT all configuration is done in the NAT instance configuration. For a full list of options that an in-kernel NAT instance can use, consult ipfw(8). The IPFW syntax follows the syntax of natd. The syntax for redirect_port is as follows:

```
redirect_port proto targetIP:targetPORT[-targetPORT] [aliasIP:]aliasPORT[-aliasPORT] [remoteIP[:remotePORT[-remotePORT]]]
```

To configure the above example setup, the arguments should be:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcp 192.168.0.3:80 80
```

After adding these arguments to the configuration of NAT instance 1 in the above ruleset, the TCP ports will be port forwarded to the LAN client machines running the IRC and HTTP services.

```
ipfw -q nat 1 config if $pif same_ports unreg_only reset 
   redirect_port tcp 192.168.0.2:6667 6667 
   redirect_port tcp 192.168.0.3:80 80
```
Port ranges over individual ports can be indicated with `redirect_port`. For example, `tcp 192.168.0.2:2000-3000 2000-3000` would redirect all connections received on ports 2000 to 3000 to ports 2000 to 3000 on client A.

Address redirection is useful if more than one IP address is available. Each LAN client can be assigned its own external IP address by `ipfw(8)`, which will then rewrite outgoing packets from the LAN clients with the proper external IP address and redirects all traffic incoming on that particular IP address back to the specific LAN client. This is also known as static NAT. For example, if IP addresses 128.1.1.1, 128.1.1.2, and 128.1.1.3 are available, 128.1.1.1 can be used as the `ipfw(8)` machine's external IP address, while 128.1.1.2 and 128.1.1.3 are forwarded back to LAN clients A and B.

The `redirect_address` syntax is as below, where `localIP` is the internal IP address of the LAN client, and `publicIP` the external IP address corresponding to the LAN client.

```
redirect_address localIP publicIP
```

In the example, the arguments would read:

```
redirect_address 192.168.0.2 128.1.1.2
redirect_address 192.168.0.3 128.1.1.3
```

Like `redirect_port`, these arguments are placed in a NAT instance configuration. With address redirection, there is no need for port redirection, as all data received on a particular IP address is redirected.

The external IP addresses on the `ipfw(8)` machine must be active and aliased to the external interface. Refer to `rc.conf(5)` for details.

### 30.4.4.3. Userspace NAT

Let us start with a statement: the userspace NAT implementation: `natd(8)`, has more overhead than in-kernel NAT. For `natd(8)` to translate packets, the packets have to be copied from the kernel to userspace and back which brings in extra overhead that is not present with in-kernel NAT.

要**開機時**啟動 Userspace 的 NAT daemon `natd(8)` 需在 `/etc/rc.conf` 中做以下最**小設定**，其中 `natd_interface` 要設成連接到網路路的 NIC 名稱，`rc(8)` script of `natd(8)` 會自動檢查是否有使用動態 IP 位址，並且自行設定並處理。

```
gateway_enable="YES"
natd_enable="YES"
natd_interface="rl0"
```

In general, the above ruleset as explained for in-kernel NAT can also be used together with `natd(8)`. The only exceptions are the configuration of the in-kernel NAT instance (ipfw -q nat 1 config …) not being applicable any more, rule number 100 and 1000 will have to change slightly as below, and reassemble rule 99 is not needed anymore as the `divert` action is used which covers fragmentation.

```
$cmd 100 divert natd ip from any to any in via $pif
$cmd 1000 divert natd ip from any to any out via $pif
```
To configure port or address redirection, a similar syntax as with in-kernel NAT is used. Although, now, instead of specifying the configuration in our ruleset script like with in-kernel NAT, configuration of `natd(8)` is best done in a configuration file. To do this, an extra flag must be passed via `/etc/rc.conf` which specifies the path of the configuration file.

```
natd_flags=-f /etc/natd.conf
```

The specified file must contain a list of configuration options, one per line. For more information about the configuration file and possible variables, consult `natd(8)`. Below are two example entries, one per line:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_address 192.168.0.3 128.1.1.3
```

IPFW `ipfw` can be used to make manual, single rule additions or deletions to the active firewall while it is running. The problem with using this method is that all the changes are lost when the system reboots. It is recommended to instead write all the rules in a file and to use that file to load the rules at boot time and to replace the currently running firewall rules whenever that file changes.

`ipfw` is a useful way to display the running firewall rules to the console screen. The IPFW accounting facility dynamically creates a counter for each rule that counts each packet that matches the rule. During the process of testing a rule, listing the rule with its counter is one way to determine if the rule is functioning as expected.

To list all the running rules in sequence:

```
# ipfw list
```

To list all the running rules with a time stamp of when the last time the rule was matched:

```
# ipfw -t list
```

The next example lists accounting information and the packet count for matched rules along with the rules themselves. The first column is the rule number, followed by the number of matched packets and bytes, followed by the rule itself.

```
# ipfw -a list
```

To list dynamic rules in addition to static rules:

```
# ipfw -d list
```

To also show the expired dynamic rules:

```
# ipfw -d -e list
```
To zero the counters:

```
# ipfw zero
```

To zero the counters for just the rule with number `NUM`:

```
# ipfw zero NUM
```

Even with the logging facility enabled, IPFW will not generate any rule logging on its own. The firewall administrator decides which rules in the ruleset will be logged, and adds the `log` keyword to those rules. Normally only deny rules are logged. It is customary to duplicate the "ipfw default deny everything" rule with the `log` keyword included as the last rule in the ruleset. This way, it is possible to see all the packets that did not match any of the rules in the ruleset.

Logging is a two edged sword. If one is not careful, an over abundance of log data or a DoS attack can fill the disk with log files. Log messages are not only written to syslogd, but also are displayed on the root console screen and soon become annoying.

The `IPFIREWALL_VERBOSE_LIMIT=5` kernel option limits the number of consecutive messages sent to `syslogd(8)`, concerning the packet matching of a given rule. When this option is enabled in the kernel, the number of consecutive messages concerning a particular rule is capped at the number specified. There is nothing to be gained from 200 identical log messages. With this option set to five, five consecutive messages concerning a particular rule would be logged to `syslogd` and the remainder identical consecutive messages would be counted and posted to `syslogd` with a phrase like the following:

```
last message repeated 45 times
```

All logged packets messages are written by default to `/var/log/security`, which is defined in `/etc/syslog.conf`.

### 30.4.5.2. 建立规则 Script

Most experienced IPFW users create a file containing the rules and code them in a manner compatible with running them as a script. The major benefit of doing this is the firewall rules can be refreshed in mass without the need of rebooting the system to activate them. This method is convenient in testing new rules as the procedure can be executed as many times as needed. Being a script, symbolic substitution can be used for frequently used values to be substituted into multiple rules.

This example script is compatible with the syntax used by the `sh(1)`, `csh(1)`, and `tcsh(1)` shells.

Symbolic substitution fields are prefixed with a dollar sign ($). Symbolic fields do not have the $ prefix. The value to populate the symbolic field must be enclosed in double quotes ("").

Start the rules file like this:

```
############### start of example ipfw rules script #############
#
ipfw -q -f flush       # Delete all rules
# Set defaults
oif="tun0"             # out interface

# Customize rules

# Additional rules
```

30.4.5.3. 執行腳本
The rules are not important as the focus of this example is how the symbolic substitution fields are populated.

If the above example was in `/etc/ipfw.rules`, the rules could be reloaded by the following command:

```
# sh /etc/ipfw.rules
```

`/etc/ipfw.rules` can be located anywhere and the file can have any name. The same thing could be accomplished by running these commands by hand:

```
# ipfw -q -f flush
# ipfw -q add check-state
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

30.4.6. IPFW核心選項

In order to statically compile IPFW support into a custom kernel, refer to the instructions in FreeBSD核心設定. The following options are available for the custom kernel configuration file:

- `options    IPFIREWALL           # enables IPFW`
- `options    IPFIREWALL_VERBOSE       # enables logging for rules with log keyword to syslogd(8)`
- `options    IPFIREWALL_VERBOSE_LIMIT=5   # limits number of logged packets per-entry`
- `options    IPFIREWALL_DEFAULT_TO_ACCEPT # sets default policy to pass what is not explicitly denied`
- `options    IPFIREWALL_NAT       # enables in-kernel NAT support`
- `options    IPFIREWALL_NAT64     # enables in-kernel NAT64 support`
- `options    IPFIREWALL_NPTV6     # enables in-kernel IPv6 NPT support`
- `options    IPFIREWALL_PMOD      # enables protocols modification module support`
30.5. IPFILTER (IPF)

IPFILTER, also known as IPF, is a cross-platform, open-source firewall that has been ported to various operating systems, including FreeBSD, NetBSD, OpenBSD, and Solaris™.

IPFILTER is a core (Kernel-side) firewall and NAT mechanisms can be controlled and monitored by Userland programs. Firewall rules can be set or removed using `ipf`, while NAT rules can be set or removed using `ipnat`. `ipfstat` can be used to list IPFILTER rules in the kernel part, `ipmon` can be used to record IPFILTER activities to system logs.

IPF was originally written with the "last match wins" rule processing logic and could only use stateless (Stateless) rules. Later, IPF began to support quick (Quick) and keep state (Keep state) options.

IPF FAQ is located at [http://www.phildev.net/ipf/index.html](http://www.phildev.net/ipf/index.html), and IPFilter mailing list archives can be obtained from [http://marc.info/?l=ipfilter](http://marc.info/?l=ipfilter).

Due to the FreeBSD also supports IPF, the operations manual is special in this chapter to introduce. This section provides several examples of rules using quick (Quick) and keep state (Keep state) options.

### 30.5.1. Enable IPF

IPF is included in the basic FreeBSD install as a kernel loadable module, meaning that a custom kernel is not needed in order to enable IPF.

For users who prefer to statically compile IPF support into a custom kernel, refer to the instructions in **FreeBSD Core**. The following kernel options are available:

- `options IPFILTER` enables IPFILTER support,
- `options IPFILTER_LOG` enables IPF logging using the `ipl` packet logging pseudo-device for every rule that has the `log` keyword,
- `options IPFILTER_LOOKUP` enables IP pools to speed up IP lookups,
- `options IPFILTER_DEFAULT_BLOCK` changes the default behavior so that any packet not matching a firewall pass rule gets blocked.

To configure the system to enable IPF at boot time, add the following entries to `/etc/rc.conf`.

```
# Start ipf firewall
ipfilter_enable="YES"
# loads rules definition text file
ipfilter_rules="/etc/ipf.rules"
# Start IP monitor log
ipmon_enable="YES"
# start as daemon
ipmon_flags="-Ds"
```

Remember to add a `block all` rule at the end of the ruleset. If you don't want to compile a custom kernel, add a `block all` rule at the end of the ruleset.
If NAT functionality is needed, also add these lines:

```
# n = map IP & port to names

gateway_enable="YES"              # Enable as LAN gateway
ipnat_enable="YES"                # Start ipnat function
ipnat_rules="/etc/ipnat.rules"    # rules definition file for ipnat

Then, to start IPF now:

# service ipfilter start
```

To load the firewall rules, specify the name of the ruleset file using `ipf`. The following command can be used to replace the currently running firewall rules:

```
# ipf -Fa -f /etc/ipf.rules
```

where `-Fa` flushes all the internal rules tables and `-f` specifies the file containing the rules to load. This provides the ability to make changes to a custom ruleset and update the running firewall with a fresh copy of the rules without having to reboot the system. This method is convenient for testing new rules as the procedure can be executed as many times as needed.

Refer to `ipf(8)` for details on the other flags available with this command.

### 30.5.2. IPF 規則語法

This section describes the IPF rule syntax used to create stateful rules. When creating rules, keep in mind that unless the `quick` keyword appears in a rule, every rule is read in order, with the last matching rule being the one that is applied. This means that even if the first rule to match a packet is a `pass`, if there is a later matching rule that is a `block`, the packet will be dropped.

Sample rulesets can be found in `/usr/shared/examples/ipfilter`.

When creating rules, a `#` character is used to mark the start of a comment and may appear at the end of a rule, to explain that rule's function, or on its own line. Any blank lines are ignored.

The keywords which are used in rules must be written in a specific order, from left to right. Some keywords are mandatory while others are optional. Some keywords have sub-options which may be keywords themselves and also include more sub-options. The keyword order is as follows, where the words shown in uppercase represent a variable and the words shown in lowercase must precede the variable that follows it:

```
ACTION DIRECTION OPTIONS proto PROTO_TYPE from SRC_ADDR SRC_PORT to DST_ADDR
DST_PORT TCP_FLAG|ICMP_TYPE keep state STATE
```

This section describes each of these keywords and their options. It is not an exhaustive list of every possible option. Refer to `ipf(5)` for a complete description of the rule syntax that can be used when creating IPF rules and examples for using each keyword.

**ACTION**

The action keyword indicates what to do with the packet if it matches that rule. Every rule must have an action. The following actions are recognized:

- `block`: drops the packet.
- `accept`: forwards the packet.
- `reject`: drops the packet and sends a reply.
- `send`: passes the packet to the transport layer.
- `error`: passthrough (send the packet to the IP layer).
- `drop`: drops the packet.
- `logging`: logs the packet.
- `log`: logs the packet.
- `pass`: forwards the packet.
- `copy`: copies the packet.
- `reject-undefined`: sends a reply.
- `verify`: verifies the packet.
- `redirect`: redirects the packet.
- `reject-unsafe`: sends a reply.
- `return`: passback as raw.

Other actions such as `capture` or `proxy` are also possible, but IPF does not log or drop them by default. You can add an `action` clause that includes `log`, `drop`, or `logging` to log or drop packets with these actions.
pass: allows the packet.

log: generates a log record.

count: counts the number of packets and bytes which can provide an indication of how often a rule is used.

auth: queues the packet for further processing by another program.

call: provides access to functions built into IPF that allow more complex actions.

decapsulate: removes any headers in order to process the contents of the packet.

DIRECTION

Next, each rule must explicitly state the direction of traffic using one of these keywords:
in: the rule is applied against an inbound packet.

out: the rule is applied against an outbound packet.

all: the rule applies to either direction.

If the system has multiple interfaces, the interface can be specified along with the direction. An example would be in on fxp0.

OPTIONS

Options are optional. However, if multiple options are specified, they must be used in the order shown here.

log: when performing the specified ACTION, the contents of the packet's headers will be written to the ipl(4) packet log pseudo-device.

quick: if a packet matches this rule, the ACTION specified by the rule occurs and no further processing of any following rules will occur for this packet.

on: must be followed by the interface name as displayed by ifconfig(8). The rule will only match if the packet is going through the specified interface in the specified direction.

When using the log keyword, the following qualifiers may be used in this order:

body: indicates that the first 128 bytes of the packet contents will be logged after the headers.

first: if the log keyword is being used in conjunction with a keep state option, this option is recommended so that only the triggering packet is logged and not every packet which matches the stateful connection.

Additional options are available to specify error return messages. Refer to ipf(5) for more details.

PROTO_TYPE

The protocol type is optional. However, it is mandatory if the rule needs to specify a SRC_PORT or a DST_PORT as it defines the type of protocol. When specifying the type of protocol, use the proto keyword followed by either a protocol number or name from /etc/protocols. Example protocol names include tcp, udp, or icmp. If PROTO_TYPE is specified but no SRC_PORT or DST_PORT is specified, all port numbers for that protocol will match that rule.

SRC_ADDR

The from keyword is mandatory and is followed by a keyword which represents the source of the packet. The source can be a hostname, an IP address followed by the CIDR mask, an address pool, or the keyword all. Refer to ipf(5) for examples.

There is no way to match ranges of IP addresses which do not express themselves easily using the dotted numeric form / mask-length notation. The net-mgmt/ipcalc package or port may be used.
The port number of the source is optional. However, if it is used, it requires PROTO_TYPE to be first defined in the rule. The port number must also be preceded by the `proto` keyword.

A number of different comparison operators are supported: `=`, `!=`, `<`, `>`, `<=` and `>=`.

To specify port ranges, place the two port numbers between `<>`, `><`, or `:`.

The `to` keyword is mandatory and is followed by a keyword which represents the destination of the packet. Similar to SRC_ADDR, it can be a hostname, an IP address followed by the CIDR mask, an address pool, or the keyword `all`.

Similar to SRC_PORT, the port number of the destination is optional. However, if it is used, it requires PROTO_TYPE to be first defined in the rule. The port number must also be preceded by the `proto` keyword.

If `tcp` is specified as the PROTO_TYPE, flags can be specified as letters, where each letter represents one of the possible TCP flags used to determine the state of a connection. Possible values are: `S` (SYN), `A` (ACK), `P` (PSH), `F` (FIN), `U` (URG), `R` (RST), `C` (CWN), and `E` (ECN).

If `icmp` is specified as the PROTO_TYPE, the ICMP type to match can be specified. Refer to `ipf(5)` for the allowable types.

If a `pass` rule contains `keep state`, IPF will add an entry to its dynamic state table and allow subsequent packets that match the connection. IPF can track state for TCP, UDP, and ICMP sessions. Any packet that IPF can be certain is part of an active session, even if it is a different protocol, will be allowed.

In IPF, packets destined to go out through the interface connected to the public Internet are first checked against the dynamic state table. If the packet matches the next expected packet comprising an active session conversation, it exits the firewall and the state of the session conversation flow is updated in the dynamic state table. Packets that do not belong to an already active session are checked against the outbound ruleset. Packets coming in from the interface connected to the public Internet are first checked against the dynamic state table. If the packet matches the next expected packet comprising an active session, it exits the firewall and the state of the session conversation flow is updated in the dynamic state table. Packets that do not belong to an already active session are checked against the inbound ruleset.

Several keywords can be added after `keep state`. If used, these keywords set various options that control stateful filtering, such as setting connection limits or connection age. Refer to `ipf(5)` for the list of available options and their descriptions.

This section demonstrates how to create an example ruleset which only allows services matching `pass` rules and blocks all others.

FreeBSD uses the loopback interface (`lo0`) and the IP address `127.0.0.1` for internal communication.

The firewall ruleset must contain rules to allow free movement of these internally used packets:

```
# no restrictions on loopback interface
```
The public interface connected to the Internet is used to authorize and control access of all outbound and inbound connections. If one or more interfaces are cabled to private networks, those internal interfaces may require rules to allow packets originating from the LAN to flow between the internal networks or to the interface attached to the Internet. The ruleset should be organized into three major sections: any trusted internal interfaces, outbound connections through the public interface, and inbound connections through the public interface.

These two rules allow all traffic to pass through a trusted LAN interface named `xl0`:

```
# no restrictions on inside LAN interface for private network
pass out quick on xl0 all
pass in quick on xl0 all
```

The rules for the public interface's outbound and inbound sections should have the most frequently matched rules placed before less commonly matched rules, with the last rule in the section blocking and logging all packets for that interface and direction.

This set of rules defines the outbound section of the public interface named `dc0`. These rules keep state and identify the specific services that internal systems are authorized for public Internet access. All the rules use `quick` and specify the appropriate port numbers and, where applicable, destination addresses.

```
# interface facing Internet (outbound)
# Matches session start requests originating from or behind the firewall, destined for the Internet.
# Allow outbound access to public DNS servers.
# Replace x.x.x. with address listed in /etc/resolv.conf.
# Repeat for each DNS server.
pass out quick on dc0 proto tcp from any to x.x.x. port = 53 flags S keep state
pass out quick on dc0 proto udp from any to xxx port = 53 keep state

# Allow access to ISP's specified DHCP server for cable or DSL networks.
# Use the first rule, then check log for the IP address of DHCP server.
# Then, uncomment the second rule, replace z.z.z.z with the IP address,
# and comment out the first rule
pass out log quick on dc0 proto udp from any to any port = 67 keep state
#pass out quick on dc0 proto udp from any to z.z.z.z port = 67 keep state

# Allow HTTP and HTTPS
pass out quick on dc0 proto tcp from any to any port = 80 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 443 flags S keep state

# Allow email
```
pass out quick on dc0 proto tcp from any to any port = 110 flags S keep state

pass out quick on dc0 proto tcp from any to any port = 25 flags S keep state

# Allow NTP
pass out quick on dc0 proto tcp from any to any port = 37 flags S keep state

# Allow FTP
pass out quick on dc0 proto tcp from any to any port = 21 flags S keep state

# Allow SSH
pass out quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Allow ping
pass out quick on dc0 proto icmp from any to any icmp-type 8 keep state

# Block and log everything else
block out log first quick on dc0 all

This example of the rules in the inbound section of the public interface blocks all undesirable packets first. This reduces the number of packets that are logged by the last rule.

# interface facing Internet (inbound)
# Block all inbound traffic from non-routable or reserved address spaces
block in quick on dc0 from 192.168.0.0/16 to any    #RFC 1918 private IP
block in quick on dc0 from 172.16.0.0/12 to any     #RFC 1918 private IP
block in quick on dc0 from 10.0.0.0/8 to any        #RFC 1918 private IP
block in quick on dc0 from 127.0.0.0/8 to any       #loopback
block in quick on dc0 from 0.0.0.0/8 to any         #loopback
block in quick on dc0 from 169.254.0.0/16 to any    #DHCP auto-config
block in quick on dc0 from 192.0.2.0/24 to any      #reserved for docs
block in quick on dc0 from 204.152.64.0/23 to any   #Sun cluster interconnect
block in quick on dc0 from 224.0.0.0/3 to any       #Class D & E multicast

# Block fragments and too short tcp packets
block in quick on dc0 all with frags
block in quick on dc0 proto tcp all with short

# block source routed packets
block in quick on dc0 all with opt lsrr
block in quick on dc0 all with opt ssrr

# Block OS fingerprint attempts and log first occurrence
block in log first quick on dc0 proto tcp from any to any flags FUP
# Block anything with special options

block in quick on dc0 all with ipopts

# Block public pings and ident

block in quick on dc0 proto icmp all icmp-type 8

block in quick on dc0 proto tcp from any to any port = 113

# Block incoming Netbios services

block in log first quick on dc0 proto tcp/udp from any to any port = 137

block in log first quick on dc0 proto tcp/udp from any to any port = 138

block in log first quick on dc0 proto tcp/udp from any to any port = 139

block in log first quick on dc0 proto tcp/udp from any to any port = 81

Any time there are logged messages on a rule with the log first option, run ipfstat -hio to evaluate how many times the rule has been matched. A large number of matches may indicate that the system is under attack.

The rest of the rules in the inbound section define which connections are allowed to be initiated from the Internet. The last rule denies all connections which were not explicitly allowed by previous rules in this section.

# Allow traffic in from ISP's DHCP server. Replace z.z.z.z with the same IP address used in the outbound section.

pass in quick on dc0 proto udp from z.z.z.z to any port = 68 keep state

# Allow public connections to specified internal web server

pass in quick on dc0 proto tcp from any to x.x.x.x port = 80 flags S keep state

# Block and log only first occurrence of all remaining traffic.

block in log first quick on dc0 all

30.5.4.設定

To enable NAT, add these statements to /etc/rc.conf and specify the name of the file containing the NAT rules:

```
gateway_enable="YES"
ipnat_enable="YES"
ipnat_rules="/etc/ipnat.rules"
```

NAT rules are flexible and can accomplish many different things to fit the needs of both commercial and home users. The rule syntax presented here has been simplified to demonstrate common usage. For a complete rule syntax description, refer to ipnat(5).

The basic syntax for a NAT rule is as follows, where map starts the rule and IF should be replaced with the name of the external interface:

```
map IF
```

672
map IF LAN_IP_RANGE -> PUBLIC_ADDRESS

The LAN_IP_RANGE is the range of IP addresses used by internal clients. Usually, it is a private address range such as 192.168.1.0/24. The PUBLIC_ADDRESS can either be the static external IP address or the keyword 0/32 which represents the IP address assigned to IF.

In IPF, when a packet arrives at the firewall from the LAN with a public destination, it first passes through the outbound rules of the firewall ruleset. Then, the packet is passed to the NAT ruleset which is read from the top down, where the first matching rule wins. IPF tests each NAT rule against the packet's interface name and source IP address. When a packet's interface name matches a NAT rule, the packet's source IP address in the private LAN is checked to see if it falls within the IP address range specified in LAN_IP_RANGE. On a match, the packet has its source IP address rewritten with the public IP address specified by PUBLIC_ADDRESS. IPF posts an entry in its internal NAT table so that when the packet returns from the Internet, it can be mapped back to its original private IP address before being passed to the firewall rules for further processing.

For networks that have large numbers of internal systems or multiple subnets, the process of funneling every private IP address into a single public IP address becomes a resource problem. Two methods are available to relieve this issue.

The first method is to assign a range of ports to use as source ports. By adding the portmap keyword, NAT can be directed to only use source ports in the specified range:

map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp 20000:60000

Alternately, use the auto keyword which tells NAT to determine the ports that are available for use:

map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp auto

The second method is to use a pool of public addresses. This is useful when there are too many LAN addresses to fit into a single public address and a block of public IP addresses is available. These public addresses can be used as a pool from which NAT selects an IP address as a packet's address is mapped on its way out.

The range of public IP addresses can be specified using a netmask or CIDR notation. These two rules are equivalent:

map dc0 192.168.1.0/24 -> 204.134.75.0/255.255.255.0
map dc0 192.168.1.0/24 -> 204.134.75.0/24

A common practice is to have a publicly accessible web server or mail server segregated to an internal network segment. The traffic from these servers still has to undergo NAT, but port redirection is needed to direct inbound traffic to the correct server. For example, to map a web server using the internal address 10.0.10.25 to its public IP address of 20.20.20.5, use this rule:

rdr dc0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80

If it is the only web server, this rule would also work as it redirects all external HTTP requests to 10.0.10.25:

rdr dc0 0.0.0.0/0 port 80 -> 10.0.10.25 port 80
IPF has a built in FTP proxy which can be used with NAT. It monitors all outbound traffic for active or passive FTP connection requests and dynamically creates temporary filter rules containing the port number used by the FTP data channel. This eliminates the need to open large ranges of high order ports for FTP connections.

In this example, the first rule calls the proxy for outbound FTP traffic from the internal LAN. The second rule passes the FTP traffic from the firewall to the Internet, and the third rule handles all non-FTP traffic from the internal LAN:

```
map dc0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
map dc0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
map dc0 10.0.10.0/29 -> 0/32
```

The FTP map rules go before the NAT rule so that when a packet matches an FTP rule, the FTP proxy creates temporary filter rules to let the FTP session packets pass and undergo NAT. All LAN packets that are not FTP will not match the FTP rules but will undergo NAT if they match the third rule.

Without the FTP proxy, the following firewall rules would instead be needed. Note that without the proxy, all ports above 1024 need to be allowed:

```
# Allow out LAN PC client FTP to public Internet
# Active and passive modes
pass out quick on rl0 proto tcp from any to any port = 21 flags S keep state
# Allow out passive mode data channel high order port numbers
pass out quick on rl0 proto tcp from any to any port > 1024 flags S keep state
# Active mode let data channel in from FTP server
pass in quick on rl0 proto tcp from any to any port = 20 flags S keep state
```

Whenever the file containing the NAT rules is edited, run `ipnat` with `-CF` to delete the current NAT rules and flush the contents of the dynamic translation table. Include `-f` and specify the name of the NAT ruleset to load:

```
# ipnat -CF -f /etc/ipnat.rules
```

To display the NAT statistics:

```
# ipnat -s
```

To list the NAT table's current mappings:

```
# ipnat -l
```

To turn verbose mode on and display information relating to rule processing and active rules and table entries:
ipnat -v

30.5.5.

視

IPF統計資訊

IPF includes `ipfstat(8)` which can be used to retrieve and display statistics which are gathered as packets match rules as they go through the firewall. Statistics are accumulated since the firewall was last started or since the last time they were reset to zero using `ipf -Z`.

The default `ipfstat` output looks like this:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
input packets logged: blocked 99286 passed 0
output packets logged: blocked 0 passed 0
packets logged: input 0 output 0
log failures: input 3898 output 0
fragment state (in): kept 0 lost 0
fragment state (out): kept 0 lost 0
packet state (in): kept 169364 lost 0
packet state (out): kept 431395 lost 0
ICMP replies: 0 TCP RSTs sent: 0
Result cache hits (in): 1215208 (out): 1098963
IN Pullups succeeded: 2 failed: 0
OUT Pullups succeeded: 0 failed: 0
Fastroute successes: 0 failures: 0
TCP cksum fails (in): 0 (out): 0
Packet log flags set:
```

Several options are available. When supplied with either `-i` for inbound or `-o` for outbound, the command will retrieve and display the appropriate list of filter rules currently installed and in use by the kernel. To also see the rule numbers, include `-n`. For example, `ipfstat -on` displays the outbound rules table with rule numbers:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Include `-h` to prefix each rule with a count of how many times the rule was matched. For example, `ipfstat -oh` displays the outbound internal rules table, prefixing each rule with its usage count:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

To display the state table in a format similar to `top(1)`, use `ipfstat -t`. When the firewall is under

675
attack, this option provides the ability to identify and see the attacking packets. The optional sub-
flags give the ability to select the destination or source IP, port, or protocol to be monitored in real
time. Refer to ipfstat(8) for details.

30.5.6. IPF

日

誌

IPF provides ipmon, which can be used to write the firewall's logging information in a human
readable format. It requires that options IPFILTER_LOG be first added to a custom kernel using the
instructions in設定FreeBSD核心.

This command is typically run in daemon mode in order to provide a continuous system log file so
that logging of past events may be reviewed. Since FreeBSD has a built in syslogd(8) facility to
automatically rotate system logs, the default rc.conf ipmon_flags statement uses

ipmon_flags="-Ds" # D = start as daemon
# s = log to syslog
# v = log tcp window, ack, seq
# n = map IP & port to names

Logging provides the ability to review, after the fact, information such as which packets were
dropped, what addresses they came from, and where they were going. This information is useful in
tracking down attackers.

Once the logging facility is enabled in rc.conf and started with service ipmon start, IPF will only log
the rules which contain the log keyword. The firewall administrator decides which rules in the
ruleset should be logged and normally only deny rules are logged. It is customary to include the
log keyword in the last rule in the ruleset. This makes it possible to see all the packets that did not
match any of the rules in the ruleset.

By default, ipmon -Ds mode uses local0 as the logging facility. The following logging levels can be
used to further segregate the logged data:

LOG_INFO - packets logged using the "log" keyword as the action rather than pass or
block.
LOG_NOTICE - packets logged which are also passed
LOG_WARNING - packets logged which are also blocked
LOG_ERR - packets which have been logged and which can be considered short due to an
incomplete header

In order to setup IPF to log all data to /var/log/ipfilter.log, first create the empty file:

# touch /var/log/ipfilter.log

Then, to write all logged messages to the specified file, add the following statement to
/etc/syslog.conf:

local0.* /var/log/ipfilter.log

To activate the changes and instruct syslogd(8) to read the modified /etc/syslog.conf,
run service syslogd reload.
Do not forget to edit /etc/newsyslog.conf to rotate the new log file.

Messages generated by ipmon consist of data fields separated by white space. Fields common to all messages are:

1. The date of packet receipt.
2. The time of packet receipt. This is in the form HH:MM:SS.F, for hours, minutes, seconds, and fractions of a second.
3. The name of the interface that processed the packet.
4. The group and rule number of the rule in the format @0:17.
5. The action: p for passed, b for blocked, S for a short packet, n did not match any rules, and L for a log rule.
6. The addresses written as three fields: the source address and port separated by a comma, the → symbol, and the destination address and port. For example: 209.53.17.22,80 → 198.73.220.17,1722.
7. PR followed by the protocol name or number: for example, PR tcp.
8. len followed by the header length and total length of the packet: for example, len 20 40.

If the packet is a TCP packet, there will be an additional field starting with a hyphen followed by letters corresponding to any flags that were set. Refer to ipf(5) for a list of letters and their flags.

If the packet is an ICMP packet, there will be two fields at the end: the first always being “icmp” and the next being the ICMP message and sub-message type, separated by a slash. For example: icmp 3/3 for a port unreachable message.

30.6. Blacklistd

Blacklistd is a daemon listening to sockets to receive notifications from other daemons about connection attempts that failed or were successful. It is most widely used in blocking too many connection attempts on open ports. A prime example is SSH running on the internet getting a lot of requests from bots or scripts trying to guess passwords and gain access. Using blacklistd, the daemon can notify the firewall to create a filter rule to block excessive connection attempts from a single source after a number of tries. Blacklistd was first developed on NetBSD and appeared there in version 7. FreeBSD 11 imported blacklistd from NetBSD.

This chapter describes how to set up blacklistd, configure it, and provides examples on how to use it. Readers should be familiar with basic firewall concepts like rules. For details, refer to the firewall chapter. PF is used in the examples, but other firewalls available on FreeBSD should be able to work with blacklistd, too.

30.6.1. 開啟 Blacklistd

The main configuration for blacklistd is stored in blacklistd.conf(5). Various command line options are also available to change blacklistd’s run-time behavior. Persistent configuration across reboots should be stored in /etc/blacklistd.conf. To enable the daemon during system boot, add a blacklistd_enable line to /etc/rc.conf like this:

```
# sysrc blacklistd_enable=yes
```

To start the service manually, run this command:

```
# service blacklistd start
```
Rules for blacklistd are configured in `blacklistd.conf(5)` with one entry per line. Each rule contains a tuple separated by spaces or tabs. Rules either belong to a local or a remote, which applies to the machine where blacklistd is running or an outside source, respectively.

30.6.2.1. 本地规则

An example blacklistd.conf entry for a local rule looks like this:

```
[local]
ssh             stream  *       *               *       3       24h
```

All rules that follow the `[local]` section are treated as local rules (which is the default), applying to the local machine. When a `[remote]` section is encountered, all rules that follow it are handled as remote machine rules.

Seven fields define a rule separated by either tabs or spaces. The first four fields identify the traffic that should be blacklisted. The three fields that follow define blacklistd's behavior. Wildcards are denoted as asterisks (`*`), matching anything in this field. The first field defines the location. In local rules, these are the network ports. The syntax for the location field is as follows:

```
[address|interface]/[mask]/[port]
```

Addresses can be specified as IPv4 in numeric format or IPv6 in square brackets. An interface name like `em0` can also be used.

The socket type is defined by the second field. TCP sockets are of type `stream`, whereas UDP is denoted as `dgram`. The example above uses TCP, since SSH is using that protocol.

A protocol can be used in the third field of a blacklistd rule. The following protocols can be used: `tcp`, `udp`, `tcp6`, `udp6`, or numeric. A wildcard, like in the example, is typically used to match all protocols unless there is a reason to distinguish traffic by a certain protocol.

In the fourth field, the effective user or owner of the daemon process that is reporting the event is defined. The username or UID can be used here, as well as a wildcard (see example rule above).

The packet filter rule name is declared by the fifth field, which starts the behavior part of the rule. By default, blacklistd puts all blocks under a pf anchor called `blacklistd` in `pf.conf` like this:

```
anchor "blacklistd/*" in on $ext_if
block in
pass out
```

For separate blacklists, an anchor name can be used in this field. In other cases, the wildcard will suffice. When a name starts with a hyphen (`-`) it means that an anchor with the default rule name prepended should be used. A modified example from the above using the hyphen would look like this:

```
ssh             stream  *       *               -ssh       3       24h
```

With such a rule, any new blacklist rules are added to an anchor called `blacklistd-ssh`.

To block whole subnets for a single rule violation, a `/` in the rule name can be used. This causes the
remaining portion of the name to be interpreted as the mask to be applied to the address specified in the rule. For example, this rule would block every address adjoining /24.

It is important to specify the proper protocol here. IPv4 and IPv6 treat /24 differently, that is the reason why * cannot be used in the third field for this rule.

This rule defines that if any one host in that network is misbehaving, everything else on that network will be blocked, too.

The sixth field, called `nfail`, sets the number of login failures required to blacklist the remote IP in question. When a wildcard is used at this position, it means that blocks will never happen. In the example rule above, a limit of three is defined meaning that after three attempts to log into SSH on one connection, the IP is blocked.

The last field in a blacklist rule definition specifies how long a host is blacklisted. The default unit is seconds, but suffixes like `m`, `h`, and `d` can also be specified for minutes, hours, and days, respectively.

The example rule in its entirety means that after three times authenticating to SSH will result in a new PF block rule for that host. Rule matches are performed by first checking local rules one after another, from most specific to least specific. When a match occurs, the remote rules are applied and the name, `nfail`, and disable fields are changed by the remote rule that matched.

Remote rules are used to specify how blacklistd changes its behavior depending on the remote host currently being evaluated. Each field in a remote rule is the same as in a local rule. The only difference is in the way blacklistd is using them. To explain it, this example rule is used:

```
[remote]
203.0.113.128/25 *      *       *               =/25    =       48h
```

The address field can be an IP address (either v4 or v6), a port or both. This allows setting special rules for a specific remote address range like in this example. The fields for type, protocol and owner are identically interpreted as in the local rule.

The name fields is different though: the equal sign (`=`) in a remote rule tells blacklistd to use the value from the matching local rule. It means that the firewall rule entry is taken and the /25 prefix (a netmask of 255.255.255.128) is added. When a connection from that address range is blacklisted, the entire subnet is affected. A PF anchor name can also be used here, in which case blacklistd will add rules for this address block to the anchor of that name. The default table is used when a wildcard is specified.

A custom number of failures in the `nfail` column can be defined for an address. This is useful for exceptions to a specific rule, to maybe allow someone a less strict application of rules or a bit more leniency in login tries. Blocking is disabled when an asterisk is used in this sixth field.

Remote rules allow a stricter enforcement of limits on attempts to log in compared to attempts coming from a local network like an office.

30.6.3. Blacklistd

There are a few software packages in FreeBSD that can utilize blacklistd's functionality. The two most prominent ones are `ftpd(8)` and `sshd(8)` to block excessive connection attempts. To activate blacklistd in the SSH daemon, add the following line to `/etc/ssh/sshd_config`:

```
679
```
接著重新啟動sshd來使變更生效。

Blacklisting for ftplib(8) is enabled using -B, either in /etc/inetd.conf or as a flag in /etc/rc.conf like this:

```
ftpd_flags=-B
```

That is all that is needed to make these programs talk to blacklistd.

### 30.6.4. Blacklistd

Blacklistd provides the user with a management utility called blacklistctl(8). It displays blocked addresses and networks that are blacklisted by the rules defined in blacklistd.conf(5). To see the list of currently blocked hosts, use `dump` combined with `-b` like this.

```
# blacklistctl dump -b
```

address/ma:port id nfail last access

| 213.0.123.128/25:22 | OK     | 6/3 | 2019/06/08 14:30:19 |

This example shows that there were 6 out of three permitted attempts on port 22 coming from the address range 213.0.123.128/25. There are more attempts listed than are allowed because SSH allows a client to try multiple logins on a single TCP connection. A connection that is currently going on is not stopped by blacklistd. The last connection attempt is listed in the `last access` column of the output.

To see the remaining time that this host will be on the blacklist, add `-r` to the previous command.

```
# blacklistctl dump -br
```

address/ma:port id nfail remaining time

| 213.0.123.128/25:22 | OK      | 6/3 | 36s |

In this example, there are 36s seconds left until this host will not be blocked any more.

### 30.6.5. 从封鎖清單移除主機

Sometimes it is necessary to remove a host from the block list before the remaining time expires. Unfortunately, there is no functionality in blacklistd to do that. However, it is possible to remove the address from the PF table using pfctl. For each blocked port, there is a child anchor inside the blacklistd anchor defined in /etc/pf.conf. For example, if there is a child anchor for blocking port 22 it is called `blacklistd/22`. There is a table inside that child anchor that contains the blocked addresses. This table is called `port` followed by the port number. In this example, it would be called `port22`. With that information at hand, it is now possible to use pfctl(8) to display all addresses listed like this:

```
# pfctl -a blacklistd/22 -t port22 -T show
```

...
After identifying the address to be unblocked from the list, the following command removes it from the list:

```
# pfctl -a blacklistd/22 -T delete 213.0.123.128/25
```

The address is now removed from PF, but will still show up in the blacklistctl list, since it does not know about any changes made in PF. The entry in blacklistd's database will eventually expire and be removed from its output eventually. The entry will be added again if the host is matching one of the block rules in blacklistd again.
Chapter 31. 頸階網路設定

31.1. 概述
This chapter covers a number of advanced networking topics.

- The basics of gateways and routes.
- How to set up USB tethering.
- How to set up IEEE™ 802.11 and Bluetooth™ devices.
- How to make FreeBSD act as a bridge.
- How to set up network PXE booting.
- How to set up IPv6 on a FreeBSD machine.
- 如何在FreeBSD上設定多個VLAN。
- Configure bluetooth headset.

在開始閱讀這章之前，您需要:
- Understand the basics of the /etc/rc scripts.
- 熟悉基本網路術語。
- Know how to configure and install a new FreeBSD kernel (設定FreeBSD核心).
- 了解如何安裝其他第三方軟體 (安裝應用程式:套件與Port)。

31.2. 通訊閘與路由
Routing is the mechanism that allows a system to find the network path to another system. A route is a defined pair of addresses which represent the “destination” and a “gateway”. The route indicates that when trying to get to the specified destination, send the packets through the specified gateway. There are three types of destinations: individual hosts, subnets, and “default”. The “default route” is used if no other routes apply. There are also three types of gateways: individual hosts, interfaces, also called links, and Ethernet hardware (MAC) addresses. Known routes are stored in a routing table.

This section provides an overview of routing basics. It then demonstrates how to configure a FreeBSD system as a router and offers some troubleshooting tips.

31.2.1. 路由基礎概念
To view the routing table of a FreeBSD system, use netstat(1):

```
% netstat -r
```

Routing tables
Internet:
Destination      Gateway            Flags     Refs     Use     Netif Expire
default          outside-gw         UGS        37      418       em0
localhost        localhost          UH          0      181       lo0
682
The entries in this example are as follows:

The first route in this table specifies the default route. When the local system needs to make a connection to a remote host, it checks the routing table to determine if a known path exists. If the remote host matches an entry in the table, the system checks to see if it can connect using the interface specified in that entry. If the destination does not match an entry, or if all known paths fail, the system uses the entry for the default route. For hosts on a local area network, the Gateway field in the default route is set to the system which has a direct connection to the Internet. When reading this entry, verify that the Flags column indicates that the gateway is usable (UG).

The default route for a machine which itself is functioning as the gateway to the outside world will be the gateway machine at the Internet Service Provider (ISP).

localhost

The second route is the localhost route. The interface specified in the Netif column for localhost is lo0, also known as the loopback device. This indicates that all traffic for this destination should be internal, rather than sending it out over the network.

MAC address

The addresses beginning with 0:e0: are MAC addresses. FreeBSD will automatically identify any hosts, test0 in the example, on the local Ethernet and add a route for that host over the Ethernet interface, re0. This type of route has a timeout, seen in the Expire column, which is used if the host does not respond in a specific amount of time. When this happens, the route to this host will be automatically deleted. These hosts are identified using the Routing Information Protocol (RIP), which calculates routes to local hosts based upon a shortest path determination.

subnet

FreeBSD will automatically add subnet routes for the local subnet. In this example, 10.20.30.255 is the broadcast address for the subnet 10.20.30 and example.com is the domain name associated with that subnet. The designation link#1 refers to the first Ethernet card in the machine.

Local network hosts and local subnets have their routes automatically configured by a daemon called routed(8). If it is not running, only routes which are statically defined by the administrator will exist.

host

The host1 line refers to the host by its Ethernet address. Since it is the sending host, FreeBSD knows to use the loopback interface (lo0) rather than the Ethernet interface. The two host2 lines represent aliases which were created using ifconfig(8). The ⇒ symbol after the lo0 interface says that an alias has been set in addition to the loopback address. Such routes only show up on the host that supports the alias and all other hosts on the local network will have a link#1 line for such routes.
The final line (destination subnet 224) deals with multicasting. Various attributes of each route can be seen in the Flags column. *常見路由表* summarizes some of these flags and their meanings:

- **U**: The route is active (up).
- **H**: The route destination is a single host.
- **G**: Send anything for this destination on to this gateway, which will figure out from there where to send it.
- **S**: This route was statically configured.
- **C**: Clones a new route based upon this route for machines to connect to. This type of route is normally used for local networks.
- **W**: The route was auto-configured based upon a local area network (clone) route.
- **L**: Route involves references to Ethernet (link) hardware.

On a FreeBSD system, the default route can be defined in /etc/rc.conf by specifying the IP address of the default gateway:

```
defaultrouter="10.20.30.1"
```

It is also possible to manually add the route using `route`:

```
# route add default 10.20.30.1
```

Note that manually added routes will not survive a reboot. For more information on manual manipulation of network routing tables, refer to `route(8)`.

31.2.2. 設定路由器使用靜態路由

A FreeBSD system can be configured as the default gateway, or router, for a network if it is a dual-homed system. A dual-homed system is a host which resides on at least two different networks. Typically, each network is connected to a separate network interface, though IP aliasing can be used to bind multiple addresses, each on a different subnet, to one physical interface.

In order for the system to forward packets between interfaces, FreeBSD must be configured as a router. Internet standards and good engineering practice prevent the FreeBSD Project from enabling this feature by default, but it can be configured to start at boot by adding this line to `/etc/rc.conf`:

```
gateway_enable="YES"          # Set to YES if this host will be a gateway
```

To enable routing now, set the `sysctl(8)` variable `net.inet.ip.forwarding` to 1. To stop routing, reset this variable to 0.

The routing table of a router needs additional routes so it knows how to reach other networks.
Routes can be either added manually using static routes or routes can be automatically learned using a routing protocol. Static routes are appropriate for small networks and this section describes how to add a static routing entry for a small network.

For large networks, static routes quickly become unscalable. FreeBSD comes with the standard BSD routing daemon `routed(8)`, which provides the routing protocols RIP, versions 1 and 2, and IRDP. Support for the BGP and OSPF routing protocols can be installed using the `net/zebra` package or port.

Consider the following network:

In this scenario, **RouterA** is a FreeBSD machine that is acting as a router to the rest of the Internet. It has a default route set to 10.0.0.1 which allows it to connect with the outside world. **RouterB** is already configured to use 192.168.1.1 as its default gateway.

Before adding any static routes, the routing table on **RouterA** looks like this:

```
% netstat -nr
Routing tables
Internet:
Destination        Gateway            Flags    Refs      Use  Netif  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0.0/24        link1             UC          0        0    xl0
192.168.1.0/24     link2             UC          0        0    xl1
```

With the current routing table, **RouterA** does not have a route to the 192.168.2.0/24 network. The following command adds the Internal Net 2 network to **RouterA**’s routing table using 192.168.1.2 as the next hop:
Now, RouterA can reach any host on the 192.168.2.0/24 network. However, the routing information will not persist if the FreeBSD system reboots. If a static route needs to be persistent, add it to /etc/rc.conf:

```
# Add Internal Net 2 as a persistent static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

The `static_routes` configuration variable is a list of strings separated by a space, where each string references a route name. The variable `route_internalnet2` contains the static route for that route name.

Using more than one string in `static_routes` creates multiple static routes. The following shows an example of adding static routes for the 192.168.0.0/24 and 192.168.1.0/24 networks:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

When an address space is assigned to a network, the service provider configures their routing tables so that all traffic for the network will be sent to the link for the site. But how do external sites know to send their packets to the network's ISP?

There is a system that keeps track of all assigned address spaces and defines their point of connection to the Internet backbone, or the main trunk lines that carry Internet traffic across the country and around the world. Each backbone machine has a copy of a master set of tables, which direct traffic for a particular network to a specific backbone carrier, and from there down the chain of service providers until it reaches a particular network.

It is the task of the service provider to advertise to the backbone sites that they are the point of connection, and thus the path inward, for a site. This is known as route propagation.

Sometimes, there is a problem with route propagation and some sites are unable to connect. Perhaps the most useful command for trying to figure out where routing is breaking down is `traceroute`. It is useful when `ping` fails.

When using `traceroute`, include the address of the remote host to connect to. The output will show the gateway hosts along the path of the attempt, eventually either reaching the target host, or terminating because of a lack of connection. For more information, refer to `traceroute(8)`.

FreeBSD natively supports both multicast applications and multicast routing. Multicast applications do not require any special configuration in order to run on FreeBSD. Support for multicast routing requires that the following option be compiled into a custom kernel:

```
options MROUTING
```
The multicast routing daemon, mrouted, can be installed using the net/mrouted package or port. This daemon implements the DVMRP multicast routing protocol and is configured by editing /usr/local/etc/mrouted.conf in order to set up the tunnels and DVMRP. The installation of mrouted also installs map-mbone and mrinfo, as well as their associated man pages. Refer to these for configuration examples.

DVMRP has largely been replaced by the PIM protocol in many multicast installations. Refer to pim(4) for more information.

31.3. Wireless Networks

Most wireless networks are based on the IEEE™ 802.11 standards. A basic wireless network consists of multiple stations communicating with radios that broadcast in either the 2.4GHz or 5GHz band, though this varies according to the locale and is also changing to enable communication in the 2.3GHz and 4.9GHz ranges.

802.11 networks are organized in two ways. In infrastructure mode, one station acts as a master with all the other stations associating to it, the network is known as a BSS, and the master station is termed an access point (AP). In a BSS, all communication passes through the AP; even when one station wants to communicate with another wireless station, messages must go through the AP. In the second form of network, there is no master and stations communicate directly. This form of network is termed an IBSS and is commonly known as an ad-hoc network.

802.11 networks were first deployed in the 2.4GHz band using protocols defined by the IEEE™ 802.11 and 802.11b standard. These specifications include the operating frequencies and the MAC layer characteristics, including framing and transmission rates, as communication can occur at various rates. Later, the 802.11a standard defined operation in the 5GHz band, including different signaling mechanisms and higher transmission rates. Still later, the 802.11g standard defined the use of 802.11a signaling and transmission mechanisms in the 2.4GHz band in such a way as to be backwards compatible with 802.11b networks.

Separate from the underlying transmission techniques, 802.11 networks have a variety of security mechanisms. The original 802.11 specifications defined a simple security protocol called WEP. This protocol uses a fixed pre-shared key and the RC4 cryptographic cipher to encode data transmitted on a network. Stations must all agree on the fixed key in order to communicate. This scheme was shown to be easily broken and is now rarely used except to discourage transient users from joining networks. Current security practice is given by the IEEE™ 802.11i specification that defines new cryptographic ciphers and an additional protocol to authenticate stations to an access point and exchange keys for data communication. Cryptographic keys are periodically refreshed and there are mechanisms for detecting and countering intrusion attempts. Another security protocol specification commonly used in wireless networks is termed WPA, which was a precursor to 802.11i. WPA specifies a subset of the requirements found in 802.11i and is designed for implementation on legacy hardware. Specifically, WPA requires only the TKIP cipher that is derived from the original WEP cipher. 802.11i permits use of TKIP but also requires support for a stronger cipher, AES-CCM, for encrypting data. The AES cipher was not required in WPA because it was deemed too computationally costly to be implemented on legacy hardware.

The other standard to be aware of is 802.11e. It defines protocols for deploying multimedia applications, such as streaming video and voice over IP (VoIP), in an 802.11 network. Like 802.11i, 802.11e also has a precursor specification termed WME (later renamed WMM) that has been defined by an industry group as a subset of 802.11e that can be deployed now to enable multimedia applications while waiting for the final ratification of 802.11e. The most important thing to know about 802.11e and WME/WMM is that it enables prioritized traffic over a wireless network through Quality of Service (QoS) protocols and enhanced media access protocols. Proper implementation of these protocols enables high speed bursting of data and prioritized traffic flow.

FreeBSD supports networks that operate using 802.11a, 802.11b, and 802.11g. The WPA and 802.11i security protocols are likewise supported (in conjunction with any of 11a, 11b, and 11g) and QoS and traffic prioritization required by the WME/WMM protocols are supported for a limited set of...
Connecting a computer to an existing wireless network is a very common situation. This procedure shows the steps required.

1. Obtain the SSID (Service Set Identifier) and PSK (Pre-Shared Key) for the wireless network from the network administrator.

2. Identify the wireless adapter. The FreeBSD GENERIC kernel includes drivers for many common wireless adapters. If the wireless adapter is one of those models, it will be shown in the output from `ifconfig(8)`:
   ```
   % ifconfig | grep -B3 -i wireless
   ```
   On FreeBSD 11 or higher, use this command instead:
   ```
   % sysctl net.wlan.devices
   ```
   If a wireless adapter is not listed, an additional kernel module might be required, or it might be a model not supported by FreeBSD. This example shows the Atheros ath0 wireless adapter.

3. Add an entry for this network to `/etc/wpa_supplicant.conf`. If the file does not exist, create it. Replace `myssid` and `mypsk` with the SSID and PSK provided by the network administrator.
   ```
   network={
   ssid="myssid"
   psk="mypsk"
   }
   ```

4. Add entries to `/etc/rc.conf` to configure the network on startup:
   ```
   wlans_ath0="wlan0"
   ifconfig_wlan0="WPA SYNCDHCP"
   ```

5. Restart the computer, or restart the network service to connect to the network:
   ```
   # service netif restart
   ```
modules so that only the required support needs to be configured. The most commonly used wireless devices are those that use parts made by Atheros. These devices are supported by `ath(4)` and require the following line to be added to `/boot/loader.conf`:

```
if_ath_load="YES"
```

The Atheros driver is split up into three separate pieces: the driver (`ath(4)`), the hardware support layer that handles chip-specific functions (`ath_hal(4)`), and an algorithm for selecting the rate for transmitting frames. When this support is loaded as kernel modules, any dependencies are automatically handled. To load support for a different type of wireless device, specify the module for that device. This example is for devices based on the Intersil Prism parts (`wi(4)`) driver:

```
if_wi_load="YES"
```

The examples in this section use an `ath(4)` device and the device name in the examples must be changed according to the configuration. A list of available wireless drivers and supported adapters can be found in the FreeBSD Hardware Notes, available on the [Release Information](http://www.freebsd.org) page of the FreeBSD website. If a native FreeBSD driver for the wireless device does not exist, it may be possible to use the Windows™ driver with the help of the NDIS driver wrapper.

In addition, the modules that implement cryptographic support for the security protocols to use must be loaded. These are intended to be dynamically loaded on demand by the `wlan(4)` module, but for now they must be manually configured. The following modules are available:

- `wlan_wep(4)`
- `wlan_ccmp(4)`
- `wlan_tkip(4)`

The `wlan_ccmp(4)` and `wlan_tkip(4)` drivers are only needed when using the WPA or 802.11i security protocols. If the network does not use encryption, `wlan_wep(4)` support is not needed. To load these modules at boot time, add the following lines to `/boot/loader.conf`:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
wlan_tkip_load="YES"
```

Once this information has been added to `/boot/loader.conf`, reboot the FreeBSD box. Alternately, load the modules by hand using `kldload(8)`.

For users who do not want to use modules, it is possible to compile these drivers into the kernel by adding the following lines to a custom kernel configuration file:

```
device wlan              # 802.11 support
device wlan_wep          # 802.11 WEP support
device wlan_ccmp         # 802.11 CCMP support
device wlan_tkip         # 802.11 TKIP support
device wlan_amrr         # AMRR transmit rate control algorithm
device ath               # Atheros pci/cardbus NIC's
device ath_hal           # pci/cardbus chip support
options AH_SUPPORT_AR5416 # enable AR5416 tx/rx descriptors
device ath_rate_sample   # SampleRate tx rate control for ath
```
With this information in the kernel configuration file, recompile the kernel and reboot the FreeBSD machine.

Information about the wireless device should appear in the boot messages, like this:

```
ath0: <Atheros 5212> mem 0x88000000-0x8800ffff irq 11 at device 0.0 on cardbus1
ath0: [ITHREAD]
ath0: AR2413 mac 7.9 RF2413 phy 4.5
```

Since the regulatory situation is different in various parts of the world, it is necessary to correctly set the domains that apply to your location to have the correct information about what channels can be used.

The available region definitions can be found in `/etc/regdomain.xml`. To set the data at runtime, use `ifconfig`:

```
# ifconfig wlan0 regdomain ETSI country AT
```

To persist the settings, add it to `/etc/rc.conf`:

```
# sysrc create_args_wlan0="country AT regdomain ETSI"
```

31.3.4. 主從式 (Infrastructure)

Infrastructure (BSS) mode is the mode that is typically used. In this mode, a number of wireless access points are connected to a wired network. Each wireless network has its own name, called the SSID. Wireless clients connect to the wireless access points.

31.3.4.1. FreeBSD 客戶端

31.3.4.1.1. 如何尋找存取點

To scan for available networks, use `ifconfig(8)`. This request may take a few moments to complete as it requires the system to switch to each available wireless frequency and probe for available access points. Only the superuser can initiate a scan:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
```

```
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
```

```
dlinkap         00:13:46:49:41:76   11   54M           -90
                          :96   100 EPS  WPA WME
freebsdap       00:11:95:c3:0d:ac    1   54M           -83
                          :96   100 EPS  WPA
```

The interface must be up before it can scan. Subsequent scan requests do not require the interface to be marked as up again.

The output of a scan request lists each BSS/IBSS network found. Besides listing the name of the network, the SSID, the output also shows the BSSID, which is the MAC address of the access point.

The CAPS field identifies the type of each network and the capabilities of the stations operating.
Extended Service Set (ESS). Indicates that the station is part of an infrastructure network rather than an IBSS/ad-hoc network.

IBSS/ad-hoc network. Indicates that the station is part of an ad-hoc network rather than an ESS network.

Privacy. Encryption is required for all data frames exchanged within the BSS using cryptographic means such as WEP, TKIP or AES-CCMP.

Short Preamble. Indicates that the network is using short preambles, defined in 802.11b High Rate/DSSS PHY, and utilizes a 56 bit sync field rather than the 128 bit field used in long preamble mode.

Short slot time. Indicates that the 802.11g network is using a short slot time because there are no legacy (802.11b) stations present.

One can also display the current list of known networks with:

```
# ifconfig wlan0 list scan
```

This information may be updated automatically by the adapter or manually with a scan request.

Old data is automatically removed from the cache, so over time this list may shrink unless more scans are done.

This section provides a simple example of how to make the wireless network adapter work in FreeBSD without encryption. Once familiar with these concepts, it is strongly recommend to use WPA to set up the wireless network.

There are three basic steps to configure a wireless network: select an access point, authenticate the station, and configure an IP address. The following sections discuss each step.

### 31.3.4.1.2.1. 選擇存取點

Most of the time, it is sufficient to let the system choose an access point using the builtin heuristics. This is the default behavior when an interface is marked as up or it is listed in /etc/rc.conf:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

If there are multiple access points, a specific one can be selected by its SSID:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid your_ssid_here DHCP"
```
In an environment where there are multiple access points with the same SSID, which is often done to simplify roaming, it may be necessary to associate to one specific device. In this case, the BSSID of the access point can be specified, with or without the SSID:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid your_ssid_here bssid xx:xx:xx:xx:xx:xx DHCP"
```

There are other ways to constrain the choice of an access point, such as limiting the set of frequencies the system will scan on. This may be useful for a multi-band wireless card as scanning all the possible channels can be time-consuming. To limit operation to a specific band, use the `mode` parameter:

```
wlans_ath0="wlan0"
ifconfig_wlan0="mode 11g ssid your_ssid_here DHCP"
```

This example will force the card to operate in 802.11g, which is defined only for 2.4GHz frequencies so any 5GHz channels will not be considered. This can also be achieved with the `channel` parameter, which locks operation to one specific frequency, and the `chanlist` parameter, to specify a list of channels for scanning. More information about these parameters can be found in `ifconfig(8)`.

Once an access point is selected, the station needs to authenticate before it can pass data. Authentication can happen in several ways. The most common scheme, open authentication, allows any station to join the network and communicate. This is the authentication to use for test purposes the first time a wireless network is setup. Other schemes require cryptographic handshakes to be completed before data traffic can flow, either using pre-shared keys or secrets, or more complex schemes that involve backend services such as RADIUS. Open authentication is the default setting. The next most common setup is WPA-PSK, also known as WPA Personal, which is described in `WPA-PSK`.

If using an Apple™AirPort™ Extreme base station for an access point, shared-key authentication together with a WEP key needs to be configured. This can be configured in `/etc/rc.conf` or by using `wpa_supplicant(8)`. For a single AirPort™ base station, access can be configured with:

```
wlans_ath0="wlan0"
ifconfig_wlan0="authmode shared wepmode on weptxkey 1 wepkey 01234567 DHCP"
```

In general, shared key authentication should be avoided because it uses the WEP key material in a highly-constrained manner, making it even easier to crack the key. If WEP must be used for compatibility with legacy devices, it is better to use WEP with open authentication. More information regarding WEP can be found in `WEP`.

Once an access point is selected and the authentication parameters are set, an IP address must be obtained in order to communicate. Most of the time, the IP address is obtained via DHCP. To achieve that, edit `/etc/rc.conf` and add `DHCP` to the configuration for the device:
The wireless interface is now ready to bring up:

```
# service netif start
```

Once the interface is running, use `ifconfig(8)` to see the status of the interface `ath0`:

```
# ifconfig wlan0
wlan0:   flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:11:95:d5:43:62
        inet 192.168.1.100 netmask 0xffffff00 broadcast 192.168.1.255
        media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
        status: associated
        ssid dlinkap channel 11
            (2462 Mhz 11g)
        bssid 00:13:46:49:41:76
        country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
        scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
        roam:rate 5 protmode CTS wme burst
```

The `status: associated` line means that it is connected to the wireless network. The `bssid 00:13:46:49:41:76` is the MAC address of the access point and `authmode OPEN` indicates that the communication is not encrypted.

If an IP address cannot be obtained from a DHCP server, set a fixed IP address. Replace the `DHCP` keyword shown above with the address information. Be sure to retain any other parameters for selecting the access point:

```
wlans_ath0="wlan0"
ifconfig_wlan0="inet 192.168.1.100 netmask 255.255.255.0 ssid your_ssid_here"
```

### 3.3.4.1.2. WPA

Wi-Fi Protected Access (WPA) is a security protocol used together with 802.11 networks to address the lack of proper authentication and the weakness of WEP. WPA leverages the 802.1X authentication protocol and uses one of several ciphers instead of WEP for data integrity. The only cipher required by WPA is the Temporary Key Integrity Protocol (TKIP). TKIP is a cipher that extends the basic RC4 cipher used by WEP by adding integrity checking, tamper detection, and measures for responding to detected intrusions. TKIP is designed to work on legacy hardware with only software modification. It represents a compromise that improves security but is still not entirely immune to attack. WPA also specifies the AES-CCMP cipher as an alternative to TKIP, and that is preferred when possible. For this specification, the term WPA2 or RSN is commonly used.

WPA defines authentication and encryption protocols. Authentication is most commonly done using one of two techniques: by 802.1X and a backend authentication service such as RADIUS, or by a minimal handshake between the station and the access point using a pre-shared secret. The former is commonly termed WPA Enterprise and the latter is known as WPA Personal. Since most people will not set up a RADIUS backend server for their wireless network, WPA-PSK is by far the most popular.
The control of the wireless connection and the key negotiation or authentication with a server is done using `wpa_supplicant(8)`. This program requires a configuration file, `/etc/wpa_supplicant.conf`, to run. More information regarding this file can be found in `wpa_supplicant.conf(5)`.

### 31.3.4.1.3.1. WPA-PSK

WPA-PSK, also known as WPA Personal, is based on a pre-shared key (PSK) which is generated from a given password and used as the master key in the wireless network. This means every wireless user will share the same key. WPA-PSK is intended for small networks where the use of an authentication server is not possible or desired.

---

#### Always use strong passwords that are sufficiently long and made from a rich alphabet so that they will not be easily guessed or attacked.

The first step is the configuration of `/etc/wpa_supplicant.conf` with the SSID and the pre-shared key of the network:

```plaintext
network={
    ssid="freebsdap"
    psk="freebsdmall"
}
```

Then, in `/etc/rc.conf`, indicate that the wireless device configuration will be done with WPA and the IP address will be obtained with DHCP:

```plaintext
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Then, bring up the interface:

```bash
# service netif start
Starting wpa_supplicant.
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.0.1
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254
--
renewal in 300 seconds.

wlan0:
    flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:11:95:d5:43:62
    inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
    media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
    status: associated
    ssid freebsdap channel 1
    (2412 Mhz 11g)
    bssid 00:11:95:c3:0d:ac
country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
```
AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS

Or, try to configure the interface manually using the information in /etc/wpa_supplicant.conf:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
```

Trying to associate with 00:11:95:c3:0d:ac (SSID='freebsdap' freq=2412 MHz)

Associated with 00:11:95:c3:0d:ac
WPA: Key negotiation completed with 00:11:95:c3:0d:ac [PTK=CCMP GTK=CCMP]
CTRL-EVENT-CONNECTED - Connection to 00:11:95:c3:0d:ac completed (auth)
[id=0 id_str=]

The next operation is to launch dhclient(8) to get the IP address from the DHCP server:

```
# dhclient wlan0
```

DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.

```
# ifconfig wlan0
```

wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:11:95:d5:43:62
inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
status: associated ssid freebsdap channel 1 (2412 Mhz 11g)
bssid 00:11:95:c3:0d:ac
country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS

If /etc/rc.conf has an ifconfig_wlan0="DHCP" entry, dhclient(8) will be launched automatically after wpa_supplicant(8) associates with the access point.

If DHCP is not possible or desired, set a static IP address after wpa_supplicant(8) has authenticated the station:

```
# ifconfig wlan0 inet 192.168.0.100 netmask 255.255.255.0
```

```
# ifconfig wlan0
```

wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:11:95:d5:43:62
inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
status: associated

695
When DHCP is not used, the default gateway and the nameserver also have to be manually set:

```bash
# route add default your_default_router
# echo "nameserver your_DNS_server" >> /etc/resolv.conf
```

WPA

WPA Enterprise:

- EAP does not come with an encryption method. Instead, EAP is embedded inside an encrypted tunnel.
- Many EAP authentication methods exist, but EAP-TLS, EAP-TTLS, and EAP-PEAP are the most common.
- EAP with Transport Layer Security (EAP-TLS) is well-supported since it was the first EAP method to be certified by the Wi-Fi Alliance.
- Requires three certificates: the certificate of the Certificate Authority (CA) installed on all machines, the server certificate for the authentication server, and one client certificate for each wireless client.
- Both the authentication server and wireless client authenticate each other by presenting their respective certificates, and then verify that these certificates were signed by the organization's CA.

Configuration:

```bash
network=
    ssid="freebsdap"
    proto=RSN
    key_mgmt=WPA-EAP
    eap=TLS
    identity="loader"
    ca_cert="/etc/certs/cacert.pem"
    client_cert="/etc/certs/clientcert.pem"
    private_key="/etc/certs/clientkey.pem"
    private_key_passwd="freebsdmallclient"
}```

- `ssid`: network name (SSID).
- `proto`: RSN IEEE™ 802.11i protocol, also known as WPA2.
- `key_mgmt`: key management protocol used. WPA using EAP authentication.
- `eap`: EAP method for the connection.
- `identity`: identity string for EAP.

EAP-TLS:

- First EAP method to be certified by the Wi-Fi Alliance.
- Requires three certificates.
- Both the authentication server and wireless client authenticate each other by presenting their respective certificates, and then verify that these certificates were signed by the organization's CA.
The `ca_cert` field indicates the pathname of the CA certificate file. This file is needed to verify the server certificate.

The `client_cert` line gives the pathname to the client certificate file. This certificate is unique to each wireless client of the network.

The `private_key` field is the pathname to the client certificate private key file.

The `private_key_passwd` field contains the passphrase for the private key.

Then, add the following lines to `/etc/rc.conf`:

wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"

The next step is to bring up the interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPACK from 192.168.0.20
bound to 192.168.0.254
-- renewal in 300 seconds.
```

```
wlan0:
  flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
  ether 00:11:95:d5:43:62
  inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
  media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
  status: associated
  ssid freebsdap channel 1 (2412 Mhz 11g)
  bssid 00:11:95:c3:0d:ac
  country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
  AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
  bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
  wme burst roaming MANUAL
```

It is also possible to bring up the interface manually using `wpa_supplicant(8)` and `ifconfig(8)`.

### 31.3.4.1.3.3 WPA

With EAP-TTLS, a client certificate is optional. This method is similar to a web server which creates a secure SSL tunnel even if visitors do not have client-side certificates. EAP-TTLS uses an encrypted TLS tunnel for safe transport of the authentication data.

The required configuration can be added to `/etc/wpa_supplicant.conf`:

```
network=
    ssid="freebsdap"
    proto=RSN
    key_mgmt=WPA-EAP
    eap=TTLS
```

This field specifies the EAP method for the connection.

The `identity` field contains the identity string for EAP authentication inside the encrypted TLS tunnel.

The `password` field contains the passphrase for the EAP authentication.

The `ca_cert` field indicates the pathname of the CA certificate file. This file is needed to verify the server certificate.

This field specifies the authentication method used in the encrypted TLS tunnel. In this example, EAP with MD5-Challenge is used. The "inner authentication" phase is often called "phase2".

Next, add the following lines to `/etc/rc.conf`:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

The next step is to bring up the interface:

```
# service netif start
Starting wpa_supplicant.
```

```
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254
-- renewal in 300 seconds.
```

```
wlan0:
flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:11:95:d5:43:62
inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
status: associated
ssid freebsdap channel 1 (2412 Mhz 11g)
bssid 00:11:95:c3:0d:ac
country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
wme burst roaming MANUAL
```

31.3.4.1.3.4. WPA

EAP-PEAP

PEAPv0/EAP-MSCHAPv2 is the most common PEAP method. In this chapter, the term PEAP is used to refer to that method.

Protected EAP (PEAP) is designed as an alternative to EAP-TTLS and is the most used EAP standard.
In a network with mixed operating systems, PEAP should be the most supported standard after EAP-TLS.

PEAP is similar to EAP-TTLS as it uses a server-side certificate to authenticate clients by creating an encrypted TLS tunnel between the client and the authentication server, which protects the ensuing exchange of authentication information. PEAP authentication differs from EAP-TTLS as it broadcasts the username in the clear and only the password is sent in the encrypted TLS tunnel. EAP-TTLS will use the TLS tunnel for both the username and password.

Add the following lines to `/etc/wpa_supplicant.conf` to configure the EAP-PEAP related settings:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP
①
  identity="test"
②
  password="test"
③
  ca_cert="/etc/certs/cacert.pem"
④
  phase1="peaplabel=0"
⑤
  phase2="auth=MSCHAPV2"
⑥
}
```

① This field specifies the EAP method for the connection.
② The `identity` field contains the identity string for EAP authentication inside the encrypted TLS tunnel.
③ The `password` field contains the passphrase for the EAP authentication.
④ The `ca_cert` field indicates the pathname of the CA certificate file. This file is needed to verify the server certificate.
⑤ This field contains the parameters for the first phase of authentication, the TLS tunnel. According to the authentication server used, specify a specific label for authentication. Most of the time, the label will be "client EAP encryption" which is set by using `peaplabel=0`. More information can be found in `wpa_supplicant.conf(5)`.
⑥ This field specifies the authentication protocol used in the encrypted TLS tunnel. In the case of PEAP, it is `auth=MSCHAPV2`.

將以下參數加到 `/etc/rc.conf`:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Then, bring up the interface:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
```

699
DHCPACK from 192.168.0.20 bound to 192.168.0.254

-- renewal in 300 seconds.

wlan0: flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
ether 00:11:95:d5:43:62
inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
status: associated
ssid freebsdap channel 1 (2412 Mhz 11g)
bssid 00:11:95:c3:0d:ac
country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS wme burst roaming MANUAL

WEP

Wired Equivalent Privacy (WEP) is part of the original 802.11 standard. There is no authentication mechanism, only a weak form of access control which is easily cracked.

WEP can be set up using `ifconfig(8)`:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 inet 192.168.1.100 netmask 255.255.255.0 
  ssid my_net wepmode on weptxkey 3 wepkey 3:0x3456789012
```

• The `weptxkey` specifies which WEP key will be used in the transmission. This example uses the third key. This must match the setting on the access point. When unsure which key is used by the access point, try `1` (the first key) for this value.

• The `wepkey` selects one of the WEP keys. It should be in the format `index:key`. Key 1 is used by default; the index only needs to be set when using a key other than the first key.

Replace the `0x3456789012` with the key configured for use on the access point.

Refer to `ifconfig(8)` for further information.

The `wpa_supplicant(8)` facility can be used to configure a wireless interface with WEP. The example above can be set up by adding the following lines to `/etc/wpa_supplicant.conf`:

```
network=
  ssid="my_net"
  key_mgmt=NONE
  wep_key3=3456789012
  wep_tx_keyidx=3
}
```

Then:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
```
Trying to associate with 00:13:46:49:41:76

(SSID = 'dlinkap', freq = 2437 MHz)

Associated with 00:13:46:49:41:76

IBSS mode, also called ad-hoc mode, is designed for point to point connections. For example, to establish an ad-hoc network between the machines A and B, choose two IP addresses and a SSID.

On A:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
```

```
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether 00:11:95:c3:0d:ac
inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
status: running
ssid freebsdap channel 2 (2417 Mhz 11g)
bssid 02:11:95:c3:0d:ac
country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
protmode CTS wme burst
```

The adhoc parameter indicates that the interface is running in IBSS mode.

B should now be able to detect A:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
freebsdap       02:11:95:c3:0d:ac    2   54M -64:-96  100 IS   WME
```

The I in the output confirms that A is in ad-hoc mode. Now, configure B with a different IP address:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
```

```
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether 00:11:95:d5:43:62
inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
status: running
ssid freebsdap channel 2 (2417 Mhz 11g)
bssid 02:11:95:c3:0d:ac
country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
protmode CTS wme burst
```

Both A and B are now ready to exchange information.
FreeBSD can act as an Access Point (AP) which eliminates the need to buy a hardware AP or run an ad-hoc network. This can be particularly useful when a FreeBSD machine is acting as a gateway to another network such as the Internet.

31.3.6.1. 基礎設定

Before configuring a FreeBSD machine as an AP, the kernel must be configured with the appropriate networking support for the wireless card as well as the security protocols being used. For more details, see 基礎設定.

The NDIS driver wrapper for Windows™ drivers does not currently support AP operation. Only native FreeBSD wireless drivers support AP mode.

Once wireless networking support is loaded, check if the wireless device supports the host-based access point mode, also known as hostap mode:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 list caps
drivercaps = 6f85edc1<STA,FF,TURBOP,IBSS,HOSTAP,AHDEMO,TXPMGT,SHSLOT,SHPREAMBLE,MONITOR,MBSS,WPA1,WPA2,BURST,WME,WDS,BGSCAN,TXFRAG>
cryptocaps = 1f<WEP,TKIP,AES,AES_CCM,TKIPMIC>
```

This output displays the card's capabilities. The HOSTAP word confirms that this wireless card can act as an AP. Various supported ciphers are also listed: WEP, TKIP, and AES. This information indicates which security protocols can be used on the AP.

The wireless device can only be put into hostap mode during the creation of the network pseudo-device, so a previously created device must be destroyed first:

```
# ifconfig wlan0 destroy
```

then regenerated with the correct option before setting the other parameters:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel 1
```

Use ifconfig(8) again to see the status of the wlan0 interface:

```
# ifconfig wlan0
wlan0: flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  ether 00:11:95:c3:0d:ac
  inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
  media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
status: running
  ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
```
The *hostap* parameter indicates the interface is running in the host-based access point mode. The interface configuration can be done automatically at boot time by adding the following lines to `/etc/rc.conf`:

```bash
wlans_ath0="wlan0"
create_args_wlan0="wlanmode hostap"
ifconfig_wlan0="inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel 1"
```

Although it is not recommended to run an AP without any authentication or encryption, this is a simple way to check if the AP is working. This configuration is also important for debugging client issues.

Once the AP is configured, initiate a scan from another wireless machine to find the AP:

```bash
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
```

<table>
<thead>
<tr>
<th>SSID/MESH ID</th>
<th>BSSID</th>
<th>CHAN</th>
<th>RATE</th>
<th>S:N</th>
<th>INT</th>
<th>CAPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>freebsdap</td>
<td>00:11:95:c3:0d:ac</td>
<td>1</td>
<td>54M</td>
<td>-66</td>
<td>-96</td>
<td>ES</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WME</td>
</tr>
</tbody>
</table>

The client machine found the AP and can be associated with it:

```bash
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
```

```
<wlan0:
  flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  ether 00:11:95:d5:43:62
  inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
  media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
  status: associated
  ssid freebsdap channel 1 (2412 Mhz 11g)
  bssid 00:11:95:c3:0d:ac
country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
roam:rate 5 protmode CTS wme burst>
```

This section focuses on setting up a FreeBSD access point using the WPA2 security protocol. More details regarding WPA and the configuration of WPA-based wireless clients can be found in the [WPA documentation](#).
Before trying to configure `hostapd(8)`, first configure the basic settings introduced in `基礎設定`.

### 31.3.6.3.1 WPA2-PSK

WPA2-PSK is intended for small networks where the use of a backend authentication server is not possible or desired.

The configuration is done in `/etc/hostapd.conf`:

1. `interface=wlan0`
2. `debug=1`
3. `ctrl_interface=/var/run/hostapd`
4. `ctrl_interface_group=wheel`
5. `ssid=freebsdap`
6. `wpa=2`
7. `wpa_passphrase=freebsdmall`
8. `wpa_key_mgmt=WPA-PSK`
9. `wpa_pairwise=CCMP`

1. **Wireless interface used for the access point.**
2. **Level of verbosity used during the execution of `hostapd(8)`.** A value of 1 represents the minimal level.
3. **Pathname of the directory used by `hostapd(8)` to store domain socket files for communication with external programs such as `hostapd_cli(8)`.** The default value is used in this example.
4. **The group allowed to access the control interface files.**
5. **The wireless network name, or SSID, that will appear in wireless scans.**
6. **Enable WPA and specify which WPA authentication protocol will be required.** A value of 2 configures the AP for WPA2 and is recommended. Set to 1 only if the obsolete WPA is required.
7. **ASCII passphrase for WPA authentication.**
8. **The key management protocol to use.** This example sets WPA-PSK.
9. **Encryption algorithms accepted by the access point.** In this example, only the CCMP (AES) cipher is accepted. CCMP is an alternative to TKIP and is strongly preferred when possible. TKIP should be allowed only when there are stations incapable of using CCMP.

The next step is to start `hostapd(8)`:

```
# service hostapd forcestart
# ifconfig wlan0
```

```
wlan0:
  flags = 8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> metric 0
  mtu 1500
  ether 04:f0:21:16:8e:10
  inet6 fe80::6f0:21ff:fe16:8e10%wlan0 prefixlen 64 scopeid 0x9
704
```
Once the AP is running, the clients can associate with it. See WPA for more details. It is possible to see the stations associated with the AP using `ifconfig wlan0 list sta`.

31.3.6.4. WEP Host-based

It is not recommended to use WEP for setting up an AP since there is no authentication mechanism and the encryption is easily cracked. Some legacy wireless cards only support WEP and these cards will only support an AP without authentication or encryption.

The wireless device can now be put into hostap mode and configured with the correct SSID and IP address:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 
  ssid freebsdap wepmode on weptxkey 3 wepkey 3:0x3456789012 mode 11g
```

- The `weptxkey` indicates which WEP key will be used in the transmission. This example uses the third key as key numbering starts with 1. This parameter must be specified in order to encrypt the data.
- The `wepkey` sets the selected WEP key. It should be in the format `index:key`. If the index is not given, key 1 is set. The index needs to be set when using keys other than the first key.

Use `ifconfig(8)` to see the status of the `wlan0` interface:

```
# ifconfig wlan0
wlan0:  flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  ether 00:11:95:c3:0d:ac
  inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
  media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
  status: running
  ssid freebsdap channel 4
    (2427 Mhz 11g)
  bssid 00:11:95:c3:0d:ac
  country US ecm authmode OPEN privacy ON deftxkey 3 wepkey 3:40-bit
  txpower 21.5 scanvalid 60 protmode CTS wme burst dtimperiod 1
```

From another wireless machine, it is now possible to initiate a scan to find the AP:

```
# ifconfig wlan0 create wlandev ath0
```
In this example, the client machine found the AP and can associate with it using the correct parameters. See WEP for more details.

31.3.7. 同時使用有線及無線連線

A wired connection provides better performance and reliability, while a wireless connection provides flexibility and mobility. Laptop users typically want to roam seamlessly between the two types of connections.

On FreeBSD, it is possible to combine two or even more network interfaces together in a "failover" fashion. This type of configuration uses the most preferred and available connection from a group of network interfaces, and the operating system switches automatically when the link state changes.

Link aggregation and failover is covered in Link Aggregation and an example for using both wired and wireless connections is provided at 乙太網路與無線介面間的容錯移轉模式.

31.3.8. 疑難排解

This section describes a number of steps to help troubleshoot common wireless networking problems.

• If the access point is not listed when scanning, check that the configuration has not limited the wireless device to a limited set of channels.
• If the device cannot associate with an access point, verify that the configuration matches the settings on the access point. This includes the authentication scheme and any security protocols. Simplify the configuration as much as possible. If using a security protocol such as WPA or WEP, configure the access point for open authentication and no security to see if traffic will pass.

Debugging support is provided by wpa_supplicant(8). Try running this utility manually with -dd and look at the system logs.

• Once the system can associate with the access point, diagnose the network configuration using tools like ping(8).
• There are many lower-level debugging tools. Debugging messages can be enabled in the 802.11 protocol support layer using wlandebug(8). For example, to enable console messages related to scanning for access points and the 802.11 protocol handshakes required to arrange communication:

```bash
# wlandebug -i wlan0 +scan+auth+debug+assoc
```

Many useful statistics are maintained by the 802.11 layer and wlanstats, found in /usr/src/tools/tools/net80211, will dump this information. These statistics should display all errors identified by the 802.11 layer. However, some errors are identified in the device drivers that lie below the 802.11 layer so they may not show up. To diagnose device-specific problems, refer to the drivers' documentation.

If the above information does not help to clarify the problem, submit a problem report and include output from the above tools.
Many cellphones provide the option to share their data connection over USB (often called "tethering"). This feature uses either the RNDIS, CDC or a custom Apple™iPhone™/iPad™ protocol.

- Android™ devices generally use the `urndis(4)` driver.
- Apple™ devices use the `ipheth(4)` driver.
- Older devices will often use the `cdce(4)` driver.

Before attaching a device, load the appropriate driver into the kernel:

```
# kldload if_urndis
# kldload if_cdce
# kldload if_ipheth
```

Once the device is attached `ue0` will be available for use like a normal network device. Be sure that the "USB tethering" option is enabled on the device.

### 31.5. 藍牙
Bluetooth is a wireless technology for creating personal networks operating in the 2.4 GHz unlicensed band, with a range of 10 meters. Networks are usually formed ad-hoc from portable devices such as cellular phones, handhelds, and laptops. Unlike Wi-Fi wireless technology, Bluetooth offers higher level service profiles, such as FTP-like file servers, file pushing, voice transport, serial line emulation, and more.

This section describes the use of a USB Bluetooth dongle on a FreeBSD system. It then describes the various Bluetooth protocols and utilities.

#### 31.5.1. 載入藍牙支援
The Bluetooth stack in FreeBSD is implemented using the `netgraph(4)` framework. A broad variety of Bluetooth USB dongles is supported by `ng_ubt(4)`. Broadcom BCM2033 based Bluetooth devices are supported by the `ubtbcmfw(4)` and `ng_ubt(4)` drivers. The 3Com Bluetooth PC Card 3CRWB60-A is supported by the `ng_bt3c(4)` driver. Serial and UART based Bluetooth devices are supported by `sio(4)`, `ng_h4(4)`, and `hcseriald(8)`.

Before attaching a device, determine which of the above drivers it uses, then load the driver. For example, if the device uses the `ng_ubt(4)` driver:

```
# kldload ng_ubt
```

If the Bluetooth device will be attached to the system during system startup, the system can be configured to load the module at boot time by adding the driver to `/boot/loader.conf`:

```
ng_ubt_load="YES"
```

Once the driver is loaded, plug in the USB dongle. If the driver load was successful, output similar to the following should appear on the console and in `/var/log/messages`:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints:
        interrupt = 0x81, bulk-in = 0x82, bulk-out = 0x2
```

7070
To start and stop the Bluetooth stack, use its startup script. It is a good idea to stop the stack before unplugging the device. Starting the bluetooth stack might require `hcsecd(8)` to be started. When starting the stack, the output should be similar to the following:

```
# service bluetooth start ubt0
```

```
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

### 31.5.2. 寻找其他蓝牙装置

The Host Controller Interface (HCI) provides a uniform method for accessing Bluetooth baseband capabilities. In FreeBSD, a netgraph HCI node is created for each Bluetooth device. For more details, refer to `ng_hci(4)`.

One of the most common tasks is discovery of Bluetooth devices within RF proximity. This operation is called inquiry. Inquiry and other HCI related operations are done using `hccontrol(8)`.

The example below shows how to find out which Bluetooth devices are in range. The list of devices should be displayed in a few seconds. Note that a remote device will only answer the inquiry if it is set to discoverable mode.

```
% hccontrol -n ubt0hci inquiry
```

```
Inquiry result, num_responses = 1
Inquiry result #0

BD_ADDR: 00:80:37:29:19:a4
Page Scan Rep. Mode: 0x1
Page Scan Period Mode: 00
Page Scan Mode: 00
Class: 52:02:04
Clock offset: 0x78ef
Inquiry complete. Status: No error
```

The `BD_ADDR` is the unique address of a Bluetooth device, similar to the MAC address of a network card. This address is needed for further communication with a device and it is possible to assign a human readable name to a `BD_ADDR`. Information regarding the known Bluetooth hosts is contained in `/etc/bluetooth/hosts`. The following example shows how to obtain the human readable name that was assigned to the remote device:
If an inquiry is performed on a remote Bluetooth device, it will find the computer as "your.host.name (ubt0)". The name assigned to the local device can be changed at any time. Remote devices can be assigned aliases in /etc/bluetooth/hosts. More information about /etc/bluetooth/hosts file might be found in bluetooth.hosts(5).

The Bluetooth system provides a point-to-point connection between two Bluetooth units, or a point-to-multipoint connection which is shared among several Bluetooth devices. The following example shows how to create a connection to a remote device:

```
% hccontrol -n ubt0hci create_connection BT_ADDR
```

create_connection accepts BT_ADDR as well as host aliases in /etc/bluetooth/hosts.

The following example shows how to obtain the list of active baseband connections for the local device:

```
% hccontrol -n ubt0hci read_connection_list
```

Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN

A connection handle is useful when termination of the baseband connection is required, though it is normally not required to do this by hand. The stack will automatically terminate inactive baseband connections.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Type hccontrol help for a complete listing of available HCI commands. Most of the HCI commands do not require superuser privileges.

31.5.3. 裝置配對

By default, Bluetooth communication is not authenticated, and any device can talk to any other device. A Bluetooth device, such as a cellular phone, may choose to require authentication to provide a particular service. Bluetooth authentication is normally done with a PIN code, an ASCII string up to 16 characters in length. The user is required to enter the same PIN code on both devices. Once the user has entered the PIN code, both devices will generate a link key. After that, the link key can be stored either in the devices or in a persistent storage. Next time, both devices will use the previously generated link key. This procedure is called pairing. Note that if the link key is lost by either device, the pairing must be repeated.

The hcsecd(8) daemon is responsible for handling Bluetooth authentication requests. The default configuration file is /etc/bluetooth/hcsecd.conf. An example section for a cellular phone with the PIN code set to 1234 is shown below:

```
```
bdaddr 00:80:37:29:19:a4;
name  "Pav's T39";
key nokey;
pin "1234";

The only limitation on PIN codes is length. Some devices, such as Bluetooth headsets, may have a fixed PIN code built in. The -d switch forces hcsecd(8) to stay in the foreground, so it is easy to see what is happening. Set the remote device to receive pairing and initiate the Bluetooth connection to the remote device. The remote device should indicate that pairing was accepted and request the PIN code. Enter the same PIN code listed in hcsecd.conf. Now the computer and the remote device are paired. Alternatively, pairing can be initiated on the remote device.

The following line can be added to /etc/rc.conf to configure hcsecd(8) to start automatically on system start:

hcsecd_enable="YES"

The following is a sample of the hcsecd(8) daemon output:

hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', PIN code exists
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4

31.5.4.

使用PPP Profile存储网络

A Dial-Up Networking (DUN) profile can be used to configure a cellular phone as a wireless modem for connecting to a dial-up Internet access server. It can also be used to configure a computer to receive data calls from a cellular phone.

Network access with a PPP profile can be used to provide LAN access for a single Bluetooth device or multiple Bluetooth devices. It can also provide PC to PC connection using PPP networking over serial cable emulation.

In FreeBSD, these profiles are implemented with ppp(8) and the rfcomm_pppd(8) wrapper which converts a Bluetooth connection into something PPP can use. Before a profile can be used, a new PPP label must be created in /etc/ppp/ppp.conf. Consult rfcomm_pppd(8) for examples.

In this example, rfcomm_pppd(8) is used to open a connection to a remote device with a BD_ADDR of 00:80:37:29:19:a4 on a DUNRFCOMM channel:
The actual channel number will be obtained from the remote device using the SDP protocol. It is possible to specify the RFCOMM channel by hand, and in this case \texttt{rfcomm_pppd(8)} will not perform the SDP query. Use \texttt{sdpcontrol(8)} to find out the RFCOMM channel on the remote device.

In order to provide network access with the PPPLAN service, \texttt{sdpd(8)} must be running and a new entry for LAN clients must be created in \texttt{/etc/ppp/ppp.conf}. Consult \texttt{rfcomm_pppd(8)} for examples.

Finally, start the RFCOMMPPP server on a valid RFCOMM channel number. The RFCOMMPPP server will automatically register the Bluetooth LAN service with the local SDP daemon. The example below shows how to start the RFCOMMPPP server.

\begin{verbatim}
# rfcomm_pppd -s -C 7 -l rfcomm-server
\end{verbatim}

31.5.5. Bluetooth

This section provides an overview of the various Bluetooth protocols, their function, and associated utilities.

31.5.5.1. Logical Link Control and Adaptation Protocol (L2CAP)

The Logical Link Control and Adaptation Protocol (L2CAP) provides connection-oriented and connectionless data services to upper layer protocols. L2CAP permits higher level protocols and applications to transmit and receive L2CAP data packets up to 64 kilobytes in length.

L2CAP is based around the concept of \textit{channels}. A channel is a logical connection on top of a baseband connection, where each channel is bound to a single protocol in a many-to-one fashion. Multiple channels can be bound to the same protocol, but a channel cannot be bound to multiple protocols. Each L2CAP packet received on a channel is directed to the appropriate higher level protocol. Multiple channels can share the same baseband connection.

In FreeBSD, a netgraph L2CAP node is created for each Bluetooth device. This node is normally connected to the downstream Bluetooth HCI node and upstream Bluetooth socket nodes. The default name for the L2CAP node is "devicel2cap". For more details refer to \texttt{ng_l2cap(4)}.

A useful command is \texttt{l2ping(8)}, which can be used to ping other devices. Some Bluetooth implementations might not return all of the data sent to them, so 0 bytes in the following example is normal.

\begin{verbatim}
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4
  seq_no = 0
  time = 48.633 ms
  result = 0

0 bytes from 0:80:37:29:19:a4
  seq_no = 1
  time = 37.551 ms
  result = 0

0 bytes from 0:80:37:29:19:a4
  seq_no = 2
  time = 28.324 ms
  result = 0

0 bytes from 0:80:37:29:19:a4
  seq_no = 3
  time = 46.150 ms
  result = 0
\end{verbatim}

The \texttt{l2control(8)} utility is used to perform various operations on L2CAP nodes. This example shows how to obtain the list of logical connections (channels) and the list of baseband connections for the local device:

\begin{verbatim}
% l2control
-a
00:02:72:00:d4:1a read_channel_list
L2CAP channels:
   Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
      711
\end{verbatim}
Another diagnostic tool is `btsockstat(1)`. It is similar to `netstat(1)`, but for Bluetooth network-related data structures. The example below shows the same logical connection as `l2control(8)` above.

```
% btsockstat
```

### Active L2CAP sockets

<table>
<thead>
<tr>
<th>PCB</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local address/PSM</th>
<th>Foreign address</th>
<th>CID</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2afe900</td>
<td>0</td>
<td>0</td>
<td>00:02:72:00:d4:1a/3</td>
<td>00:07:e0:00:0b:ca</td>
<td>66</td>
<td>OPEN</td>
</tr>
</tbody>
</table>

### Active RFCOMM sessions

<table>
<thead>
<tr>
<th>L2PCB</th>
<th>PCB</th>
<th>Flag</th>
<th>MTU</th>
<th>Out-Q</th>
<th>DLCs</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2afe900</td>
<td>c2b53380</td>
<td>1</td>
<td>127</td>
<td>0</td>
<td>Yes</td>
<td>OPEN</td>
</tr>
</tbody>
</table>

### Active RFCOMM sockets

<table>
<thead>
<tr>
<th>PCB</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local address</th>
<th>Foreign address</th>
<th>Chan</th>
<th>DLCI</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>c2e8bc80</td>
<td>0</td>
<td>250</td>
<td>00:02:72:00:d4:1a</td>
<td>00:07:e0:00:0b:ca</td>
<td>3</td>
<td>6</td>
<td>OPEN</td>
</tr>
</tbody>
</table>

31.5.5.2. Radio Frequency Communication (RFCOMM)

The RFCOMM protocol provides emulation of serial ports over the L2CAP protocol. RFCOMM is a simple transport protocol, with additional provisions for emulating the 9 circuits of RS-232 (EIATIA-232-E) serial ports. It supports up to 60 simultaneous connections (RFCOMM channels) between two Bluetooth devices.

For the purposes of RFCOMM, a complete communication path involves two applications running on the communication endpoints with a communication segment between them. RFCOMM is intended to cover applications that make use of the serial ports of the devices in which they reside. The communication segment is a direct connect Bluetooth link from one device to another. RFCOMM is only concerned with the connection between the devices in the direct connect case, or between the device and a modem in the network case. RFCOMM can support other configurations, such as modules that communicate via Bluetooth wireless technology on one side and provide a wired interface on the other side.

In FreeBSD, RFCOMM is implemented at the Bluetooth sockets layer.

31.5.5.3. Service Discovery Protocol (SDP)

The Service Discovery Protocol (SDP) provides the means for client applications to discover the existence of services provided by server applications as well as the attributes of those services. The attributes of a service include the type or class of service offered and the mechanism or protocol information needed to utilize the service.

SDP involves communication between a SDP server and a SDP client. The server maintains a list of service records that describe the characteristics of services associated with the server. Each service record contains information about a single service. A client may retrieve information from a service record maintained by the SDP server by issuing a SDP request. If the client, or an application associated with the client, decides to use a service, it must open a separate connection to the service provider in order to utilize the service. SDP provides a mechanism for discovering services and their attributes, but it does not provide a mechanism for utilizing those services.
Normally, a SDP client searches for services based on some desired characteristics of the services. However, there are times when it is desirable to discover which types of services are described by an SDP server's service records without any prior information about the services. This process of looking for any offered services is called **browsing**.

The Bluetooth SDP server, `sdpd(8)`, and command line client, `sdpcontrol(8)`, are included in the standard FreeBSD installation. The following example shows how to perform a SDP browse query.

```sh
% sdpcontrol -a 00:01:03:fc:6e:ec browse
```

Record Handle: 00000000
Service Class ID List:
- Service Discovery Server (0x1000)
Protocol Descriptor List:
- L2CAP (0x0100)
  - Protocol specific parameter #1: u/int/uuid16 1
  - Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
- Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
- LAN Access Using PPP (0x1102)
  Protocol Descriptor List:
- L2CAP (0x0100)
- RFCOMM (0x0003)
  - Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
- LAN Access Using PPP (0x1102) ver. 1.0

Note that each service has a list of attributes, such as the RFCOMM channel. Depending on the service, the user might need to make note of some of the attributes. Some Bluetooth implementations do not support service browsing and may return an empty list. In this case, it is possible to search for the specific service. The example below shows how to search for the OBEX Object Push (OPUSH) service:

```sh
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Offering services on FreeBSD to Bluetooth clients is done with the `sdpd(8)` server. The following line can be added to `/etc/rc.conf`:

```conf
sdpd_enable="YES"
```

Then the `sdpd(8)` daemon can be started with:
The local server application that wants to provide a Bluetooth service to remote clients will register
the service with the local SDP daemon. An example of such an application is
rfcomm_pppd(8). Once
started, it will register the Bluetooth LAN service with the local SDP daemon.

The list of services registered with the local SDP server can be obtained by issuing a SDP browse
query via the local control channel:

```
# sdpcontrol -l browse
```

31.5.5.4. OBEX Object Push (OPUSH)

Object Exchange (OBEX) is a widely used protocol for simple file transfers between mobile devices.
Its main use is in infrared communication, where it is used for generic file transfers between
notebooks or PDAs, and for sending business cards or calendar entries between cellular phones and
other devices with Personal Information Manager (PIM) applications.

The OBEX server and client are implemented by obexapp, which can be installed using the
comms/obexapp package or port.

The OBEX client is used to push and/or pull objects from the OBEX server. An example object is a
business card or an appointment. The OBEX client can obtain the RFCOMM channel number from
the remote device via SDP. This can be done by specifying the service name instead of the RFCOMM
channel number. Supported service names are:

- IrMC
- FTRN
- OPUSH

It is also possible to
specify the RFCOMM channel as a number. Below is an example of an OBEX session where the
device information object is pulled from the cellular phone, and a new object, the business card, is
pushed into the phone's directory.

```
% obexapp
    -a 00:80:37:29:19:a4
    -C IrMC

obex> get telecom/devinfo.txt devinfo-t39.txt
Success, response: OK, Success (0x20)

obex> put new.vcf
Success, response: OK, Success (0x20)

obex> di
Success, response: OK, Success (0x20)
```

In order to provide the OPUSH service, sdpd(8) must be running and a root folder, where all
incoming objects will be stored, must be created. The default path to the root folder is
/var/spool/obex. Finally, start the OBEX server on a valid RFCOMM channel number. The OBEX
server will automatically register the OPUSH service with the local SDP daemon. The example
below shows how to start the OBEX server.

```
# obexapp -s -C 10
```

31.5.5.5. Serial Port Profile (SPP)

The Serial Port Profile (SPP) allows Bluetooth devices to perform serial cable emulation. This
profile allows legacy applications to use Bluetooth as a cable replacement, through a virtual serial
port abstraction.

In FreeBSD, rfcomm_sppd(1) implements SPP and a pseudo tty is used as a virtual serial port
abstraction. The example below shows how to connect to a remote device's serial port service. A RFCOMM channel does not have to be specified as `rfcomm_sppd(1)` can obtain it from the remote device via SDP. To override this, specify a RFCOMM channel on the command line.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t
rfcomm_sppd[94692]: Starting on /dev/pts/6...
```

Once connected, the pseudo tty can be used as serial port:

```
# cu -l /dev/pts/6
```

The pseudo tty is printed on stdout and can be read by wrapper scripts:

```
PTS=`rfcomm_sppd -a 00:07:E0:00:0B:CA -t`
cu -l $PTS
```

31.5.6. 疑難排解

By default, when FreeBSD is accepting a new connection, it tries to perform a role switch and become master. Some older Bluetooth devices which do not support role switching will not be able to connect. Since role switching is performed when a new connection is being established, it is not possible to ask the remote device if it supports role switching. However, there is a HCI option to disable role switching on the local side:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

To display Bluetooth packets, use the third-party package hcidump, which can be installed using the `comms/hcidump` package or port. This utility is similar to `tcpdump(1)` and can be used to display the contents of Bluetooth packets on the terminal and to dump the Bluetooth packets to a file.

31.6. 橋接

It is sometimes useful to divide a network, such as an Ethernet segment, into network segments without having to create IP subnets and use a router to connect the segments together. A device that connects two networks together in this fashion is called a "bridge." A bridge works by learning the MAC addresses of the devices on each of its network interfaces. It forwards traffic between networks only when the source and destination MAC addresses are on different networks. In many respects, a bridge is like an Ethernet switch with very few ports. A FreeBSD system with multiple network interfaces can be configured to act as a bridge.

Bridging can be useful in the following situations:

### Connecting Networks

The basic operation of a bridge is to join two or more network segments. There are many reasons to use a host-based bridge instead of networking equipment, such as cabling constraints or firewalling. A bridge can also connect a wireless interface running in hostap mode to a wired network and act as an access point.
Filtering/Traffic Shaping Firewall

A bridge can be used when firewall functionality is needed without routing or Network Address Translation (NAT). An example is a small company that is connected via DSL or ISDN to an ISP. There are thirteen public IP addresses from the ISP and ten computers on the network. In this situation, using a router-based firewall is difficult because of subnetting issues. A bridge-based firewall can be configured without any IP addressing issues.

Network Tap

A bridge can join two network segments in order to inspect all Ethernet frames that pass between them using `bpf(4)` and `tcpdump(1)` on the bridge interface or by sending a copy of all frames out an additional interface known as a span port.

Layer 2 VPN

Two Ethernet networks can be joined across an IP link by bridging the networks to an EtherIP tunnel or a `tap(4)` based solution such as OpenVPN.

Layer 2 Redundancy

A network can be connected together with multiple links and use the Spanning Tree Protocol (STP) to block redundant paths.

This section describes how to configure a FreeBSD system as a bridge using `if_bridge(4)`. A netgraph bridging driver is also available, and is described in `ng_bridge(4)`.

Packet filtering can be used with any firewall package that hooks into the `pfil(9)` framework. The bridge can be used as a traffic shaper with `altq(4)` or `dummynet(4)`.

In FreeBSD, `if_bridge(4)` is a kernel module which is automatically loaded by `ifconfig(8)` when creating a bridge interface. It is also possible to compile bridge support into a custom kernel by adding `device if_bridge` to the custom kernel configuration file.

The bridge is created using interface cloning. To create the bridge interface:

```
# ifconfig bridge create
bridge0
# ifconfig bridge0
bridge0:
  flags = 8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
  ether 96:3d:4b:f1:79:7a
  id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
  maxage 20 holdcnt 6 proto rstp maxaddr 100
  timeout 1200
  root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
```

When a bridge interface is created, it is automatically assigned a randomly generated Ethernet address. The `maxaddr` and `timeout` parameters control how many MAC addresses the bridge will keep in its forwarding table and how many seconds before each entry is removed after it is last seen. The other parameters control how STP operates.

Next, specify which network interfaces to add as members of the bridge. For the bridge to forward packets, all member interfaces and the bridge need to be up:
The bridge can now forward Ethernet frames between `fxp0` and `fxp1`. Add the following lines to `/etc/rc.conf` so the bridge is created at startup:

```
cloned_interfaces="bridge0"
ifconfig_bridge0="addm fxp0 addm fxp1 up"
ifconfig_fxp0="up"
ifconfig_fxp1="up"
```

If the bridge host needs an IP address, set it on the bridge interface, not on the member interfaces. The address can be set statically or via DHCP. This example sets a static IP address:

```
# ifconfig bridge0 inet 192.168.0.1/24
```

It is also possible to assign an IPv6 address to a bridge interface. To make the changes permanent, add the addressing information to `/etc/rc.conf`.

When packet filtering is enabled, bridged packets will pass through the filter inbound on the originating interface on the bridge interface, and outbound on the appropriate interfaces. Either stage can be disabled. When direction of the packet flow is important, it is best to firewall on the member interfaces rather than the bridge itself.

The bridge has several configurable settings for passing non-IP and IP packets, and layer2 firewalling with `ipfw(8)`. See `if_bridge(4)` for more information.

31.6.2. 開啟 Spanning Tree

For an Ethernet network to function properly, only one active path can exist between two devices. The STP protocol detects loops and puts redundant links into a blocked state. Should one of the active links fail, STP calculates a different tree and enables one of the blocked paths to restore connectivity to all points in the network.

The Rapid Spanning Tree Protocol (RSTP or 802.1w) provides backwards compatibility with legacy STP. RSTP provides faster convergence and exchanges information with neighboring switches to quickly transition to forwarding mode without creating loops. FreeBSD supports RSTP and STP as operating modes, with RSTP being the default mode.

STP can be enabled on member interfaces using `ifconfig(8)`. For a bridge with `fxp0` and `fxp1` as the current interfaces, enable STP with:

```
# ifconfig bridge0 stp fxp0 stp fxp1
```

```
bridge0:
flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether d6:cf:d5:a0:94:6d
id 00:01:02:4b:d4:50 priority 32768 hellotime 2 fwddelay 15
maxage 20 holdcnt 6 proto rstp maxaddr 100
```

```
root id 00:01:02:4b:d4:50 priority 32768 ifcost 0 port 0
```

```
717
```
This bridge has a spanning tree ID of `00:01:02:4b:d4:50` and a priority of `32768`. As the root id is the same, it indicates that this is the root bridge for the tree.

Another bridge on the network also has STP enabled:

```
bridge0:
flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
ether 96:3d:4b:f1:79:7a
id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15
maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4
member: fxp0
flags = 1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
port 4 priority 128 path cost 200000 proto rstp
role root state forwarding
member: fxp1
flags = 1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
port 5 priority 128 path cost 200000 proto rstp
role designated state forwarding
```

The line `root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4` shows that the root bridge is `00:01:02:4b:d4:50` and has a path cost of `400000` from this bridge. The path to the root bridge is via port `4` which is `fxp0`.

31.6.3. 橋接介面參數

Several ifconfig parameters are unique to bridge interfaces. This section summarizes some common uses for these parameters. The complete list of available parameters is described in ifconfig(8).

**private**

A private interface does not forward any traffic to any other port that is also designated as a private interface. The traffic is blocked unconditionally so no Ethernet frames will be forwarded, including ARP packets. If traffic needs to be selectively blocked, a firewall should be used instead.

**span**

A span port transmits a copy of every Ethernet frame received by the bridge. The number of span ports configured on a bridge is unlimited, but if an interface is designated as a span port, it cannot also be used as a regular bridge port. This is most useful for snooping a bridged network passively on another host connected to one of the span ports of the bridge. For example, to send a copy of all frames out the interface named `fxp4`:

```
# ifconfig bridge0 span fxp4
```
If a bridge member interface is marked as sticky, dynamically learned address entries are treated as static entries in the forwarding cache. Sticky entries are never aged out of the cache or replaced, even if the address is seen on a different interface. This gives the benefit of static address entries without the need to pre-populate the forwarding table. Clients learned on a particular segment of the bridge cannot roam to another segment.

An example of using sticky addresses is to combine the bridge with VLANs in order to isolate customer networks without wasting IP address space. Consider that CustomerA is on vlan100, CustomerB is on vlan101, and the bridge has the address 192.168.0.1:

```
# ifconfig bridge0 addm vlan100 sticky
# ifconfig bridge0 addm vlan101 sticky
# ifconfig bridge0 inet 192.168.0.1/24
```

In this example, both clients see 192.168.0.1 as their default gateway. Since the bridge cache is sticky, one host cannot spoof the MAC address of the other customer in order to intercept their traffic.

Any communication between the VLANs can be blocked using a firewall or, as seen in this example, private interfaces:

```
# ifconfig bridge0 private vlan100
# ifconfig bridge0 private vlan101
```

The customers are completely isolated from each other and the full /24 address range can be allocated without subnetting.

The number of unique source MAC addresses behind an interface can be limited. Once the limit is reached, packets with unknown source addresses are dropped until an existing host cache entry expires or is removed.

The following example sets the maximum number of Ethernet devices for CustomerA on vlan100 to 10:

```
# ifconfig bridge0 ifmaxaddr vlan100 10
```

Bridge interfaces also support monitor mode, where the packets are discarded after bpf(4) processing and are not processed or forwarded further. This can be used to multiplex the input of two or more interfaces into a single bpf(4) stream. This is useful for reconstructing the traffic for network taps that transmit the RX/TX signals out through two separate interfaces. For example, to read the input from four network interfaces as one stream:

```
# ifconfig bridge0 addm fxp0 addm fxp1 addm fxp2 addm fxp3 monitor up
# tcpdump -i bridge0
```

SNMP

The bridge interface and STP parameters can be monitored via bsnmpd(1) which is included in the FreeBSD base system. The exported bridge MIBs conform to IETF standards so any SNMP client or monitoring package can be used to retrieve the data.

To enable monitoring on the bridge, uncomment this line in /etc/snmpd.config by removing the beginning # symbol:
Other configuration settings, such as community names and access lists, may need to be modified in this file. See `bsnmpd(1)` and `snmp_bridge(3)` for more information. Once these edits are saved, add this line to `/etc/rc.conf`:

```
bsnmpd_enable="YES"
```

Then, start `bsnmpd(1)`:

```
# service bsnmpd start
```

The following examples use the Net-SNMP software (`net-mgmt/net-snmp`) to query a bridge from a client system. The `net-mgmt/bsnmptools` port can also be used. From the SNMP client which is running Net-SNMP, add the following lines to `$HOME/.snmp/snmp.conf` in order to import the bridge MIB definitions:

```
mibdirs +/usr/shared/snmp/mibs
mibs +BRIDGE-MIB:RSTP-MIB:BEGEMOT-MIB:BEGEMOT-BRIDGE-MIB
```

To monitor a single bridge using the IETF BRIDGE-MIB (RFC4188):

```
% snmpwalk -v 2c -c public bridge1.example.com mib-2.dot1dBridge
```

```
BRIDGE-MIB::dot1dBaseBridgeAddress.0 = STRING: 66:fb:9b:6e:5c:44
BRIDGE-MIB::dot1dBaseNumPorts.0 = INTEGER: 1 ports
BRIDGE-MIB::dot1dStpTimeSinceTopologyChange.0 = Timeticks: (189959) 0:31:39.59 centiseconds
BRIDGE-MIB::dot1dStpTopChanges.0 = Counter32: 2
BRIDGE-MIB::dot1dStpDesignatedRoot.0 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortState.3 = INTEGER: forwarding (5)
BRIDGE-MIB::dot1dStpPortEnable.3 = INTEGER: enabled (1)
BRIDGE-MIB::dot1dStpPortPathCost.3 = INTEGER: 200000
BRIDGE-MIB::dot1dStpPortDesignatedRoot.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedCost.3 = INTEGER: 0
BRIDGE-MIB::dot1dStpPortDesignatedBridge.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedPort.3 = Hex-STRING: 03 80
BRIDGE-MIB::dot1dStpPortForwardTransitions.3 = Counter32: 1
RSTP-MIB::dot1dStpVersion.0 = INTEGER: rstp (2)
```

The `dot1dStpTopChanges.0` value is two, indicating that the STP bridge topology has changed twice. A topology change means that one or more links in the network have changed or failed and a new tree has been calculated. The `dot1dStpTimeSinceTopologyChange.0` value will show when this happened.
To monitor multiple bridge interfaces, the private BEGEMOT-BRIDGE-MIB can be used:

```
% snmpwalk -v 2c -c public bridge1.example.com enterprises.fokus.begemot.begemotBridge BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName.
```

```
"bridge0" = STRING: bridge0
```

```
"bridge2" = STRING: bridge2
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress.
```

```
"bridge0" = STRING: e:ce:3b:5a:9e:13
```

```
"bridge2" = STRING: 12:5e:4d:74:d:fc
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts.
```

```
"bridge0" = INTEGER: 1
```

```
"bridge2" = INTEGER: 1
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange.
```

```
"bridge0" = Timeticks: (116927) 0:19:29.27 centi-seconds
```

```
"bridge2" = Timeticks: (82773) 0:13:47.73 centi-seconds
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges.
```

```
"bridge0" = Counter32: 1
```

```
"bridge2" = Counter32: 1
```

```
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot.
```

```
"bridge0" = Hex-STRING: 80 00 00 40 95 30 5E 31
```

```
"bridge2" = Hex-STRING: 80 00 00 50 8B B8 C6 A9
```

To change the bridge interface being monitored via the `mib-2.dot1dBridge` subtree:

```
% snmpset -v 2c -c private bridge1.example.com BEGEMOT-BRIDGE-MIB::begemotBridgeDefaultBridgeIf.0 bridge2
```

---

FreeBSD provides the `lagg(4)` interface which can be used to aggregate multiple network interfaces into one virtual interface in order to provide failover and link aggregation. Failover allows traffic to continue to flow as long as at least one aggregated network interface has an established link. Link aggregation works best on switches which support LACP, as this protocol distributes traffic bi-directionally while responding to the failure of individual links.

The aggregation protocols supported by the lagg interface determine which ports are used for outgoing traffic and whether or not a specific port accepts incoming traffic. The following protocols are supported by `lagg(4)`:

- **failover**
  - This mode sends and receives traffic only through the master port. If the master port becomes unavailable, the next active port is used. The first interface added to the virtual interface is the master port and all subsequently added interfaces are used as failover devices. If failover to a non-master port occurs, the original port becomes master once it becomes available again.

- **fec / loadbalance**
  - Cisco™ Fast EtherChannel™ (FEC) is found on older Cisco™ switches. It provides a static setup and does not negotiate aggregation with the peer or exchange frames to monitor the link. If the switch supports LACP, that should be used instead.
The IEEE™ 802.3ad Link Aggregation Control Protocol (LACP) negotiates a set of aggregable links with the peer into one or more Link Aggregated Groups (LAGs). Each LAG is composed of ports of the same speed, set to full-duplex operation, and traffic is balanced across the ports in the LAG with the greatest total speed. Typically, there is only one LAG which contains all the ports. In the event of changes in physical connectivity, LACP will quickly converge to a new configuration.

LACP balances outgoing traffic across the active ports based on hashed protocol header information and accepts incoming traffic from any active port. The hash includes the Ethernet source and destination address and, if available, the VLAN tag, and the IPv4 or IPv6 source and destination address.

This mode distributes outgoing traffic using a round-robin scheduler through all active ports and accepts incoming traffic from any active port. Since this mode violates Ethernet frame ordering, it should be used with caution.

This section demonstrates how to configure a Cisco™ switch and a FreeBSD system for LACP load balancing. It then shows how to configure two Ethernet interfaces in failover mode as well as how to configure failover mode between an Ethernet and a wireless interface.

Example 50. Cisco™ switch LACP Aggregation

This example connects two fxp(4) Ethernet interfaces on a FreeBSD machine to the first two Ethernet ports on a Cisco™ switch as a single load balanced and fault tolerant link. More interfaces can be added to increase throughput and fault tolerance. Replace the names of the Cisco™ ports, Ethernet devices, channel group number, and IP address shown in the example to match the local configuration.

Frame ordering is mandatory on Ethernet links and any traffic between two stations always flows over the same physical link, limiting the maximum speed to that of one interface. The transmit algorithm attempts to use as much information as it can to distinguish different traffic flows and balance the flows across the available interfaces.

On the Cisco™ switch, add the FastEthernet0/1 and FastEthernet0/2 interfaces to channel group 1:

```
interface FastEthernet0/1
channel-group 1 mode active
channel-protocol lacp

interface FastEthernet0/2
channel-group 1 mode active
channel-protocol lacp
```

On the FreeBSD system, create the lagg(4) interface using the physical interfaces fxp0 and fxp1 and bring the interfaces up with an IP address of 10.0.0.3/24:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24
```
Next, verify the status of the virtual interface:

```
# ifconfig lagg0
lagg0:
  flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  options = 8<VLAN_MTU>
  ether 00:05:5d:71:8d:b8
  inet 10.0.0.3 netmask 0xffffff00 broadcast 10.0.0.255
  media: Ethernet autoselect
  status: active
  laggproto lacp
  laggport: fxp1
    flags = 1c<ACTIVE,COLLECTING,DISTRIBUTING>
  laggport: fxp0
    flags = 1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

Ports marked as **ACTIVE** are part of the LAG that has been negotiated with the remote switch. Traffic will be transmitted and received through these active ports. Add `-v` to the above command to view the LAG identifiers.

To see the port status on the Cisco™ switch:

```
switch# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors
Partner's information:
  LACP port                        Oper    Port     Port
  Port      Flags   Priority  Dev ID         Age     Key     Number   State
  Fa0/1     SA      32768     0005.5d71.8db8  29s    0x146   0x3      0x3D
  Fa0/2     SA      32768     0005.5d71.8db8  29s    0x146   0x4      0x3D
```

For more detail, type `show lacp neighbor detail`.

To retain this configuration across reboots, add the following entries to `/etc/rc.conf` on the FreeBSD system:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24"
```
Failover mode can be used to switch over to a secondary interface if the link is lost on the master interface. To configure failover, make sure that the underlying physical interfaces are up, then create the `lagg(4)` interface. In this example, `fxp0` is the master interface, `fxp1` is the secondary interface, and the virtual interface is assigned an IP address of 10.0.0.15/24:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24
```

The virtual interface should look something like this:

```
# ifconfig lagg0
lagg0:
  flags = 8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
  options = 8<VLAN_MTU>
  ether 00:05:5d:71:8d:b8
  inet 10.0.0.15 netmask 0xffffff00 broadcast 10.0.0.255
  media: Ethernet autoselect
  status: active
  laggproto failover
  laggport: fxp1
    flags = 0<>
  laggport: fxp0
    flags = 5<MASTER,ACTIVE>
```

Traffic will be transmitted and received on `fxp0`. If the link is lost on `fxp0`, `fxp1` will become the active link. If the link is restored on the master interface, it will once again become the active link.

To retain this configuration across reboots, add the following entries to `/etc/rc.conf`:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0"
ifconfig_lagg0="laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24"
```

For laptop users, it is usually desirable to configure the wireless device as a secondary which is only used when the Ethernet connection is not available. With `lagg(4)`, it is possible to configure a failover which prefers the Ethernet connection for both performance and security reasons, while maintaining the ability to transfer data over the wireless connection. This is achieved by overriding the physical wireless interface's MAC address with that of the Ethernet interface.

In this example, the Ethernet interface, `bge0`, is the master and the wireless interface, `wlan0`, is the failover. The `wlan0` device was created from `iwn0` wireless interface, which will be configured with the MAC address of the Ethernet interface. First, determine the MAC address of the Ethernet interface.
Replace `bge0` to match the system's Ethernet interface name. The `ether` line will contain the MAC address of the specified interface. Now, change the MAC address of the underlying wireless interface:

```
# ifconfig iwn0 ether 00:21:70:da:ae:37
```

Bring the wireless interface up, but do not set an IP address:

```
# ifconfig wlan0 create wlandev iwn0 ssid my_router up
```

Make sure the `bge0` interface is up, then create the `lagg(4)` interface with `bge0` as master with failover to `wlan0`:

```
# ifconfig bge0 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport bge0 laggport wlan0
```

The virtual interface should look something like this:

```
# ifconfig lagg0
lagg0:flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
options=8<VLAN_MTU>
ether 00:21:70:da:ae:37
media: Ethernet autoselect
status: active
laggproto failover
laggport: wlan0
flags=0<>
laggport: bge0
flags=5<MASTER,ACTIVE>
```

Then, start the DHCP client to obtain an IP address:
To retain this configuration across reboots, add the following entries to /etc/rc.conf:

```bash
ifconfig_bge0="up"
wlan0_iwn0="wlan0"
ifconfig_wlan0="WPA"
create_args_wlan0="wlanaddr 00:21:70:da:ae:37"
cloned_interfaces="lagg0"
ifconfig_lagg0="up laggproto failover laggport bge0 laggport wlan0 DHCP"
```

The Intel™ Preboot eXecution Environment (PXE) allows an operating system to boot over the network. For example, a FreeBSD system can boot over the network and operate without a local disk, using file systems mounted from an NFS server. PXE support is usually available in the BIOS.

To use PXE when the machine starts, select the **Boot from network** option in the BIOS setup or type a function key during system initialization.

In order to provide the files needed for an operating system to boot over the network, a PXE setup also requires properly configured DHCP, TFTP, and NFS servers, where:

- Initial parameters, such as an IP address, executable boot filename and location, server name, and root path are obtained from the DHCP server.
- The operating system loader file is booted using TFTP.
- The file systems are loaded using NFS.

When a computer PXE boots, it receives information over DHCP about where to obtain the initial boot loader file. After the host computer receives this information, it downloads the boot loader via TFTP and then executes the boot loader. In FreeBSD, the boot loader file is `/boot/pxeboot`.

After `/boot/pxeboot` executes, the FreeBSD kernel is loaded and the rest of the FreeBSD bootup sequence proceeds, as described in the FreeBSD boot procedures.

This section describes how to configure these services on a FreeBSD system so that other systems can PXE boot into FreeBSD. Refer to `diskless(8)` for more information.

As described, the system providing these services is insecure. It should live in a protected area of a network and be untrusted by other hosts.

### 31.8.1. Setting up the PXE Environment

The steps shown in this section configure the built-in NFS and TFTP servers. The next section demonstrates how to install and configure the DHCP server.

In this example, the directory which will contain the files used by PXE users is `/b/tftpboot/FreeBSD/install`. It is important that this directory exists and that the same directory name is set in both `/etc/inetd.conf` and `/usr/local/etc/dhcpd.conf`.

1. Create the root directory which will contain a FreeBSD installation to be NFS mounted:

   ```bash
   # export NFSROOTDIR=/b/tftpboot/FreeBSD/install
   ```
1. Create the directory structure that will be used as the root file system:

   ```bash
   # mkdir -p ${NFSROOTDIR}
   ```

2. Enable the NFS server by adding this line to `/etc/rc.conf`:

   ```bash
   nfs_server_enable="YES"
   ```

3. Export the diskless root directory via NFS by adding the following to `/etc/exports`:

   ```bash
   /b -ro -alldirs -maproot=root
   ```

4. Start the NFS server:

   ```bash
   # service nfsd start
   ```

5. Enable `inetd(8)` by adding the following line to `/etc/rc.conf`:

   ```bash
   inetd_enable="YES"
   ```

6. Uncomment the following line in `/etc/inetd.conf` by making sure it does not start with a `#` symbol:

   ```bash
   tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /b/tftpboot
   ```

   Some PXE versions require the TCP version of TFTP. In this case, uncomment the second `tftp` line which contains `stream tcp`.

7. Start `inetd(8)`:

   ```bash
   # service inetd start
   ```

8. Install the base system into `${NFSROOTDIR}`, either by decompressing the official archives or by rebuilding the FreeBSD kernel and userland (refer to FreeBSD for more detailed instructions, but do not forget to add `DESTDIR=/${NFSROOTDIR}` when running the `make installkernel` and `make installworld` commands.

9. Test that the TFTP server works and can download the boot loader which will be obtained via PXE:

   ```bash
   # tftp localhost
tftp> get FreeBSD/install/boot/pxeboot
   Received 264951 bytes in 0.1 seconds
   ```

10. Edit `${NFSROOTDIR}/etc/fstab` and create an entry to mount the root file system over NFS:

    ```bash
    # Device                                         Mountpoint    FSType   Options  Dump Pass
    727
    ```

    ```bash
    Device                                         Mountpoint    FSType   Options  Dump Pass
    727
    ```
Replace `myhost.example.com` with the hostname or IP address of the NFS server. In this example, the root file system is mounted read-only in order to prevent NFS clients from potentially deleting the contents of the root file system.

11. Set the root password in the PXE environment for client machines which are PXE booting:
   ```
   # chroot ${NFSROOTDIR}
   # passwd
   ```
12. If needed, enable `ssh(1)` root logins for client machines which are PXE booting by editing `${NFSROOTDIR}/etc/ssh/sshd_config` and enabling `PermitRootLogin`. This option is documented in `sshd_config(5)`.
13. Perform any other needed customizations of the PXE environment in `${NFSROOTDIR}`. These customizations could include things like installing packages or editing the password file with `vipw(8)`.

When booting from an NFS root volume, `/etc/rc` detects the NFS boot and runs `/etc/rc.initdiskless`. In this case, `/etc` and `/var` need to be memory backed file systems so that these directories are writable but the NFS root directory is read-only:
   ```
   # chroot ${NFSROOTDIR}
   # mkdir -p conf/base
   # tar -c -v -f conf/base/etc.cpio.gz --format cpio --gzip etc
   # tar -c -v -f conf/base/var.cpio.gz --format cpio --gzip var
   ```

When the system boots, memory file systems for `/etc` and `/var` will be created and mounted and the contents of the `cpio.gz` files will be copied into them. By default, these file systems have a maximum capacity of 5 megabytes. If your archives do not fit, which is usually the case for `/var` when binary packages have been installed, request a larger size by putting the number of 512 byte sectors needed (e.g., 5 megabytes is 10240 sectors) in `${NFSROOTDIR}/conf/base/etc/md_size` and `${NFSROOTDIR}/conf/base/var/md_size` files for `/etc` and `/var` file systems respectively.

31.8.2. 設定 DHCP 伺服器

The DHCP server does not need to be the same machine as the TFTP and NFS server, but it needs to be accessible in the network.

DHCP is not part of the FreeBSD base system but can be installed using the `net/isc-dhcp44-server` port or package.

Once installed, edit the configuration file, `/usr/local/etc/dhcpd.conf`. Configure the `next-server`, `filename`, and `root-path` settings as seen in this example:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
  range 192.168.0.2 192.168.0.3 ;
  option subnet-mask 255.255.255.0 ;
  option routers 192.168.0.1 ;
  option broadcast-address 192.168.0.255 ;
```
The next-server directive is used to specify the IP address of the TFTP server. The filename directive defines the path to /boot/pxeboot. A relative filename is used, meaning that /b/tftpboot is not included in the path. The root-path option defines the path to the NFS root file system. Once the edits are saved, enable DHCP at boot time by adding the following line to /etc/rc.conf:

dhcpd_enable="YES"

Then start the DHCP service:

# service isc-dhcpd start

Once all of the services are configured and started, PXE clients should be able to automatically load FreeBSD over the network. If a particular client is unable to connect, when that client machine boots up, enter the BIOS configuration menu and confirm that it is set to boot from the network. This section describes some troubleshooting tips for isolating the source of the configuration problem should no clients be able to PXE boot.

1. Use the net/wireshark package or port to debug the network traffic involved during the PXE booting process, which is illustrated in the diagram below.
2. On the TFTP server, read /var/log/xferlog to ensure that pxeboot is being retrieved from the correct location. To test this example configuration:

```
# tftp 192.168.0.1
tftp> get FreeBSD/install/boot/pxeboot
```

Received 264951 bytes in 0.1 seconds

The BUGS sections in tftpd(8) and tftp(1) document some limitations with TFTP.

3. Make sure that the root file system can be mounted via NFS. To test this example configuration:

```
# mount -t nfs 192.168.0.1:/b/tftpboot/FreeBSD/install /mnt
```

IPv6

IPv6 is the new version of the well known IP protocol, also known as IPv4. IPv6 provides several advantages over IPv4 as well as many new features:

- Its 128-bit address space allows for 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses. This addresses the IPv4 address shortage and eventual IPv4 address exhaustion.
- Routers only store network aggregation addresses in their routing tables, thus reducing the average space of a routing table to 8192 entries. This addresses the scalability issues associated with IPv4, which required every allocated block of IPv4 addresses to be exchanged between Internet routers, causing their routing tables to become too large to allow efficient routing.
- Address autoconfiguration (RFC2462).
- Mandatory multicast addresses.
- Built-in IPsec (IP security).
- Simplified header structure.
- Support for mobile IP.
- IPv6-to-IPv4 transition mechanisms.

FreeBSD includes the http://www.kame.net/ IPv6 reference implementation and comes with everything needed to use IPv6. This section focuses on getting IPv6 configured and running.
There are three different types of IPv6 addresses:

- **Unicast**: A packet sent to a unicast address arrives at the interface belonging to the address.

- **Anycast**: These addresses are syntactically indistinguishable from unicast addresses but they address a group of interfaces. The packet destined for an anycast address will arrive at the nearest router interface. Anycast addresses are only used by routers.

- **Multicast**: These addresses identify a group of interfaces. A packet destined for a multicast address will arrive at all interfaces belonging to the multicast group. The IPv4 broadcast address, usually `xxx.xxx.xxx.255`, is expressed by multicast addresses in IPv6.

When reading an IPv6 address, the canonical form is represented as `x:x:x:x:x:x:x:x`, where each `x` represents a 16 bit hex value. An example is `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`.

Often, an address will have long substrings of all zeros. A `::` (double colon) can be used to replace one substring per address. Also, up to three leading zeros per hex value can be omitted. For example, `fe80::1` corresponds to the canonical form `fe80:0000:0000:0000:0000:0000:0000:0001`.

A third form is to write the last 32 bits using the well known IPv4 notation. For example, `2002::10.0.0.1` corresponds to the hexadecimal canonical representation `2002:0000:0000:0000:0000:0000:0a00:0001`, which in turn is equivalent to `2002::a00:1`.

To view a FreeBSD system's IPv6 address, use `ifconfig(8)`:

```
# ifconfig
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
ether 00:00:21:03:08:e1
media: Ethernet autoselect (100baseTX)
status: active
```

In this example, the `rl0` interface is using `fe80::200:21ff:fe03:8e1%rl0`, an auto-configured link-local address which was automatically generated from the MAC address.

Some IPv6 addresses are reserved. A summary of these reserved addresses is seen in [Reserved IPv6 Addresses](#):
IPv6 address  Prefix length (Bits)  說明
::00:xx:xx:xx:xx  96 bits  說明
::ff:xx:xx:xx:xx  96 bits  IPv4 mapped IPv6 address

The lower 32 bits are the compatible IPv4 address.

fe80::/10  10 bits  link-local
Equivalent to 169.254.0.0/16 in IPv4.

fc00::/7  7 bits  unique-local
Unique local addresses are intended for local communication and are only routable within a set of cooperating sites.

ff00::  8 bits  multicast

2000::-3fff:  3 bits  global unicast
All global unicast addresses are assigned from this pool. The first 3 bits are 001.

For further information on the structure of IPv6 addresses, refer to RFC3513.

31.9.2.設定IPv6
To configure a FreeBSD system as an IPv6 client, add these two lines to rc.conf:
ifconfig_rl0_ipv6="inet6 accept_rtadv"
rtsold_enable="YES"
The first line enables the specified interface to receive router advertisement messages. The second line enables the router solicitation daemon, rtsol(8).

If the interface needs a statically assigned IPv6 address, add an entry to specify the static address and associated prefix length:
ifconfig_rl0_ipv6="inet6 2001:db8:4672:6565:2026:5043:2d42:5344 prefixlen 64"
To assign a default router, specify its address:
ipv6_defaultrouter="2001:db8:4672:6565::1"

31.9.3.連線到Provider
In order to connect to other IPv6 networks, one must have a provider or a tunnel that supports IPv6:
• Contact an Internet Service Provider to see if they offer IPv6.
• Hurricane Electric offers tunnels with end-points all around the globe.

Install the net/freenet6 package or port for a dial-up connection.
This section demonstrates how to take the directions from a tunnel provider and convert them into /etc/rc.conf settings that will persist through reboots.

The first /etc/rc.conf entry creates the generic tunneling interface gif0:

```
cloned_interfaces="gif0"
```

Next, configure that interface with the IPv4 addresses of the local and remote endpoints. Replace MY_IPv4_ADDR and REMOTE_IPv4_ADDR with the actual IPv4 addresses:

```
create_args_gif0="tunnel MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

To apply the IPv6 address that has been assigned for use as the IPv6 tunnel endpoint, add this line, replacing MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR with the assigned address:

```
ifconfig_gif0_ipv6="inet6 MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Then, set the default route for the other side of the IPv6 tunnel. Replace MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR with the default gateway address assigned by the provider:

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

If the FreeBSD system will route IPv6 packets between the rest of the network and the world, enable the gateway using this line:

```
ipv6_gateway_enable="YES"
```

31.9.4. Router Advertisement

This section demonstrates how to setup rtadvd(8) to advertise the IPv6 default route.

To enable rtadvd(8), add the following to /etc/rc.conf:

```
rtadvd_enable="YES"
```

It is important to specify the interface on which to do IPv6 router advertisement. For example, to tell rtadvd(8) to use rl0:

```
rtadvdInterfaces="rl0"
```

Next, create the configuration file, /etc/rtadvd.conf as seen in this example:

```
rl0:
        addrs#1: addr="2001:db8:1f11:246::": prefixlen#64: tc=ether:
```

Replace rl0 with the interface to be used and 2001:db8:1f11:246:: with the prefix of the allocation. For a dedicated /64 subnet, nothing else needs to be changed. Otherwise, change the prefixlen to the correct value.

31.9.5. IPv6

與 IPv6 位址對應

When IPv6 is enabled on a server, there may be a need to enable IPv4 mapped IPv6 address communication. This compatibility option allows for IPv4 addresses to be represented as IPv6 addresses. Permitting IPv6 applications to communicate with IPv4 and vice versa may be a security issue. This option may not be required in most cases and is available only for compatibility. This option will allow IPv6-only applications to work with IPv4 in a dual stack environment. This is most useful for third party applications which may not support an IPv6-only environment. To enable this feature, add the following to /etc/rc.conf:

```
ipv6_ipv4mapping="YES"
```

Reviewing the information in RFC 3493, section 3.6 and 3.7 as well as RFC 4038 section 4.2 may be useful to some administrators.

31.10. 共用位址備援協定 (CARP)

The Common Address Redundancy Protocol (CARP) allows multiple hosts to share the same IP address and Virtual Host ID (VHID) in order to provide high availability for one or more services. This means that one or more hosts can fail, and the other hosts will transparently take over so that users do not see a service failure.

In addition to the shared IP address, each host has its own IP address for management and configuration. All of the machines that share an IP address have the same VHID. The VHID for each virtual IP address must be unique across the broadcast domain of the network interface.

High availability using CARP is built into FreeBSD, though the steps to configure it vary slightly depending upon the FreeBSD version. This section provides the same example configuration for versions before and equal to or after FreeBSD 10.

This example configures failover support with three hosts, all with unique IP addresses, but providing the same web content. It has two different masters named hosta.example.org and hostb.example.org, with a shared backup named hostc.example.org.

These machines are load balanced with a Round Robin DNS configuration. The master and backup machines are configured identically except for their hostnames and management IP addresses. These servers must have the same configuration and run the same services. When the failover occurs, requests to the service on the shared IP address can only be answered correctly if the backup server has access to the same content. The backup machine has two additional CARP interfaces, one for each of the master content server's IP addresses. When a failure occurs, the backup server will pick up the failed master machine's IP address.

31.10.1. 使用 CARP 於 FreeBSD 10 及之後版本

Enable boot-time support for CARP by adding an entry for the carp.ko kernel module in /boot/loader.conf:

```
carp_load="YES"
```

To load the module now without rebooting:

```
734
```
For users who prefer to use a custom kernel, include the following line in the custom kernel configuration file and compile the kernel as described in FreeBSD:

device carp

The hostname, management IP address and subnet mask, shared IP address, and VHID are all set by adding entries to /etc/rc.conf. This example is for hosta.example.org:

hostname="hosta.example.org"
ifconfig_em0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 1 pass testpass alias 192.168.1.50/32"

The next set of entries are for hostb.example.org. Since it represents a second master, it uses a different shared IP address and VHID. However, the passwords specified with pass must be identical as CARP will only listen to and accept advertisements from machines with the correct password.

hostname="hostb.example.org"
ifconfig_em0="inet 192.168.1.4 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 2 pass testpass alias 192.168.1.51/32"

The third machine, hostc.example.org, is configured to handle failover from either master. This machine is configured with two CARPVHIDs, one to handle the virtual IP address for each of the master hosts. The CARP advertising skew, advskew, is set to ensure that the backup host advertises later than the master, since advskew controls the order of precedence when there are multiple backup servers.

hostname="hostc.example.org"
ifconfig_em0="inet 192.168.1.5 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 1 advskew 100 pass testpass alias 192.168.1.50/32"
ifconfig_em0_alias1="inet vhid 2 advskew 100 pass testpass alias 192.168.1.51/32"

Having two CARPVHIDs configured means that hostc.example.org will notice if either of the master servers becomes unavailable. If a master fails to advertise before the backup server, the backup server will pick up the shared IP address until the master becomes available again.

If the original master server becomes available again, hostc.example.org will not release the virtual IP address back to it automatically. For this to happen, preemption has to be enabled. The feature is disabled by default, it is controlled via the sysctl(8) variable net.inet.carp.preempt. The administrator can force the backup server to return the IP address to the master:

# ifconfig em0 vhid 1 state backup

Once the configuration is complete, either restart networking or reboot each system. If the configuration is successful, the master will have the shared IP address. If not, check the output of ifconfig and make sure that the IP address is assigned to the correct interface.
availability is now enabled.

CARP functionality can be controlled via several `sysctl(8)` variables documented in the `carp(4)` manual pages. Other actions can be triggered from CARP events by using `devd(8)`.

31.10.2. 使用 CARP 於 FreeBSD 9 及先前版本

The configuration for these versions of FreeBSD is similar to the one described in the previous section, except that a CARP device must first be created and referred to in the configuration.

Enable boot-time support for CARP by loading the `if_carp.ko` kernel module in `/boot/loader.conf`:

```
if_carp_load="YES"
```

To load the module now without rebooting:

```
# kldload carp
```

For users who prefer to use a custom kernel, include the following line in the custom kernel configuration file and compile the kernel as described in `設定 FreeBSD 核心`:

```
device carp
```

Next, on each host, create a CARP device:

```
# ifconfig carp0 create
```

Set the hostname, management IP address, the shared IP address, and VHID by adding the required lines to `/etc/rc.conf`. Since a virtual CARP device is used instead of an alias, the actual subnet mask of `/24` is used instead of `/32`. Here are the entries for `hosta.example.org`:

```
hostname="hosta.example.org"
ifconfig_fxp0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass testpass 192.168.1.50/24"
```

On `hostb.example.org`:

```
hostname="hostb.example.org"
ifconfig_fxp0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass testpass 192.168.1.51/24"
```

The third machine, `hostc.example.org`, is configured to handle failover from either of the master hosts:

```
hostname="hostc.example.org"
```

736
Preemption is disabled in the GENERIC FreeBSD kernel. If preemption has been enabled with a custom kernel, hostc.example.org may not release the IP address back to the original content server. The administrator can force the backup server to return the IP address to the master with the command:

```
# ifconfig carp0 down && ifconfig carp0 up
```

This should be done on the carp interface which corresponds to the correct host.

Once the configuration is complete, either restart networking or reboot each system. High availability is now enabled.

31.11. VLANs

VLANs are a way of virtually dividing up a network into many different subnetworks, also referred to as segmenting. Each segment will have its own broadcast domain and be isolated from other VLANs.

To configure VLANs at run time, with a NIC of em0 and a VLAN tag of 5 the command would look like this:

```
# ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.20.20/24
```

See how the interface name includes the NIC driver name and the VLAN tag, separated by a period? This is a best practice to make maintaining the VLAN configuration easy when many VLANs are present on a machine.

To configure VLANs at boot time, /etc/rc.conf must be updated. To duplicate the configuration above, the following will need to be added:

```
vlans_em0="5"
ifconfig_em0_5="inet 192.168.20.20/24"
```

Additional VLANs may be added, by simply adding the tag to the vlans field and adding an additional line configuring the network on that VLAN tag's interface.

It is useful to assign a symbolic name to an interface so that when the associated hardware is changed, only a few configuration variables need to be updated. For example, security cameras need to be run over VLAN 1 on em0. Later, if the em0 card is replaced with a card that uses the ixgb(4) driver, all references to em0.1 will not have to change to ixgb0.1.
To configure VLAN 5, on the NIC em0, assign the interface name cameras, and assign the interface an IP address of 192.168.20.20 with a 24-bit prefix, use this command:

```
# ifconfig em0.5 create vlan 5 vlandev em0 name cameras inet 192.168.20.20/24
```

For an interface named video, use the following:

```
# ifconfig video.5 create vlan 5 vlandev video name cameras inet 192.168.20.20/24
```

To apply the changes at boot time, add the following lines to `/etc/rc.conf`:

```
vlans_video="camera"
create_args_camera="vlan 5"
ifconfig_camera="inet 192.168.20.20/24"
```
A.1. CD

FreeBSD CD 以及 DVD 合集可從以下幾個線上零售商取得:

- FreeBSD Mall, Inc.
  2420 Sand Creek Rd C-1 #347
  Brentwood, CA 94513 USA
  Phone: +1 925 240-6652
  Fax: +1 925 674-0821
  Email: info@freebsdmall.com
  WWW: https://www.freebsdmall.com

- Getlinux
  78 Rue de la Croix Rochopt
  Épinay-sous-Sénart 91860 France
  Email: contact@getlinux.fr
  WWW: http://www.getlinux.fr/

- Dr. Hinner EDV
  Kochelseestr. 11
  D-81371 München Germany
  Phone: (0177) 428 419 0
  Email: infow@hinner.de
  WWW: http://www.hinner.de/ linux/freebsd.html

- Linux Center
  Galernaya Street, 55
  Saint-Petersburg 190000 Russia
  Phone: +7-812-309-06-86
  Email: info@linuxcenter.ru
  WWW: http://linuxcenter.ru/shop/freebsd

A.2. FTP

FreeBSD 的官方原始碼可從全體任一鏡像站透過匿名 FTP 取得。其中包括 ftp://ftp.FreeBSD.org/pub/FreeBSD/ 站可使用 HTTP 及 FTP, 该站是由多台由計畫叢集管理员所維護的主機所組成, 且在 GeoDNS 之後, 可導向使用者到最近可用的鏡像站。除此之外, FreeBSD 也可透過匿名 FTP 從下列鏡像站取得。要透過匿名 FTP 取得 FreeBSD 時, 請先嘗試使用臨近的站台。列在“主要鏡像站”中的鏡像站通常會有完整的 FreeBSD 封存檔(每一個架構目前所有可用的版本), 但若要考慮下載速度, 可能要使用您所在國家或區域的站台。區域的站台會有熱門架構最近期的版本, 但不會有完整的 FreeBSD 封存檔。所有站台皆提供匿名 FTP 存取, 但某些站台會以其他方式提供存取。每個站台可用的存取方式會列在主機名稱後的括號當中。
Central Servers

Primary Mirror Sites

Armenia

Australia

Austria

Brazil

Czech Republic

Denmark

Estonia

Finland

France

Germany

Greece

Hong Kong

Ireland

Japan

Korea

Latvia

Lithuania

Netherlands

New Zealand

Norway

Poland

Russia

Saudi Arabia

Slovenia

South Africa

Spain

Sweden

Switzerland

Taiwan

Ukraine

United Kingdom

United States of America

(as of UTC)

Central Servers


Primary Mirror Sites

In case of problems, please contact the hostmaster <mirror-admin@FreeBSD.org> for this domain.


Armenia

In case of problems, please contact the hostmaster <hostmaster@am.FreeBSD.org> for this domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD
- ftp://ftp2.am.FreeBSD.org/pub/FreeBSD

Australia

In case of problems, please contact the hostmaster <hostmaster@au.FreeBSD.org> for this domain.


Austria

In case of problems, please contact the hostmaster <hostmaster@at.FreeBSD.org> for this domain.


http://ftp.at.FreeBSD.org/pub/FreeBSD/
Brazil
In case of problems, please contact the hostmaster <hostmaster@br.FreeBSD.org>

Czech Republic
In case of problems, please contact the hostmaster <hostmaster@cz.FreeBSD.org>

Denmark
In case of problems, please contact the hostmaster <staff@dotsrc.org>

Estonia
In case of problems, please contact the hostmaster <hostmaster@ee.FreeBSD.org>

Finland
In case of problems, please contact the hostmaster <hostmaster@fi.FreeBSD.org>

France


Germany
In case of problems, please contact the hostmaster <de-bsd-hubs@de.FreeBSD.org>
In case of problems, please contact the hostmaster <hostmaster@gr.FreeBSD.org> for this domain.

In case of problems, please contact the hostmaster <hostmaster@ie.FreeBSD.org> for this domain.

In case of problems, please contact the hostmaster <hostmaster@jp.FreeBSD.org> for this domain.

In case of problems, please contact the hostmaster <hostmaster@kr.FreeBSD.org> for this domain.
Latvia
In case of problems, please contact the hostmaster <hostmaster@lv.FreeBSD.org> for this domain.


Lithuania
In case of problems, please contact the hostmaster <hostmaster@lt.FreeBSD.org> for this domain.


Netherlands
In case of problems, please contact the hostmaster <hostmaster@nl.FreeBSD.org> for this domain.


New Zealand

Norway
In case of problems, please contact the hostmaster <hostmaster@no.FreeBSD.org> for this domain.


Poland
In case of problems, please contact the hostmaster <hostmaster@pl.FreeBSD.org> for this domain.


Russia
In case of problems, please contact the hostmaster <hostmaster@ru.FreeBSD.org> for this domain.


Saudi Arabia
In case of problems, please contact the hostmaster <ftpadmin@isu.net.sa> for this domain.

ftp://ftp.isu.net.sa/pub/

Slovenia
In case of problems, please contact the hostmaster <hostmaster@si.FreeBSD.org> for this domain.

South Africa
In case of problems, please contact the hostmaster <hostmaster@za.FreeBSD.org> for this domain.

Spain
In case of problems, please contact the hostmaster <hostmaster@es.FreeBSD.org> for this domain.

Sweden
In case of problems, please contact the hostmaster <hostmaster@se.FreeBSD.org> for this domain.
• ftp://ftp2.se.FreeBSD.org/pub/
• ftp://ftp3.se.FreeBSD.org/pub/

Switzerland
In case of problems, please contact the hostmaster <hostmaster@ch.FreeBSD.org> for this domain.

Taiwan
In case of problems, please contact the hostmaster <hostmaster@tw.FreeBSD.org> for this domain.
• ftp://ftp2.tw.FreeBSD.org/pub/
• ftp://ftp5.tw.FreeBSD.org/pub/
A.3. 使用 Subversion 746
A.3.1. 簡介
自2012年7月起，FreeBSD儲存所有FreeBSD的原始碼、文件與Port套件集均使用Subversion作為其唯一版本控制系統。

Subversion只是一套開發人員工具。一般使用者可能會較喜歡使用freebsd-update（FreeBSD更新）來更新FreeBSD基礎系統及portsnap（使用Port套件集）來更新FreeBSD Port套件集。

本節將示範如何在FreeBSD系統安裝Subversion及使用它建立一個本地的FreeBSD檔案庫複本，也包含使用Subversion的其他資訊。

A.3.2. 根SSL憑證
安裝security/ca_root_nss可讓Subversion能夠驗証HTTPS檔案庫伺服器的身份。

root SSL憑証可從Port安裝:
```
# cd /usr/ports/security/ca_root_nss
# make install clean
```
或從套件:
```
# pkg install ca_root_nss
```

A.3.3. Svnlite
輕量化版的Subversionsvnlite已會隨FreeBSD安裝。Port或套件版的Subversion僅在要使用其Python或Perl API時需要，或是新想使用最新版本Subversion時才需要。與正常Subversion唯一的差別只是指令名稱改為svnlite。

A.3.4. 安裝
若無法使用svnlite或需要完整的版本的Subversion就必須安裝。Subversion可從Port套件集安裝:
```
# cd /usr/ports/devel/subversion
# make install clean
```
Subversion也可以以套件安裝:
```
# pkg install subversion
```

A.3.5. 執行
Subversion要下載原始碼乾淨的複本到本地目錄可使用svn。在此目錄中的檔案稱作本地工作複本（Local working copy）。

在第一次使用checkout前請先移動或刪除目地現有的目錄。在現有非svn目錄存儲在的情況下做取出（Checkout）
Subversion 使用 URL 来指定档案库，使用的格式为
protocol://hostname/path。路径的第一个部份为要存放的 FreeBSD 档案库，目前有三个档案库，base 为 FreeBSD 基础系统原始码、ports 为 Port 套件集以及 doc 为说明文件。举例来说，URL https://svn.FreeBSD.org/ports/head 代表 Port 档案库的主要分支，使用 https 通讯协定。

使用指令从指定的档案库取出 (Checkout) 原始码如下:

```
# svn checkout https://svn.FreeBSD.org/repository/branch lwcdir
```

where:

- repository 要是下列专案档案库其中之一：base, ports 或 doc。
- branch 则依据使用的档案库来决定。
- ports 与 doc 大部份的更新皆在 head 分支，而 base 则会将 -CURRENT 的最新版本存放于 head 下，-STABLE 分支各自最新的版本则会放于 stable/9(9.x) 与 stable/10(10.x) 下。
- lwcdir 则是要存放指定分支内容的目标目录，通常 ports 会置於 /usr/ports，base 会置於 /usr/src 以及 doc 会置於 /usr/doc。

以下范例会使用 HTTPS 协定从 FreeBSD 的档案库取出 Port 套件集，并将本地工作複本放置於 /usr/ports。若 /usr/ports 已存在，且不是由 svn 所建立的，记得要在取出之前重新命名或删除。

```
# svn checkout https://svn.FreeBSD.org/ports/head /usr/ports
```

由於首次取出的动作必须下载远程档案库中完整的分支，会需要花费一段时间，请耐心等候。

首次取出之后，往后要更新本地工作複本可以执行:

```
# svn update lwcdir
```

要更新上述范例所建立的 /usr/ports 可执行:

```
# svn update /usr/ports
```

因为只会传输有更新过的档案，更新的动作会比取出还要快速。

另一种在取出之后更新本地工作複本的方式是透过 /usr/ports, /usr/src 以及 /usr/doc 目录所提供的 Makefile。设定 SVN_UPDATE 并使用 update 目标。例如要更新 /usr/src:

```
# cd /usr/src
# make update SVN_UPDATE=yes
```
這是公開存取的鏡像站，使用了GeoDNS會自動選擇適合的後端伺服器。若要由瀏覽器檢視Subversion檔案庫可以使用https://svnweb.FreeBSD.org/。

HTTPS是的預備協定，但security/ca_root_nss包需要安裝才能自動驗證憑证。

A.3.7.要取得其他有關使用Subversion的資訊請參考"Subversion書"，其書名為Version Control with Subversion或是Subversion Documentation。

A.4.使用rsync這些站台讓FreeBSD可透過rsync通訊協定取得。rsync工具只會傳輸兩個檔案集之間的差異，所以能夠大大地加快在網路上同步的速度，這對大多數FreeBSD FTP伺服器的鏡像站非常有用。rsync在許多作業系統上也可以使用，在FreeBSD上請參考net/rsyncPort或使用套件。

捷克(Czech Republic)rsync://ftp.cz.FreeBSD.org/可用的檔案集:
- ftp: FreeBSD FTP伺服器的部份鏡像。
- FreeBSD: FreeBSD FTP伺服器的完整鏡像。

荷蘭(Netherlands)rsync://ftp.nl.FreeBSD.org/可用的檔案集:
- FreeBSD: FreeBSD FTP伺服器的完整鏡像。

俄羅斯(Russia)rsync://ftp.mtu.ru/可用的檔案集:
- FreeBSD: FreeBSD FTP伺服器的完整鏡像。
- FreeBSD-Archive: FreeBSD封存FTP伺服器的鏡像。

瑞典(Sweden)rsync://ftp4.se.freebsd.org/可用的檔案集:
- FreeBSD: FreeBSD FTP伺服器的完整鏡像。

- FreeBSD: FreeBSD FTP伺服器的完整鏡像。

英國(United Kingdom)rsync://rsync.mirrorservice.org/
可用的档案集：
• ftp.freebsd.org: FreeBSD FTP 服务器的完整镜像。

• rsync://ftp-master.FreeBSD.org/
此服务器仅供 FreeBSD 主要镜像站使用。

可用的档案集：
• FreeBSD: FreeBSD FTP 服务器的主要存储。
• acl: FreeBSD 主要 ACL 清单。

• rsync://ftp13.FreeBSD.org/
可用的档案集：
• FreeBSD: FreeBSD FTP 服务器的完整镜像。
雖然操作手冊提供FreeBSD作業系統各個部分完整的說明，卻難免有「小學而大遺」之憾，像是如何讓整個作業系統運作順暢。因此，身邊有UNIX™系統管理的好書以及好的使用手冊是不可或缺的。

B.1. FreeBSD相關書籍

<table>
<thead>
<tr>
<th>英文書籍</th>
<th>繁體中文書籍</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute FreeBSD, 2nd Edition: The Complete Guide to FreeBSD</td>
<td>FreeBSD入門與應用(光碟豪華版)</td>
</tr>
<tr>
<td>The Complete FreeBSD</td>
<td>FreeBSD技術內幕(FreeBSD Unleashed簡體中文譯版)</td>
</tr>
<tr>
<td>The FreeBSD Corporate Networker’s Guide</td>
<td>FreeBSD使用大全第二版(簡體中文)</td>
</tr>
<tr>
<td>FreeBSD: An Open-Source Operating System for Your Personal Computer</td>
<td>FreeBSD Handbook第二版(簡體中文譯版)</td>
</tr>
<tr>
<td>Teach Yourself FreeBSD in 24 Hours</td>
<td>FreeBSD Handbook(日譯版)</td>
</tr>
<tr>
<td>FreeBSD 6 Unleashed</td>
<td>FreeBSD &amp; Windows集成組網實務(簡體中文)</td>
</tr>
<tr>
<td>FreeBSD website架設實務(簡體中文)</td>
<td>FreeBSD網站架設實務(簡體中文)</td>
</tr>
<tr>
<td>FreeBSD (日文)</td>
<td>FreeBSD (日文)</td>
</tr>
<tr>
<td>Complete Introduction to FreeBSD</td>
<td>FreeBSD mit Methode (德文)</td>
</tr>
<tr>
<td>Personal UNIX Starter Kit FreeBSD</td>
<td>FreeBSD de Luxe (德文)</td>
</tr>
<tr>
<td>FreeBSD Handbook (日譯版)</td>
<td>FreeBSD Handbook (日譯版)</td>
</tr>
<tr>
<td>FreeBSD mit Methode (德文)</td>
<td>FreeBSD mit Methode (德文)</td>
</tr>
<tr>
<td>FreeBSD de Luxe (德文)</td>
<td>FreeBSD de Luxe (德文)</td>
</tr>
</tbody>
</table>

英語書籍:

<table>
<thead>
<tr>
<th>英文書籍</th>
<th>ISBN</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Complete FreeBSD</td>
<td>0596005164</td>
</tr>
<tr>
<td>The FreeBSD Corporate Networker’s Guide</td>
<td>0201704811</td>
</tr>
<tr>
<td>FreeBSD: An Open-Source Operating System for Your Personal Computer</td>
<td>0971204500</td>
</tr>
<tr>
<td>Teach Yourself FreeBSD in 24 Hours</td>
<td>0672324245</td>
</tr>
<tr>
<td>FreeBSD 6 Unleashed</td>
<td>0672328755</td>
</tr>
<tr>
<td>FreeBSD: The Complete Reference</td>
<td>0072224096</td>
</tr>
</tbody>
</table>

B.2. 使用指南

Ohio State University has written a UNIX Introductory Course which is available online in HTML and PostScript format. An Italian translation of this document is available as part of the FreeBSD Italian Documentation.
Edinburgh University has written an Online Guide for newcomers to the UNIX environment.


Old BSD releases from the Computer Systems Research group (CSRG) http://www.mckusick.com/csrg/: The 4CD set covers all BSD versions from 1BSD to 4.4BSD and 4.4BSD-Lite2 (but not 2.11BSD, unfortunately). The last disk also holds the final sources plus the SCCS files.

B.9.最新期刊與雜誌

- Admin Magazin (in German), published by Medialinx AG. ISSN: 2190-1066
- BSD Magazine, published by Software Press Sp. z o.o. ISSN: 1898-9144
- BSD Now—Video Podcast, published by Jupiter Broadcasting LLC
- BSD Talk Podcast, by Will Backman
The rapid pace of FreeBSD progress makes print media impractical as a means of following the latest developments. Electronic resources are the best, if not often the only, way to stay informed of the latest advances. Since FreeBSD is a volunteer effort, the user community itself also generally serves as a "technical support department" of sorts, with electronic mail, web forums, and USENET news being the most effective way of reaching that community.

The most important points of contact with the FreeBSD user community are outlined below. Please send other resources not mentioned here to the FreeBSD documentation project mailing list so that they may also be included.

C.1. 網站
• The FreeBSD Forums provide a web based discussion forum for FreeBSD questions and technical discussion.
• The FreeBSDConferences YouTube Channel provides a collection of high quality videos from BSD conferences around the world. This is a great way to watch key developers give presentations about new work in FreeBSD.

C.2. 郵遞論壇 (Mailing List)
The mailing lists are the most direct way of addressing questions or opening a technical discussion to a concentrated FreeBSD audience. There are a wide variety of lists on a number of different FreeBSD topics. Sending questions to the most appropriate mailing list will invariably assure a faster and more accurate response.

The charters for the various lists are given at the bottom of this document. Please read the charter before joining or sending mail to any list. Most list subscribers receive many hundreds of FreeBSD related messages every day, and the charters and rules for use are meant to keep the signal-to-noise ratio of the lists high. To do less would see the mailing lists ultimately fail as an effective communications medium for the Project.

To test the ability to send email to FreeBSD lists, send a test message to freebsd-test. Please do not send test messages to any other list.

When in doubt about what list to post a question to, see How to get best results from the FreeBSD-questions mailing list.

Before posting to any list, please learn about how to best use the mailing lists, such as how to help avoid frequently-repeated discussions, by reading the Mailing List Frequently Asked Questions (FAQ) document.

Archives are kept for all of the mailing lists and can be searched using the FreeBSD World Wide Web server. The keyword searchable archive offers an excellent way of finding answers to frequently asked questions and should be consulted before posting a question. Note that this also means that messages sent to FreeBSD mailing lists are archived in perpetuity. When protecting privacy is a concern, consider using a disposable secondary email address and posting only public information.

C.2.1. 論壇摘要
General lists:
The following are general lists which anyone is free (and encouraged) to join:
- freebsd-advocacy: FreeBSD Evangelism
- freebsd-announce: Important events and Project milestones (moderated)
List

Architecture and design discussions

Discussions pertaining to the maintenance of the FreeBSD problem report database and related tools

Bug reports

Non-technical items related to the FreeBSD community

FreeBSD-specific Chromium issues

Discussion concerning the use of FreeBSD-CURRENT

Issues for Internet Service Providers using FreeBSD

FreeBSD employment and consulting opportunities

User questions and technical support

Security notifications (moderated)

Discussion concerning the use of FreeBSD-STABLE

Where to send test messages instead of to one of the actual lists

FreeBSD advocacy for women

Technical lists:
The following lists are for technical discussion. Read the charter for each list carefully before joining or sending mail to one as there are firm guidelines for their use and content.

ACPI and power management development

Porting AFS to FreeBSD

Porting FreeBSD to AMD64 systems (moderated)

Discussion about Apache related ports

Porting FreeBSD to ARM™ processors

Using ATM networking with FreeBSD

Using Bluetooth™ technology in FreeBSD

FreeBSD on cloud platforms (EC2, GCE, Azure, etc.)

Using FreeBSD in a clustered environment

Discussing database use and development under FreeBSD

Using and improving FreeBSD on the desktop

Build and test reports from the Continuous Integration servers

Notifications of the FreeBSD review system

Creating FreeBSD related documents

Writing device drivers for FreeBSD

Using and working on DTrace in FreeBSD
FreeBSD users of Eclipse IDE, tools, rich client applications and ports.

FreeBSD-specific ElasticSearch discussions

Using FreeBSD in embedded applications

Peer support of FreeBSD-related software that is no longer supported by the FreeBSD Project.

Emulation of other systems such as Linux/MS-DOS™/Windows™

Porting Enlightenment and Enlightenment applications

FreeBSD-specific Erlang discussions

FreeBSD FireWire™ (iLink, IEEE 1394) technical discussion

Fortran on FreeBSD

File systems

Support for Games on FreeBSD

Gecko Rendering Engine issues

GEOM-specific discussions and implementations

Discussion of git use in the FreeBSD project

Porting GNOME and GNOME applications

General technical discussion

FreeBSD-specific Haskell issues and discussions

General discussion of hardware for running FreeBSD

FreeBSD Internationalization

FreeBSD on the IA-32 (Intel™ x86) platform

Porting FreeBSD to Intel™'s upcoming IA64 systems

Infiniband on FreeBSD

Technical discussion concerning the redesign of the IP firewall code

ISDN developers

Discussion about the jail(8) facility

Java™ developers and people porting JDK™s to FreeBSD

Porting KDE and KDE applications

Porting LFS to FreeBSD

Porting FreeBSD to MIPS™

Discussions about mobile computing

Mono and C# applications on FreeBSD

Multimedia applications

Technical discussions about bus architecture

Networking discussion and TCP/IP source code

757
freebsd-numerics
Discussions of high quality implementation of libm functions

freebsd-ocaml
FreeBSD-specific OCaml discussions

freebsd-office
Office applications on FreeBSD

freebsd-performance
Performance tuning questions for high performance/load installations

freebsd-perl
Maintenance of a number of Perl-related ports

freebsd-pf
Discussion and questions about the packet filter firewall system

freebsd-pkg
discussion
Binary package management and package tools discussion

freebsd-pkg-fallout
Fallout logs from package building

freebsd-pkgbase
Packaging the FreeBSD base system

freebsd-platforms
Concerning ports to non Intel™ architecture platforms

freebsd-ports
Discussion of the Ports Collection

freebsd-ports-announce
Important news and instructions about the Ports Collection (moderated)

freebsd-ports-bugs
Discussion of the ports bugs/PRs

freebsd-ppc
Porting FreeBSD to the PowerPC™

freebsd-proliant
Technical discussion of FreeBSD on HP ProLiant server platforms

freebsd-python
FreeBSD-specific Python issues

freebsd-rc
Discussion related to the rc.d system and its development

freebsd-realtime
Development of realtime extensions to FreeBSD

freebsd-ruby
FreeBSD-specific Ruby discussions

freebsd-scsi
The SCSI subsystem

freebsd-security
Security issues affecting FreeBSD

freebsd-small
Using FreeBSD in embedded applications
(obsolete; use freebsd-embedded instead)

freebsd-snapshots
FreeBSD Development Snapshot Announcements

freebsd-sparc64
Porting FreeBSD to SPARC™ based systems

freebsd-standards
FreeBSD's conformance to the C99 and the POSIX™ standards

freebsd-sysinstall
sysinstall(8) development

freebsd-tcltk
FreeBSD-specific Tcl/Tk discussions

freebsd-testing
Testing on FreeBSD

freebsd-tex
Porting TeX and its applications to FreeBSD

freebsd-threads
Threading in FreeBSD

freebsd-tilera
Porting FreeBSD to the Tilera family of CPUs

freebsd-tokenring
Support Token Ring in FreeBSD

freebsd-toolchain
Maintenance of FreeBSD's integrated toolchain
Limited lists:
The following lists are for more specialized (and demanding) audiences and are probably not of interest to the general public. It is also a good idea to establish a presence in the technical lists before joining one of these limited lists in order to understand the communications etiquette involved.

Digest lists:
All of the above lists are available in a digest format. Once subscribed to a list, the digest options can be changed in the account options section.

SVN lists:
The following lists are for people interested in seeing the log messages for changes to various areas of the source tree. They are Read-Only lists and should not have mail sent to them.
List Source area

svn-ports-head
/usr/ports
All changes to the "head" branch of the ports Subversion repository

svn-ports-svnadmin
/usr/ports
All changes to the administrative scripts, hooks, and other configuration data of the ports Subversion repository

svn-src-all
/usr/src
All changes to the src Subversion repository (except for user and projects)

svn-src-head
/usr/src
All changes to the "head" branch of the src Subversion repository (the FreeBSD-CURRENT branch)

svn-src-projects
/usr/projects
All changes to the projects area of the src Subversion repository

svn-src-release
/usr/src
All changes to the releases area of the src Subversion repository

svn-src-releng
/usr/src
All changes to the releng branches of the src Subversion repository (the security / release engineering branches)

svn-src-stable
/usr/src
All changes to the all stable branches of the src Subversion repository

svn-src-stable-6
/usr/src
All changes to the stable/6 branch of the src Subversion repository

svn-src-stable-7
/usr/src
All changes to the stable/7 branch of the src Subversion repository

svn-src-stable-8
/usr/src
All changes to the stable/8 branch of the src Subversion repository

svn-src-stable-9
/usr/src
All changes to the stable/9 branch of the src Subversion repository

svn-src-stable-10
/usr/src
All changes to the stable/10 branch of the src Subversion repository

svn-src-stable-11
/usr/src
All changes to the stable/11 branch of the src Subversion repository

svn-src-stable-12
/usr/src
All changes to the stable/12 branch of the src Subversion repository

svn-src-stable-other
/usr/src
All changes to the older stable branches of the src Subversion repository

svn-src-svnadmin
/usr/src
All changes to the administrative scripts, hooks, and other configuration data of the src Subversion repository
To subscribe to a list, click the list name at http://lists.FreeBSD.org/mailman/listinfo. The page that is displayed should contain all of the necessary subscription instructions for that list.

To actually post to a given list, send mail to listname@FreeBSD.org. It will then be redistributed to mailing list members worldwide.

To unsubscribe from a list, click on the URL found at the bottom of every email received from the list. It is also possible to send an email to listname-unsubscribe@FreeBSD.org to unsubscribe.

It is important to keep discussion in the technical mailing lists on a technical track. To only receive important announcements, instead join the FreeBSD announcements mailing list, which is intended for infrequent traffic.

C.2.3. 福toJson

All FreeBSD mailing lists have certain basic rules which must be adhered to by anyone using them. Failure to comply with these guidelines will result in two (2) written warnings from the FreeBSD Postmaster postmaster@FreeBSD.org, after which, on a third offense, the poster will removed from all FreeBSD mailing lists and filtered from further posting to them. We regret that such rules and measures are necessary at all, but today's Internet is a pretty harsh environment, it would seem, and many fail to appreciate just how fragile some of its mechanisms are.

Rules of the road:

• The topic of any posting should adhere to the basic charter of the list it is posted to. If the list is about technical issues, the posting should contain technical discussion. Ongoing irrelevant chatter or flaming only detracts from the value of the mailing list for everyone on it and will not be tolerated. For free-form discussion on no particular topic, the FreeBSD chat mailing list is freely available and should be used instead.

• No posting should be made to more than 2 mailing lists, and only to 2 when a clear and obvious need to post to both lists exists. For most lists, there is already a great deal of subscriber overlap and except for the most esoteric mixes (say "-stable & -scsi"), there really is no reason to post to more than one list at a time. If a message is received with multiple mailing lists on the Cc line, trim the Cc line before replying. The person who replies is still responsible for cross-posting, no matter who the originator might have been.

• Personal attacks and profanity (in the context of an argument) are not allowed, and that includes users and developers alike. Gross breaches of netiquette, like excerpting or reposting private mail when permission to do so was not and would not be forthcoming, are frowned upon but not specifically enforced. However, there are also very few cases where such content would fit within the charter of a list and it would therefore probably rate a warning (or ban) on that basis alone.

• Advertising of non-FreeBSD related products or services is strictly prohibited and will result in an immediate ban if it is clear that the offender is advertising by spam.

Individual list charters:

freebsd-acpi
ACPI and power management development
freebsd-afs
Andrew File System
This list is for discussion on porting and using AFS from CMU/Transarc

freebsd-announce
Important events / milestones
This is the mailing list for people interested only in occasional announcements of significant FreeBSD events. This includes announcements about snapshots and other releases. It contains announcements of new FreeBSD capabilities. It may contain calls for volunteers etc. This is a low volume, strictly moderated mailing list.

freebsd-arch
Architecture and design discussions
This list is for discussion of the FreeBSD architecture. Messages will mostly be kept strictly technical in nature. Examples of suitable topics are:
• How to re-vamp the build system to have several customized builds running at the same time.
• What needs to be fixed with VFS to make Heidemann layers work.
• How do we change the device driver interface to be able to use the same drivers cleanly on many buses and architectures.
• How to write a network driver.

freebsd-bluetooth
Bluetooth™ in FreeBSD
This is the forum where FreeBSD’s Bluetooth™ users congregate. Design issues, implementation details, patches, bug reports, status reports, feature requests, and all matters related to Bluetooth™ are fair game.

freebsd-bugbusters
Coordination of the Problem Report handling effort
The purpose of this list is to serve as a coordination and discussion forum for the Bugmeister, his Bugbusters, and any other parties who have a genuine interest in the PR database. This list is not for discussions about specific bugs, patches or PRs.

freebsd-bugs
Bug reports
This is the mailing list for reporting bugs in FreeBSD. Whenever possible, bugs should be submitted using the web interface to it.

freebsd-chat
Non technical items related to the FreeBSD community
This list contains the overflow from the other lists about non-technical, social information. It includes discussion about whether Jordan looks like a toon ferret or not, whether or not to type in capitals, who is drinking too much coffee, where the best beer is brewed, who is brewing beer in their basement, and so on. Occasional announcements of important events (such as upcoming parties, weddings, births, new jobs, etc) can be made to the technical lists, but the follow ups should be directed to this -chat list.

freebsd-chromium
FreeBSD-specific Chromium issues
This is a list for the discussion of Chromium support for FreeBSD. This is a technical list to
Running FreeBSD on various cloud platforms

This list discusses running FreeBSD on Amazon EC2, Google Compute Engine, Microsoft Azure, and other cloud computing platforms.

FreeBSD core team

This is an internal mailing list for use by the core members. Messages can be sent to it when a serious FreeBSD-related matter requires arbitration or high-level scrutiny.

FreeBSD-current

Discussions about the use of FreeBSD-CURRENT

This is the mailing list for users of FreeBSD-CURRENT. It includes warnings about new features coming out in -CURRENT that will affect the users, and instructions on steps that must be taken to remain -CURRENT. Anyone running "CURRENT" must subscribe to this list. This is a technical mailing list for which strictly technical content is expected.

FreeBSD-desktop

Using and improving FreeBSD on the desktop

This is a forum for discussion of FreeBSD on the desktop. It is primarily a place for desktop porters and users to discuss issues and improve FreeBSD's desktop support.

Dev-ci

Continuous Integration reports of build and test results

All Continuous Integration reports of build and test results

Dev-reviews

Notifications of work in progress in FreeBSD's review tool

Automated notifications of work in progress for review in FreeBSD's review tools, including patches.

FreeBSD-doc

Documentation Project

This mailing list is for the discussion of issues and projects related to the creation of documentation for FreeBSD. The members of this mailing list are collectively referred to as "The FreeBSD Documentation Project". It is an open list; feel free to join and contribute!

FreeBSD-drivers

Writing device drivers for FreeBSD

This is a forum for technical discussions related to device drivers on FreeBSD. It is primarily a place for device driver writers to ask questions about how to write device drivers using the APIs in the FreeBSD kernel.

FreeBSD-dtrace

Using and working on DTrace in FreeBSD

DTrace is an integrated component of FreeBSD that provides a framework for understanding the kernel as well as user space programs at run time. The mailing list is an archived discussion for developers of the code as well as those using it.
FreeBSD users of Eclipse IDE, tools, rich client applications and ports. The intention of this list is to provide mutual support for everything to do with choosing, installing, using, developing and maintaining the Eclipse IDE, tools, rich client applications on the FreeBSD platform and assisting with the porting of Eclipse IDE and plugins to the FreeBSD environment. The intention is also to facilitate exchange of information between the Eclipse community and the FreeBSD community to the mutual benefit of both.

Although this list is focused primarily on the needs of Eclipse users it will also provide a forum for those who would like to develop FreeBSD specific applications using the Eclipse framework.

Using FreeBSD in embedded applications
This list discusses topics related to using FreeBSD in embedded systems. This is a technical mailing list for which strictly technical content is expected. For the purpose of this list, embedded systems are those computing devices which are not desktops and which usually serve a single purpose as opposed to being general computing environments. Examples include, but are not limited to, all kinds of phone handsets, network equipment such as routers, switches and PBXs, remote measuring equipment, PDAs, Point Of Sale systems, and so on.

Emulation of other systems such as Linux/MS-DOS™/Windows™
This is a forum for technical discussions related to running programs written for other operating systems on FreeBSD.

Enlightenment
Discussions concerning the Enlightenment Desktop Environment for FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

Peer support of FreeBSD-related software that is no longer supported by the FreeBSD Project.
This list is for those interested in providing or making use of peer support of FreeBSD-related software for which the FreeBSD Project no longer provides official support in the form of security advisories and patches.

FireWire™ (iLink, IEEE 1394)
This is a mailing list for discussion of the design and implementation of a FireWire™ (aka IEEE 1394 aka iLink) subsystem for FreeBSD. Relevant topics specifically include the standards, bus devices and their protocols, adapter boards/cards/chips sets, and the architecture and implementation of code for their proper support.

Fortran on FreeBSD
This is the mailing list for discussion of Fortran related ports on FreeBSD: compilers, libraries, scientific and engineering applications from laptops to HPC clusters.

File systems
Discussions concerning FreeBSD filesystems. This is a technical mailing list for which strictly technical content is expected.
This is a technical list for discussions related to bringing games to FreeBSD. It is for individuals actively working on porting games to FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

Gecko Rendering Engine

This is a forum about Gecko applications using FreeBSD. Discussion centers around Gecko Ports applications, their installation, their development and their support within FreeBSD.

GEOM

Discussions specific to GEOM and related implementations. This is a technical mailing list for which strictly technical content is expected.

Use of git in the FreeBSD project

Discussions of how to use git in FreeBSD infrastructure including the github mirror and other uses of git for project collaboration. Discussion area for people using git against the FreeBSD github mirror. People wanting to get started with the mirror or git in general on FreeBSD can ask here.

GNOME

Discussions concerning The GNOME Desktop Environment for FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

Infiniband on FreeBSD

Technical mailing list discussing Infiniband, OFED, and OpenSM on FreeBSD.

IP Firewall

This is the forum for technical discussions concerning the redesign of the IP firewall code in FreeBSD. This is a technical mailing list for which strictly technical content is expected.

Porting FreeBSD to IA64

This is a technical mailing list for individuals actively working on porting FreeBSD to the IA-64 platform from Intel™, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

ISDN Communications

This is the mailing list for people discussing the development of ISDN support for FreeBSD.
freebsd-java

Java™ Development
This is the mailing list for people discussing the development of significant Java™ applications for FreeBSD and the porting and maintenance of JDK™s.

freebsd-jobs

Jobs offered and sought
This is a forum for posting employment notices specifically related to FreeBSD and resumes from those seeking FreeBSD-related employment. This is not a mailing list for general employment issues since adequate forums for that already exist elsewhere.

Note that this list, like other FreeBSD.org mailing lists, is distributed worldwide. Be clear about the geographic location and the extent to which telecommuting or assistance with relocation is available.

Email should use open formats only — preferably plain text, but basic Portable Document Format (PDF), HTML, and a few others are acceptable to many readers. Closed formats such as Microsoft™ Word (.doc) will be rejected by the mailing list server.

freebsd-kde

KDE
Discussions concerning KDE on FreeBSD systems. This is a technical mailing list for which strictly technical content is expected.

freebsd-hackers

Technical discussions
This is a forum for technical discussions related to FreeBSD. This is the primary technical mailing list. It is for individuals actively working on FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome. This is a technical mailing list for which strictly technical content is expected.

freebsd-hardware

General discussion of FreeBSD hardware
General discussion about the types of hardware that FreeBSD runs on, various problems and suggestions concerning what to buy or avoid.

freebsd-hubs

Mirror sites
Announcements and discussion for people who run FreeBSD mirror sites.

freebsd-isp

Issues for Internet Service Providers
This mailing list is for discussing topics relevant to Internet Service Providers (ISPs) using FreeBSD. This is a technical mailing list for which strictly technical content is expected.

freebsd-mono

Mono and C# applications on FreeBSD
This is a list for discussions related to the Mono development framework on FreeBSD. This is a technical mailing list. It is for individuals actively working on porting Mono or C# applications to FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.
FreeBSD-specific OCaml discussions

This is a list for discussions related to the OCaml support on FreeBSD. This is a technical mailing list. It is for individuals working on OCaml ports, 3rd party libraries and frameworks. Individuals interested in the technical discussion are also welcome.

Office applications on FreeBSD

Discussion centers around office applications, their installation, their development and their support within FreeBSD.

Project Infrastructure Announcements

This is the mailing list for people interested in changes and issues related to the FreeBSD.org Project infrastructure. This moderated list is strictly for announcements: no replies, requests, discussions, or opinions.

Discussions about tuning or speeding up FreeBSD

This mailing list exists to provide a place for hackers, administrators, and/or concerned parties to discuss performance related topics pertaining to FreeBSD. Acceptable topics includes talking about FreeBSD installations that are either under high load, are experiencing performance problems, or are pushing the limits of FreeBSD. Concerned parties that are willing to work toward improving the performance of FreeBSD are highly encouraged to subscribe to this list. This is a highly technical list ideally suited for experienced FreeBSD users, hackers, or administrators interested in keeping FreeBSD fast, robust, and scalable. This list is not a question-and-answer list that replaces reading through documentation, but it is a place to make contributions or inquire about unanswered performance related topics.

Discussion and questions about the packet filter firewall system

Discussion concerning the packet filter (pf) firewall system in terms of FreeBSD. Technical discussion and user questions are both welcome. This list is also a place to discuss the ALTQ QoS framework.

Binary package management and package tools discussion

Discussion of all aspects of managing FreeBSD systems by using binary packages to install software, including binary package toolkits and formats, their development and support within FreeBSD, package repository management, and third party packages. Note that discussion of ports which fail to generate packages correctly should generally be considered as ports problems, and so inappropriate for this list.

Fallout logs from package building

All packages building failures logs from the package building clusters.

Packaging the FreeBSD base system.

Discussions surrounding implementation and issues regarding packaging the FreeBSD base system.
This is the mailing list for people working on the SCSI subsystem for FreeBSD. This is a technical mailing list for which strictly technical content is expected.

freebsd-security

Security issues
FreeBSD computer security issues (DES, Kerberos, known security holes and fixes, etc). This is a technical mailing list for which strictly technical discussion is expected. Note that this is not a question-and-answer list, but that contributions (BOTH question AND answer) to the FAQ are welcome.

freebsd-security-notifications

Security Notifications
Notifications of FreeBSD security problems and fixes. This is not a discussion list. The discussion list is FreeBSD-security.

freebsd-small

Using FreeBSD in embedded applications
This list discusses topics related to unusually small and embedded FreeBSD installations. This is a technical mailing list for which strictly technical content is expected.

freebsd-embedded

This list has been obsoleted by freebsd-embedded.

freebsd-snapshots

FreeBSD Development Snapshot Announcements
This list provides notifications about the availability of new FreeBSD development snapshots for the head/ and stable/ branches.

freebsd-stable

Discussions about the use of FreeBSD-STABLE
This is the mailing list for users of FreeBSD-STABLE. "STABLE" is the branch where development continues after a RELEASE, including bug fixes and new features. The ABI is kept stable for binary compatibility. It includes warnings about new features coming out in -STABLE that will affect the users, and instructions on steps that must be taken to remain -STABLE. Anyone running "STABLE" should subscribe to this list. This is a technical mailing list for which strictly technical content is expected.

freebsd-standards

C99 & POSIX Conformance
This is a forum for technical discussions related to FreeBSD Conformance to the C99 and the POSIX standards.

freebsd-teaching

Teaching with FreeBSD
Non technical mailing list discussing teaching with FreeBSD.

freebsd-testing

Testing on FreeBSD
Technical mailing list discussing testing on FreeBSD, including ATF/Kyua, test build infrastructure, port tests to FreeBSD from other operating systems (NetBSD, … etc.).
freebsd-tex
Porting TeX and its applications to FreeBSD
This is a technical mailing list for discussions related to TeX and its applications on FreeBSD. It is for individuals actively working on porting TeX to FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

freebsd-toolchain
Maintenance of FreeBSD's integrated toolchain
This is the mailing list for discussions related to the maintenance of the toolchain shipped with FreeBSD. This could include the state of Clang and GCC, but also pieces of software such as assemblers, linkers and debuggers.

freebsd-transport
Discussions of transport level network protocols in FreeBSD
The transport mailing list exists for the discussion of issues and designs around the transport level protocols in the FreeBSD network stack, including TCP, SCTP and UDP. Other networking topics, including driver specific and network protocol issues should be discussed on the FreeBSD networking mailing list.

freebsd-translators
Translating FreeBSD documents and programs
A discussion list where translators of FreeBSD documents from English into other languages can talk about translation methods and tools. New members are asked to introduce themselves and mention the languages they are interested in translating.

freebsd-usb
Discussing FreeBSD support for USB
This is a mailing list for technical discussions related to FreeBSD support for USB.

freebsd-user-groups
User Group Coordination List
This is the mailing list for the coordinators from each of the local area Users Groups to discuss matters with each other and a designated individual from the Core Team. This mail list should be limited to meeting synopsis and coordination of projects that span User Groups.

freebsd-virtualization
Discussion of various virtualization techniques supported by FreeBSD
A list to discuss the various virtualization techniques supported by FreeBSD. On one hand the focus will be on the implementation of the basic functionality as well as adding new features. On the other hand users will have a forum to ask for help in case of problems or to discuss their use cases.

freebsd-wip-status
FreeBSD Work-In-Progress Status
This mailing list can be used by developers to announce the creation and progress of FreeBSD related work. Messages will be moderated. It is suggested to send the message "To:" a more topical FreeBSD list and only "BCC:" this list. This way the WIP can also be discussed on the topical list, as no discussion is allowed on this list.

An editorial digest of the messages to this list might be posted to the FreeBSD website every few 770
Discussions of 802.11 stack, tools, device driver development

The FreeBSD-wireless list focuses on 802.11 stack (sys/net80211), device driver and tools development. This includes bugs, new features and maintenance.

Discussion of the FreeBSD port to Xen™ implementation and usage

A list that focuses on the FreeBSD Xen™ port. The anticipated traffic level is small enough that it is intended as a forum for both technical discussions of the implementation and design details as well as administrative deployment issues.

XFCE

This is a forum for discussions related to bringing the XFCE environment to FreeBSD. This is a technical mailing list. It is for individuals actively working on porting XFCE to FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

Zope

This is a forum for discussions related to bringing the Zope environment to FreeBSD. This is a technical mailing list. It is for individuals actively working on porting Zope to FreeBSD, to bring up problems or discuss alternative solutions. Individuals interested in following the technical discussion are also welcome.

The FreeBSD mailing lists are filtered in multiple ways to avoid the distribution of spam, viruses, and other unwanted emails. The filtering actions described in this section do not include all those used to protect the mailing lists.

Only certain types of attachments are allowed on the mailing lists. All attachments with a MIME content type not found in the list below will be stripped before an email is distributed on the mailing lists.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch

Some of the mailing lists might allow attachments of other MIME content types.
the above list should be applicable for most of the mailing lists. If an email contains both an HTML and a plain text version, the HTML version will be removed. If an email contains only an HTML version, it will be converted to plain text.

C.3. Usenet

In addition to two FreeBSD specific newsgroups, there are many others in which FreeBSD is discussed or are otherwise relevant to FreeBSD users.

C.3.1. BSD 專属新聞群

- comp.unix.bsd.freebsd.announce
- comp.unix.bsd.freebsd.misc
- de.comp.os.unix.bsd (German)
- fr.comp.os.bsd (French)

C.3.2. 其他相關的UNIX™新新聞群

- comp.unix
- comp.unix.questions
- comp.unix.admin
- comp.unix.programmer
- comp.unix.shell
- comp.unix.misc
- comp.unix.bsd

C.3.3. X 視窗系統

- comp.windows.x

C.4. 官方鏡像站

Central Servers

<table>
<thead>
<tr>
<th>Country</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Armenia</td>
<td><a href="http://www.at.FreeBSD.org/">http://www.at.FreeBSD.org/</a> (IPv6)</td>
</tr>
<tr>
<td>Australia</td>
<td><a href="http://www.au.FreeBSD.org/">http://www.au.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://www2.au.FreeBSD.org/">http://www2.au.FreeBSD.org/</a></td>
</tr>
<tr>
<td>Austria</td>
<td></td>
</tr>
<tr>
<td>Czech</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>Denmark</td>
<td>Finland</td>
</tr>
<tr>
<td></td>
<td>France</td>
</tr>
<tr>
<td></td>
<td>Germany</td>
</tr>
<tr>
<td></td>
<td>Hong Kong</td>
</tr>
<tr>
<td></td>
<td>Ireland</td>
</tr>
<tr>
<td></td>
<td>Japan</td>
</tr>
<tr>
<td></td>
<td>Latvia</td>
</tr>
<tr>
<td></td>
<td>Lithuania</td>
</tr>
<tr>
<td></td>
<td>Netherlands</td>
</tr>
<tr>
<td></td>
<td>Norway</td>
</tr>
<tr>
<td></td>
<td>Russia</td>
</tr>
<tr>
<td></td>
<td>Slovenia</td>
</tr>
<tr>
<td></td>
<td>South Africa</td>
</tr>
<tr>
<td></td>
<td>Spain</td>
</tr>
<tr>
<td></td>
<td>Sweden</td>
</tr>
<tr>
<td></td>
<td>Switzerland</td>
</tr>
<tr>
<td></td>
<td>Taiwan</td>
</tr>
<tr>
<td></td>
<td>United Kingdom</td>
</tr>
<tr>
<td></td>
<td>United States of America</td>
</tr>
</tbody>
</table>

(as of UTC)
Czech Republic • http://www.cz.FreeBSD.org/ (IPv6)

Denmark • http://www.dk.FreeBSD.org/ (IPv6)

Finland • http://www.fi.FreeBSD.org/

France • http://www1.fr.FreeBSD.org/

Germany • http://www.de.FreeBSD.org/ (IPv6)

Hong Kong • http://www.hk.FreeBSD.org/

Ireland • http://www.ie.FreeBSD.org/

Japan • http://www.jp.FreeBSD.org/ (IPv6)

Latvia • http://www.lv.FreeBSD.org/

Lithuania • http://www.lt.FreeBSD.org/

Netherlands • http://www.nl.FreeBSD.org/

Norway • http://www.no.FreeBSD.org/

Russia • http://www.ru.FreeBSD.org/ (IPv6)

Slovenia • http://www.si.FreeBSD.org/

South Africa • http://www.za.FreeBSD.org/
<table>
<thead>
<tr>
<th>Country</th>
<th>Websites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spain</td>
<td>• <a href="http://www.es.FreeBSD.org/">http://www.es.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www2.es.FreeBSD.org/">http://www2.es.FreeBSD.org/</a></td>
</tr>
<tr>
<td>Sweden</td>
<td>• <a href="http://www.se.FreeBSD.org/">http://www.se.FreeBSD.org/</a></td>
</tr>
<tr>
<td>Switzerland</td>
<td>• <a href="http://www.ch.FreeBSD.org/">http://www.ch.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www2.ch.FreeBSD.org/">http://www2.ch.FreeBSD.org/</a></td>
</tr>
<tr>
<td>(IPv6)</td>
<td></td>
</tr>
<tr>
<td>Taiwan</td>
<td>• <a href="http://www.tw.FreeBSD.org/">http://www.tw.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www2.tw.FreeBSD.org/">http://www2.tw.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www4.tw.FreeBSD.org/">http://www4.tw.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www5.tw.FreeBSD.org/">http://www5.tw.FreeBSD.org/</a></td>
</tr>
<tr>
<td>(IPv6)</td>
<td></td>
</tr>
<tr>
<td>United Kingdom</td>
<td>• <a href="http://www1.uk.FreeBSD.org">http://www1.uk.FreeBSD.org</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www3.uk.FreeBSD.org/">http://www3.uk.FreeBSD.org/</a></td>
</tr>
<tr>
<td>United States of America</td>
<td>• <a href="http://www5.us.FreeBSD.org/">http://www5.us.FreeBSD.org/</a></td>
</tr>
<tr>
<td></td>
<td>• <a href="http://www3.us.FreeBSD.org/">http://www3.us.FreeBSD.org/</a></td>
</tr>
<tr>
<td>(IPv6)</td>
<td></td>
</tr>
</tbody>
</table>
The OpenPGP keys of the FreeBSD.org officers are shown here. These keys can be used to verify a signature or send encrypted email to one of the officers. A full list of FreeBSD OpenPGP keys is available in the PGP Keys article. The complete keyring can be downloaded at pgpkeyring.txt.

D.1. Officers

D.1.1. Security Officer Team

<security-officer@FreeBSD.org>

*pub* rsa4096/D9AD2A18057474CB 2022-12-11 [C] [expires: 2026-01-24]

Key fingerprint = 0BE3 3275 D74C 953C 79F8 1107 D9AD 2A18 0574 74CB

uid                            FreeBSD Security Officer <security-officer@freebsd.org>

*sub* rsa4096/6E58DE901F001AEF 2022-12-11 [S] [expires: 2025-01-15]

*sub* rsa4096/46DB26D62F6039B7 2022-12-11 [E] [expires: 2025-01-15]

-----BEGIN PGP PUBLIC KEY BLOCK-----