

SUSE Linux Enterprise Desktop

11

www.novell.com

2009 1 15

管理指南



管理指南

所有內容的版權皆屬於 © 2006- 2009 Novell, Inc.

法律事項

本手冊受 Novell 智慧財產權保護。重製、複製或散發本手冊即表示您已明確同意遵循本授權合約的條款及細則。

若不違反下列條件，本手冊便得以使用原樣或隨附於電子或列印格式的套件方式，自由地重製、複製和散發：

作者和撰稿者的著作權標示和名稱清楚標示於所有重製、複製和散發的副本上。本手冊 (尤其是列印版本) 的重製和/或散發並不以營利為目的。在使用本手冊的任何部分之前，必須獲得 Novell, Inc 的明確授權。

如需 Novell 的商標，請參閱位於 <http://www.novell.com/company/legal/trademarks/tmlist.html> 的 Novell 商標和服務標誌清單。* Linux 是 Linus Torvalds 的註冊商標。所有其他協力廠商商標，為各所有人所有之財產。商標符號(®、™等)表示 Novell 的商標；星號(*) 則代表協力廠商的商標。

本手冊中所有資訊在編輯時，都已全力注意各項細節。但這不保證百分之百的正確性。因此，Novell, Inc.、SUSE LINUX Products GMBH、作者或譯者都不需對任何錯誤或造成的結果負責。

目錄

關於本指南	ix
部 I 支援與一般任務	1
1 YaST 線上更新	3
1.1 使用 Qt 介面手動安裝修補程式	4
1.2 使用 gtk 介面手動安裝修補程式	6
1.3 自動線上更新	8
2 收集系統資訊以供支援所用	9
2.1 Novell 支援連結綜覽	9
2.2 使用 Supportconfig	10
2.3 向 Novell 提交資訊	12
2.4 如需更多資訊	13
3 文字模式的 YaST	15
3.1 在模組中瀏覽	16
3.2 組合鍵的限制	17
3.3 YaST 指令行選項	18
4 使用指令行工具管理軟體	21
4.1 使用 Zypper	21
4.2 RPM — 套件管理員	26

5	使用 Nomad 存取遠端桌面	37
5.1	Nomad 的先決條件	38
5.2	安裝與設定	39
5.3	使用 Nomad	40
5.4	疑難排解	40
5.5	如需更多資訊	41
6	Bash 和 Bash 程序檔	43
6.1	什麼是「外圍程序」?	43
6.2	寫入外圍程序程序檔	49
6.3	重新指向指令事件	49
6.4	使用別名	50
6.5	使用 Bash 中的變數	51
6.6	分組與組合指令	53
6.7	使用通用流程建構元	54
6.8	如需更多資訊	55
部 II	系統	57
7	64 位元系統環境的 32 位元和 64 位元應用程式	59
7.1	執行期間支援	59
7.2	軟體開發	60
7.3	Biarch 平台的軟體編譯	60
7.4	核心規格	61
8	啟動及設定 Linux 系統	63
8.1	Linux 開機程序	63
8.2	init 程序	66
8.3	透過 /etc/sysconfig 設定系統	74
9	開機載入程式 GRUB	77
9.1	使用 GRUB 開機	78
9.2	使用 YaST 設定開機載入程式	85
9.3	解除安裝 Linux 開機載入程式	91
9.4	建立開機 CD	91
9.5	圖形化 SUSE 畫面	93
9.6	疑難排解	93
9.7	如需更多資訊	94

10	特殊系統功能	95
10.1	特殊軟體套件的資訊	95
10.2	虛擬主控台	102
10.3	鍵盤配置	102
10.4	語言與國家專用的設定	103
11	印表機操作	107
11.1	列印系統的工作流程	108
11.2	連接印表機的方法和通訊協定	109
11.3	安裝軟體	109
11.4	網路印表機	110
11.5	圖形列印介面	112
11.6	由指令行開始列印	113
11.7	SUSE Linux Enterprise Desktop 中的特殊功能	113
11.8	疑難排解	115
12	使用 udev 進行動態核心設備管理	123
12.1	/dev 目錄	123
12.2	核心 uevent 和 udev	124
12.3	驅動程式、核心模組和設備	124
12.4	開機和初始設備設定	125
12.5	監控執行中的 udev 精靈	125
12.6	透過 udev 規則影響核心設備事件處理	126
12.7	永久設備命名	133
12.8	udev 使用的檔案	133
12.9	如需更多資訊	134
13	X Window System	135
13.1	手動設定 X Window System	135
13.2	安裝與設定字型	141
13.3	如需更多資訊	146
14	使用 FUSE 存取檔案系統	147
14.1	設定 FUSE	147
14.2	裝載 NTFS 分割區	147
14.3	使用 SSHFS 裝載遠端檔案系統	148
14.4	裝載 ISO 檔案系統	149
14.5	可用的 FUSE 外掛程式	149
14.6	如需更多資訊	149

部 III 行動電腦 151

15 Linux 的行動計算功能 153

15.1	筆記型電腦	153
15.2	行動硬體	160
15.3	行動電話和 PDA	161
15.4	如需更多資訊	161

16 電源管理 163

16.1	省電功能	163
16.2	ACPI	164
16.3	硬碟的休眠	168
16.4	疑難排解	169
16.5	如需更多資訊	171

17 使用 Tablet PC 173

17.1	安裝 Tablet PC 套件	174
17.2	設定手寫板設備	175
17.3	使用虛擬鍵盤	176
17.4	旋轉顯示器	176
17.5	使用筆勢辨識	177
17.6	使用筆來做筆記和素描	179
17.7	疑難排解	181
17.8	如需更多資訊	183

部 IV 服務 185

18 基本網路 187

18.1	IP 位址與路由	190
18.2	IPv6—下一代的網際網路	193
18.3	名稱解析	200
18.4	使用 YaST 手動設定網路連線	202
18.5	NetworkManager	220
18.6	手動設定網路連線	221
18.7	smpppd 做為撥號助理	235

19 無線通訊 237

19.1	無線區域網路	237
------	------------------	-----

20	網路中的 SLP 服務	247
20.1	安裝	247
20.2	啟用 SLP	248
20.3	SUSE Linux Enterprise Desktop 中的 SLP 前端	248
20.4	透過 SLP 提供服務	248
20.5	如需更多資訊	249
21	使用 NTP 進行時間同步化	251
21.1	使用 YaST 設定 NTP 用戶端	251
21.2	手動設定網路中的 ntp	254
21.3	設定本地參考時鐘	254
22	使用 NetworkManager	257
22.1	NetworkManager 的使用案例	257
22.2	啟用 NetworkManager	258
22.3	設定網路連線	258
22.4	使用 KDE NetworkManager 工具集	260
22.5	使用 GNOME NetworkManager Applet	260
22.6	NetworkManager 與 VPN	262
22.7	NetworkManager 和安全性	263
22.8	常見問題解答	265
22.9	疑難排解	266
22.10	如需更多資訊	267
23	Samba	269
23.1	術語	269
23.2	設定 Samba 伺服器	270
23.3	設定用戶端	271
23.4	做為登入伺服器的 Samba	271
23.5	如需更多資訊	272
24	使用 NFS 共享檔案系統	273
24.1	安裝必要軟體	273
24.2	以 YaST 輸入檔案系統	273
24.3	手動輸入檔案系統	274
24.4	NFS 搭配使用 Kerberos	276
24.5	如需更多資訊	276

25	檔案同步化	279
25.1	可用的資料同步化軟體	279
25.2	選取程式時所要考慮的決定性因素	280
25.3	CVS 簡介	283
25.4	rsync 簡介	286
25.5	如需更多資訊	287

關於本指南

本指南的主要對象為操作 SUSE® Linux Enterprise 期間的專業網路和系統管理員。因此，本指南的重點只在於確保 SUSE Linux Enterprise 的設定正確，而且網路所需服務都已備妥，讓它可以像初始安裝一樣正常運作。至於如何確保 SUSE Linux Enterprise 能夠與您企業的應用程式軟體正確相容，或其核心功能能否符合您的要求，則不在本指南的討論範圍。本文假設您可符合完整需求，且需要安裝或檢測是否符合完整需求的測試安裝。

本指南包含下列內容：

管理

SUSE Linux Enterprise 提供範圍廣泛的工具，可針對系統的各個方面進行自定。這個部分介紹其中幾項。

系統

學習這部份，進一步瞭解作業系統基礎。SUSE Linux Enterprise 支援多種硬體結構，您可以此自行打造可執行於 SUSE Linux Enterprise 上的應用程式。開機載入程式和開機程序資訊會協助您瞭解 Linux 系統的運作方式，以及您個人的自定程序檔和應用程式如何與作業系統融合。

行動運算

需要特別注意筆記型電腦以及行動設備 (例如 PDA 或行動電話與 SUSE Linux Enterprise) 之間的通訊。請注意省電，並留心不同設備與不斷變化的網路環境的整合。另外，請瞭解提供所需功能的背景技術。

服務

SUSE Linux Enterprise 的設計目的是要做為網路作業系統。SUSE® Linux Enterprise Desktop 包含支援許多網路服務的用戶端。它可與異質環境完全整合，包括 MS Windows 伺服器 and 用戶端。

本手冊的許多章節包含連到其他文件資源的連結。這包括系統和網際網路上所提供的其他文件。

如需適用於產品的文件與最新文件更新的綜覽，請參閱 <http://www.novell.com/documentation>。

1 可用文件

我們提供不同語言的 HTML 和 PDF 版本的書籍。此產品提供了下列使用者和管理員手冊：

GNOME 使用者指南 (↑*GNOME 使用者指南*)

介紹 SUSE Linux Enterprise Desktop 的 GNOME 桌面。本指南將引導您使用與設定桌面，並協助您執行主要任務。主要適用對象為希望將 GNOME 桌面做為預設桌面有效利用的終端使用者。

應用程式指南 (↑*應用程式指南*)

瞭解如何使用與設定 SUSE Linux Enterprise Desktop 上的主要桌面應用程式。本指南介紹瀏覽器、電子郵件用戶端及辦公室應用程式與協作工具。同時也介紹圖形與多媒體應用程式。

部署指南 (↑*部署指南*)

顯示如何安裝單個或多個系統，以及如何利用產品內在功能部署基礎結構。從本地安裝或使用網路安裝伺服器，到使用遠端控制、高度自定及自動安裝技術進行大量部署，有各式各樣的做法供您選擇。

管理指南 [1頁]

描述系統管理任務，如維護、監控及自定初始安裝的系統。

Security Guide (↑*Security Guide*)

介紹系統安全性的基本概念，涵蓋了本地安全性與網路安全性方面。顯示如何利用產品內含的 Novell AppArmor 等安全性軟體 (可讓您根據程式指定該程式可能會讀取、寫入及執行的檔案) 或可以可靠地收集關於所有安全性相關事件資訊的稽核系統。

系統分析與調整指南 (↑*系統分析與調整指南*)

用於偵測、解決及最佳化問題的管理員指南。其中描述了如何透過監控工具檢查並最佳化系統，以及如何有效管理資源。該指南還概述了常見問題與解決方案，以及其他說明與文件資源。

使用 Xen 的虛擬系統 (↑*使用 Xen 的虛擬系統*)

介紹了產品的的虛擬化技術。它提供各種應用程式和每個 SUSE Linux Enterprise Server 支援平台的安裝類型概觀，以及安裝程序的簡短說明。

除綜合性手冊外，還提供了以下幾個快速入門指南：

快速安裝 (↑快速安裝)

列出系統要求，並提供透過 DVD 或 ISO 影像安裝 SUSE Linux Enterprise Desktop 的逐步指南。

Linux Audit 快速入門

簡要說明如何啟用和設定稽核系統，以及如何執行關鍵任務，例如設定稽核規則、產生報告和分析記錄檔案等。

Novell AppArmor 快速入門

可幫助您理解 Novell® AppArmor 背後的主要概念。

大部分 SUSE Linux Enterprise Desktop 手冊的 HTML 版本位於所安裝系統的 `/usr/share/doc/manual` 目錄下，或桌面的說明中心中。<http://www.novell.com/documentation> 中提供了最新的文件更新，您可於此處下載 PDF 或 HTML 版本的產品手冊。

2 意見反應

以下為可供使用的數種意見回應管道：

- 若要報告產品元件的錯誤，或是提交增強功能要求，請使用 <https://bugzilla.novell.com/>。如果您不熟悉 Bugzilla，Novell Bugzilla 首頁中的 *Bug Writing FAQs* 或許會有幫助。
- 我們希望得到您對本手冊以及本產品隨附之其他文件的意見和建議。請使用線上文件每頁下方的使用備註功能，並在其中輸入您的意見。

3 文件慣例

本手冊使用下列印刷慣例：

- `/etc/passwd`：檔案名稱和目錄名稱
- `placeholder`：以實際的值來取代 `placeholder`
- `PATH`：環境變數 `PATH`

- `ls, --help`: 指令、選項和參數
- `user`: 使用者或群組
- `Alt, Alt+F1`: 供人按下的按鍵或案件組合；顯示的按鍵與鍵盤上一樣為大寫
- 「檔案」、「檔案」>「另存新檔」: 功能表項目、按鈕
- *Dancing Penguins* (*Penguins* 章節, ↑另一本手冊): 這是另一本手冊之章節的參照。

第 I. 支援與一般任務 部分

YaST 線上更新

Novell 為您的產品提供持續的軟體安全性更新。依預設，openSUSE Updater 用於保持您的系統為最新狀態。如需有關 openSUSE Updater 的詳細資訊，請參閱第 7.5 節「Keeping the System Up-to-date」(第 7 章 *Installing or Removing Software*, ↑部署指南)。本章介紹了用於更新軟體套件的替代工具：YaST 線上更新。

適用於 SUSE® Linux Enterprise Desktop 的目前修補程式位於更新軟體儲存庫中。如果在安裝期間已註冊了產品，則表明已設定更新儲存庫。如果未註冊 SUSE Linux Enterprise Desktop，則可以按照以下方式進行註冊：在 YaST 中執行「軟體」>「線上更新組態」，然後啟動「進階」>「註冊以獲得支援，並獲得更新儲存庫」。或者，您可以從信任的來源手動新增更新儲存庫。若要新增或移除儲存庫，請透過 YaST 中的「軟體」>「軟體儲存庫」啟動儲存庫管理員。第 7.4 節「Adding Software Repositories」(第 7 章 *Installing or Removing Software*, ↑部署指南) 中提供了有關儲存庫管理員的詳細資訊。

注：存取更新目錄時發生錯誤

如果您無法存取更新目錄，可能表示訂閱已過期。SUSE Linux Enterprise Desktop 通常提供一年或三年的訂閱，您可以在這個時間段內存取更新目錄。一旦訂閱期結束，此存取權將被拒絕。

存取更新目錄遭拒絕時，系統會顯示一則警告訊息，建議您造訪 Novell Customer Center 以查看您的訂閱。可在 <http://www.novell.com/center/> 上找到 Novell Customer Center。

提供了具有不同關聯層級的更新。安全性更新可修復嚴重的安全性威脅，必須予以安裝。建議使用更新可修復可能危害電腦的問題，而選擇性更新可修復非安全性相關問題或僅提供增強功能。

若要使用 YaST 安裝更新與改良，請在 YaST 中執行「軟體」>「線上更新」。目前您系統可用的所有新修補程式(選購程式除外)均已標示為即將安裝。按一下「接受」或「套用」自動安裝這些修補程式。安裝完成之後，請按「結束」確認。您的系統現在已是最新狀態。

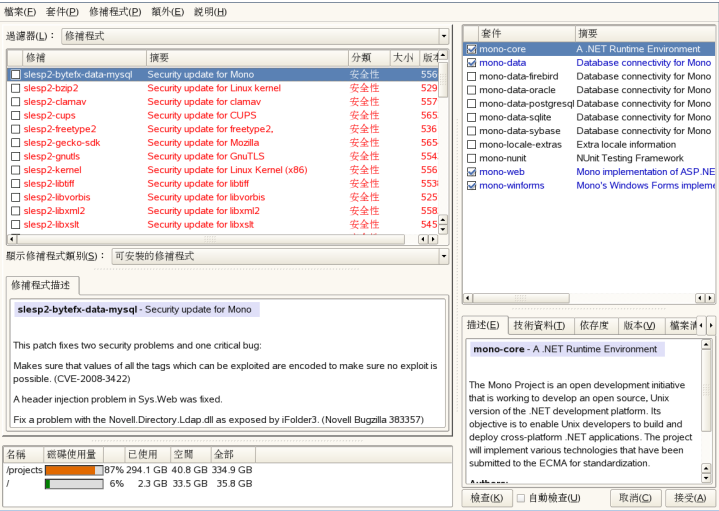
提示: 禁用 `deltarpm`

依預設更新將做為 `deltarpm` 下載。由於從 `deltarpm` 重建 `rpm` 套件需要佔用大量的記憶體和 CPU 時間，出於效能考量，某些設定或硬體組態可能要求禁用 `deltarpm`。若要禁用 `deltarpm`，請編輯檔案 `/etc/zypp/zypp.conf`，並將 `download.use_deltarpm` 設定為 `false`。

1.1 使用 Qt 介面手動安裝修補程式

「線上更新」視窗包含四個區段。左邊是所有可用的修補程式清單。修補程式清單下方會顯示所選修補程式的描述。右欄列出所選修補程式所包含的套件(修補程式可包含多組套件)，且下方會列出所選套件的詳細描述。或者，磁碟使用情形可以顯示在左欄底部(此畫面依預設為淡出—使用點狀滑動軸可使其顯示出來)。

圖形 1.1 YaST 線上更新



修補程式畫面會列出適用於 SUSE Linux Enterprise Desktop 的可用修補程式。修補程式根據安全性關聯程度排序。安全性、建議使用與選擇性。修補程式存在三種不同的檢視窗。使用「顯示修補程式類別」可以切換不同的檢視窗：

需要的修補程式 (預設檢視窗)

適用於系統上安裝的套件但目前尚未安裝的修補程式。

不需要的修補程式

適用於系統上未安裝之套件的修補程式，或已滿足其需求的修補程式 (因為已從其他來源進行更新)。

所有修補程式

可用於 SUSE Linux Enterprise Desktop 的所有修補程式。

清單項目包含符號與修補程式名稱。如需可能符號清單，請按 **Shift + F1**。「安全性」和「建議」狀態的修補程式所要求的動作已自動預設。這些動作是「自動安裝」、「自動更新」或「自動刪除」。「選擇性」修補程式的動作未預設—用滑鼠右鍵按一下修補程式，然後從清單中選擇一個動作。

如果從更新儲存庫以外的其他儲存庫安裝最新的套件，則可以使用此安裝來滿足此套件之修補程式的需求。在此情況下，會有一個核取標記顯示在修補程式摘要的前面。僅當您將修補程式標示為已安裝時，該修補程式才會顯示在清單

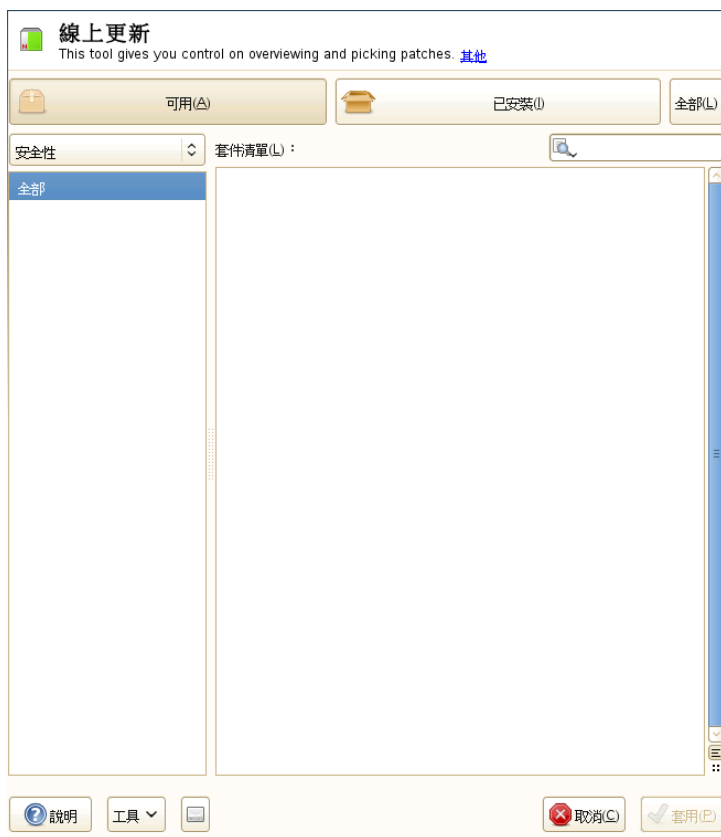
中。事實上這並沒有安裝修補程式 (因為套件已經是最新的)，而是將修補程式標示為已安裝。

大部分的修補程式都會包含多套件的更新。若要變更對單一套件所執行的動作，請在套件視窗中的套件上按一下滑鼠右鍵，並選擇動作。當您標示完所有要處理的修補程式與套件後，請按一下「接受」繼續。

1.2 使用 gtk 介面手動安裝修補程式

「線上更新」視窗包含兩個主要部份。左側窗格列出所有修補程式，並為修補程式清單提供不同的過濾器。右側窗格顯示變更清單，當您按一下「套用」後，便會執行這些變更。

圖形 1.2 YaST 線上更新



修補程式清單過濾器

可用的

適用於系統上安裝的套件但目前尚未安裝的修補程式。

已安裝

已安裝的修補程式。

全部

已安裝或可用的修補程式。

嚴重程度

僅顯示「選擇性」、「推薦」或「安全性」修補程式。預設會顯示「所有」修補程式。

「儲存機制」

此過濾器可讓您按儲存庫顯示修補程式。

「套件清單」

在此處套用您的自定過濾器。

按一下修補程式項目可開啟詳細描述左側窗格底部區域中修補程式的資訊列。此處可檢視修補程式的詳細描述及可用版本。您也可以選擇「安裝」選擇性修補程式 — 系統已預先選取安全性與推薦的修補程式以進行安裝。

1.3 自動線上更新

YaST 也可讓您設定自動更新。開啟「軟體」>「線上更新組態」。核取「自動線上更新」並選擇更新頻率，即「每日」、「每週」及「每月」。某些修補程式 (例如核心更新) 需要使用者介入，而這可能會讓自動更新程序停止下來。因此，若希望更新程序完全自動執行，應核取「略過互動式修補程式」。核取後，應不時執行手動「線上更新」以安裝需要互動的修補程式。

收集系統資訊以供支援所用

若有問題出現，可使用 `supportconfig` 收集系統資訊，例如目前使用的核心版本、硬體、RPM 資料庫、分割區及其他。所得結果可用來協助 Novell 支援中心找出問題所在。

2.1 Novell 支援連結綜覽

Novell 支援連結 (NSL) 是 SUSE Linux Enterprise Desktop 的新功能，此工具會收集系統資訊，並可讓您將該資訊上傳到其他伺服器以進行進一步分析。Novell 支援中心使用 Novell 支援連結來收集問題伺服器的系統資訊，並將資訊傳送至 Novell 的公用 FTP 伺服器。收集的系統資訊包括：目前正在使用的核心版本、硬體、RPM 資料庫、分割區及其他。其分析結果可用於幫助 Novell 支援中心解決您開啟的服務要求。

有兩種方式可以使用 Novell 支援連結：

1. 使用 YaST 支援模組，
2. 使用指令行公用程式 `supportconfig`。

YaST 支援模組會呼叫 `supportconfig` 來收集系統資訊。

2.2 使用 Supportconfig

以下幾節描述如何透過 YaST 或從指令行使用 supportconfig，以及您可以使用的選項。

2.2.1 使用 YaST 收集資訊

要使用 YaST 收集系統資訊，請執行下列步驟：

- 1 開啟 URL <http://www.novell.com/center/eservice>，然後建立服務要求號碼。
- 2 啟動 YaST。
- 3 開啟「支援」模組。
- 4 按一下「建立報告 Tar 聚合檔」。
- 5 從選項按鈕清單中選取選項。如果要先進行測試，請使用「僅收集最少量的資訊」。按「下一步」繼續。
- 6 輸入您的聯絡人資訊。使用來自 **步驟 1** [10頁] 的服務要求號碼，將其輸入到標示為「Novell 11 位數服務要求號碼」的文字欄位中。按「下一步」繼續。
- 7 資訊收集即會開始。該程序完成後，按「下一步」繼續。
- 8 檢閱資料集合，如果不需要某個檔案名稱，請使用「從資料移除」。按「下一步」繼續。
- 9 儲存 Tar 聚合檔。如果要上載至 Novell 客戶中心，請確定啟用「將記錄檔案 Tar 聚合檔上載至 URL」。按「下一步」完成操作。

2.2.2 直接使用 Supportconfig 收集資訊

若要在指令行中使用 supportconfig，請執行下列步驟：

- 1 開啟外圍程序，切換為 root 身份。
- 2 在不使用任何選項的情況下執行 `supportconfig` 來收集預設系統資訊。
- 3 等待此工具完成操作。
- 4 預設歸檔位於 `/var/log` 中，檔案名稱格式為 `nts_主機_日期_時間.tbz`

2.2.3 Supportconfig 通用選項

`supportconfig` 公用程式有多個啟動選項。您可以使用 `supportconfig -h` 或線上文件檢視這些選項。一般情況下，`supportconfig` 執行時不使用任何選項。以下是一些其他通用啟動選項的摘要：

- 使用精簡選項 (`-m`) 縮小所收集之資訊的大小：

```
supportconfig -m
```

- 在輸出中包含其他聯絡人資訊 (全部在一行上)：

```
supportconfig -E tux@example.org -N "Tux Penguin" -O "Penguin Inc." ...
```

- 進行疑難排解時，您可能只想收集目前要解決之問題所在區域的資訊。例如，如果 LVM 有問題，且您是在最近使用預設 `supportconfig` 輸出時發現的該問題。在對其進行變更之後，您希望收集目前的 LVM 資訊。以下指令僅收集最精簡的 `supportconfig` 資訊與 LVM。

```
supportconfig -i LVM
```

要檢視完整的功能清單，請執行：

```
supportconfig -F
```

- 使用 `-u` 與 `-r` 選項上載附有關聯服務要求號碼的 `supportconfig` Tar 聚合檔。例如，如果您已開啟 Novell 服務要求，追蹤號碼是 12345678901，則請執行以下指令：

```
supportconfig -ur 12345678901
```

2.3 向 Novell 提交資訊

可以使用 YaST 支援模組或 `supportconfig` 指令行公用程式向 Novell 提交系統資訊。當您遇到伺服器問題，希望獲取 Novell 的協助時，您將需要開啟服務要求並向 Novell 提交您的伺服器資訊。下文將描述使用 YaST 與指令行這兩種方法。

過程 2.1 使用 YaST 向 Novell 提交資訊

- 1 開啟 URL <http://www.novell.com/center/eservice>，然後建立服務要求號碼。
 - 2 記下您的 11 位數服務要求號碼。以下範例假設服務要求號碼為 12345678901。
 - 3 在 YaST 的支援模組視窗中按一下「建立報告 Tar 聚合檔」。
 - 4 選取「使用自定」選項圓鈕。按「下一步」繼續。
 - 5 輸入聯絡人資訊，填寫「Novell 11 位數服務要求號碼」及 Novell 的上載目標 URL。
 - 對於安全上載目標，請使用：<https://secure-www.novell.com/upload?apname=supportconfig&file={tarball}>。
 - 對於一般 FTP 上載目標，請使用：<ftp://ftp.novell.com/incoming>。
- 按「下一步」繼續。系統即開始收集資訊。該程序完成後，按「下一步」繼續。
- 6 檢閱資料集合，並使用「從資料移除」移除任何您想要從上載到 Novell 的 Tar 聚合檔排除的檔案。按「下一步」繼續。
 - 7 依預設，`/root` 中會儲存一份 Tar 聚合檔的副本。確認您使用的是上文描述的其中一個 Novell 上載目標，並且「將記錄檔案 Tar 聚合檔上載至 URL」已啟用。按「下一步」完成操作。
 - 8 按一下「完成」。

過程 2.2 使用 *supportconfig* 將資訊提交至 Novell

- 1 開啟 URL <http://www.novell.com/center/eservice>，然後建立服務要求號碼。
- 2 記下您的 11 位數服務要求號碼。以下範例假設服務要求號碼為 12345678901。
- 3 伺服器連接至網際網路：

3a 要使用預設上載目標，請執行：

```
supportconfig -ur 12345678901
```

3b 對於安全上載目標，請在同一行上使用以下指令：

```
supportconfig -r 12345678901 -U  
'https://secure-www.novell.com/upload?appname=supportconfig&file={tarball}'
```

- 4 伺服器未連接至網際網路

4a 執行以下指令：

```
supportconfig -r 12345678901
```

4b 手動將 `/var/log/nts_SR12345678901*tbz` Tar 聚合檔上載至 Novell 的 FTP 伺服器 (<ftp://ftp.novell.com/incoming>)。

4c 也可以使用服務要求 URL <http://www.novell.com/center/eservice> 將 Tar 聚合檔附加至服務要求。

- 5 Tar 聚合檔位於 <ftp://ftp.novell.com/incoming> 目錄中後，便會被自動附加到您的服務要求中。

2.4 如需更多資訊

下列文件提供了關於收集系統資訊的詳細資訊：

- `man supportconfig` — `supportconfig` 的線上文件

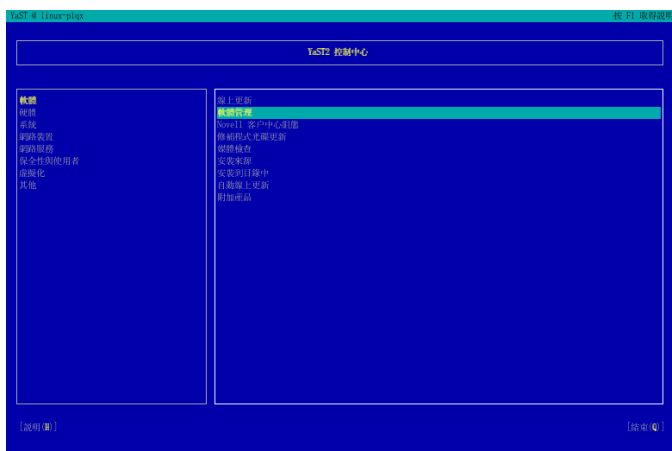
- `man supportconfig.conf` — `supportconfig` 組態檔案的線上文件
- <http://www.novell.com/communities/print/node/4097> — 使用 `Supportconfig` 進行基本伺服器狀態檢查
- <http://www.novell.com/communities/print/node/4827> — 建立自己的 `Supportconfig` 外掛程式
- <http://www.novell.com/communities/print/node/4800> — 建立中心 `Supportconfig` 儲存庫

文字模式的 YaST

本章節對象為未在其系統上執行 X 伺服器，且依賴以文字為基礎的安裝工具的系統管理員及進階使用者。這裡提供了一些基本資訊，說明如何在文字模式中啟動與操作 YaST。

文字模式中的 YaST 使用 ncurses 程式庫來提供簡易的虛擬圖形化使用者介面。依預設，ncurses 程式庫已安裝。要執行 YaST 之最小受支援的終端模擬器大小為 80x25 個字元。

圖形 3.1 文字模式下的 YaST 主視窗



在文字模式中啟動 YaST 時會先出現 YaST 控制中心，請參閱圖形 3.1。主要視窗包含 3 個區域。由粗的白色框線所圍繞的左框架，內含一些不同模組所屬的類別。會以彩色背景來表示作用中的類別。由細的白色框線所圍繞的右框架，

內含作用中類別的可用模組的綜覽。下方框架中有「說明」按鈕與「結束」按鈕。

啟動 YaST 控制中心時，會自動選取「軟體」類別。您可以使用↓與↑來變更類別。若要啟動所選取的類別中的某個模組，請按→。此模組選項會加上一個粗的框線。您可以使用↓與↑以選取想要的模組。您可以按住方向鍵不放來捲動可用模組清單。選取模組後，模組標題會以彩色背景顯示。

您可以按 **Enter** 以啟動想要的模組。模組中的各個按鈕或選項欄位中，都有一個不同顏色的字母(預設為黃色)。您可以使用 **Alt + yellow_letter** 來直接選取按鈕，而毋須使用 **Tab** 進行瀏覽。按 **Alt + Q** 或選取「結束」並按 **Enter** 來結束 YaST 控制中心。

3.1 在模組中瀏覽

以下對 YaST 模組的控制元件做出說明，假設所有的功能鍵及 **Alt** 組合鍵都起作用，且未被指定不同的全域功能。如需有關可能的例外狀況的資訊，請參閱第 3.2 節「組合鍵的限制」[17頁]。

在按鈕與選項清單中瀏覽

使用 **Tab** 可以在按鈕與包含選項清單的框架之間進行瀏覽。若要反向瀏覽，請使用 **Alt + Tab** 組合或 **Shift + Tab** 組合。

在選項清單中瀏覽

在作用中且包含選項清單的框架中，您可以使用方向鍵(↑和↓)以便在其中的個別元件之間進行瀏覽。如果框架中個別項目超出其寬度，您可以使用 **Shift + →** 或 **Shift + ←** 以水平方式向右及向左捲動。或者使用 **Ctrl + E** 或 **Ctrl + A**。如果使用 **→** 或 **←**，會導致作用中的框架或目前的選項清單變更，如同在控制中心內一樣，則您也可以使用此組合鍵。

按鈕、圓形按鈕以及核取方塊

如果要選取有空白方括號(核取方塊)的按鈕，或是有空白括號(選項圓鈕)的按鈕，請按空格鍵或 **Enter** 鍵。或者，直接使用 **Alt + yellow_letter** 來選取選項圓鈕與核取方塊。在此狀況中，您不需按 **Enter** 來做確認。如果您使用 **Tab** 瀏覽至某個項目，按 **Enter** 即可執行所選取的動作或啟用個別的功能表項目。

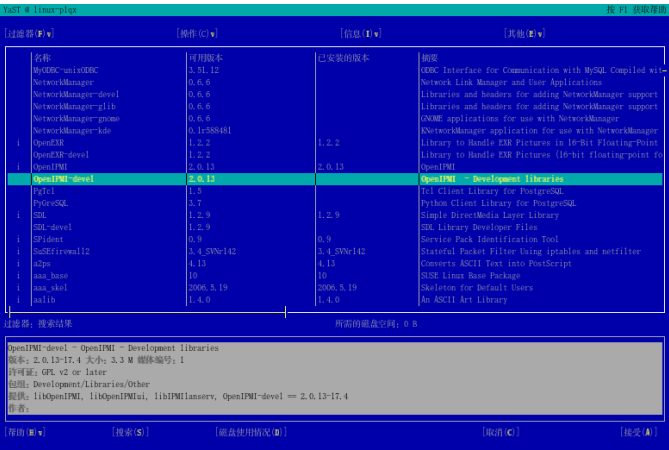
功能鍵

F 鍵 (F1 到 F12) 可用來快速存取不同的按鈕。可用的 F 鍵捷徑顯示在 YaST 螢幕的結算底線中。因為不同的模組提供不同的按鈕設定 (詳細資訊、資訊、新增、刪除等等)，所以各個功能鍵所實際對應的按鈕是依照作用中的 YaST 模組而定。您可以將 F10 當作「接受」、「確定」、「下一步」以及「完成」來使用。按 F1 可存取 YaST 說明。

在 ncurses 模式中使用導覽樹狀結構

某些 YaST 模組在視窗的左側使用導覽樹狀結構以便選取組態對話方塊。在 ncurses 模式中，於導覽樹狀結構中選取後必須按 Enter 才能顯示選取的對話方塊。此行為的目的是為了在瀏覽導覽樹狀結構時節省費時的重新描繪時間。

圖形 3.2 軟體安裝模組



3.2 組合鍵的限制

如果您的視窗管理員使用了全部的 Alt 組合，則 YaST 中的 Alt 組合可能無效。像是 Alt 或 Shift 等鍵也可能事先已由終端機的設定所佔用。

以 Esc 取代 Alt

您可以使用 Esc 來取代 Alt，而不使用 Alt。例如，Esc-H 可取代 Alt+H。(先按 Esc，然後按 H。)

您可以使用 **Alt** 與來往前瀏覽和往後瀏覽 **Ctrl + F** **Ctrl + B**

如果 **Alt** 和 **Shift** 組合已先由視窗管理員或終端機所佔用，則請使用 **Ctrl + F** 組合 (往前) 與 **Ctrl + B** 組合 (往後) 來代替。

功能鍵的限制

有些功能會使用 **F** 鍵。有些特定的功能鍵可能已由終端機所佔用，無法供 YaST 使用。不過，在純文字主控台中，應該都可以使用各種的 **Alt** 組合鍵與功能鍵。

3.3 YaST 指令行選項

除文字模式介面之外，YaST 還提供了純指令行介面。若要取得 YaST 指令行選項的清單，請輸入：

```
yast -h
```

3.3.1 啟動個別模組

如果要節省時間，您可以直接啟動個別的 YaST 模組。若要啟動模組，請輸入：

```
yast <module_name>
```

使用 `yast -l` 或 `yast --list`，則可以檢視一個清單，其中包含您系統中所有可用的模組名稱。例如，使用 `yast lan` 可啟動網路模組。

3.3.2 透過指令行安裝套件

如果您知道套件名稱，並且此套件由任何使用中的安裝儲存庫所提供，則可以使用指令行選項 `-i` 來安裝套件。

```
yast -i <package_name>
```

或

```
yast --install <package_name>
```

套件名稱可以是單一短套件名稱，例如 `gvim` (安裝時進行相依性檢查)，也可以是 `rpm` 套件的完整路徑 (安裝時不進行相依性檢查)。

如果您需要以指令行為基礎的軟體管理公用程式以及 YaST 無法提供的功能，可以考慮使用 `zypper`。這個新的公用程式使用相同的軟體管理程式庫，此程式庫也是 YaST 套件管理員的基礎。[第 4.1 節「使用 Zypper」](#) [21 頁] 中介紹了 `zypper` 的基本用法。

3.3.3 YaST 模組的指令行參數

為使用程序檔中的 YaST 功能，YaST 提供了可支援個別模組的指令行。並不是所有模組都有指令行支援。若要顯示某模組的可用選項，請輸入：

```
yast <module_name> help
```

如果某模組未提供指令行支援，則此模組將在文字模式中啟動，並且系統會顯示以下訊息：

```
This YaST module does not support the command line interface.
```


使用指令行工具管理軟體

本章介紹了兩個用於管理軟體的指令行工具：Zypper 與 RPM。

4.1 使用 Zypper

Zypper 是用於安裝與更新套件的指令行工具。zypper 語法與 `rug` 語法類似。與 `rug` 相比，`zypper` 不需要視情境執行 `zmd` 精靈。如需有關 `rug` 相容性的詳細資訊，請參閱 http://en.opensuse.org/Zypper/Usage#Compatibility_with_Rug。它在完成遠端軟體管理任務或透過外圍程序程序檔管理軟體時非常有用。

`zypper` 內建有說明綜覽：

```
zypper help
```

4.1.1 一般使用情形

`zypper` 的一般語法為：

```
zypper [global-options] command [command-options] [arguments] ...
```

括號中的部分為非必需。執行 `zypper` 的最簡單的方法就是輸入其名稱，後面跟著指令。例如，若要將所有需要的修補程式套用至系統類型：

```
zypper update
```

此外，您還可以在指令之前輸入全域選項來從一或多個全域選項中進行選擇。例如，根據您的需要，`--non-interactive` 表示執行指令時不詢問任何問題：

```
zypper --non-interactive update
```

若要使用特定於特殊指令的選項，請在指令之後輸入這些選項。例如，`--auto-agree-with-licenses` 表示將所有需要的修補程式套用至系統時不要求確認任何授權—系統已提前讀取所有授權。

```
zypper update --auto-agree-with-licenses
```

某些指令需要一個或多個引數：

```
zypper install mplayer
```

某些選項也需要引數。以下指令行表示以較新套件更新系統：

```
zypper update -t package
```

以上所有組合均表示，僅使用 `factory` 儲存庫進行安裝且詳加說明：

```
zypper -v install --repo factory mplayer amarok
```

您也可以使用 `+/-` 或 `~/!` 切換鍵一次性安裝及移除套件。

```
zypper install emacs -vim
```

或：

```
zypper remove emacs +vim
```

或者如果您選擇對指定的第一個套件使用 `-`，則必須在它前面輸入 `--` 以防止系統將其當成指令選項：

```
zypper install -- -vim emacs
```

4.1.2 使用 Zypper 安裝和移除軟體

若要透過註冊的儲存庫安裝套件，請使用

```
zypper install package_name
```

`zypper` 還支援萬用字元。若要安裝以 *套件名稱* 開頭的所有套件，請使用

```
zypper install package_name*
```

您也可以直接安裝本地或遠端 RPM — Zypper 也會根據情況自動安裝套件 `package_name`：

```
zypper install http://www.example.com/package_name.rpm
```

要避免解析相依性，請使用 `--no-recommends` 或 `--no-force-resolution`。

若要移除已安裝的套件，請使用

```
zypper remove package_name
```

警告：不要移除系統的必要套件

不要移除 `glibc`、`zypper`、`kernel` 或類似套件。這些套件都是系統的必要套件，如若缺少則系統可能會停止工作。

依預設，`zypper` 會在安裝或移除選定套件之前要求您確認。使用 `--non-interactive` 選項可覆寫此行為。此選項必須位於實際模式 (安裝、移除與更新) 之前，例如

```
zypper --non-interactive install package_name
```

此選項允許在程序檔與 `cron` 工作中使用 `zypper`。

若要安裝某套件所對應的來源套件，請使用

```
zypper source-install package_name
```

使用此指令還可安裝指定套件的相依版本。如果不想如此，請新增開關 `--no-build-deps`，如下所示：

```
zypper source-install --no-build-deps package_name
```

當然，僅當您擁有其來源套件已新增至儲存庫清單的儲存庫時，此指令行才有效。如需有關新增儲存庫的詳細資訊，請參閱[第 4.1.4 節「管理儲存庫」](#) [24頁]。

變更已安裝的軟體基礎後，請驗證是否仍然滿足所有相依性：

```
zypper verify
```

4.1.3 使用 Zypper 更新軟體

使用 `zypper` 可以兩種不同的方式更新軟體。若要將所有正式發佈的修補程式整合至您的系統，只需執行

```
zypper update
```

在此情況下，系統會檢查儲存庫中所有可用修補程式的關聯性，並視需要進行安裝。

如果儲存庫中有新套件但未提供修補程式，則 `zypper update` 不起任何作用。若要更新所有這些套件，必須指定安裝 `package` 類型的更新。

```
zypper update -t package
```

若要更新個別套件，請使用安裝指令：

```
zypper install package_name
```

所有可用新套件的清單可透過以下指令獲得：

```
zypper list-updates -t package
```

4.1.4 管理儲存庫

`zypper` 的所有安裝或更新指令均依賴於 `zypper` 可識別的儲存庫清單。若要列出系統可識別的所有儲存庫，請使用指令：

```
zypper repos
```

結果類似於以下輸出：

#	Enabled	Refresh	Type	Alias	Name
1	Yes	Yes	yast2	openSUSE-DVD 11.0	openSUSE-DVD 11.0
2	Yes	No	yast2	Main (OSS)	Main (OSS)
3	Yes	No	yast2	Main (Non-OSS)	Main (Non-OSS)

若要從清單中移除儲存庫，請使用 `zypper renamerepo` 指令加上要刪除的儲存庫別名。若要從範例中移除主要儲存庫（非 OSS），請使用以下指令：

```
zypper renamerepo Main Repository (Non-OSS)
```

若要新增儲存庫，請執行

```
zypper addrepo URI Alias
```

URI 可以是網際網路儲存庫 (請造訪 http://en.opensuse.org/Additional_YaST_Package_Repositories 獲取可用儲存庫的清單), 也可以是目錄、CD 或 DVD。別名是儲存庫的縮寫與唯一識別碼。您可以隨意選擇, 前提是它必須是唯一的。如果指定了已使用的別名, **zypper** 會發出警告。

4.1.5 查詢

您可以使用各種查詢指令, 如 `search`、`info` 或 `what-provides`。

`search` 用於查詢套件名稱並在輸出第一欄中顯示狀態 (s) 資訊。

`info` 使用套件名稱做為引數顯示有關套件的詳細資訊。

`what-provides` 套件類似於 `rpm -q --whatprovides` 套件, 但是 `rpm` 只能查詢 RPM 資料庫 (即所有已安裝套件的資料庫)。另外, **Zypper** 會告知您任何儲存庫功能的提供者, 而不是只有已安裝的儲存庫。

特別是若要進行除錯, 可以使用 `--plus-repo`、`--disable-repositories` 與 `--disable-system-resolvables` 開關。如果想僅在一個儲存庫內進行搜尋, 可以使用這些開關。如需詳細的使用情形資訊, 請參閱 **zypper** 線上文件 (`man zypper`)。

4.1.6 使用 Zypper 外圍程序

有時必須按照序列執行數個不同的 **zypper** 指令。若要阻止 **zypper** 重新讀取每個 **zypper** 指令的所有資料庫, 可以在外圍程序模式中執行 **zypper**。

```
zypper shell
```

當外圍程序在執行中時, 只需使用各自的參數執行 **zypper** 指令：

```
zypper> in zsh
...
zypper> exit
```

使用 **zypper** 外圍程序通常速度比較快, 因為所有相關資料都保留在記憶體內。

Zypper 支援讀取行程式庫。也就是說，您可以在 Zypper 外圍程序中使用所有指令行編輯功能，這些功能也能在 Bash 外圍程序中使用。Zypper 可在 `~/.zypper_history` 檔案中維護其指令歷程。

4.1.7 如需更多資訊

如需有關透過指令行進行更新的詳細資訊，請輸入 `zypper --help` 或參閱 `zypper(8)` 線上文件。如需範例及詳細資訊，請造訪 <http://en.opensuse.org/Zypper/Usage>。

4.2 RPM — 套件管理員

RPM (RPM 套件管理員) 用於管理軟體套件。主要指令為 `rpm` 及 `rpmbuild`。使用者、系統管理員和套件建立者可在威力強大的 RPM 資料庫中查詢已安裝軟體的詳細資訊。

`rpm` 主要包括五種模式：安裝、解除安裝或更新軟體套件；重建 RPM 資料庫；查詢 RPM 基礎或個別的 RPM 歸檔；套件完整性檢查；簽署套件。`rpmbuild` 可用於建立初始來源的可安裝套件。

可安裝的 RPM 歸檔以特殊二進位格式包裝封裝。這些歸檔由要安裝的程式檔和 `rpm` 在安裝期間用來設定軟體套件或儲存在 RPM 資料庫中供記錄之用的特定中繼資訊所組成。RPM 歸檔的副檔名通常為 `.rpm`。

提示: 軟體開發套件

對於許多套件，軟體開發所需的元件 (程式庫、標頭、Include 檔案等) 分別封裝在單獨的套件中。只有在您想要自行編譯軟體 (例如最新的 GNOME 套件) 時，才需要這些開發套件。由副檔名 `-devel` 即可識別出開發套件，例如 `alsa-devel`、`gimp-devel` 和 `kdelibs3-devel` 套件。

4.2.1 確認套件驗證性

RPM 套件具有 GnuPG 簽章。包含指紋的金鑰為：

```
1024D/9C800ACA 2000-10-19 SuSE Package Signing Key <build@suse.de>  
Key fingerprint = 79C1 79B2 E1C8 20C1 890F 9994 A84E DAE8 9C80 0ACA
```

`rpm --checksig package-1.2.3.rpm` 指令可用來驗證 RPM 套件的簽章，以判斷它確實來自 SUSE，還是來自其他可信任的設備。特別建議在從網際網路更新套件時使用此指令。SUSE 公用套件簽章金鑰通常位於 `/root/.gnupg/` 中。在 `/usr/lib/rpm/gnupg/` 目錄中也額外放置了金鑰，以便讓一般使用者確認 RPM 套件的簽名。

4.2.2 管理套件：安裝、更新和解除安裝

一般而言，安裝 RPM 歸檔很簡單：`rpm -i package.rpm`。使用此指令可安裝套件，但是必須滿足套件的相依性，而且套件不可與其他套件衝突。如果 rpm 要求要安裝的套件必須符合相依性要求，會顯示錯誤訊息。RPM 資料庫會在背景中確認沒有產生衝突——特定的檔案僅可屬於一個套件。藉由選擇不同選項，您可以強迫 rpm 忽略這些預設，但只有進階使用者才可以這樣做。否則，會有危害系統完整性的風險，而且可能會危害更新系統的能力。

選項 `-U` 或 `--upgrade` 和 `-F` 或 `--freshen` 可用來更新套件，例如，`rpm -F package.rpm`。此指令會移除舊版的檔案，並立刻安裝新檔案。兩個版本之間的不同在於 `-U` 會安裝先前系統中沒有的套件，而 `-F` 僅更新先前安裝的套件。在更新時，rpm 會使用下列策略小心地更新組態檔：

- 如果系統管理員未變更組態檔，rpm 會安裝新版本的相應檔案。系統管理員不需要做任何動作。
- 如果系統管理員在更新之前變更了組態檔，rpm 會將變更的檔案以副檔名 `.rpmorig` 或 `.rpmsave` (備份檔案) 儲存，並安裝新套件的版本，但此操作僅在原始安裝的檔案與新版本不同時才會發生。在這種情況下，請比較備份檔案 (`.rpmorig` 或 `.rpmsave`) 與新安裝的檔案，然後再對新檔案做一次變更。之後，請確定刪除所有 `.rpmorig` 和 `.rpmsave` 檔案以避免未來更新的問題。
- `.spec` 如果組態檔已存在，且如果在檔案中指定了 `noreplace` 標籤，便會出現 `.rpmnew` 檔案。

在更新之後，應該在比較完 `.rpmsave` 和 `.rpmnew` 之後將它們移除，才不會妨礙未來的更新。如果 RPM 資料庫之前無法辨識檔案，會指定 `.rpmorig` 副檔名。

否則，會使用 `.rpmsave`。換言之，`.rpmorig` 是在將外來格式更新為 RPM 後產生的。`.rpmsave` 是在將舊版 RPM 更新為新版 RPM 後產生的。`.rpmnew` 不會透露任何關於系統管理員是否曾對組態檔做過任何變更的資訊。可在 `/var/adm/rpmconfigcheck` 找到這些檔案的清單。部分組態檔 (如 `/etc/httpd/httpd.conf`) 不會覆寫以允許後續操作。

`-U` 切換參數的功能不不完全等同於使用 `-e` 選項進行解除安裝以及使用 `-i` 選項進行安裝。如果可能，請使用 `-U`。

若要移除套件，請輸入 `rpm -e package`。如果沒有無法解析的相依性，rpm 僅會刪除套件。只要其他應用程式還需要它，理論上無法刪除 Tcl/Tk。即使是這種情況下，RPM 還是可從資料庫呼叫以得到協助。如果此種刪除是 (不論出於何種原因或處於不尋常的狀況下) 不可行的 — 即使不存在額外的相依性，使用選項 `--rebuilddb` 來重建 RPM 資料庫可能有幫助。

4.2.3 RPM 與修補程式

為了確保系統的操作安全性，必須經常在系統上安裝更新套件。以前，要除去套件中的錯誤，只能夠更換整個套件。在大型套件中，如果有包含錯誤的小檔案就很容易產生大量的資料。但是，SUSE RPM 提供在套件中安裝修補程式的功能。

最重要的考量可用 `pine` 當作範例：

修補程式 RPM 是否適用於我的系統？

若要進行檢查，首先請查詢安裝的套件版本。以 `pine` 為例，可使用指令

```
rpm -q pine
pine-4.44-188
```

然後檢查修補程式 RPM 是否適用於此版本的 `pine`：

```
rpm -qp --basedon pine-4.44-224.i586.patch.rpm
pine = 4.44-188
pine = 4.44-195
pine = 4.44-207
```


此修補程式適用於三種不同版本的 **Pine**。範例中也列出安裝的版本，因此可安裝此修補程式。

修補程式會更換哪些檔案？

受到修補程式影響的檔案可在修補程式 **RPM** 中輕易地看出。`rpm` 參數 `-P` 可讓您選擇特殊的修補程式功能。可使用以下指令顯示檔案清單：

```
rpm -qpPl pine-4.44-224.i586.patch.rpm
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

或者，如果已經安裝修補程式，可使用以下指令：

```
rpm -qPl pine
/etc/pine.conf
/etc/pine.conf.fixed
/usr/bin/pine
```

修補程式 **RPM** 如何安裝於系統中？

修補程式 **RPM** 可做為一般 **RPM** 使用。唯一的不同是必須已經安裝適合的 **RPM**。

系統中已經安裝哪個修補程式，是用於哪個套件版本？

使用指令 `rpm -qPa` 可顯示系統中已經安裝的所有修補程式清單。如果新系統中僅安裝一個修補程式 (如本範例)，則清單顯示如下：

```
rpm -qPa
pine-4.44-224
```

如果在日後，您想要知道原始安裝的套件版本，可在 **RPM** 資料庫中找到此資訊。以 `pine` 為例，可使用以下指令顯示此資訊：

```
rpm -q --basedon pine
pine = 4.44-188
```

可在 `rpm` 以及 `rpmbuild` 的 `man` 頁面中找到包括 **RPM** 修補程式功能的相關資訊。

4.2.4 Delta RPM 套件

Delta RPM 套件包含舊版與新版 **RPM** 套件之間的差異。在舊版 **RPM** 上套用 `delta RPM` 會產生完整的新版 **RPM**。但是您不需要取得舊版的 **RPM**，因為 `delta RPM` 也可以和已安裝的 **RPM** 搭配使用。`delta RPM` 套件的大小比修補程式 **RPM** 還

小，這一特點有利於透過網際網路傳送更新套件。缺點是使用 **delta RPM** 的更新作業會比一般或修補程式 **RPM** 消耗更多的 CPU 週期。

`prepdeltarpm`、`writedeltarpm` 以及 `applydeltarpm` 二進位檔屬於 **delta RPM** 套裝軟體 (`deltarpm` 套件) 的一部分，可協助您建立和套用 **delta RPM** 套件。使用下列指令可建立名稱為 `new.delta.rpm` 的 **delta RPM**。下列指令假設 `old.rpm` 和 `new.rpm` 都已存在：

```
prepdeltarpm -s seq -i info old.rpm > old.cpio
prepdeltarpm -f new.rpm > new.cpio
xdelta delta -0 old.cpio new.cpio delta
writedeltarpm new.rpm delta info new.delta.rpm
```

最後，移除暫存工作檔案 `old.cpio`、`new.cpio` 以及 `delta`。

如果已經安裝舊套件，使用 `applydeltarpm` 即可從檔案系統重新建構新 **RPM**：

```
applydeltarpm new.delta.rpm new.rpm
```

若不要存取檔案系統，而要從舊 **RPM** 產生新 **RPM**，請使用 `-r` 選項：

```
applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

請參閱 `/usr/share/doc/packages/deltarpm/README` 以取得技術細節。

4.2.5 RPM 查詢

使用 `-q` 選項，`rpm` 會啟動查詢，可檢查 **RPM** 歸檔 (藉由新增選項 `-p`)，也可以查詢安裝套件的 **RPM** 資料庫。有多個切換參數可用於指定所需的資訊類型。請參閱 [表格 4.1 「最重要的 RPM 查詢選項」](#) [30頁]。

表格 4.1 最重要的 *RPM* 查詢選項

<code>-i</code>	套件資訊
<code>-l</code>	檔案清單
<code>-f FILE</code>	查詢包含 <i>FILE</i> 檔案的套件 (完整的路徑必須以 <i>FILE</i> 指定)

-s	含有狀態資訊的檔案清單 (隱含 -l)
-d	只列出文件檔案 (隱含 -l)
-c	只列出組態檔案 (隱含 -l)
--dump	含有完整詳細資訊的檔案清單 (與 -l、-c 或 -d 一起搭配使用)
--provides	列出另一個套件可以使用 --requires 要求的套件功能
--requires、-R	套件所需的功能
--scripts	安裝程序檔 (預先安裝、後續安裝、解除安裝)

例如，`rpm -q -i wget` 指令可顯示如 **範例 4.1** 「`rpm -q -i wget`」[31頁] 中所顯示的資訊。

範例 4.1 `rpm -q -i wget`

```

Name           : wget                               Relocations: (not relocatable)
Version        : 1.9.1                               Vendor: SUSE LINUX AG,
Nuernberg, Germany
Release        : 50                                   Build Date: Sat 02 Oct 2004
03:49:13 AM CEST
Install date: Mon 11 Oct 2004 10:24:56 AM CEST      Build Host: f53.suse.de
Group          : Productivity/Networking/Web/Utilities  Source RPM:
wget-1.9.1-50.src.rpm
Size           : 1637514                             License: GPL
Signature      : DSA/SHA1, Sat 02 Oct 2004 03:59:56 AM CEST, Key ID
a84edae89c800aca
Packager       : http://www.suse.de/feedback
URL            : http://wget.sunsite.dk/
Summary        : A tool for mirroring FTP and HTTP servers
Description    :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
[...]
```

只有在您指定完整檔案名稱及完整路徑時，選項 `-f` 才會有作用。提供盡可能多的檔案名稱。例如，以下指令

```
rpm -q -f /bin/rpm /usr/bin/wget
```

會產生：

```
rpm-4.1.1-191
wget-1.9.1-50
```

如果只知道檔案名稱的一部分，可使用**範例 4.2「搜尋套件的程序檔」** [32頁] 中所示的外圍程序程序檔。執行時，可將部份檔案名稱當作參數傳給程序檔。

範例 4.2 搜尋套件的程序檔

```
#!/bin/sh
for i in $(rpm -q -a -l | grep $1); do
    echo "\"$i\" is in package:"
    rpm -q -f $i
    echo ""
done
```

`rpm -q --changelog rpm` 指令會依日期排序，顯示特定套件之相關變更資訊的詳細清單。此範例顯示套件 `rpm` 的相關資訊。

藉由安裝的 **RPM** 資料庫協助，可執行驗證檢查。以 `-v`、`-y` 或 `--verify` 啟動檢查。使用此選項，`rpm` 可顯示從安裝開始，套件中所有變更過的檔案。`rpm` 使用八個字元的符號來提供下列變更的提示：

表格 4.2 RPM 驗證選項

5	MD5 檢查總數
S	檔案大小
L	符號連結
T	修改時間
D	主要和次要的設備編號
U	擁有者
G	群組
M	模式 (許可權和檔案類型)

如果是組態檔，會印出字母 `c`。例如，若 `/etc/wgetrc` (`wget`) 有變更：

```
rpm -V wget
S.5....T c /etc/wgetrc
```

RPM 資料庫的檔案放在 `/var/lib/rpm`。如果分割區 `/usr` 的大小為 1 GB，此資料庫將佔用 30 MB 左右的空間，尤其是在完整更新之後。如果資料庫遠大於預期，使用選項 `--rebuilddb` 來重建資料庫很有用。在執行之前，請備份舊的資料庫。`cron` 程序檔 `cron.daily` 會對資料庫做每日備份(以 `gzip` 封裝)，並將備份儲存在 `/var/adm/backup/rpmdb` 中。副本數量由 `/etc/sysconfig/backup` 中的變數 `MAX_RPMDDB_BACKUPS` (預設值：5) 控制。單一備份的大小大約是 1 GB 的 `/usr` 備份成 1 MB。

4.2.6 安裝與編譯來源套件

所有來源套件均帶有副檔名 `.src.rpm` (來源 RPM)。

提示

來源套件可從安裝媒體複製到硬碟，並用 YaST 解壓縮。但是，在套件管理員中，它們不會被標示為已安裝 (`[i]`)。這是因為來源套件沒有輸入 RPM 資料庫中。只有已安裝的作業系統軟體會列在 RPM 資料庫中。您在「安裝」來源套件時，僅會將原始程式碼新增到系統中。

在 `/usr/src/packages` 中必須可以找到 `rpm` 和 `rpmbuild` 的下列目錄 (除非您在如 `/etc/rpmrc` 的檔案中指定自定設定)：

SOURCES

用於原始來源 (`.tar.bz2` 或 `.tar.gz` 檔案等) 和配送特定調整 (大部份是 `.diff` 或 `.patch` 檔案)

SPECS

用於 `.spec` 檔案，和中繼 Makefile 相似，可控制 `build` 程序

BUILD

所有來源在此目錄中解壓縮、修補和編譯

RPMS

儲存完整二進位套件的地方

SRPMS

此處為來源 RPM

當您使用 YaST 安裝來源套件時，所有需要的元件都會安裝在 `/usr/src/packages` 中：SOURCES 中的來源和調整以及 SPECS 中的相關 `.spec`。

警告

請勿試驗系統元件 (glibc、rpm、sysvinit 等)，因為這樣會危害系統的操作性。

以下範例使用 `wget.src.rpm` 套件。在使用 YaST 安裝套件之後，應該有類似下列清單的檔案：

```
/usr/src/packages/SOURCES/nops_doc.diff
/usr/src/packages/SOURCES/toplev_destdir.diff
/usr/src/packages/SOURCES/wget-1.9.1+ipvmisc.patch
/usr/src/packages/SOURCES/wget-1.9.1-brokentime.patch
/usr/src/packages/SOURCES/wget-1.9.1-passive_ftp.diff
/usr/src/packages/SOURCES/wget-LFS-20040909.tar.bz2
/usr/src/packages/SOURCES/wget-wrong_charset.patch
/usr/src/packages/SPECS/wget.spec
```

`rpmbuild -b X /usr/src/packages/SPECS/wget.spec` 可開始編譯。
`X` 代表建立程序各種階段的萬用字元 (請參閱 `--help` 的輸出或 **RPM** 文件以取得詳細資訊)。以下僅為簡略的說明：

`-bp`

在 `/usr/src/packages/BUILD` 中準備來源：解壓縮和修補。

`-bc`

執行與 `-bp` 相同動作，但是會額外編譯。

`-bi`

執行與 `-bp` 相同的動作，但是會額外安裝建立的軟體。警告：如果套件不支援 **BuildRoot** 功能，您可能會覆寫組態檔。

`-bb`

執行與 `-bi` 相同的動作，但是會額外建立二進位套件。如果編譯成功，二進位應該在 `/usr/src/packages/RPMS`。

-ba

執行與 -bb 相同的動作，但是會額外建立來源 RPM。如果編譯成功，二進位應該在 /usr/src/packages/SRPMS。

--short-circuit

略過部分步驟。

現在可使用 rpm -i (最好使用 rpm -U) 來安裝所建立的二進位 RPM。使用 rpm 來安裝會讓它出現在 RPM 資料庫中。

4.2.7 以 build 編譯 RPM 套件

許多套件中都包含不想要的檔案，它們會在 build 程序中增到執行系統中，因為導致危險產生。為了避免發生此狀況，可使用 build，它會建立要在其中建立套件的已定義環境。若要建立此 chroot 環境，必須提供 build 程序檔與完整的套件樹狀結構。此樹狀結構可在硬碟上、透過 NFS 或從 DVD 取得。用 build --rpms directory 設定位置。和 rpm 不同，build 指令會在來源目錄中尋找 SPEC 檔。若要以裝載在系統中 /media/dvd 之下的 DVD 建立 wget (如上面的範例)，請以 root 的身分執行下列指令：

```
cd /usr/src/packages/SOURCES/  
mv ../SPECS/wget.spec .  
build --rpms /media/dvd/suse/ wget.spec
```

之後，系統便會在 /var/tmp/build-root 中建立一個最小的環境。套件將於此環境中建立。完成時，結果套件位於 /var/tmp/build-root/usr/src/packages/RPMS 中。

build 程序檔可提供數個額外選項。例如，讓程序檔偏好您自己的 RPM、省略建置環境的啟始化，或將 rpm 指令限制在上述階段之一。可使用 build --help 以及參閱 build man 頁面來存取其他資訊。

4.2.8 RPM 歸檔和 RPM 資料庫工具

Midnight Commander (mc) 可顯示 RPM 歸檔的內容，並複製部分內容。它將歸檔以虛擬檔案系統呈現，提供 Midnight Commander 的所有常見功能表選項。使用 F3 可顯示 HEADER。使用游標和 Enter 可檢視歸檔結構。使用 F5 可複製歸檔元件。

KDE 提供 `kpackage` 工具做為 `rpm` 的前端。完整功能的套件管理員是以 YaST 模組的方式提供 (請參閱 第 7 章 *Installing or Removing Software* (↑部署指南))。

使用 Nomad 存取遠端桌面

Nomad (Novell Open Mobile Agile Desktop) 隨附有 SUSE® Linux Enterprise Desktop，可讓您從任何圖形硬體執行暫停連線的桌面工作階段。它包含以下核心元件：

代理 X 伺服器

支援先進的 X 延伸，如 Composite、XVideo 與 RANDR。

工作階段管理程式

負責建立並追蹤可遠端存取的桌面工作階段。

連線處理器

將遠端桌面通訊協定 (RDP) 做位輪送與安全性層級使用。RDP 是可讓使用者連接到執行 Microsoft 終端機服務之電腦的多通道協定。不過，如果用戶端軟體支援，連線處理器會使用虛擬 X11 通道 (rdpx11)，將未過濾的 X11 流量輸送到顯示桌面的本地 X 伺服器。連線處理器總是會在必要時退回到純 RDP 指令，這表示可從任何現有的 RDP 用戶端存取遠端桌面。

用戶端程式

系統提供了 SUSE Linux Enterprise Desktop 的特殊 RDP 用戶端。此用戶端可實作 Nomad 專用延伸以進行 X11 協定轉遞，並可在載入合適的組和管理員外掛程式時，本地組合遠端桌面。

組和管理員延伸

使用組合可對應用程式視窗設定進階視覺效果，例如透明度、淡出/淡入、縮放、變形、拖曳以及重改方向等。

Nomad 可讓您從不同的實體位置遠端存取桌面，例如，您可從家中或工作地點存取相同的工作階段。工作階段中斷後，只需移到其他終端機，然後恢復工作即可。您也可以將目前執行的環境複製到行動設備，如筆記型電腦。使用 Nomad 還可在協作或培訓場合共享桌面，以允許遠端控制與管理。

5.1 Nomad 的先決條件

您必須在本地機器上安裝 `rdesktop` 套件，才能使用 Nomad。另外還可安裝以下套件：

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`
- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`
- `tsclient`

在提供桌面的遠端機器上，需要安裝包含開放式來源遠端桌面通訊協定 (RDP) 伺服器的 `xrdp` 套件。

另外還可安裝以下套件：

- `compiz`
- `compiz-plugins-dmx`
- `compiz-fusion-plugins-main`
- `libcompizconfig`

- `python-compizconfig`
- `compiz-manager`
- `simple-ccsm`

5.2 安裝與設定

做為主機的本地機器不需要任何特別組態。只要安裝了 `rdesktop` 套件，即可使用 `rdesktop` 指令行工具連接提供桌面的遠端機器。如果您偏好圖形使用者介面，另外還請安裝 `tsclient` 套件。`tsclient` (終端伺服器用戶端) 是 `rdesktop` 與其他遠端桌面工具的 GNOME 前端，同樣支援 Xnest 與 VNC 用戶端 (`vncviewer`)。若要提高效能並使用桌面特效，請安裝其他 `compiz` 套件。

不過，您需要以如下方式準備提供桌面的遠端機器：

- 1 安裝 `xrdp` 套件。這會自動將 `xrpd` 伺服器新增至 `runlevel 5`。要手動啟動或停止服務，請以 `root` 身份執行 `/etc/init.d/xrdp start` 或 `/etc/init.d/xrdp stop`。
- 2 將防火牆設定為允許連線連接到埠 3389 (此埠用於 RDP 連線)。啟動 YaST 並選取「安全性與使用者」>「防火牆」。按一下「允許的服務」，然後選取其中允許服務的區域。按一下「進階」並輸入 3389 做為「TCP 埠」。確認您在 YaST 中所做的設定。
- 3 若要使用 3D 桌面特效，請安裝其他 `compiz` 套件。如此，當使用支援虛擬通道的用戶端時，效能將會明顯改善。透過啟用本地與遠端桌面的桌面特效，本地的組合管理員能夠將特效套用於遠端桌面的元素。

注：桌面效果

如果要在遠端桌面上使用桌面特效，請確定 `compiz-plugins-dmx` 套件已安裝於兩端系統中：提供遠端桌面的系統與存取遠端桌面的本地系統。

5.3 使用 Nomad

只要遠端機器上正在執行 `xrpd` 並且開啟了埠 3389，您便可使用 RDP 用戶端連接到遠端主機。要進行連接，請使用 `rdesktop` 指令行工具或提供圖形使用者介面的 `tsclient`。

5.3.1 使用 `rdesktop` 連接伺服器

要為使用者 `tux` 建立壓縮模式的連線，請在外圍程序中執行以下指令：

```
rdesktop -u tux -z server
```

其中，`server` 是遠端機器的主機名稱或 IP 位址。

此指令會為指定使用者啟動登入畫面，他可從此處登入到遠端桌面。透過 `xrdp` 執行的桌面工作階段是相互獨立的，不會與一般顯示管理員 (如 GDM 或 KDM) 衝突。

建立連線時，可以設定多個選項。例如，可以使用全螢幕模式、選擇某種鍵盤配置或調整規格。使用 `rdesktop --help` 瞭解可用 `rdesktop` 選項的詳細資訊。

5.4 疑難排解

如果您無法建立連線，請根據下列清單繼續。

遠端機器上是否啟動並正在執行 `xrdp` 伺服器？

1. 檢查提供桌面的遠端機器上是否已安裝 `xrdp` 套件。
2. 檢查 `xrdp` 服務是否正在執行。
3. 如果不是，請以 `root` 身份執行指令 `/etc/init.d/xrdp start` 或 `/etc/init.d/xrdp restart` 手動啟動或重新啟動該服務。

啟動 `xrdp` 服務後，應該會執行兩個程序：`xrdp` 與 `xrdp-sesman`。如果其中一個程序因為某種原因而無法啟動，那麼在前景中手動啟動這兩個程序時，系統很可能會告訴您問題所在。

4. 要手動啟動程序，請切換到 `root` 身份，然後執行
`/usr/sbin/xrdp-sesman -n` 與 `/usr/sbin/xrdp -nodaemon`。
5. 另外，請檢查 `/var/log/xrdp-sesman.log` 中的 `xrdp-sesman` 輸出與
`/var/log/messages` 中的 `xrdp` 輸出以獲取詳細資訊。

5.5 如需更多資訊

如需有關 Nomad 的詳細資訊，請參閱 <http://en.opensuse.org/Nomad>。

Bash 和 Bash 程序檔

如今，許多人使用具有圖形使用者介面 (GUI) 的電腦，例如 KDE 或 GNOME 介面。儘管這些介面提供了很多功能，使用它們執行自動任務時，還是會有限制。外圍程序是 GUI 的有效補充，本章概述了外圍程序的一些方面，在此例中指 Bash。

6.1 什麼是「外圍程序」？

一般而言，外圍程序就是 Bash (Bourne again Shell)。本章所述的「外圍程序」指的是 Bash。實際上，除了 Bash 以外，還存在其他具有不同功能與特性的外圍程序。如需有關其他外圍程序的詳細資訊，請在 YaST 中搜尋外圍程序。

6.1.1 瞭解 Bash 組態檔案

可使用以下方式呼叫外圍程序：

1. 做為互動式登入外圍程序。當使用 `--login` 選項呼叫 Bash 以登入機器，或使用 SSH 登入遠端機器時採用這種方式。
2. 做為「一般」的互動式外圍程序。啟動 `xterm`、`konsole` 或類似工具時通常會使用這種方式。
3. 做為非互動式外圍程序。在指令行中呼叫外圍程序程序檔時使用這種方式。

系統會讀取不同的組態檔案，視所使用的外圍程序類型而定。下面的表格顯示了登入與非登入外圍程序組態檔案。

表格 6.1 登入外圍程序的 *Bash* 組態檔案

檔案	描述
<code>/etc/profile</code>	請勿修改此檔案，否則您的修改在下次更新時可能會被破壞！
<code>/etc/profile.local</code>	擴充 <code>/etc/profile</code> 時，使用此檔案
<code>/etc/profile.d/</code>	包含特定程式的系統層級組態檔案
<code>~/.profile</code>	在此處插入登入外圍程序的使用者特定組態

表格 6.2 非登入外圍程序的 *Bash* 組態檔案

<code>/etc/bash.bashrc</code>	請勿修改此檔案，否則您的修改在下次更新時可能會被破壞！
<code>/etc/bash.bashrc.local</code>	使用此檔案僅插入 Bash 的系統層級修改
<code>~/bashrc</code>	在此處插入使用者特定的組態

此外，**Bash** 還使用以下其他檔案：

表格 6.3 *Bash* 的特殊檔案

檔案	描述
<code>~/.bash_history</code>	包含您鍵入的所有指令清單
<code>~/.bash_logout</code>	登出時使用

6.1.2 目錄結構

下表概述了可在 Linux 系統中找到的最重要的較高層目錄。下列清單中提供了關於目錄與重要子目錄的更多詳細資訊。

表格 6.4 標準目錄網路樹的綜覽

目錄	內容
/	根目錄 — 目錄樹的起點。
/bin	基本的二進位檔案，例如系統管理員與一般使用者都需要使用的指令。通常還包含 Bash 等外圍程序。
/boot	開機載入程式的靜態檔案。
/dev	存取主機特定設備所需的檔案。
/etc	主機特定系統的組態檔案。
/home	存放系統中所有擁有帳戶之使用者的主目錄。只有 <code>root</code> 的主目錄不在 <code>/home</code> 中，而是位於 <code>/root</code> 中。
/lib	基本的共享程式庫與核心模組。
/media	抽取式媒體的定點。
/mnt	用於暫時裝載檔案系統的定點。
/opt	附加應用程式軟體套件。
/root	超級使用者 <code>root</code> 的主目錄。
/sbin	基本的系統二進位檔案。
/srv	系統所提供之服務的資料。

目錄	內容
/tmp	暫存檔案。
/usr	包含唯讀資料的次要階層。
/var	可變資料，例如記錄檔案。
/windows	僅當系統中同時安裝了 Microsoft Windows* 與 Linux 才可以使用。包含 Windows 資料。

以下清單提供了更多詳細資訊，以及可在其中找到檔案與子目錄的一些目錄範例：

/bin

包含 root 及其他使用者可能會使用的基本外圍程序指令。這些指令包括 ls、mkdir、cp、mv、rm 以及 rmdir。/bin 還包含 SUSE Linux Enterprise Desktop 中的預設外圍程序 Bash。

/boot

包含開機所需的資料，例如開機載入程式、核心及核心開始執行使用者模式程式之前所使用的其他資料。

/dev

存放代表硬體元件的設備檔案。

/etc

包含控制 X Window System 等程式的操作的本地組態檔案。/etc/init.d 子目錄包含開機期間執行的程序檔。

/home/*使用者名稱*

存放系統中每個擁有帳戶之使用者的個人資料。只有檔案擁有者或系統管理員才能修改位於此處的檔案。依預設，電子郵件目錄與個人桌面組態以隱藏檔案與目錄的形式存放於此處。KDE 使用者可分別在 .kde 或 .kde4 中找到其桌面的個人組態資料，而 GNOME 使用者可在 .gconf 中找到相應資料。

注：網路環境中的主目錄

如果您是在網路環境中工作，您的主目錄可能會對應至檔案系統中 `/home` 以外的目錄。

`/lib`

包含啟動系統及執行根檔案系統中的指令所需的基本共享程式庫。在 Windows 中，對應的共享程式庫為 DLL 檔案。

`/media`

包含 CD-ROM、USB 晶片組及數位相機 (若使用 USB) 等抽取式媒體的定點。`/media` 通常存放系統硬碟之外的任何類型磁碟機或光碟機。抽取式媒體插入或連接到系統並進行裝載後，您就可以從此處存取該媒體。

`/mnt`

此目錄提供了暫時裝載之檔案系統的定點。`root` 可在此處裝載檔案系統。

`/opt`

為安裝其他軟體而保留。可在該處找到選擇性軟體與大型附加程式套件。KDE3 位於該處，而 KDE4 與 GNOME 現已移至 `/usr`。

`/root`

`root` 使用者的主目錄。`root` 的個人資料位於此處。

`/sbin`

如 `s` 所指示，此目錄存放適用於超級使用者的公用程式。`/sbin` 不僅包含 `/bin` 中的二進位檔案，還包含啟動、還原及復原系統所必需的二進位檔案。

`/srv`

存放系統所提供之服務的資料，例如 FTP 與 HTTP。

`/tmp`

需要檔案暫時儲存區的程式會使用此目錄。

`/usr`

`/usr` 與使用者無關，而是 UNIX 系統資源 (UNIX system resource) 的縮寫。`/usr` 中的資料是靜態唯讀的資料，可依照檔案系統階層標準 (Filesystem Hierarchy Standard, FHS) 在不同的主機之間共享。此目錄包含所有應用程

式，並且會在檔案系統中建立次要階層。KDE4 與 GNOME 也位於此處。`/usr` 存放了許多子目錄，例如 `/usr/bin`、`/usr/sbin`、`/usr/local` 以及 `/usr/share/doc`。

`/usr/bin`

包含一般情況下可存取的程式。

`/usr/sbin`

包含為系統管理員保留的程式，例如修復功能。

`/usr/local`

在此目錄中，系統管理員可安裝獨立版本的本地延伸。

`/usr/share/doc`

存放系統的各種文件檔案與版本說明。在 `manual` 子目錄中，可找到此手冊的線上版本。如果安裝了多種語言，此目錄可能會包含不同語言的手冊版本。

在 `packages` 下面，可找到系統上已安裝軟體套件中所包含的文件。每個套件都有一個子目錄 `/usr/share/doc/packages/套件名稱`，該目錄通常用於存放套件的讀我檔案，有時也會存放範例、組態檔案或其他程序檔。

如果系統中安裝了 **HOWTO**，`/usr/share/doc` 還會存放 `howto` 子目錄，在其中提供許多與 **Linux** 軟體設定及操作相關之任務的其他文件。

`/var`

`/usr` 存放的是靜態唯讀資料，而 `/var` 存放的是系統操作時寫入的資料，因此為可變資料，例如記錄檔案或多工緩衝處理資料。例如，系統的記錄檔案位於 `/var/log/messages` 中 (只有 `root` 才能存取)。

`/windows`

僅當系統中同時安裝了 Microsoft Windows 與 Linux 才可以使用。包含可在系統的 Windows 分割區上使用的 Windows 資料。您是否可以編輯此目錄中的資料取決於 Windows 分割區使用的檔案系統。如果是 FAT32，則您可以開啟並編輯此目錄中的檔案。如果是 NTFS 檔案系統，您只能讀取 Linux 中的 Windows 檔案，但不能對其進行修改。

6.2 寫入外圍程序程序檔

使用外圍程序程序檔可以方便地完成各種任務：收集資料、搜尋文字中的單字或片語，以及執行很多其他有用的操作。以下範例顯示了一個列印文字的小型外圍程序程序檔：

範例 6.1 列印文字的外圍程序程序檔

```
#!/bin/sh ❶  
# Output the following line: ❷  
echo "Hello World" ❸
```

- ❶ 第一行以 *Shebang* 字元(#!) 開頭，指出此檔案為程序檔。程序檔透過 *Shebang* 後面指定的解譯器執行，在此例中為 `/bin/sh`。
- ❷ 第二行為備註，以 `#` 開頭。建議將較為複雜的行設為備註，以便記住其作用。
- ❸ 第三行使用內建指令 `echo` 列印相應的文字。

您需要符合一些先決條件才能執行此程序檔：

1. 每個程序檔都應包含一行 *Shebang* 行(上面的範例即是如此。)如果程序檔未包含此行，您需要自行呼叫解譯器。
2. 您可以將程序檔儲存於任何位置。但是，最好將其儲存於外圍程序可以搜尋的目錄中。外圍程序中的搜尋路徑由環境變數 `PATH` 決定。例如，將程序檔儲存於目錄 `~/bin/` 中名為 `hello.sh` 的檔案中。
3. 程序檔需要執行權限。使用下列指令設定權限：

```
chmod +x ~/bin/hello.sh
```

如果滿足了上述所有先決條件，您就可以使用 `~/bin/hello.sh` 或 `hello.sh` 執执行程序檔。第一次呼叫會使用絕對路徑，而第二次呼叫會在由 `PATH` 環境變數指定的每個目錄中搜尋指令。

6.3 重新指向指令事件

每條指令可以使用三個通道用於輸入或輸出：

- **標準輸出** 這是預設的輸出通道。指令進行列印時會使用標準輸出通道。
- **標準輸入** 如果指令需要使用者或其他指令的輸入，將會使用此通道。
- **標準錯誤** 指令使用此通道報告錯誤。

要重新指向這些通道，可以使用以下幾種方式：

指令 `>` 檔案

將指令輸出儲存為檔案，現有的檔案將會刪除。例如，`ls` 指令將輸出寫入到檔案 `listing.txt` 中：

```
ls > listing.txt
```

指令 `>>` 檔案

將指令輸出附加至檔案。例如，`ls` 指令將輸出附加至檔案 `listing.txt` 中：

```
ls >> listing.txt
```

指令 `<` 檔案

讀取檔案，將其做為指定指令的輸入。例如，`read` 指令會將檔案內容讀取至變數中：

```
read a < foo
```

指令1 `|` 指令2

將左邊指令的輸出重新指向為右邊指令的輸入。

每個通道都有一個**檔案描述子**：`0` (零) 代表標準輸入，`1` 代表標準輸出，`2` 代表標準錯誤。您可以將此檔案描述子插入到 `<` 或 `>` 字元的前面。例如，下行將搜尋以 `foo` 開始的檔案，但透過將檔案重新指向至 `/dev/null` 隱藏了錯誤：

```
find / -name "foo*" 2>/dev/null
```

6.4 使用別名

別名為一條或多條指令的捷徑定義。別名的語法為：

```
alias NAME=DEFINITION
```

例如，下行定義了別名 `lt`，輸出一份較長的清單 (選項 `-l`)，按照修改時間排序 (`-t`)，並且在排序時按反向列印 (`-r`)：

```
alias lt='ls -ltr'
```

要檢視所有的別名定義，請使用 `alias`。

6.5 使用 Bash 中的變數

外圍程序變數可以是全域變數或本地變數。您可以在所有外圍程序中存取全域變數或環境變數。與此相反，本地變數僅顯示於目前的外圍程序中。

要檢視所有環境變數，請使用 `printenv` 指令。如需特殊變數，請將變數名稱做為引數插入。

```
printenv PATH
```

也可以使用 `echo` 檢視變數：

```
echo $PATH
```

此指令可列印 `PATH` 變數。要設定本地變數，請使用變數名稱，後面跟上等號，再跟上值：

```
PROJECT="SLED"
```

請不要在等號兩邊插入空格，否則將會出錯。要設定環境變數，請使用 `export`：

```
export NAME="tux"
```

要移除變數，請使用 `unset`：

```
unset NAME
```

下表包含了部分可在外圍程序程序檔中使用的常用環境變數：

表格 6.5 有用的環境變數

HOME	目前使用者的主目錄
HOST	目前的主機名稱

LANG	工具當地化以後，會使用此環境變數指定的語言。也可將英語設定為 C。
PATH	外圍程序的搜尋路徑，即以冒號分隔的目錄清單。
PS1	指定在每條指令前列印的一般提示
PS2	指定執行多行指令時列印的輔助提示
PWD	目前的工作目錄
USER	目前的使用者

6.5.1 使用引數變數

例如，如果您有程序檔 `foo.sh`，就可以按以下格式執行該程序檔：

```
foo.sh "Tux Penguin" 2000
```

要存取傳送至程序檔的所有引數，您需要使用位置參數。`$1` 代表第一個引數的位置參數，`$2` 代表第二個引數的位置參數，依此類推。最多可以使用九個參數。要獲取程序檔名稱，請使用 `$0`。

下面的程序檔 `foo.sh` 可列印從 1 到 4 的所有引數：

```
#!/bin/sh
echo \"$1\" \"$2\" \"$3\" \"$4\"
```

如果您使用以上引數執行此程序檔，所得結果為：

```
"Tux Penguin" "2000" "" ""
```

6.5.2 使用變數替代項

變數替代項會從左側或右側將模式套用至變數內容。以下清單包含了可用的語法格式：

```
${VAR#pattern}
```

從左側移除最短的相符項：


```
file=/home/tux/book/book.tar.bz2
echo ${file#*/}
home/tux/book/book.tar.bz2
```

`${VAR##pattern}`
從左側移除最長的相符項：

```
file=/home/tux/book/book.tar.bz2
echo ${file##*/}
book.tar.bz2
```

`${VAR%pattern}`
從右側移除最短的相符項：

```
file=/home/tux/book/book.tar.bz2
echo ${file%.*}
/home/tux/book/book.tar
```

`${VAR%%pattern}`
從右側移除最長的相符項：

```
file=/home/tux/book/book.tar.bz2
echo ${file%%.*}
/home/tux/book/book
```

6.6 分組與組合指令

外圍程序允許您組合及分組指令，以進行條件式執行。每條指令都會傳回決定操作成功與否的離開碼。如果為0(零)，則說明指令成功，任何其他離開碼都代表特定於指令的錯誤。

以下清單顯示了可對指令分組的方式：

指令1 ; 指令2

以順序執行指令。不檢查離開碼。下行透過 `cat` 顯示檔案內容，然後透過 `ls` 列印其檔案內容，而不管其離開碼為何：

```
cat filelist.txt ; ls -l filelist.txt
```

指令1 && 指令2

如果左邊的指令成功，即會執行右邊的指令(邏輯「與」)。下行顯示檔案內容，並且僅在前面的指令成功時才會列印其檔案內容(將其與此清單中的上一個項目進行比較)：

```
cat filelist.txt && ls -l filelist.txt
```

指令1 || 指令2

如果左邊的指令失敗，即會執行右邊的指令(邏輯「或」)。下行將只會在於 /home/tux/foo 中建立目錄失敗時，才會在 /home/wilber/bar 中建立目錄：

```
mkdir /home/tux/foo || mkdir /home/wilber/bar
```

```
funcname() { ... }
```

建立外圍程序函數。可以使用位置參數存取其引數。下行定義了可列印較短訊息的函數 hello：

```
hello() { echo "Hello $1"; }
```

可以按以下格式呼叫此函數：

```
hello Tux
```

將會列印：

```
Hello Tux
```

6.7 使用通用流程建構元

要控制程序檔的流程，外圍程序需包含 while、if、for 及 case 建構元。

6.7.1 If 控制指令

If 用於檢查運算式。例如，以下程式碼將測試目前的使用者是否為 Tux：

```
if test $USER = "tux" then
    echo "Hello Tux."
else
    echo "You are not Tux."
fi
```

測試運算式可以很複雜，也可以很簡單。下面的運算式會檢查檔案 foo.txt 是否存在：

```
if test -e /tmp/foo.txt
then
    echo "Found foo.txt"
fi
```

<http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html> 中提供了更多運算式。

6.7.2 使用 For 指令建立迴路

For迴路可讓您對一組項目執行指令。例如，以下程式碼會列印目前目錄中關於 PNG 檔案的部分資訊：

```
for i in *.png; do
  ls -l $i
done
```

6.8 如需更多資訊

man 頁面 `man sh` 中提供了關於 Bash 的重要資訊。以下清單中提供了更多關於此主題的資訊：

- <http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html> — Bash 初級使用者指南
- <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html> — BASH 程式設計 - 方法介紹
- <http://tldp.org/LDP/abs/html/index.html> — Bash - 程序檔進階指南
- <http://www.grymoire.com/Unix/Sh.html> — Sh - Bourne 外圍程序

第 II. 系統 部分

64 位元系統環境的 32 位元和 64 位元應用程式

SUSE® Linux Enterprise Desktop 可用於 64 位元平台。但這並不表示所有包含的應用程式都已移植到 64 位元平台。SUSE Linux Enterprise Desktop 支援在 64 位元系統環境中使用 32 位元應用程式。本章概述了有關 SUSE Linux Enterprise Desktop 在 64 位元平台下的支援方式。它說明 32 位元應用程式的執行方式 (執行期間支援) 以及如何編譯 32 位元應用程式，讓它們在 32 位元和 64 位元兩種系統環境都可執行。另外，您可找到關於核心 API 和 32 位元應用程式如何在 64 位元核心下執行的相關資訊。

針對 amd64 與 Intel 64 等 64 位元平台所設計的 SUSE Linux Enterprise Desktop，可讓現有的 32 位元應用程式不需額外的設定即可在 64 位元環境中執行。這項支援意謂您可以繼續使用偏好的 32 位元應用程式，無需等到對應的 64 位元連接埠上市。

7.1 執行期間支援

重要: 不同應用程式版本之間的衝突

如果應用程式有 32 位元和 64 位元兩種版本，同時安裝二種版本，一定會發生問題。在這種狀況下，可在兩種版本中選定一種來安裝並使用。

此規則的例外是 PAM (可插式驗證模組)。SUSE Linux Enterprise Desktop 在驗證程序中將 PAM 當做使用者與應用程式之間的溝通層。在也可以執行 32 位元應用程式的 64 位元作業系統上，始終有必要安裝 PAM 模組的兩個版本。

要正確執行，每一個應用程式都需要一些程式庫。不幸的是，這些程式庫的 32 位元和 64 位元版本，名稱都一樣。它們必須透過其他方法來區分彼此。

要保留與 32 位元版本的相容性，程式庫儲存在系統中的位置，與在 32 位元環境中的位置相同。在 32 位元和 64 位元環境中，`libc.so.6` 的 32 位元版本都位於 `/lib/libc.so.6`。

所有 64 位元程式庫和物件檔案都位於名為 `lib64` 的目錄。您通常預期在 `/lib` 和 `/usr/lib` 之下找到的 64 位元物件檔案，現在放在 `/lib64` 以及 `/usr/lib64` 底下。這表示在 `/lib` 和 `/usr/lib` 之下，有預留空間給 32 位元程式庫使用，因而兩種版本的檔案名稱能夠保持不變。

32 位元 `/lib` 目錄的子目錄，如果其資料內容不取決於字組大小，也不會移動。此配置與 LSB (Linux Standards Base) 以及 FHS (File System Hierarchy Standard) 相容。

7.2 軟體開發

Biarch 開發工具鏈可以產生 32 位元和 64 位元物件。預設是編譯 64 位元物件。使用特殊旗標可以產生 32 位元物件。GCC 的特殊旗標是 `-m32`。

所有標題檔案必須使用與結構無關的形式來編寫。安裝的 32 位元和 64 位元程式庫，必須有一個與已安裝標頭檔案相符的 API (應用程式設計介面)。一般的 SUSE Linux Enterprise Desktop 環境是根據此原則而設計。對於手動更新的程式庫，請自行解決這些問題。

7.3 Biarch 平台的軟體編譯

若要在 biarch 架構上，為其他結構開發二進位程式，必須為第二個結構額外安裝相關程式庫。這些套件稱為 `rpmname-32bit`。您也需要 `rpmname-devel` 套件的個別標頭和程式庫，以及 `rpmname-devel-32bit` 第二個架構的開發程式庫。

大部份開放原始碼程式使用的程式組態是以 `autoconf` 為基礎。若要使用 `autoconf` 為第二個結構設定程式，請覆寫 `autoconf` 的一般編譯器和連結器設定，方法是執行包含其他環境變數的 `configure` 程序檔。

以下範例，將 x86 的 x86_64 系統指示為第二個架構：

1 使用 32 位元編譯器：

```
CC="gcc -m32"
```

2 指示連結器處理 32 位元物件 (務必使用 gcc 做為連結器前端工具)：

```
LD="gcc -m32"
```

3 設定組合器來產生 32 位元物件：

```
AS="gcc -c -m32"
```

4 判斷 libtool 等等的程式庫是否來自 /usr/lib：

```
LDFLAGS="-L/usr/lib"
```

5 決定程式庫是否儲存在 lib 子目錄中：

```
--libdir=/usr/lib
```

6 決定是否使用 32 位元 X 程式庫：

```
--x-libraries=/usr/lib/xorg
```

並非每一個程式都需要所有這些變數。將它們配合各程式使用。

```
CC="gcc -m32" \
LDFLAGS="-L/usr/lib;" \
    .configure \
        --prefix=/usr \
        --libdir=/usr/lib
make
make install
```

7.4 核心規格

x86_64 適用的 _64 位元核心，提供 64 位元和 32 位元兩種核心 ABI (應用程式二進位介面)。後者與相對應 32 位元核心的 ABI 是相同的。這表示 32 位元應用程式可以用與 32 位元核心溝通相同的方式，來與 64 位元核心溝通。

32 位元系統模擬的 64 位元核心呼叫，不支援系統程式使用的所有 API。這要視平台而定。基於這個原因，少數應用程式 (例如 `lspci`) 就必須進行編譯。

64 位元核心只可以載入為此核心特別編譯的 64 位元核心模組。它無法使用 32 位元核心模組。

提示

部份應用程式需要個別的核心可載入式模組。如果您想在 64 位元系統環境使用這種 32 位元應用程式，請洽詢此應用程式的提供者以及 Novell，確定是否可以取得此模組的核心可載入式模組的 64 位元版本以及核心 API 的 32 位元編譯版本。

啟動及設定 Linux 系統

啟動 Linux 系統需要各種不同的元件。硬體自身是由 BIOS 啟動的。BIOS 會藉由開機載入程式啟動核心。此後，關於 `init` 和 `runlevel` 的開機程序完全由作業系統控制。憑藉 `runlevel` 的概念，您可以保持日常使用的設定，以及對系統執行維護任務。

8.1 Linux 開機程序

Linux 開機程序由數個階段所組成，每個階段分別由元件所代表。以下清單簡短概述開機程序，以及所有相關主要元件的功能。

1. **BIOS** 啟動電腦之後，BIOS 會啟始螢幕和鍵盤並測試主記憶體。在此階段中，機器不會存取大量儲存媒體。接著，會從 CMOS 值載入目前日期、時間和最重要的周邊。識別第一個硬碟及其幾何之後，系統控制將會從 BIOS 轉到開機載入程式。
2. **開機載入程式** 第一顆硬碟的第一個實體 512 位元組資料磁區，會載入主要記憶體。接著，在此磁區開頭的開機載入程式會接管開機程序。開機載入程式執行的指令，決定其他部份的開機程序。因此，第一個硬碟的前 512 位元組是主開機紀錄 (MBR)。然後，開機載入程式會將控制傳送到實際作業系統，在這裡是指 Linux 核心。如需更多關於 GRUB (也就是 Linux 開機載入程式) 的詳細資訊，請參閱第 9 章開機載入程式 GRUB [77 頁]。
3. **核心和初始 RAM 檔案系統** 為了送出系統控制，開機載入程式會將核心和初始 RAM 式檔案系統 (`initramfs`) 兩者都載入記憶體。核心可以直接使用 `initramfs` 的內容。`initramfs` 包含一個名為 `init` 的小執行檔，它可以處理實體

根目錄檔案系統的裝載。若存取大量儲存之前需要什麼特殊硬體驅動程式的話，那一定就是 `initramfs` 了。如需有關 `initramfs` 的詳細資訊，請參閱第 8.1.1 節「`initramfs`」[64頁]。

4. **`initramfs` 上的 `init`** 本程式將執行裝載適當根目錄檔案系統所需的全部動作，例如提供必要檔案的系統核心功能，並且提供包含 `udev` 之大量儲存控制器的設備驅動程式。找到根目錄檔案系統後，將會檢查是否有錯誤並進行裝載。若裝載成功，將會清除 `initramfs` 並執行根目錄檔案系統上的 `init` 程式。如需有關 `init` 的詳細資訊，請參閱第 8.1.2 節「`initramfs` 上的 `init`」[65頁]。如需更多有關 `udev` 的詳細資訊，請參閱第 12 章使用 `udev` 進行動態核心設備管理[123頁]。
5. **`init`** `init` 會透過提供數個不同層級所提供的不同功能來負責系統實際開機作業。第 8.2 節「`init` 程序」[66頁] 中會描述 `init`。

8.1.1 `initramfs`

`initramfs` 是一個小型 `cpio` 歸檔，其可由核心載入 RAM 磁碟。在實際根目錄檔案系統裝載之前，它提供可讓程式執行的最小 Linux 環境。BIOS 常式會將這個最小的 Linux 環境載入記憶體，且除了記憶體要求外沒有其他特定硬體需求。此外，`initramfs` 還必須提供一個名為 `init` 的執行檔，這個執行檔會在處理開機程序的根目錄檔案系統上執行實際的 `init` 程式。

在根目錄檔案系統能夠裝載以及作業系統可以啟動之前，核心需要相應的驅動程式來存取根目錄檔案系統所在的設備。這些驅動程式可能包含特定類型硬碟的特殊驅動程式，或者甚至包含存取網路檔案系統的網路驅動程式。`initramfs` 上的 `init` 還會載入根目錄檔案系統所需的模組。當模組載入完成之後，`udev` 便會為 `initramfs` 提供所需的設備。在後來的開機程序中，變更根目錄檔案系統後，必須重新產生這些設備。結合使用 `boot.udev` 和指令 `udevtrigger` 可以實現此目的。

如果您要在安裝的系統上變更硬體 (例如硬碟)，而開機時此硬體需要核心中存在相應的驅動程式，那麼您就必須更新 `initramfs`。更新 `initramfs` 與更新其前身 `initrd` 是採用一樣的更新方式，即是呼叫 `mkinitrd`。不使用任何引數來呼叫 `mkinitrd` 將會建立一個 `initramfs`。呼叫 `mkinitrd -R` 則會建立一個 `initrd`。在 SUSE® Linux Enterprise Desktop 中，`/etc/sysconfig/kernel` 中的變數 `INITRD_MODULES` 會指定要載入的模組。安裝之後，此變數會自動設定為正確值。模組會確實依據它們在 `INITRD_MODULES` 中出現的順序來載入。如果您不依賴於 `/dev/sd?` 設備檔案的正確設定，這一點將無關緊要。不過，在目

前系統中，您也可以使用 `/dev/disk/` 下以幾個不同子目錄分類的設備檔案，這些子目錄的名稱為 `by-id`、`by-path` 和 `by-uuid`，這些目錄總是代表相同磁碟。在安裝時也可以透過指定相應的裝載選項來實現此目的。

重要: 更新 `initramfs` 或 `initrd`

開機載入程式將採用與核心相同的方式載入 `initramfs` 或 `initrd`。更新 `initramfs` 或 `initrd` 之後不需要重新安裝 `GRUB`，因為開機時 `GRUB` 會在目錄搜尋正確的檔案。

8.1.2 `initramfs` 上的 `init`

`Initramfs` 上的 `init` 其主要目的是準備裝載實際根目錄檔案系統，以及存取實際根目錄檔案系統。根據您的系統組態，`init` 負責下列任務。

載入核心模組

存取您電腦的硬體元件可能需要特殊的驅動程式(最重要的元件是硬碟)，具體取決於您的硬體組態。若要存取最後根目錄檔案系統，核心需載入適當的檔案系統驅動程式。

提供區塊特殊檔案

對於每個載入的模組，核心均會產生設備事件。`udev` 會處理這些事件，並在 `RAM` 檔案系統的 `/dev` 中產生所需的區塊專用檔案。如果沒有這些專用檔案，便無法存取檔案系統和其他設備。

管理 RAID 和 LVM 設定

如果您將系統設定為 `RAID` 或 `LVM` 之下的根目錄檔案系統，`init` 會設定 `LVM` 或 `RAID`，以便之後能夠存取檔案根目錄系統。如需有關 `RAID` 與 `LVM` 的資訊，請參閱第 14 章 *Advanced Disk Setup* (↑ 部署指南)。

管理網路組態

如果您將系統設為使用網路裝載的根目錄檔案系統(透過 `NFS` 裝載)，那麼 `init` 必須確認是否已載入適當的網路驅動程式，還有是否設定為允許存取根目錄檔案系統。

安裝程序中，當 `init` 在初始開機時被呼叫，它的任務會與之前討論的不同：

尋找安裝媒體

啟動安裝程序時，您的機器會從安裝媒體使用 YaST 安裝程式，來載入一個安裝核心，以及一個特殊 `initrd`。在 RAM 檔案系統中執行的 YaST 安裝程式，必需具備安裝媒體的實際位置相關資訊，以便存取該程式和安裝作業系統。

啟動硬體辨識並載入適當核心模組

如同在 [第 8.1.1 節「initramfs」](#) [64頁]所述，開機程序會以最少驅動程式啟動，供大部份的硬體組態使用。`init` 將啟動一個硬體掃描程序，該程序會判斷驅動程式是否適用您的硬體組態。開機程序所需的模組名稱會寫入 `/etc/sysconfig/kernel` 的 `INITRD_MODULES` 中。這些名稱是用來產生系統開機所需的自定 `initramfs`。如果模組不是開機所需，而是供 `coldplug` 使用，則模組會寫入 `/etc/sysconfig/hardware/hwconfig-*`。用此目錄中的組態檔案描述的所有設備都會在開機程序中啟動。

載入安裝系統或救援系統

在硬體妥善識別、適當驅動程式完成，以及 `udev` 已建立設備特殊檔案之後，`init` 就會啟動安裝系統，其中包含實際 YaST 安裝程式或救援系統。

啟動 YaST

最後，`init` 將會啟動 YaST，而 YaST 會啟動套件安裝和系統組態。

8.2 init 程序

`init` 程式是程序 ID 1 的程序，其負責以指定方式來啟動系統。`init` 由核心直接啟動，並拒絕訊號 9，這個訊號通常會刪除程序。所有其他程式是直接透過 `init` 或它其中一個子程序啟動。

`init` 主要是在 `/etc/inittab` 檔案中設定，*runlevels* 即是在該檔案中定義的 (請參閱 [第 8.2.1 節「Runlevel」](#) [67頁])。該檔案還會指定每個 *runlevel* 可以使用的服務和精靈。視 `/etc/inittab` 中的項目而定，`init` 會執行數個程序檔。依預設，開機後啟動的第一個程序檔是 `/etc/init.d/boot`。系統啟動化階段一旦完成，系統就會使用 `/etc/init.d/rc` 程序檔將 *runlevel* 變更為其預設 *runlevel*。為避免混淆，這些稱做 *init* 程序檔的程序檔都位於目錄 `/etc/init.d` 中 (請參閱 [第 8.2.2 節「Init 程序檔」](#) [69頁])。

系統啟動和關閉的程序，是由 `init` 維護。依此觀點，核心可以視為背景程序，它的任務是維護所有其他程序，並根據其他程式的要求來調整 CPU 時間和硬體存取。

8.2.1 Runlevel

在 Linux 中是由 *runlevel* 定義啟動系統的方式，以及在所執行的系統可以使用哪些服務。開機之後，系統會按照 `/etc/inittab` 中的 `initdefault` 這一行文字的定義而啟動。一般是 3 或 5。請參閱表格 8.1 「可用的 Runlevel」[67頁]。還有一種方法是，*runlevel* 可以在開機期間指定 (例如，在開機提示時新增 *runlevel* 編號)。所有不經過核心自身直接評估的參數，都會傳遞至 `init`。若要開機進入 Runlevel 3，只需在開機提示符處新增數字 3。

表格 8.1 可用的 *Runlevel*

Runlevel	描述
0	系統暫停
S or 1	單一使用者模式
2	本地多重使用者模式，不包含遠端網路 (NFS 等)
3	完整的多重使用者模式，包含網路
4	「使用者定義」，除非管理員設定了此 <i>runlevel</i> ，否則此項將不可用。
5	完整多重使用者模式，包含網路、X 顯示管理員—KDM、GDM 或 XDM
6	系統重新開機

重要: 避免透過 NFS 裝載分割區的 Runlevel 2

如果系統會透過 NFS 裝載 `/usr` 一類分割區，您就不應該使用 *runlevel* 2。如果程式檔案或是程式庫因 *runlevel* 2 (本地多重使用者模式，無遠端網路) 無法提供 NFS 服務而發生遺失，系統可能無法正常運作。

若要在系統執行時變更 `runlevel`，請輸入 `telinit` 以及當成引數的對應數字。只有系統管理員可以執行此動作。以下清單列出 `runlevel` 區域中最重要指令摘要。

`telinit 1` 或 `shutdown now`

系統變更為單一使用者模式。此模式是用於系統維護和管理任務。

`telinit 3`

可以啟動所有主要的程式和服務(包括網路)，且可讓一般使用者登入並在非圖形環境下使用該系統。

`telinit 5`

啟用圖形式環境。通常這時會啟動顯示管理員，如 `XDM`、`GDM` 或 `KDM`。如果已啟用自動登入，本地使用者就可以登入事先選定的視窗管理員(`GNOME`、`KDE` 或任何其他視窗管理員)。

`telinit 0` 或 `shutdown -h now`

暫停系統。

`telinit 6` 或 `shutdown -r now`

暫停系統後重新開機

所有 `SUSE Linux Enterprise Desktop` 標準安裝中，預設都使用 `runlevel 5`。使用者會在提示之下使用圖形介面登入或預設的使用者會自動登入。如果預設 `runlevel` 是 3，這時 `X Window System` 必須依據第 13 章 *`X Window System`* [135 頁] 所述適當設定，才能將 `runlevel` 切換至 5。完成這個動作之後，請輸入 `telinit 5` 來檢查系統是否已依指定方式運作。如果一切都如預期，您可以使用 `YaST`，將預設 `runlevel` 設定成 5。

警告: `/etc/inittab` 中有錯誤可能會造成系統開機發生錯誤

如果 `/etc/inittab` 損毀，系統可能無法正常開機。因此，在編輯 `/etc/inittab` 時要格外小心。將機器重新開機前，一定要用 `telinit q` 指令讓 `init` 重新讀取 `/etc/inittab`。

通常，當您變更 `runlevel` 時會發生兩件事。首先，啟動目前 `runlevel` 中的停止程序檔，關閉對目前 `runlevel` 很重要的一些程式。然後啟動新 `runlevel` 的啟動程序檔。在大部份的情況下，此時也會啟動一些程式。例如，從 `runlevel 3` 變更成 5 時，會發生以下事件：

1. 管理員 (root) 可以輸入 `telinit 5`，要求 `init` 變更為不同的 `runlevel`。
2. `init` 會檢查目前的 `runlevel` (`runlevel`) 並決定是否要以新 `runlevel` 作為啟動 `/etc/init.d/rc` 的參數。
3. 現在，如果新 `runlevel` 沒有啟動程序檔，`rc` 會呼叫目前 `runlevel` 的停止程序檔。在此範例中的所有程序檔，都位於 `/etc/init.d/rc3.d` (舊的 `runlevel` 是 3)，而且開頭是 `K`。`K` 後面的數字表示以 `stop` 參數執行程序檔的順序，因為還有其他的因素要考慮。
4. 新的 `runlevel` 啟動程序檔，會最後才啟動。在此範例中的所有程序檔，都位於 `/etc/init.d/rc5.d`，而且開頭是 `S`。同時，`S` 後面的數字決定要啟動程序檔的序列。

變改成與目前 `runlevel` 相同的 `runlevel` 時，`init` 只會檢查 `/etc/inittab` 是否變更，並啟動適當的步驟，例如，在另一個介面啟動 `getty`。使用指令 `telinit q` 也可以完成相同功能。

8.2.2 Init 程序檔

`/etc/init.d` 中的程序檔有兩種類型：

由 `init` 直接執行的程序檔

這種狀況僅出現於開機程序或立即關閉系統 (電源中斷或使用者按下 `Ctrl + Alt + Del`) 時。此程序檔定義於 `/etc/inittab`。

由 `init` 間接執行的程序檔

這些程序檔在變更 `runlevel` 時就會執行，而且永遠會呼叫主要程序檔 `/etc/init.d/rc`，以保證相關程序檔的順序正確。

所有程序檔都位於 `/etc/init.d`。在開機時執行的程序檔會透過 `/etc/init.d/boot.d` 的符號連結來呼叫。用於變更 `runlevel` 的程序也可以透過其中一個子目錄 (`/etc/init.d/rc0.d` 到 `/etc/init.d/rc6.d`) 的符號連結來呼叫。這樣的安排是為了明確執行，避免當程序檔用於多個 `runlevel` 時的重複執行。因為每一個程序檔都可當成啟動程序檔和停止程序檔來執行，所以這些程序檔必須了解參數 `start` 和 `stop`。程序檔也了解 `restart`、`reload`、`force-reload` 和 `status` 選項。這些不同選項在 [表格 8.2 「可能的 init 程序](#)

檔選項」[70頁] 都有說明。直接由 `init` 執行的程序檔沒有這些連結。他們可以根據需要從 `runlevel` 獨立執行。

表格 8.2 可能的 `init` 程序檔選項

選項	描述
<code>start</code>	啟動服務。
<code>stop</code>	停止服務。
<code>restart</code>	如果服務在執行中，先停止，再重新啟動。如果服務沒有執行，請啟動它。
<code>reload</code>	不需停止和重新啟動服務，就可以重新載入組態。
<code>force-reload</code>	如果服務支援，請重新載入組態。否則，執行與 <code>restart</code> 相同的動作。
<code>status</code>	顯示目前狀態。

每一個特定 `runlevel` 子目錄中的連結，可以將程序檔與不同 `runlevel` 產生關聯。安裝或解除安裝套件時，可以透過程式 `insserv` 的協助，新增和移除這些連結(或透過 `/usr/lib/lsb/install_initd`，它是一個會呼叫該程式的程序檔)。如需詳細資訊，請參閱 `insserv(8)` 線上文件。

這些設定也都可以利用 `YaST` 模組來變更。如果您必須在指令行上檢查狀態，請使用 `chkconfig(8)` 線上文件中所說明的 `chkconfig` 工具。

簡介最先啟動的開機及最後啟動的停止程序檔，並說明維護程序檔。

`boot`

使用 `init` 直接啟動系統時會執行。它與所選的 `runlevel` 無關，而且只會執行一次。此時，會裝載 `/proc` 和 `/dev/pts` 檔案系統，啟用 `blogd` (開機記錄精靈)。如果系統是更新或安裝之後第一次開機，會啟動啟始系統組態。

任何其他服務啟動之前，`blogd` 精靈是透過開機和 `rc` 啟動的服務。當這些程序檔（會執行多子程序檔，如使特殊檔案區塊能夠使用）啟動的動作完成之後就會停止。`blogd` 會將螢幕輸出的所有內容寫入記錄檔案 `/var/log/boot`

.msg，但唯有當 /var 裝載為可讀寫，才會發生。否則，blogd 會緩衝處理所有螢幕資料，直到 /var 可以使用為止。您可以在 blogd(8) 線上文件取得進一步資訊。

boot 程序檔也負責啟動 /etc/init.d/boot.d 中名稱以 s 開頭的所有程序檔。在該處，會檢查檔案系統，並在需要時設定迴圈設備。也會設定系統時間。如果自動檢查和修復檔案系統時發生錯誤，系統管理員輸入管理員密碼後即可介入。最後執行的程序檔是 boot.local。

boot.local

在此輸入開機時要執行的其他指令 (進入 runlevel 之前)。它就像是 DOS 系統的 AUTOEXEC.BAT。

halt

只有進入 runlevel 0 或 6 時，才會執行此程序檔。在此，它是以 halt 或 reboot 的方式執行。系統是否關閉還是重新開機，取決於呼叫 halt 的方式。如果在關機期間需要特殊指令，請將這些指令新增到 halt.local 程序檔。

rc

此程序檔會呼叫目前 runlevel 的適當停止程序檔，以及新選取 runlevel 的啟動程序檔。與 /etc/init.d/boot 程序檔相似，系統從 /etc/inittab 中呼叫此程序檔，並使用所需的 runlevel 做為參數。

您可以建立自己的程序檔，並輕鬆將它們整合至上述配置。如需關於格式化、命名以及組織自定程序檔的指示，請參閱 LSB 的規格和 init、init.d、chkconfig 及 insserv 的線上文件。另請參閱 startproc 和 killproc 的 man 頁面。

警告: 錯誤的 init 程序檔可能會暫停系統

錯誤的 init 程序檔可能會讓您的機器暫停。編輯類似程序檔要格外小心，可能的話，讓它們在多重使用者環境下密集測試。如需關於 init 程序檔的一些有用資訊，請參閱第 8.2.1 節「Runlevel」[67頁]。

若要為指定的程式或服務建立自定 init 程序檔，請將檔案 /etc/init.d/skeleton 當成樣板。使用新名稱儲存此檔案副本，編輯所需相關程式和檔案名稱、路徑，以及其他詳細資料。您可能需要使用自己的組件來強化程式檔，好讓 init 程序觸發正確的動作。

上方的 INIT INFO 區塊是程序檔的必要組件，因此必須進行編輯。請參閱 [範例 8.1 「迷你 INIT INFO 區塊」](#) [72頁]。

範例 8.1 迷你 INIT INFO 區塊

```
### BEGIN INIT INFO
# Provides:          FOO
# Required-Start:    $syslog $remote_fs
# Required-Stop:     $syslog $remote_fs
# Default-Start:     3 5
# Default-Stop:      0 1 2 6
# Description:       Start FOO to allow XY and provide YZ
### END INIT INFO
```

請在 INFO 區塊的第一行中、Provides: 後面，為此 init 程序檔所控制的程式或服務指定名稱。Required-Start: 與 Required-Stop: 這兩行，會指定服務本身啟動或停止之前，必須先啟動或停止的所有服務。此資訊以後會用來產生程序檔名稱的編號，這些就是在 runlevel 目錄可以找到的編號。Default-Start: 和 Default-Stop: 之後，指定應該自動啟動或停止服務的 runlevel。最後，會在 Description: 簡要說明討論中的服務。

若要從 runlevel 目錄 (/etc/init.d/rc?.d/) 建立對應到 /etc/init.d/ 中的程序檔，請輸入 `insserv new-script-name` 指令。insserv 程式會評估 INIT INFO 標題，為 runlevel 目錄 (/etc/init.d/rc?.d/) 中的啟動和停止程序檔建立必要的連結。該程式也會為這些連結的名稱加上必要的編號，即可依正確順序啟動和停止每一個 runlevel。如果您偏好以圖形工具來建立類似連結，請使用 YaST 提供的 runlevel 編輯器，請見 [第 8.2.3 節「使用 YaST 設定系統服務 \(Runlevel\)」](#) [72頁]。

如果 /etc/init.d/ 裡已經有程序檔，應該整合至現有的 runlevel 配置，使用 insserv 在 runlevel 目錄中立即建立連結，或者在 YaST 的 runlevel 編輯器中啟用對應的服務。下次重新啟動時，將會套用您所做的變更—新服務會自動啟動。

請勿手動設定這些連結。如果 INFO 區塊發生錯誤，稍後執行某些其他服務的 insserv 將會發生問題。為此程序檔手動新增的服務將會在下次執行 insserv 時予以移除。

8.2.3 使用 YaST 設定系統服務 (Runlevel)

使用「YaST」>「系統系統服務 (Runlevel)」>「啟動 YaST」模組後，會出現一個概觀清單，其中會列出所有可用的服務和每個服務目前的狀態 (停用或啟

用)。決定要以「簡單模式」或「進階模式」使用模組。大部份情況下，預設的「簡單模式」應該都已夠用。左欄顯示服務的名稱，中間欄顯示它的目前狀態，而右欄提供簡短說明。視窗下方為選取的服務提供更詳細的說明。若要啟用服務，在表格中選取它，然後選取「啟用」。停用服務的步驟也一樣。

要更仔細控制啟動或停止服務的 runlevel，或者變更預設的 runlevel，請先選取「進階模式」。目前預設的 Runlevel 或「initdefault」(系統開機時預設裝載的 Runlevel) 會出現在視窗上方。一般情況下，SUSE Linux Enterprise Desktop 系統的預設 runlevel 是 runlevel 5 (含網路和 X 的完整多重使用者模式)。合適的替代方法可能是 runlevel 3 (含網路的完整多重使用者模式)。

此 YaST 對話方塊允許其中一個 runlevel 選項 (如表格 8.1 「可用的 Runlevel」 [67頁] 所列) 做為新預設。還可使用此視窗中的表格，啟用或停用個別服務和精靈。表格會列示可用的服務和精靈，顯示目前在您的系統上它們是否啟用，如果啟用，是哪一個 runlevel。使用滑鼠選取其中一列後，按一下代表 Runlevel 的核取方塊(「B」、「0」、「1」、「2」、「3」、「5」、「6」和「S」)來定義 Runlevel，以便在該處執行選取的服務或精靈。Runlevel 4 並未定義，以便建立自定 runlevel。表格概觀的正下方，提供目前所選服務或精靈的簡要說明。

警告: 錯誤的 Runlevel 設定可能會造成系統損害

錯誤的 Runlevel 設定可能會導致系統無法使用。在您套用變更之前，請務必確定您知道它們的後果。

圖形 8.1 系統服務 (Runlevel)



使用「啟動、停止或重新整理」，決定是否啟動服務。「重新整理狀態」檢查目前狀態。您可以使用「設定或重設」來做選擇，將您的變更套用至系統，或復原啟動runlevel編輯器之前的設定。選取「完成」，就會將變更的設定儲存至磁碟。

8.3 透過 /etc/sysconfig 設定系統

SUSE Linux Enterprise Desktop 的主要組態是由 /etc/sysconfig 中的組態檔案來控制。/etc/sysconfig 中個別的檔案只由相關的程式檔讀取。這可以確保例如網路設定，只由網路相關的程序檔來剖析。

有兩種方式可以編輯系統組態：使用 YaST Sysconfig 編輯器或手動編輯組態檔。

8.3.1 使用 YaST Sysconfig 編輯器變更系統組態

YaST Sysconfig 編輯器會提供一個易於使用的前端工具，方便您設定系統。假使您對要變更的組態變數實際位置不甚了解，您只要使用此模組內建的搜尋功能即可變更所需的組態變數值，接著 YaST 會套用這些變更、根據 Sysconfig 中設定的值更新組態，然後重新啟動服務。

警告: 修改 `/etc/sysconfig/*` 檔案可能會損壞您的安裝檔案

如果您先前並無經驗和相關知識，請勿修改 `/etc/sysconfig` 檔案。它可能會嚴重破壞您的系統。`/etc/sysconfig` 中的檔案包含每一個變數的簡短註解，說明它們的實際作用。

圖形 8.2 使用 `sysconfig` 編輯器的系統組態



YaST sysconfig 對話方塊分割成三個部份。對話方塊的左側顯示所有可設定變數的樹狀結構檢視。當您選取變數時，右側會顯示目前的選擇，以及此變數的目前設定。在下方，第三個視窗顯示變數用途的簡短說明、可能值、預設值以及產生此變數的實際組態檔。對話方塊也提供關於變更變數後要執行的組態程序檔，以及變更結果會啟動什麼新服務等等的相關資訊。YaST 會要求您確認變更，並告訴您在選取「完成」並離開對話方塊後會執行的程序檔。請選取目前要略過的服務和程序檔，它們會在稍後啟動。為了讓變更生效，YaST 會自動套用所有變更並重新啟動已變更組態的服務。

8.3.2 手動變更系統組態

若要手動變更系統組態，請執行下列步驟：

- 1 以 `root` 使用者身份登入。
- 2 使用 `telinit 1`，將系統置於單一使用者模式 (`runlevel 1`)。
- 3 使用選擇的編輯器對組態檔進行所需變更。

如果未使用 YaST 來變更 `/etc/sysconfig` 中的組態檔，請確定空的變數值是由兩個引號所表示 (`KEYTABLE=""`)，而且該值與其中的空格包夾在引號中。由一個文字組成的值不需要包夾在引號中。

- 4 執行 `SUSEconfig` 來確定變更生效。
- 5 使用 `telinit` 預設 `runlevel` 指令，將系統還原成先前的 `Runlevel`。採用系統預設的 `Runlevel` 來取代 `default_runlevel`。若要返回有網路和 X 的完整多重使用者模式，請選擇 5；若要在有網路的完整多重使用者模式下工作，請選擇 3。

變更整個系統設定，例如網路組態時，才會用到此程序。小幅度變更不必進入單一使用者模式，不過您也可以進入單一使用者模式，即可百分之百確定所有相關程式都正確重新啟動。

提示: 設定自動系統組態

若要停用由 `SUSEconfig` 自動設定的系統組態，請將 `/etc/sysconfig/suseconfig` 中的 `ENABLE_SUSECONFIG` 變數設為 `no`。如果您想使用 `SUSE` 安裝支援，請勿停用 `SUSEconfig` 也可以停用部份自動組態。

開機載入程式 GRUB

本章將描述如何設定 GRUB，即 SUSE® Linux Enterprise Desktop 使用的開機載入程式。一種特殊的 YaST 模組可以用於執行所有設定。如果您不熟悉 Linux 開機的要點，請閱讀以下章節，取得部份背景資訊。本章節也敘述使用 GRUB 開機時，經常遇到的部份問題以及它們的解決方案。

注：使用 UEFI 的機器上未安裝 GRUB

依照慣例，GRUB 將會安裝在配有傳統 BIOS 的機器上與使用相容支援模組 (CSM) 的 UEFI (統一可擴充韌體介面) 機器上。未啟用 CSM 的 UEFI 機器會自動安裝 eLILO (如果 DVD1 開機成功)。如需詳細資訊，請參閱系統上 `/usr/share/doc/packages/elilo/` 中的 **eLILO** 文件。

本章節著重於開機管理以及開機載入程式 GRUB 的組態。整個開機程序簡述於 [第 8 章啟動及設定 Linux 系統](#) [63 頁]。開機載入程式是機器 (BIOS) 與作業系統 (SUSE Linux Enterprise Desktop) 之間的介面。開機載入程式的組態會直接影響作業系統的開機。

下列詞彙將在本章節中經常出現，而且可能需要說明：

主開機記錄

MBR 的結構是由作業系統 (獨立會議) 所定義。前 446 位元組是保留給程式碼。它們通常包含一部分開機載入程式或作業系統選取器。接下來的 64 位元組為最多包含四項的分割區表提供空間。分割區表包含磁碟分割以及檔案系統類型相關資訊。作業系統需要此表格來處理硬碟。在 MBR 中使用傳統標準程式碼的情況下，只有一個分割區必須標示為作用中。MBR 的

最後兩個位元組必須包含靜態的「魔術編號」(AA55)。包含不同值的 MBR 會被一些 BIOS 視為無效，所以開機時不列入考慮。

開機磁區

開機磁區是除了擴充分割區以外，硬碟分割區的第一個磁區，它只會當成其他分割區的「容器」。這些開機磁區有 512 位元組的空間供程式碼使用，而這些程式碼是用來啟動各自分割區中安裝的作業系統。此適用於 DOS、Windows 和 OS/2 分割區格式的開機磁區，它們也包含檔案系統的部份重要基本資料。相反的，Linux 分割區的開機磁區一開始是空白，直到設定不同於 XFS 的檔案系統之後，才會寫入記錄。因此，Linux 分割區無法自行啟動，即使它包含核心以及有效的根檔案系統。開機磁區如果包含可以啟動系統的有效程式碼，則它的魔術編號與 MBR 的最後兩個位元組相同(AA55)。

9.1 使用 GRUB 開機

GRUB (Grand Unified Bootloader) 包含兩個階段。第一個階段包含 512 個位元組，其唯一的任務就是載入開機載入程式的第二個階段。接下來，會載入第二個階段。這個階段會包含開機載入程式的主要部分。

某些組態會使用中繼階段 1.5，用來配置和載入適當檔案系統的階段 2。在適當情況下，這種方法在安裝或使用 YaST 初始設定 GRUB 時會是預設選項。

第二個階段可以存取許多檔案系統。目前支援 Ext2、Ext3、ReiserFS、Minix 和 Windows 使用的 DOS FAT 檔案系統。對於 XFS、UFS 以及 BSD 使用的 FFS，也支援到一定的程度。自 0.95 版開始，GRUB 也可以從包含符合「El Torito」規格的 ISO 9660 標準檔案系統的 CD 或 DVD 啟動。即使在系統啟動之前，GRUB 可以存取支援的 BIOS 磁碟機 (BIOS 偵測到的磁片或硬碟、CD 光碟機和 DVD 磁碟機) 的系統。因此，對 GRUB 組態檔案 (menu.lst) 所做的變更，將不需要重新安裝開機管理程式。啟動系統後，GRUB 會重新載入功能表檔案以及核心或起始 RAM 磁碟 (initrd) 的有效路徑和分割區資料，然後找這些檔案。

GRUB 的實際組態是以下列敘述的三個檔案為基礎：

```
/boot/grub/menu.lst
```

此檔案包含可以使用 GRUB 啟動的分割區或作業系統，所有的相關資訊。如果沒有這段資訊，GRUB 指令行就會提示使用者提供如何繼續執行 (如需要詳細資訊，請參閱[章節「在開機程序期間編輯功能表項目」](#) [83頁])。

`/boot/grub/device.map`

此檔案會從 GRUB 和 BIOS 表示法，將設備名稱轉譯成 Linux 設備名稱。

`/etc/grub.conf`

此檔案包含 GRUB 外圍程序正確安裝開機載入程式時需要的指令、參數和選項。

有多種方法可以控制 GRUB。現有組態啟動項目，可以從圖形功能表選取 (開頭顯示畫面)。組態會從檔案 `menu.lst` 載入。

在 GRUB，啟動前可以變更所有啟動參數。例如，編輯功能表檔案發生的錯誤，可以用此方法更正。開機指令也可透過輸入提示輸入 (請參閱[章節「在開機程序期間編輯功能表項目」](#) [83頁])。GRUB 提供開機前，判斷核心和 `initrd` 二者位置的可能性。以此方法，您還可以為開機載入程式組態中不存在的項目，啟動安裝的作業系統。

GRUB 事實上有兩個版本：一個是開機載入程式，一個是位於 `/usr/sbin/grub` 的一般 Linux 程式。此程式稱為 *GRUB 外圍程序*。它可以在安裝系統中提供 GRUB 模擬功能，並可用來安裝 GRUB 或是在套用之前測試設定。將 GRUB 安裝成硬碟或磁片上的開機載入程式，這種功能是以指令 `install` 和 `setup` 的形式，整合於 GRUB。Linux 載入時，可以在 GRUB 外圍程序使用此指令。

9.1.1 GRUB 開機功能表

圖形開頭顯示畫面以及開機功能表是以 GRUB 組態檔 `/boot/grub/menu.lst` 為基礎的，它包含可以透過功能表啟動的所有分割區或作業系統，全部的相關資訊。

每次啟動系統時，GRUB 會從檔案系統載入功能表檔案。基於此因素，GRUB 不需要在每次變更檔案後，重新安裝。使用 YaST 開機載入程式來修改 GRUB 組態，如[第 9.2 節「使用 YaST 設定開機載入程式」](#) [85頁]所述一般。

功能表檔案包含指令。語法相當簡單。每一行包含一個指令，後面是由像外圍程序中的空格所分開的選用參數。基於歷史因素，部份指令可以允許 = 放在第一個參數前面。註解是以井字號 (#) 開頭。

若要識別功能表綜覽中的功能表項目，請為每一個項目設定 `title`。關鍵字 `title` 後面的文字 (包括任何空格) 將在功能表中顯示成可選取的選項。當這個功能表項目被選取時，就會執行下一個 `title` 指示的所有指令。

最簡單的案例是重新導向至其他作業系統的開機載入程式。指令是 `chainloader` 及引數通常是 **GRUB** 區塊表示法中，其他分割區的啟動區塊。例如：

```
chainloader (hd0,3)+1
```

GRUB 中的設備名稱會在**章節「硬碟和分割區的命名慣例」** [80頁]說明。這個範例會指定第一個硬碟中，第四個分割區的第一個區塊。

使用指令 `kernel` 來指定核心影像。第一個引數是分割區中，核心影像的路徑。其他引數會傳送到指令行上的核心。

如果核心沒有內建驅動程式來存取根分割區、或是這時是使用包含進階 **HotPlug** 功能的最新 **Linux** 系統，`initrd` 就必須以個別的 **GRUB** 指令來指定，它唯一的引數是 `initrd` 檔案路徑。因為 `initrd` 的載入位址是記錄於載入的核心影像，所以指令 `initrd` 必須立即接在 `kernel` 指令的後面。

指令 `root` 會簡化核心和 `initrd` 檔案的指定。`root` 的唯一引數是設備或分割區。這個設備會用於所有一直到下一個 `root` 指令指定時才會出現明確指定設備的所有核心、`initrd` 或其他檔案路徑。

`boot` 指令會在每一個功能表項目最後暗示，所以它不需要寫入功能表檔案。不過，如果您使用互動式 **GRUB** 來啟動，必須在最後輸入 `boot` 指令。指令本身沒有引數。它只是啟動載入的核心影像或指定的鏈結載入器。

撰寫所有功能表項目之後，將其中一個定義為 `default` 項目。否則，會使用第一個(項目 0)做為預設項目。您也可以指定預設項目要在幾秒後啟動。`timeout` 和 `default` 通常在功能表項目前面。如需參考範例，請參閱**章節「功能表檔案範例」** [81頁]。

硬碟和分割區的命名慣例

GRUB 用於硬碟和分割區的命名慣例與用於一般 **Linux** 設備的命名慣例不同。它比較類似 **BIOS** 的簡易磁碟列舉方式，而且語法類似一些 **BSD** 衍生版本中所用的語法。在 **GRUB**，分割區的編號會從 0 開始。這表示 `(hd0, 0)` 是第一個硬碟的第一個分割區。與一般桌上型機器的 **primary master** 硬碟對應的 **Linux** 設備名稱是 `/dev/sda1`。

四個可能的主要分割區會指定分割區編號 0 到 3。邏輯分割區是從 4 開始編號：

```
(hd0,0)    first primary partition of the first hard disk
(hd0,1)    second primary partition
```

```
(hd0,2)    third primary partition
(hd0,3)    fourth primary partition (usually an extended partition)
(hd0,4)    first logical partition
(hd0,5)    second logical partition
```

由於獨立於 BIOS 設備，GRUB 不會區分 IDE、SATA、SCSI 和硬體 RAID 設備。BIOS 或其他控制器識別的所有硬碟，會按照 BIOS 的開機順序編號。

可惜的是，它通常無法將 Linux 設備名稱正確對應至 BIOS 設備名稱。它會透過演算法的協助來產生此對應，然後儲存至檔案 `device.map`，需要時可以編輯它。如需關於檔案 `device.map` 的資訊，請參閱第 9.1.2 節「檔案 `device.map`」[83頁]。

完整 GRUB 路徑包含放在括號中的設備名稱，以及指定分割區中檔案系統的檔案路徑。路徑開頭是一個斜線。例如，如果系統有一個 IDE 硬碟，它的第一個分割區包含 Linux，則可以使用下列方式設定可開機核心：

```
(hd0,0)/boot/vmlinuz
```

功能表檔案範例

以下範例顯示 GRUB 功能表檔案的結構。這個範例安裝會在 `/dev/sda5` 下安裝 Linux 開機分割區，在 `/dev/sda7` 下安裝根分割區，在 `/dev/sda1` 下進行 Windows 安裝。

```
gfxmenu (hd0,4)/boot/message
color white/blue black/light-gray
default 0
timeout 8

title linux
    root (hd0,4)
    kernel /boot/vmlinuz root=/dev/sda7 vga=791 resume=/dev/sda9
    initrd /boot/initrd

title windows
    rootnoverify (hd0,0)
    chainloader +1

title floppy
    rootnoverify (hd0,0)
    chainloader (fd0)+1

title failsafe
    root (hd0,4)
    kernel /boot/vmlinuz.shipped root=/dev/sda7 ide=nodma \
```

```
apm=off acpi=off vga=normal nosmp maxcpus=0 3 noresume  
initrd /boot/initrd.shipped
```

第一個區塊定義開頭顯示畫面的組態：

gfxmenu (hd0,4)/message

背景影像 `message` 位於 `/dev/sda5` 分割區的最上層目錄。

color white/blue black/light-gray

顏色配置：白色 (前景)、藍色 (背景)，黑色 (選擇項目)，以及淺灰色 (選擇項目的背景)。色彩配置對於開頭顯示畫面沒有作用，只對您可以存取而且可以自定的 GRUB 功能表有作用 (您可以使用 **Esc**，結束開頭顯示畫面)。

default 0

第一個功能表項目 `title linux` 是預設要啟動的項目。

timeout 8

沒有任何用者輸入，經過 8 秒後，GRUB 會自動啟動預設項目。若要停用自動開機，請刪除 `timeout`。如果設定為 `timeout 0`，GRUB 會馬上啟動預設項目。

第二個 (最大的) 區塊會列示各種可開機的作業系統。個別作業系統的區段會從 `title` 開始。

- 第一個項目 (`title linux`) 負責將 SUSE Linux Enterprise Desktop 開機。核心 (`vmlinux`) 是位在第一個硬碟的第一個邏輯分割區 (開機分割區)。核心參數，例如根分割區和 VGA 模式，會在此添加。根分割區是根據 Linux 命名慣例 (`/dev/sda7`) 指定的，因為此資訊是由核心讀取，與 GRUB 無關。`initrd` 也是位在第一個硬碟的第一個邏輯分割區。
- 第二個項目負責載入 Windows。Windows 是從第一個硬碟的第一個分割區啟動 (`hd0, 0`)。指令 `chainloader +1` 會造成 GRUB 載入並執行指定分割區的第一個磁區。
- 下一個項目允許從磁片開機，無需修改 BIOS 設定。
- 開機選項 `failsafe` 會以選擇的核心參數來啟動 Linux，可以啟動發生問題的 Linux 系統。

功能表檔案可以在需要時變更。GRUB 會在下次啟動時使用修改的設定。使用 YaST 或選擇的編輯器，永久地編輯檔案。另一種方法是，使用 GRUB 的編輯

功能，以互動方式暫時變更 (請參閱[章節「在開機程序期間編輯功能表項目」](#) [83頁])。

在開機程序期間編輯功能表項目

在圖形化開機功能表中，可以使用方向鍵選取要啟動的作業系統。如果選取 Linux 系統，可以在開機提示時輸入其他啟動參數。若要直接編輯個別功能表項目，請按 **Esc** 結束開頭顯示畫面，然後跳到 GRUB 文字功能表，再按 **E**。用此方式所做的變更，只會套用到目前開機，因此不會永久變更。

重要：開機程序期間的鍵盤配置

US 鍵盤配置是啟動時唯一可以使用的鍵盤配置。請參閱圖形 6.1「US Keyboard Layout」(↑[系統分析與調整指南](#))中的圖形。

編輯功能表項目有利於修復無法再啟動的缺陷系統，因為開機載入程式錯誤的組態檔，可以手動輸入參數，便得以解決。在開機程序期間手動輸入參數，對於測試新設定但可避免損壞原始系統，有很大的幫助。

啟用編輯模式之後，使用方向鍵選取功能表項目來編輯組態。若要讓組態變成可編輯狀態，請再按 **E** 一次。使用此方法，在對於開機程序產生負面影響之前編輯錯誤的分割區或路徑指定。按 **Enter**，結束編輯模式並返回功能表。然後按 **B** 來啟動此項目。底下的說明文字會顯示進一步可行的動作。

若要永久地輸入變更的開機選項，然後傳送至核心，以 `root` 身份開啟檔案 `menu.lst`，然後將各自的核心參數加入到現有的指令行，並以空格分隔：

```
title linux
    root(hd0,0)
    kernel /vmlinuz root=/dev/sda3 additional parameter
    initrd /initrd
```

GRUB 會在下次啟動系統時，使用新參數。另一種方法是，使用 YaST 開機載入程式模組做此變更。將新參數加入至現有的行，並用空格分開。

9.1.2 檔案 `device.map`

檔案 `device.map` 會將 GRUB 和 BIOS 設備名稱對應至 Linux 設備名稱。在包含 IDE 和 SCSI 硬碟的混合系統中，GRUB 必須透過特定程序來判斷開機順序，因為 GRUB 可能不會存取 BIOS 的開機順序資訊。GRUB 儲存分析的結果至檔

案 `/boot/grub/device.map`。在 BIOS 中的開機順序設定成 IDE 在 SCSI 前面的系統，檔案 `device.map` 會顯示如下：

```
(fd0)  /dev/fd0
(hd0)  /dev/sda
(hd1)  /dev/sdb
```

因為 IDE、SCSI 和其他硬碟的順序取決於各種因素，而且 Linux 無法識別對應，所以檔案 `device.map` 可以手動設定。如果您在啟動時發生問題，檢查此檔案中的順序是否對應至 BIOS 的順序，然後在需要時使用 GRUB 提示來暫時修改它。啟動 Linux 系統之後，檔案 `device.map` 可以透過 YaST 開機載入程式或其他選擇的編輯器，永久性編輯。

在手動變更檔案 `device.map` 之後，執行以下指令來重新安裝 GRUB。此指令會造成檔案 `device.map` 重新載入，並以 `grub.conf` 列示的指令執行：

```
grub --batch < /etc/grub.conf
```

9.1.3 檔案 `/etc/grub.conf`

除了 `menu.lst` 和 `device.map` 之外，第三重要的 GRUB 組態檔案是 `/etc/grub.conf`。此檔案包含 GRUB 外圍程序正確安裝開機載入程式時需要的指令、參數和選項：

```
setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)
quit
```

此指令會告知 GRUB 自動將開機載入程式安裝到第一個硬碟上的第二個分割區 (`hd0,1`)，安裝時使用位於同一個分割區上的開機影像。從已裝載的檔案系統安裝 `stage2` 影像時必須使用 `--stage2=/boot/grub/stage2` 參數。部分 BIOS 的 LBA 支援實作錯誤，`--force-lba` 提供了忽略這些錯誤的解決方案。

9.1.4 設定啟動密碼

即使作業系統啟動之前，GRUB 也可以存取檔案系統。沒有 root 許可權的使用者，在此時可以存取 Linux 系統中的檔案 (這些檔案在系統啟動後，他們並無法存取)。若要封鎖此類型的存取、或者防止使用者啟動特定作業系統，請設定開機密碼。

重要：開機密碼和開頭顯示畫面

如果您在 GRUB 使用啟動密碼，將不會顯示一般的開頭顯示畫面。

按照以下方式，以使用者 `root` 的身份設定開機密碼：

- 1 在 `root` 提示下，使用 `grub-md5-crypt` 將密碼加密：

```
# grub-md5-crypt
Password: ****
Retype password: ****
Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 2 將加密字串貼到檔案 `menu.lst` 的全域區段：

```
gfxmenu (hd0,4)/message
color white/blue black/light-gray
default 0
timeout 8
password --md5 $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

現在 GRUB 指令只可以在按 `P` 並輸入密碼後的啟動提示時執行。不過，使用者仍然可以從開機功能表，啟動所有作業系統。

- 3 要防止一或多個作業系統從開機功能表啟動，將項目 `lock` 新增至沒有密碼便不可以啟動的 `menu.lst` 每一個段落。例如：

```
title linux
    kernel (hd0,4)/vmlinuz root=/dev/sda7 vga=791
    initrd (hd0,4)/initrd
    lock
```

重新啟動系統並從開機功能表選取 `Linux` 之後，將會顯示以下錯誤訊息：

```
Error 32: Must be authenticated
```

按 `Enter` 進入功能表。然後按 `P`，進入密碼提示要求。輸入密碼並按 `Enter` 之後，應該就可啟動選取的作業系統 (本範例為 `Linux`)。

9.2 使用 YaST 設定開機載入程式

在 SUSE Linux Enterprise Desktop 系統中設定開機載入程式最簡單的方法是使用 YaST 模組。在「YaST 控制中心」選取「系統」|「」>「開機載入程式組態」。

正如 **圖形 9.1 「開機載入程式設定」** [86頁] 所示，這時會顯示系統目前的開機載入程式組態，並允許您進行變更。

圖形 9.1 開機載入程式設定



使用「磁區管理」索引標籤來編輯、變更和刪除個別作業系統的開機載入程式磁區。要加入選項，請按一下「新增」。若要變更現有選項的值，請先用滑鼠選取，接著按一下「編輯」。如果要移除現有項目，請選取它並按一下「刪除」。如果不熟悉開機載入程式選項，請先閱讀**第 9.1 節「使用 GRUB 開機」** [78頁]。

使用「開機載入程式安裝」索引標籤，來檢視和變更關於類型、位置和進階載入器設定的設定。

存取下拉式功能表 (按一下「其他」即可開啓) 中的進階組態選項。內建的編輯器可讓您變更 GRUB 組態檔案 (請參閱**第 9.1 節「使用 GRUB 開機」** [78頁] 以取得詳細資料)。還可以刪除現有的組態然後「從頭開始」，或者讓 YaST 「建議新的組態」。也可以將組態寫入磁碟，或者從磁碟中重新讀取組態。要還原安裝期間儲存的原始主開機記錄 (MBR)，請選擇「還原硬碟的 MBR」。

9.2.1 調整預設開機項目

若要變更預設開機的系統，請依照下列步驟執行：

過程 9.1 設定預設系統

- 1 開啟「磁區管理」索引標籤。
- 2 從清單中選取所需項目。
- 3 按一下「設定為預設值」。
- 4 按一下「完成」，啟用這些變更。

9.2.2 修改開機載入程式位置

若要修改開機載入程式的位置，請執行下列步驟：

過程 9.2 變更開機載入程式位置

- 1 選取「開機載入程式安裝」索引標籤，然後為「開機載入程式位置」選擇以下其中一個選項：

從開機分割區開機

/boot 分割區的開機磁區

從延伸分割區開機

這會在延伸分割區容器中安裝開機載入程式。

從主開機記錄開機

這樣會在第一個磁碟的 MBR 中安裝開機載入程式 (根據 BIOS 中預設的開機順序)。

從根分割區開機

這會在 / 分割區的開機磁區中安裝開機載入程式。

自定開機分割區

這個選項可讓您手動指定開機載入程式的位置。

- 2 按一下「完成」，套用變更。

9.2.3 變更開機載入程式逾時

開機載入程式不會立即啟動預設系統。在逾時期間，您可以選取系統開機，或是寫入一些核心參數。若要設定開機載入程式逾時時間，請依照下列步驟執行：

過程 9.3 變更開機載入程式逾時

- 1 開啟「開機載入程式安裝」索引標籤。
- 2 按一下「開機載入程式選項」。
- 3 輸入新的值或使用滑鼠按住適當的方向鍵、或使用鍵盤上的方向鍵，以變更「逾時秒數」的設定值。
- 4 按一下「確定」。
- 5 按一下「完成」，儲存這些變更。

9.2.4 設定啟動密碼

使用 YaST 模組，您也可以設定密碼來保護開機。這可以提供您另一層安全保護。

過程 9.4 設定開機載入程式密碼

- 1 開啟「開機載入程式安裝」索引標籤。
- 2 按一下「開機載入程式選項」。
- 3 在「功能表介面密碼」中設定您的密碼。
- 4 按一下「確定」。
- 5 按一下「完成」，儲存這些變更。

9.2.5 調整磁碟順序

如果您的電腦有多個硬碟，您可以指定磁碟開機順序採用機器 BIOS 的設定 (請參閱第 9.1.2 節「檔案 device.map」[83頁])。若要執行這個動作，請依照下列步驟執行：

過程 9.5 設定磁碟順序

- 1 開啟「開機載入程式安裝」索引標籤。
- 2 按一下「開機載入程式詳細資訊」。
- 3 如果列出超過一個磁碟，請選擇一個磁碟，然後按一下「向上」或「向下」重新排列顯示磁碟的順序。
- 4 按一下「確定」儲存變更。
- 5 按一下「完成」，儲存這些變更。

9.2.6 設定進階選項

可透過「開機載入程式安裝」>「開機載入程式選項」設定進階開機選項。一般情況下，不需要變更預設設定。

「在分割區表中為開機分割區設定使用中旗標」

啟動包含開機載入程式的分割區。有的舊作業系統 (例如 Windows 98) 只能從使用中分割區開機。

除錯旗標

將 GRUB 設定為除錯模式，以顯示描述磁碟活動的訊息。

將一般開機碼寫入 MBR

以與作業系統無關的一般代碼取代目前的 MBR。

隱藏開機功能表

隱藏開機功能表並使用預設項目開機。

使用信任的 GRUB

啟動信任的 GRUB，以支援信任的計算功能。

圖形功能表檔案

顯示開機畫面時所使用的圖形檔案路徑。

序列連接參數

如果您的機器是透過序列主控台控制的，您可以指定以何種速度來使用哪一個 COM 埠。同時請將「終端機定義」設定為「serial」。如需詳細資料，請參閱 info grub 或 <http://www.gnu.org/software/grub/manual/grub.html>。

終端機定義

若要透過序列主控台開機，請在此處輸入「serial」，否則請將其留為空白。此時還需要指定「序列連接參數」。

9.2.7 變更開機載入程式類型

在「開機載入程式安裝」中設定開機載入程式類型。SUSE Linux Enterprise Desktop 中的預設開機載入程式為 GRUB。若要使用 LILO，請繼續下列步驟：

過程 9.6 變更開機載入程式類型

- 1 開啟「開機載入程式安裝」索引標籤。
- 2 為「開機載入程式」選取「*LILO*」。
- 3 在這時開啟的對話方塊中，選取下面其中一個動作：

建議新組態

指定 YaST 建議新的組態

轉換目前的組態

指定 YaST 轉換目前的組態。轉換組態時，可能會遺失某些設定。

從頭開始設定新的組態

寫入自定的組態。安裝 SUSE Linux Enterprise Desktop 時不能使用此動作。

讀取儲存在硬碟的組態

載入您個人的 `/etc/lilo.conf`。安裝 SUSE Linux Enterprise Desktop 時不能使用此動作。

4 按一下「*確定*」來儲存變更。

5 按一下主對話方塊視窗的「*完成*」，套用變更。

在轉換過程中，舊的 GRUB 組態會儲存在硬碟。若要使用它，只要將開機載入程式類型變更回 GRUB，然後從快顯示功能表選擇「*還原轉換前儲存的組態*」。這個動作只能在已安裝系統上使用。

注: 自定開機載入程式

如果要使用 GRUB 或 LILO 以外的開機載入程式，請選取「*不要安裝任何開機載入程式*」。請先詳細閱讀開機載入程式的說明文件，再選取這個選項。

9.3 解除安裝 Linux 開機載入程式

YaST 可以用來解除安裝 Linux 開機載入程式，並將 MBR 還原回安裝 Linux 前的狀態。安裝時，YaST 會自動建立原始 MBR 的備份，並在需要時還原。

要解除安裝 GRUB，請開啟 YaST 開機載入程式模組（「*系統*」>「*開機載入程式*」）。選取「*其他*」>「*還原硬碟的 MBR*」，並以「*是，重新寫入*」進行確認。

9.4 建立開機 CD

如果使用開機管理員來啟動系統發生問題，或者開機管理員無法安裝在硬碟或磁片的 MBR 上，也可以建立一張包含 Linux 所有必要啟動檔案的開機 CD。您的系統需要有一個 CD 燒錄器。

使用 GRUB 建立可開機的 CD-ROM 只需要一個特殊形式的 *stage2* (稱為 *stage2_eltorito*)，並可以選用自定的 *menu.lst*。不需要典型的檔案 *stage1* 和 *stage2*。

過程 9.7 建立開機 CD

- 1 變更至要在其中建立 ISO 影像的目錄，例如：`cd /tmp`
- 2 為 GRUB 建立子目錄，然後變更至新建立的 *iso* 目錄：

```
mkdir -p iso/boot/grub && cd iso
```

- 3 將核心以及stage2_eltorito、initrd、menu.lst 和 message 等檔案複製至 iso/boot/:

```
cp /boot/vmlinuz boot/  
cp /boot/initrd boot/  
cp /boot/message boot/  
cp /usr/lib/grub/stage2_eltorito boot/grub  
cp /boot/grub/menu.lst boot/grub
```

- 4 調整 boot/grub/menu.lst 中的路徑項目，使它們指向光碟機。方法是以 **CD-ROM** 光碟機的裝置名稱 (cd) 來取代路徑名稱中以 (hdx,y) 格式列出的硬碟設備名稱。您可能還需要調整訊息檔案、核心和 initrd 的路徑 — 它們分別需指向 /boot/message、/boot/vmlinuz 和 /boot/initrd。調整之後，menu.lst 的內容應類似於下面的範例：

```
timeout 8  
default 0  
gfxmenu (cd)/boot/message  
  
title Linux  
    root (cd)  
    kernel /boot/vmlinuz root=/dev/sda5 vga=794 resume=/dev/sda1 \  
    splash=verbose showopts  
    initrd /boot/initrd
```

使用 splash=silent (而非使用 splash=verbose) 讓開機訊息不要出現在開機程序中。

- 5 使用以下指令建立 ISO 影像：

```
genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot \  
-boot-load-size 4 -boot-info-table -iso-level 2 -input-charset utf-8 \  
-o grub.iso /tmp/iso
```

- 6 使用您偏好的公程式，將產生的檔案 grub.iso 燒錄至光碟。請勿將 ISO 影像燒錄為資料檔案；請在您的燒錄公程式中使用燒錄光碟影像的選項。

9.5 圖形化 SUSE 畫面

如果使用 `vga=值` 選項做為核心參數，則圖形 SUSE 螢幕會顯示在第一個主控台上。如果您使用 YaST 來安裝，則會依照選取的解析度與圖形卡來自動啟用此選項。視需要，有三種方式可以停用 SUSE 畫面：

必要時停用 SUSE 畫面。

可以在指令行中輸入指令 `echo 0 >/proc/splash` 以停用圖形式畫面。若要再次啟用它，請輸入 `echo 1 >/proc/splash`。

預設停用 SUSE 畫面。

您可以新增核心參數 `splash=0` 到開機載入程式的組態。如需更多詳細資訊，請參閱第 9 章 **開機載入程式 GRUB** [77 頁]。但是，如果您想使用文字模式 (較早版本的預設值)，則請設定 `vga=normal`。

完全停用 SUSE 畫面

編譯新的核心且關閉「**框架緩衝區支援**」中的「**使用開機顯示畫面而非開機標幟**」選項。

提示

在核心中關閉 `framebuffer` 支援將會自動關閉開機顯示畫面。如果您使用自定核心來執行，SUSE 將不會為您的系統提供任何支援。

9.6 疑難排解

此章節列示使用 GRUB 啟動時，經常遇到的問題以及可能解決方案的簡要說明。部份問題已經在知識庫 (網址為 <http://support.novell.com/>) 的文章中做了說明。使用搜尋對話方塊尋找 **GRUB**、**開機** 和 **開機載入程式** 這類的關鍵字。

GRUB 和 XFS

XFS 在分割區啟動區塊中，不會保留空間給 `stage1`。因此，不要將 XFS 分割區指定成開機載入程式的位置。您可以建立不是以 XFS 格式化的個別開機分割區來解決這個問題。

GRUB 報告 GRUB Geom 錯誤

GRUB 會在系統啟動時檢查連接硬碟的位置。有時候，BIOS 會傳回不一致的資訊，而且 GRUB 會報告 GRUB 位置錯誤。若這樣的話，請更新 BIOS。

如果 Linux 安裝在其他硬碟上，而且未註冊在 BIOS，GRUB 也會傳回此錯誤訊息。開機載入程式的 *stage1* 可以正確找到和載入，不過 *stage2* 則找不到。將新磁碟登錄在 BIOS 就可以解決此問題。

具有多個硬碟的系統無法開機

安裝時，YaST 可能已經判斷硬碟的開機順序錯誤。例如，GRUB 可能會將 IDE 磁碟當成 `hd0`，將 SCSI 磁碟當成 `hd1`，即使 BIOS 中的開機順序與此相反，即 SCSI 在 IDE 之前。

發生此狀況，在啟動程序時，透過 GRUB 指令行的協助來更正硬碟。在系統啟動之後，編輯 `device.map` 來永久套用新的對應。然後檢查檔案 `/boot/grub/menu.lst` 和 `/boot/grub/device.map` 中的 GRUB 設備名稱，然後使用以下指令，重新安裝開機載入程式：

```
grub --batch < /etc/grub.conf
```

從第二顆硬碟啟動 Windows

有些作業系統，例如 Windows，只可以從第一個硬碟啟動。在第一個硬碟以外的硬碟安裝類似作業系統時，會影響個別功能表項目的邏輯變更。

```
...
title windows
    map (hd0) (hd1)
    map (hd1) (hd0)
    chainloader (hd1,0)+1
...
```

在此範例中，Windows 是從第二個硬碟啟動。基於此目的，硬碟的邏輯順序是使用 `map` 變更的。此變更不會影響 GRUB 功能表檔案中的邏輯。因此，第二個硬碟必須指定成 `chainloader`。

9.7 如需更多資訊

如需關於 GRUB 更進一步的資訊，請參閱<http://www.gnu.org/software/grub/>。另請參閱 `grub info` 頁面。您也可以在「技術資訊搜尋」搜尋關鍵字「GRUB」，取得關於特殊問題的相關資訊，網址是 <http://www.novell.com/support>。

特殊系統功能

本章節會提供關於軟體套件、虛擬主控台及鍵盤配置等資訊。介紹 `bash`、`cron` 和 `logrotate` 軟體元件，是因為這些元件自上一版後有所變更或加強。這些元件也許不很重要，但與系統的關係密切，使用者可能想變更它們的預設動作。本章最後一節則會介紹語言與國家的專用設定 (I18N 與 L10N)。

10.1 特殊軟體套件的資訊

程式 `bash`、`cron`、`logrotate`、`locate`、`ulimit` 和 `free`，對系統管理員和許多使用者而言十分重要。線上文件和 `info` 頁面是兩個很有用的指令資訊來源，但並非隨時都能使用。`GNU Emacs` 是非常普遍而且很好設定的文字編輯器。

10.1.1 `bash` 套件與 `/etc/profile`

`Bash` 是預設的系統外圍程序。如果以它做為登入外圍程序，可以讀取多種啟始化檔案。`Bash` 會以它們顯示在清單中的順序來處理。

1. /etc/profile
2. ~/.profile
3. /etc/bash.bashrc
4. ~/.bashrc

在 ~/.profile 或 ~/.bashrc 中進行自定設定。為了要確保這些檔案能正確的處理，您必須將基本設定從 /etc/skel/.profile 或 /etc/skel/.bashrc 中複製至使用者的主目錄。建議您在更新後從 /etc/skel 複製設定。請執行下列的外圍程式指令，以避免遺失您調整過的設定。

```
mv ~/.bashrc ~/.bashrc.old
cp /etc/skel/.bashrc ~/.bashrc
mv ~/.profile ~/.profile.old
cp /etc/skel/.profile ~/.profile
```

然後個人的調整設定需要從 *.old 檔案再複製回去。

10.1.2 cron 套件

如果您需要按預先定義的時間，在背景中定期自動執行指令，則可以使用 cron 工具。cron 由採用特殊格式的時間表驅動。其中某些表格是由系統提供，使用者可以視需要撰寫自己的表格。

cron 表格現在位於 /var/cron/tabs。/etc/crontab 做為整個系統的 cron 表格。在時間表格之後、指令之前，輸入要直接執行指令的使用者名稱。在 **範例 10.1「/etc/crontab 中的項目」** [96 頁] 中，則是輸入 root。位於 /etc/cron.d 的套件專用表格有相同的格式。請參閱 cron 線上文件 (man cron)。

範例 10.1 /etc/crontab 中的項目

```
1-59/5 * * * * root test -x /usr/sbin/atrun && /usr/sbin/atrun
```

您不能呼叫 crontab -e 指令來編輯 /etc/crontab。這個檔案必須直接載入編輯器中進行修改，然後儲存。

有些套件會將外圍程式程序檔安裝至 /etc/cron.hourly、/etc/cron.daily、/etc/cron.weekly 及 /etc/cron.monthly 等目錄中，其執行由 /usr/lib/cron/run-crons 控制。/usr/lib/cron/run-crons 每隔

15分鐘會從主表格 (/etc/crontab) 執行一次。這會保證被忽略的程序可以在適當的時間執行。

如果要按照自定時間，執行 hourly、daily 或其他定期維護程序檔，請定期使用 /etc/crontab 項目移除時間戳記檔案 (請參閱 **範例 10.2 「/etc/crontab: 移除時戳檔案」** [97頁]，它可以在整點前移除 hourly、在每天的 2:14 a.m. 移除 daily 等)。

範例 10.2 /etc/crontab: 移除時戳檔案

```
59 * * * * root rm -f /var/spool/cron/lastrun/cron.hourly
14 2 * * * root rm -f /var/spool/cron/lastrun/cron.daily
29 2 * * 6 root rm -f /var/spool/cron/lastrun/cron.weekly
44 2 1 * * root rm -f /var/spool/cron/lastrun/cron.monthly
```

或者，可以將 /etc/sysconfig/cron 中的 DAILY_TIME 設定為 cron.daily 啟動的時間。MAX_NOT_RUN 設定確保能夠觸發日常工作的執行，即使使用者在長時間內未按指定的 DAILY_TIME 開啟電腦。MAX_NOT_RUN 的最大值為 14 天。

為明確起見，日常系統維護工作會配送至不同的程序檔。它們包含在 aaa_base 套件中。例如，/etc/cron.daily 中有 suse.de-backup-rpmdb、suse.de-clean-tmp 或 suse.de-cron-local 等元件。

10.1.3 記錄檔：套件 logrotate

某些系統服務 (*daemon*) 以及核心本身，會定期將系統狀態與特定事件記錄到記錄檔中。這樣管理員可以定期在某個時間點檢查系統的狀態、找出錯誤或有問題的功能，並且用精確的方式來排除它們。這些記錄檔通常以 FHS 所指定的方式儲存於 /var/log，而且會日益增大。logrotate 套件有助於控制這些檔案增大的方式。

使用檔案 /etc/logrotate.conf 來設定 logrotate。尤其 include 規格主要是設定其他要讀取的檔案。產生記錄檔的程式會在 /etc/logrotate.d 中單獨安裝組態檔。例如，套件隨附的檔案，如 apache2 (/etc/logrotate.d/apache2) 與 syslogd (/etc/logrotate.d/syslog)。

範例 10.3 */etc/logrotate.conf* 的範例

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own lastlog or wtmp - we'll rotate them here
#/var/log/wtmp {
#     monthly
#     create 0664 root utmp
#     rotate 1
#}

# system-specific logs may be also be configured here.
```

logrotate 是透過 cron 來控制，並且每日是經由 */etc/cron.daily/logrotate* 來呼叫。

重要

create 選項會讀取 */etc/permissions** 中由管理員所做的所有設定。請確定個人的修改不會造成衝突。

10.1.4 locate 指令

可以快速尋找檔案的 **locate** 指令，並不包含在安裝軟體的標準範圍中。若有需要，請安裝 **findutils-locate** 套件。**updatedb** 程序會在每晚自行啟動，或啟動系統後的 15 分鐘左右啟動。

10.1.5 ulimit 指令

利用 `ulimit` (使用者限制) 指令，您可以限制系統資源的使用，並顯示這些限制。`ulimit` 對於限制應用程式可用的記憶體特別有用。利用它，可以避免應用程式使用過多的記憶體空間，使用過多的記憶體空間可能會導致系統暫停。

`ulimit` 可以搭配多種選項來使用。若要限制記憶體的使用，請利用 [表格 10.1 「ulimit：設定使用者的資源」](#) [99頁] 中所列的選項。

表格 10.1 *ulimit*：設定使用者的資源

<code>-m</code>	最大的常駐集大小
<code>-v</code>	外圍程序可用的虛擬記憶體最大容量
<code>-s</code>	堆疊的最大大小
<code>-c</code>	所建立的核心檔案的最大大小
<code>-a</code>	將會報告所有目前限制

您可以在 `/etc/profile` 中設定全系統的項目。在此可建立核心檔，以供程式設計人員除錯之用。一般使用者無法增加系統管理員在 `/etc/profile` 中所指定的值，但可以在 `~/.bashrc` 中建立特殊的設定項目。

範例 10.4 *ulimit*：~/.bashrc 中的設定

```
# Limits maximum resident set size (physical memory):
ulimit -m 98304

# Limits of virtual memory:
ulimit -v 98304
```

記憶體的單位必須為 **KB**。如需詳細資訊，請參閱 `man bash`。

重要

並非所有的外圍程序都支援 `ulimit` 指示詞。在您根據這些限制的內含設定時，**PAM** (例如 `pam_limits`) 會提供全面的調整設定。

10.1.6 free 指令

如果您的目標是要找出目前使用了多少RAM的話，則是稍微誤解 `free` 指令的用途。您可以在 `/proc/meminfo` 找到此資訊。近來，使用 Linux 之類新式作業系統的使用者，真的不太需要擔心記憶體的問題。*可用的RAM*的概念要回溯到聯合記憶體管理的年代之前。*記憶體要物盡其用的*口號非常適用於 Linux。所以，Linux 一直致力於平衡快取，而不允許有剩餘或未使用的記憶體。

基本上，核心不會有任何應用程式或使用者資料的直接知識。相反地，它會在頁面快取中管理應用程式與使用者資料。如果記憶體不足，部份的記憶體資料會寫入交換分割區或檔案中，這部份一開始就能用 `mmap` 指令的協助來讀取 (請參閱 `man mmap`)。

核心也可以有其他的快取，例如 *slab* 快取，網路存取的快取資料會儲存於此處。這可以解釋 `/proc/meminfo` 中計數器之間的不同。它們大部分但非全部都可以透過 `/proc/slabinfo` 來存取。

10.1.7 線上文件和資訊頁面

某些 GNU 應用程式 (例如 `tar`) 不再支援 `man` 頁面。針對這些指令，請使用 `--help` 選項來取得 `info` 頁面的快速綜覽，這些頁面將提供更深入詳盡的說明。`info` 是 GNU 的超連結文字系統。您可以輸入 `info info` 來讀取此系統的介紹。您可以輸入 `emacs -f info` 或直接在主控台使用 `info`，以便使用 Emacs 檢視 `info` 頁面。您也可以使用 `tkinfo`、`xinfo` 或說明系統檢視 `info` 頁面。

10.1.8 使用 man 指令選取線上文件。

使用 `man` 線上文件通常會顯示可立即閱讀的線上文件。現在，如果其它區段存在具有相同名稱的線上文件，`man` 指令就會提示使用者選擇要顯示哪個區段的頁面，使用者需輸入區段做為回答。

如果您要返回上一行為，請在外圍程序啟始化檔案 (例如 `~/.bashrc`) 中設定 `MAN_POSIXLY_CORRECT=1`。

10.1.9 GNU Emacs 的設定

GNU Emacs 是個複雜的工作環境。以下幾個小節包含在 GNU Emacs 啟動時組態檔案的處理情形。更多相關資訊可在 <http://www.gnu.org/software/emacs/> 取得。

啟動時，Emacs 會讀取多個檔案，包含使用者、系統管理員與供應商的設定，以及取得自定或預設組態的設定。啟始化檔案 `~/.emacs` 會從 `/etc/skel` 安裝至個別使用者的主目錄。`.emacs` 接著會讀取 `/etc/skel/.gnu-emacs` 檔案。如果要自定程式，請將 `.gnu-emacs` 複製到主目錄 (利用 `cp /etc/skel/.gnu-emacs ~/.gnu-emacs` 指令)，並依照您的需求來設定。

`.gnu-emacs` 定義 `~/.gnu-emacs-custom` 檔案為自定檔案。如果使用者是使用 Emacs 中的自定選項來進行設定，這些設定會儲存至 `~/.gnu-emacs-custom` 中。

透過 SUSE Linux Enterprise Desktop，emacs 套件可將檔案 `site-start.el` 安裝至目錄 `/usr/share/emacs/site-lisp` 中。`site-start.el` 檔案會在啟始化檔案 `~/.emacs` 前載入。此外，`site-start.el` 會確保那些以 Emacs 附加套件來散佈的特定組態檔案皆能自動載入，例如 `psgml`。此類型的組態檔案也位於 `/usr/share/emacs/site-lisp` 中，並且會以 `suse-start-` 為開頭。本地系統管理員可在 `default.el` 中指定整個系統的設定。

有關這些檔案的詳細資訊可在 *Init File* 下的 Emacs 資訊檔案中取得：[info://emacs/InitFile](http://www.gnu.org/software/emacs/InitFile)。關於如何關閉這些檔案的載入 (若有需要) 的資訊，也可在此取得。

Emacs 的元件分成數個套件：

- `emacs` 基本套件。
- `emacs-x11` (通常已安裝)：具有 X11 支援的程式。
- `emacs-nox`：沒有 X11 支援的程式。
- `emacs-info`：info 格式的線上文件。
- `emacs-el`：以 `emacs lisp` 編寫的未編譯文件庫檔案。執行期間用不到這類檔案。

- 需要的話可安裝多種外掛套件：emacs-auctex (LaTeX 用)、psgml (SGML 與 XML 用)、gnuserv (用戶端與伺服器作業用)，以及其他。

10.2 虛擬主控台

Linux 是多重使用者及多工的作業系統。這些功能的優點即使在獨立的個人電腦系統中一樣令人讚賞。在文字模式中，有六個虛擬主控台可用。請使用 **Alt + F1** 到 **Alt + F6** 這些鍵來切換虛擬主控台。第七個主控台保留給 X 使用，第十個主控台可以顯示核心訊息。您可以修改 `/etc/inittab` 檔案來指定較多或較少的主控台。

若要在不關閉主控台的情況下，從 X 切換到主控台，請使用 **Ctrl + Alt + F1** 到 **Ctrl + Alt + F6** 這些鍵。若要回到 x，請按 **Alt + F7**。

10.3 鍵盤配置

若要標準化程式的鍵盤配置，請變更下列的檔案：

```
/etc/inputrc
/etc/X11/Xmodmap
/etc/skel/.Xmodmap
/etc/skel/.exrc
/etc/skel/.less
/etc/skel/.lesskey
/etc/csh.cshrc
/etc/termcap
/usr/lib/terminfo/x/xterm
/usr/share/X11/app-defaults/XTerm
/usr/share/emacs/VERSION/site-lisp/term/*.el
```

這些變更僅會影響使用 `terminfo` 項目的應用程式，或直接變更其組態檔的應用程式 (`vi`、`less` 等等。)。未隨附於此系統的應用程式必須相容於這些預設值。

在 X 中，可以使用 **ctrl + shift (右邊)** 來存取組合鍵 (組合鍵)。請參閱 `/etc/X11/Xmodmap` 中的對應項目。

使用 X 鍵盤延伸程式 (XKB)，可以進行進一步的設定。GNOME (`gswitchit`) 以及 KDE (`kxkb`) 桌面環境也會使用此延伸程式。

提示: 如需更多資訊

有關 **XKB** 資訊可在 `/etc/X11/xkb/README` 檔案中取得，也可以從列於該檔案中的文件取得。

關於中文、日文以及韓文 (CJK) 輸入的詳細資訊，可在 **Mike Fabian** 網頁中取得：<http://www.suse.de/~mfabian/suse-cjk/input.html>。

10.4 語言與國家專用的設定

本系統在很大程度上是國際化的軟體，而且能以彈性的方式針對當地的需求進行修改。換句話說，國際化 (*I18N*) 允許特定的當地語系化 (*L10N*)。I18N 與 L10N 這兩個縮寫是取首尾兩個字母，兩字母中間再加上省略的字母數目。

設定位於 `/etc/sysconfig/language` 中所定義的 `LC_` 變數。設定範圍除了本地語言支援外，還包括訊息 (語言)、字元集、排序順序、時間和日期、數字及貨幣等類別。每種類別都可以用自己的變數來直接定義，或用在 `language` 檔案中的主要變數來間接定義 (請參閱 `locale man` 頁面)。

`RC_LC_MESSAGES`、`RC_LC_CTYPE`、`RC_LC_COLLATE`、`RC_LC_TIME`、`RC_LC_NUMERIC`、`RC_LC_MONETARY`

這些變數會傳送到外圍程序，但不會包含 `RC_` 字首，並代表列出的類別。相關外圍程序設定檔會列於下面。目前的設定可以用 `locale` 指令來顯示。

`RC_LC_ALL`

此變數 (如果設定) 會覆寫先前所提到的變數值。

`RC_LANG`

如果沒有設定前面的變數，則此為備用變數。依照預設，只會設定 `RC_LANG`。這讓使用者更容易輸入自己的值。

`ROOT_USES_LANG`

有 `yes` 或 `no` 兩個變數。如果設為 `no` 的話，則 `root` 永遠可在 **POSIX** 環境中作業。

變數可以用 **YaST sysconfig** 編輯器來設定 (請參閱第 8.3.1 節「**使用 YaST Sysconfig 編輯器變更系統組態**」[75頁])。這樣的變數值中包含語言碼、國碼、編碼及修飾元。個別的元件會以特定的字元來連接：

```
LANG=<language>[_<COUNTRY>].<Encoding>[@<Modifier>]]
```

10.4.1 一些範例

您必須將語言與國碼一起設定。語言設定必須符合<http://www.evertype.com/standards/iso639/iso639-en.html>和<http://www.loc.gov/standards/iso639-2/>中的標準 ISO 639。國碼列在http://www.din.de/gremien/nas/nabd/iso3166ma/codlstp1/en_listp1.html中的 ISO 3166。

只有設定那些可以在 `/usr/lib/locale` 中找到的可用描述檔案的值，才會有意義。您可以用 `localedef` 指令從 `/usr/share/i18n` 中的檔案建立其他描述檔；描述檔屬於 `glibc-i18ndata` 套件的一部份。`en_US.UTF-8` (針對美式英文) 的描述檔可以用以下指令建立：

```
localedef -i en_US -f UTF-8 en_US.UTF-8
```

```
LANG=en_US.UTF-8
```

如果安裝期間選擇美式英文的話，則此為預設設定。如果您選擇了其他語言，則仍然可以使用該語言，但會以 UTF-8 做為字元編碼。

```
LANG=en_US.ISO-8859-1
```

這會將語言設成英文、國家設成美國、字元集設成 ISO-8859-1。此字元集並不支援歐元符號，但有時對於尚未支援 UTF-8 的程式卻非常實用。然後，有些程式將會評估定義字元集的 (此例為 ISO-8859-1) 的字串，像是 Emacs。

```
LANG=en_IE@euro
```

上方範例在語言設定中明確包括歐元符號。嚴格來說，這個設定現在已不再使用，因為 UTF-8 也涵蓋歐元符號。它只有在應用程式不支援 UTF-8 而只支援 ISO-8859-15 時才有用。

SuSEconfig 會讀取 `/etc/sysconfig/language` 中的變數，並將必要的變更寫入 `/etc/SuSEconfig/profile` 與 `/etc/SuSEconfig/csh.cshrc` 中。`/etc/profile` 將讀取 `/etc/SuSEconfig/profile`，或將之做為來源。`/etc/SuSEconfig/csh.cshrc` 則是 `/etc/csh.cshrc` 的來源。這讓這些設定能在整個系統中使用。

使用者可以適當地編輯自己的 `~/.bashrc` 來覆寫系統預設值。例如，若不要讓整個系統的程式訊息皆使用 `en_US` 時，請加入 `LC_MESSAGES=es_ES`，便會改用西班牙文來顯示訊息。

10.4.2 `~/.i18n` 中的地區設定

如果您對區域設定的系統預設值不滿意，可以根據 **Bash** 指令碼語法在 `~/.i18n` 中變更設定值。`~/.i18n` 中的項目會覆寫 `/etc/sysconfig/language` 中的系統預設值。使用相同變數名稱，但不用 `RC_` 名稱空間字首，例如，以 `LANG` 取代 `RC_LANG`：

```
LANG=cs_CZ.UTF-8
LC_COLLATE=C
```

10.4.3 語言支援的設定

按照規定，在訊息類別中的檔案僅會儲存於對應的語言目錄中 (像是 `en`)，以便有備用可用。如果您將 `LANG` 設為 `en_US`，而且 `/usr/share/locale/en_US/LC_MESSAGES` 中的訊息檔案不存在的話，則它會回到 `/usr/share/locale/en/LC_MESSAGES` 中。

您也可以定義備用鍊，例如，不列塔尼文之於法文，或是加里斯亞文之於西班牙文之於葡萄牙文：

```
LANGUAGE="br_FR:fr_FR"
```

```
LANGUAGE="gl_ES:es_ES:pt_PT"
```

如有需要，請改用挪威文變體 `Nynorsk` 與 `Bokmal` (讓其他備用為否)：

```
LANG="nn_NO"
```

```
LANGUAGE="nn_NO:nb_NO:no"
```

或

```
LANG="nb_NO"
```

```
LANGUAGE="nb_NO:nn_NO:no"
```

請注意，在挪威文中，會以不同方式處理 `LC_TIME`。

如果無法適當的辨識界定位數群組的分隔符號，可能會發生問題。如果 `LANG` 設定為類似 `de` 的兩個字母的語言碼，但卻使用 `/usr/share/lib/de_DE/LC_NUMERIC` 中的定義檔 `glibc`，就會發生這種情形。因此，`LC_NUMERIC` 必須設定為 `de_DE`，讓系統能辨識分隔符號定義。

10.4.4 如需更多資訊

- *The GNU C Library Reference Manual* 的「Locales and Internationalization」一章。包含在 `glibc-info` 中。
- Markus Kuhn 所寫的 *UTF-8 and Unicode FAQ for Unix/Linux*，目前網址如下：
<http://www.cl.cam.ac.uk/~mgk25/unicode.html>。
- *Unicode-Howto*，Bruno Haible 著：`/usr/share/doc/howto/en/html/Unicode-HOWTO.html`。

印表機操作

SUSE® Linux Enterprise Desktop 可支援以多種類型的印表機進行列印，包括遠端網路印表機。印表機可經由 YaST 或手動設定。如需組態設定指示，請參閱第 6.5 節「Setting Up a Printer」(第 6 章 *Setting Up Hardware Components with YaST*, ↑*部署指南*)。圖形和指令行公用程式都可用來啟動和管理列印工作。如果您的印表機無法如預期般運作，請參閱第 11.8 節「疑難排解」[115 頁]。

CUPS 是 SUSE Linux Enterprise Desktop 中的標準列印系統。CUPS 為高度使用者導向。在許多情況下，可與 LPRng 相容，或稍做努力便可與其搭配。SUSE Linux Enterprise Desktop 中包含 LPRng 完全是基於相容性考量。

印表機可藉由介面 (例如 USB 或網路) 和印表機語言加以區分。購買印表機時，請確定印表機的介面 (例如 USB 或並列埠) 是否也可在您的硬體上找到，並請確定印表機的語言符合所需。印表機可根據下列三種印表機語言來分類：

PostScript 印表機

Linux 和 Unix 內部列印系統以 PostScript 印表機語言產生和處理大部分列印工作。此語言已經很舊但很有效率。如果印表機可直接處理 PostScript 文件，且不需轉換到列印系統中其他階段，潛在錯誤來源的次數便會減少。因為 PostScript 受限於高額的授權成本，這些印表機通常較無 PostScript 解譯器的印表機昂貴。

標準印表機 (PCL 和 ESC/P 語言)

雖然這些印表機語言很舊，它們仍在擴充以處理印表機的新功能。如果已知印表機語言，列印系統可以藉由 Ghostscript 的協助，將 PostScript 工作轉換為對應的印表機語言。此處理階段稱為解譯。最知名的語言為 PCL，大多為 HP 印表機與其複製品所使用；另一個是 ESC/P，大多為 Epson 印表機所使用。Linux 通常支援這些印表機語言，並可產生不錯的列印效果。Linux

可能無法處理非常新且新潮的印表機的部分功能，因為開放原始碼開發人員仍在研究這些功能。除了 HP 所開發的 HPLIP 之外，目前沒有印表機製造商在開發 Linux 驅動程式並將開放原始碼授權提供給 Linux 散發者。這些印表機大部分屬於中等價位。

專屬印表機 (也稱為 GDI 印表機)

這些印表機並不支援任何一般的印表機語言。它們使用自己的印表機語言，而當有新型號發行，那些語言也可能有所變更。這些印表機通常指有 Windows 驅動程式。如需相關資訊，請參閱第 11.8.1 節「沒有標準印表機語言模式支援的印表機」[116頁]。

在購買新印表機之前，請參考下列來源以檢查您想要購買的印表機之支援性：

<http://www.linuxfoundation.org/en/OpenPrinting/>
OpenPrinting.org 印表機資料庫。

<http://www.cs.wisc.edu/~ghost/>
Ghostscript 網頁。

`/usr/share/doc/packages/ghostscript-library/catalog.devices`
所含驅動程式的清單。

線上資料庫會永遠顯示最新的 Linux 支援狀態。但是，Linux 版本僅可與生產期間可用的驅動程式整合。因此，目前被評選為「完全支援」的印表機，在最新的 SUSE Linux Enterprise Desktop 版本發行之後，可能就將失去此稱號。因此，資料庫不一定能指出正確狀態，而僅提供估計值。

11.1 列印系統的工作流程

使用者會建立列印工作。列印工作的組成元素為要列印的資料加上暫存序列器的資訊 (例如印表機的名稱或是印表機佇列的名稱)，以及非必要的過濾器資訊 (例如，印表機的特定選項)。

每一台印表機都至少有一個專屬的印表機佇列。暫存序列器會在佇列中列印工作，直到所需的印表機已準備好接收資料。當印表機備妥時，暫存序列器會透過過濾器與後端，傳送資料至印表機。

過濾器會將列印應用程式所產生的資料 (通常為 PostScript 或 PDF，但也會有 ASCII、JPEG 等) 轉換為印表機特定資料 (PostScript、PCL、ESC/P 等)。印表機

的特性描述在 PPD 檔案中。PPD 檔案含有印表機特定選項以及在印表機上啟用它們所需的參數。過濾器系統可確保啟用使用者所選取的選項。

如果您是使用 PostScript 印表機，過濾器系統會將資料轉換為印表機特定的 PostScript。這並不需要印表機驅動程式。如果您使用非 PostScript 印表機，過濾器系統會將資料轉換為印表機特定資料。這將需要印表機適用的印表機驅動程式。後端會從過濾器接收印表機特定的資料，然後將它傳送至印表機。

11.2 連接印表機的方法和通訊協定

有各種方法可將印表機連接到系統。CUPS 列印系統的組態無法辨識本地印表機和透過網路連接到系統的印表機。在 Linux 中，本地印表機必須依照印表機製造商的手冊所述方式連接。CUPS 支援序列埠、USB、並列埠和 SCSI 連線。

警告： 在執行中的系統變更纜線連接

在將印表機連接到機器時，請不要忘記只有 USB 設備可在操作中插上和拔除。若要避免損壞您的系統或印表機，請先關機再變更任何非 USB 的連接。

11.3 安裝軟體

PPD (PostScript 印表機描述) 為描述內容 (如解析度) 和選項 (如雙面列印模組的可用性) 的電腦語言。這些描述是使用 CUPS 中各種印表機選項所需。沒有 PPD 檔案，列印資料會被轉送給處於「raw」狀態的印表機，這通常不是想要的狀態。在安裝 SUSE Linux Enterprise Desktop 時，會預先安裝很多 PPD 檔案。

若要設定 PostScript 印表機，最好的方法是取得適當的 PPD 檔。在標準安裝範圍中自動安裝的套件 `manufacturer-PPDs` 提供許多 PPD 檔案。請參閱第 11.7.2 節「各種套件中的 PPD 檔案」[114頁]和第 11.8.2 節「PostScript 印表機沒有可用的 PPD 檔案」[116頁]。

新的 PPD 檔案可儲存在目錄 `/usr/share/cups/model/` 中，或以 YaST (請參閱 章節「Adding Drivers with YaST」(第 6 章 *Setting Up Hardware Components with YaST*, ↑ 部署指南)) 新增到列印系統。之後，便可在安裝期間選取 PPD 檔案。

請小心，印表機製造商是否除了修改組態檔之外，要您安裝整個軟體套裝。首先，這種安裝會導致遺失 SUSE Linux Enterprise Desktop 所提供的支援，其次印表機指令可能會以不同方式運作，使系統再也無法處理其他製造商的設備。基於此原因，不建議安裝製造商軟體。

11.4 網路印表機

網路印表機可支援各種通訊協定，有些甚至可同時支援。雖然大部分支援的通訊協定為標準的，部分製造商會因為測試系統無法正確地執行標準，或因為想要提供標準無法提供的特定功能，而擴充 (修改) 標準。然後製造商僅對少數作業系統提供驅動程式，以減少那些系統的困難。不幸地，他們很少提供 Linux 驅動程式。目前的情況是，您無法以每一個通訊協定均能在 Linux 中順暢執行的假設來行事。因此，您必須試驗各種選項以達到功能性組態。

CUPS 支援 socket、LPD、IPP 和 smb 通訊協定。

插槽

Socket 指不須先執行資料信號交換，而將資料傳送到網際網路插槽的連接。經常使用的插槽連接埠號碼為 9100 或 35。設備 URI (資源識別字串) 的語法為 `socket://IP.of.the.printer:port`，例如
`socket://192.168.2.202:9100/`。

LPD (行列式印表機精靈，Line Printer Daemon)

經過實驗的 LPD 通訊協定描述於 RFC 1179 中。在此通訊協定之下，部分工作相關資料 (如印表機佇列的 ID) 會在傳送實際列印資料之前傳送。因此，在設定資料傳輸的 LPD 通訊協定時，必須指定印表機佇列。不同印表機製造商的執行具有足夠彈性接受任何名稱做為印表機佇列。如有需要，印表機手冊應該會指出要使用的名稱。通常使用 LPT、LPT1、LP1 或相似名稱。LPD 佇列也可以在 CUPS 系統中不同 Linux 或 Unix 主機上設定。LPD 服務的連接埠號碼為 515。某個設備 URI 的範例為
`lpd://192.168.2.202/LPT1`。

IPP (網際網路列印通訊協定，Internet Printing Protocol)

IPP 是相對較新的 (1999) 通訊協定，以 HTTP 通訊協定為基礎。有了 IPP，可比使用其他通訊協定傳輸更多工作相關資料。CUPS 使用 IPP 進行內部資料傳輸。這是在兩個 CUPS 伺服器之間轉送佇列偏好的通訊協定。正確設定 IPP 必須要有列印佇列的名稱。IPP 的連接埠號碼為 631。設備 URI 的範例

為 `ipp://192.168.2.202/ps` 和
`ipp://192.168.2.202/printers/ps`。

SMB (Windows 共享)

CUPS 也支援在連接到 Windows 共享的印表機上列印。此用途使用的通訊協定為 **SMB**。SMB 使用連接埠號碼 137、138 和 139。設備 URI 的範例為 `smb://user:password@workgroup/smb.example.com/printer`、`smb://user:password@smb.example.com/printer` 和 `smb://smb.example.com/printer`。

必須在設定組態之前決定印表機支援的通訊協定。如果製造商未提供所需資訊，可使用 `nmap` 指令 (`nmap` 套件) 來猜測通訊協定。`nmap` 會檢查主機上開啟的通訊埠。例如：

```
nmap -p 35,137-139,515,631,9100-10000 printerIP
```

11.4.1 以指令行工具設定 CUPS

除了使用 YaST 設定 CUPS 選項之外，設定網路印表機時，CUPS 可以由 `lpadmin` 和 `lpoptions` 之類的指令行工具進行設定。您需要包含後端 (如並列和參數) 的設備 URI。若要判斷系統中的設備 URI 是否有效，請使用以下 `lpinfo -v | grep "://"` 指令：

```
# lpinfo -v | grep "://"
direct usb://ACME/FunPrinter%20XL
direct parallel:/dev/lp0
```

使用 `lpadmin`，CUPS 伺服器管理員可新增、移除或管理類別何列印佇列。若要新增印表機佇列，請使用下列語法：

```
lpadmin -p queue -v device-URI -P PPD-file -E
```

然後設備 (`-v`) 會變為可用 `queue` (`-P`)，使用指定的 PPD 檔案 (`-P`)。這表示如果要手動設定印表機，您必須知道 PPD 檔案以及設備 URI。

請勿使用 `-E` 做為第一選項。對於所有 CUPS 指令，第一個引數 `-E` 設定使用加密連接。若要啟用印表機，必須依照下列範例所示使用 `-E`：

```
lpadmin -p ps -v parallel:/dev/lp0 -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

下列範例是設定網路印表機：

```
lpadmin -p ps -v socket://192.168.2.202:9100/ -P \
/usr/share/cups/model/Postscript-level1.ppd.gz -E
```

如需 lpadmin 的更多選項，請參閱 lpadmin(1) 的線上文件。

在印表機設定期間，某些選項會設成預設。可針對每一個列印工作修改這些選項(視所使用的列印工具而定)。也可以使用 YaST 變更這些預設選項。使用指令行工具，可依下列方式設定預設選項：

1 首先，列出所有選項：

```
lpoptions -p queue -l
```

範例：

```
Resolution/Output Resolution: 150dpi *300dpi 600dpi
```

啟用的預設選項前面會加上星號(*)，用以識別。

2 以 lpadmin 變更選項：

```
lpadmin -p queue -o Resolution=600dpi
```

3 檢查新設定：

```
lpoptions -p queue -l
```

```
Resolution/Output Resolution: 150dpi 300dpi *600dpi
```

當一般使用者執行 lpoptions 時，設定會寫入 ~/.cups/lpoptions。然而，root 設定會寫至 /etc/cups/lpoptions。

11.5 圖形列印介面

Xpp 和 KDE 程式 KPrinter 等工具備有圖形介面，以便選擇佇列，並可設定 PPD 檔案中可用的 CUPS 標準選項和印表機專用的選項。您甚至可以使用 KPrinter 做為非 KDE 應用程式的標準列印介面。在這些應用程式的列印對話方塊中，請指定 kprinter 或 kprinter --stdin 做為列印指令。要使用的指令視應用程式的資料傳輸方式而有所不同——只需試試看哪一個指令有效即可。如果設定正確，應用程式會在發出列印工作時開啟 KPrinter 對話方塊，您就可以使用對話方塊來選取佇列，並設定其他列印選項。但前提是該應用程式自有的列印設定與 KPrinter 的列印設定不相衝突，而且列印選項啟用後只會經由 KPrinter 做

變更。如需有關 KPrinter 的詳細資訊，請參閱第 6 章 *Managing Print Jobs* (↑*KDE 使用者指南*)。

11.6 由指令行開始列印

若要由指令行進行列印，請輸入 `lp -d 佇列名稱 檔案名稱`；請以相對應的名稱來取代 *佇列名稱* 及 *檔案名稱*。

有些應用程式有賴 `lp` 指令來進行列印。在此情況下，請在應用程式列印對話方塊中 (通常並未指定 *檔案名稱*) 輸入正確的指令，例如 `lp -d 佇列名稱`。

11.7 SUSE Linux Enterprise Desktop 中的特殊功能

CUPS 的許多功能已經可適用於 SUSE Linux Enterprise Desktop。此處涵蓋部份最重要的變更。

11.7.1 CUPS 與防火牆

執行 SUSE Linux Enterprise Desktop 的預設安裝後，`SuSEfirewall2` 隨即會啟用，且網路介面會設定為處於「外部區域」中，這會阻擋內送流量。使用 CUPS 時，必須調整這些預設的設定。如需 `SuSEfirewall2` 組態設定的詳細資訊，請參閱第 9.4 節「`SuSEfirewall2`」(第 9 章 *Masquerading and Firewalls*, ↑*Security Guide*)。

CUPS 用戶端

CUPS 用戶端通常在位於防火牆之後的受信任網路環境中的一般工作站上執行。在此情況下，建議將網路介面設定為處於「內部區域」中，以便可從該網路中存取工作站。

CUPS 伺服器

如果 CUPS 伺服器位於受防火牆保護的受信任網路環境中，則應將網路介面設定為處於防火牆的「內部區域」中。建議您在受信任的網路環境中設定 CUPS 伺服器，否則請留心使用特殊防火牆規則與 CUPS 組態中的安全性設定加以保護。

11.7.2 各種套件中的 PPD 檔案

YaST 印表機組態僅使用安裝於系統上 `/usr/share/cups/model/` 中的 PPD 檔案來設定 CUPS 的佇列。為了尋找適合印表機型號的 PPD 檔案，YaST 會將硬體偵測期間決定的廠商和型號與系統上 `/usr/share/cups/model/` 中提供的所有 PPD 檔案內的廠商和型號相比較。基於此原因，YaST 印表機組態將從 PPD 檔案中取出的廠商和型號資訊產生資料庫。選取印表機後，請接收符合型號清單中的廠商與型號的 PPD 檔案。

僅使用 PPD 檔案且不使用其他資訊來源的組態，好處在於 `/usr/share/cups/model/` 中的 PPD 檔案可自由修改。YaST 印表機組態可辨識變更並重新產生廠商和型號資料庫。例如，如果您只有 PostScript 印表機，通常不需要 `cups-drivers` 套件中的 Foomatic PPD 檔案，或 `gutenprint` 套件中的 Gutenprint PPD 檔案。您可以直接將 PostScript 印表機的 PPD 檔案複製到 `/usr/share/cups/model/` (如果在 `manufacturer-PPDs` 套件中尚未存在)，以達到印表機的最佳組態。

cups 套件中的 CUPS PPD 檔案

`cups` 套件中的一般 PPD 檔案已經以 PostScript Level 1 和 Level 2 印表機適當的 Foomatic PPD 檔案補充。

- `/usr/share/cups/model/Postscript-level1.ppd.gz`
- `/usr/share/cups/model/Postscript-level2.ppd.gz`

cups-drivers 套件中的 PPD 檔案

一般情況下，Foomatic 印表機過濾器 `foomatic-rip` 會與非 PostScript 印表機的 Ghostscript 搭配使用。適當的 Foomatic PPD 檔案有 `*NickName: ...`

Foomatic/Ghostscript driver 和 `*cupsFilter: ... foomatic-rip` 的項目。這些 PPD 檔案位於 cups-drivers 套件中。

YaST 通常偏好 manufacturer-PPD 檔案。但是如果沒有適合的 manufacturer-PPD 檔案，將會選取包含項目 `*NickName: ... Foomatic ...` (推薦) 的 Foomatic PPD 檔案。

gutenprint 套件中的 Gutenprint PPD 檔案

除了 foomatic-rip 之外，來自 Gutenprint (以前稱為 GIMP-Print) 的 CUPS 過濾器 `rastertogutenprint` 也可以用於許多非 PostScript 印表機。此過濾器 and 適合的 Gutenprint PPD 檔案可在 gutenprint 套件中找到。Gutenprint PPD 檔案位於 `/usr/share/cups/model/gutenprint/` 中，並含有項目 `*NickName: ... CUPS+Gutenprint` 和 `*cupsFilter: ... rastertogutenprint`。

位於 manufacturer-PPDs 套件中印表機製造廠商的 PPD 檔案

manufacturer-PPDs 套件包含具有充分自由授權的印表機製造廠商所發行的 PPD 檔案。PostScript 印表機應該以印表機製造廠商的適合 PPD 檔案來設定，因為有此檔案才可使用 PostScript 印表機的所有功能。YaST 偏好來自 manufacturer-PPDs 的 PPD 檔案。如果型號名稱不符合，YaST 就無法使用來自 manufacturer-PPDs 套件的任何 PPD 檔案。如果對於類似型號 (如 Funprinter 12xx 系列)，manufacturer-PPDs 套件僅包含一個 PPD 檔案，則可能出現此情況。在此情況下，請手動分別選取 YaST 中的 PPD 檔案。

11.8 疑難排解

下列章節涵蓋印表機硬體和軟體最常遭遇的問題，以及解決或避免這些問題的方式。涵蓋的主題包含 GDI 印表機、PPD 檔案和連接埠組態，並討論了一般網路印表機問題、列印瑕疵、佇列處理。

11.8.1 沒有標準印表機語言模式支援的印表機

這些印表機不支援任何的一般印表機語言，且只有特殊的專屬控制序列才能處理。因此它們僅可在製造廠商針對其開發驅動程式的作業系統版本上使用。GDI 是 Microsoft* 為繪圖設備所開發的程式設計介面。製造廠商通常只提供 Windows 適用的驅動程式，而因為 Windows 驅動程式使用 GDI 介面，所以這些印表機也稱為 *GDI 印表機*。實際問題不在於程式設計介面，而是這些印表機僅可使用對應印表機型號的專用印表機語言處理。

部分 GDI 印表機可切換到 GDI 模式或標準印表機語言來操作。若有可能，請參閱印表機的手冊。某些型號需要特殊的 Windows 軟體來進行切換 (請注意，從 Windows 列印時，Windows 印表機驅動程式可能會一直將印表機切回 GDI 模式)。對於其他 GDI 印表機，則可以使用標準印表機語言的延伸模組。

部分製造廠商提供其印表機的專用驅動程式。專用印表機驅動程式的壞處在於不保證可與安裝的列印系統搭配使用，且不保證適用於各種硬體平台。相反的，支援標準印表機語言的印表機不需依賴特殊的列印系統版本或特殊硬體平台。

除了不需花費時間嘗試使專用 Linux 驅動程式運作，也不需花費更多成本購買支援的印表機。如此可一次解決所有驅動程式問題、減少安裝與設定特殊驅動程式軟體以及取得列印系統中新開發所需的驅動程式更新的需要。

11.8.2 PostScript 印表機沒有可用的 PPD 檔案

如果 manufacturer-PPDs 套件不包含任何適用於 PostScript 印表機的 PPD 檔案，應該可以使用印表機製造廠商驅動程式光碟中的 PPD 檔案，或從印表機製造廠商的網頁下載適合的 PPD 檔案。

如果 PPD 檔案以壓縮保存檔 (.zip) 或自解壓縮保存檔 (.exe) 形式提供，請以 unzip 解壓縮。首先，檢閱 PPD 檔案的授權條款。然後，請使用 cupstestppd 公用程式來檢查 PPD 檔案是否符合「Adobe PostScript Printer Description File Format Specification, version 4.3」(Adobe PostScript 印表機說明檔案格式規格，版本 4.3)。如果公用程式傳回「FAIL」，就表示 PPD 檔案非常嚴重，可能造成重大問題。應該要減少 cupstestppd 所報告的問題點。若有需要，請詢問印表機製造廠商以取得適合的 PPD 檔案。

11.8.3 並列埠

最安全的方法是將印表機直接連接到第一並列埠，並在 BIOS 中選取下列並列埠設定值：

- I/O 位址：378 (十六進位)
- 中斷：無關
- 模式：Normal、Spp 或 Output Only
- DMA：停用

如果沒有這些設定值，印表機無法在並列埠上定址，請依照 BIOS 中的設定值，以 0x378 的格式在 `/etc/modprobe.conf` 中明確輸入 I/O 位址。如果有兩個並列埠，I/O 位址分別設為 378 和 278 (十六進位)，請以 0x378,0x278 格式輸入。

如果沒有使用中斷 7，可以使用 **範例 11.1** 「`/etc/modprobe.conf`：第一並列埠的中斷模式」[117頁]中所示的項目啟用。在啟用中斷模式之前，請檢查檔案 `/proc/interrupts` 以瞭解哪些中斷已經在使用中。只會顯示目前正在使用中的中斷。這可能因為作用中的硬體元件而有變化。其他任何設備都不能使用並列埠的中斷。如果您不確定，請以 `irq=none` 使用輪詢模式。

範例 11.1 `/etc/modprobe.conf`：第一並列埠的中斷模式

```
alias parport_lowlevel parport_pc
options parport_pc io=0x378 irq=7
```

11.8.4 網路印表機連接方式

查明網路問題

將印表機直接連接到電腦。基於測試因素，請將印表機設為本地印表機。如果可以，問題便與網路相關。

檢查 TCP/IP 網路

TCP/IP 網路和名稱解析必須可作用。

檢查遠端 lpd

使用以下指令來測試是否可在主機上建立到 lpd (連接埠 515) 的 TCP 連接：

```
netcat -z host 515 && echo ok || echo failed
```

如果無法建立到 lpd 的連接，可能是 lpd 不在作用中，或是有基本網路問題。

以使用者 root 的身份，使用以下指令來查詢 (可能很長) 遠端主機上佇列的狀態報告，假使對應的 lpd 在作用中且主機接受查詢：

```
echo -e "\004queue" \  
| netcat -w 2 -p 722 host 515
```

如果 lpd 沒有回應，它可能不在作用中，或是有基本網路問題。如果 lpd 有回應，回應應該會顯示主機上的佇列為何無法列印。如果您收到像 **範例 11.2「來自 lpd 的錯誤訊息」** [118頁] 中的回應，問題可能是因為遠端 lpd 所造成。

範例 11.2 來自 lpd 的錯誤訊息

```
lpd: your host does not have line printer access  
lpd: queue does not exist  
printer: spooling disabled  
printer: printing disabled
```

檢查遠端 cupsd

根據預設，CUPS 網路伺服器應該每隔三十秒在 UDP 連接埠 631 上廣播其佇列。同時，可使用以下指令來測試網路中是否有 CUPS 網路伺服器。請確定在執行指令之前停止本地 CUPS 精靈。

```
netcat -u -l -p 631 & PID=$! ; sleep 40 ; kill $PID
```

如果廣播 CUPS 網路伺服器存在，輸出將如 **範例 11.3「來自 CUPS 網路伺服器的廣播」** [118頁] 中所示。

範例 11.3 來自 CUPS 網路伺服器的廣播

```
ipp://192.168.2.202:631/printers/queue
```

可使用以下指令來測試是否可建立到主機上 cupsd (連接埠 631) 的 TCP 連接：

```
netcat -z host 631 && echo ok || echo failed
```

如果無法建立與 `cupsd` 的連接，則表明 `cupsd` 可能不在作用中或可能有基本的網路問題。假設對應的 `cupsd` 在作用中，而且主機可以接受查詢，則 `lpstat -h host -l -t` 會傳回 `host` 上所有佇列的狀態報告 (可能很大)。

此指令可用來測試主機上的佇列是否可接受由單一換行字元組成的列印工作。應該不會印出任何資料。可能會退出一張空白頁。

```
echo -en "\r" \  
| lp -d queue -h host
```

網路列印或列印伺服器盒疑難排解

在列印伺服器盒中執行的暫存序列器在執行大量列印工作時，有時會造成問題。因為這是列印伺服器盒中的暫存序列器所造成，您無法解決此問題。處理方式是，透過 **TCP** 插槽將印表機直接連接到列印伺服器盒，以規避列印伺服器盒中的暫存序列器。請參閱第 11.4 節「網路印表機」[110頁]。

利用此方法，可減少列印伺服器盒在不同資料格式之間的轉換問題 (TCP/IP 網路和本地印表機連接)。若要使用此方法，您必須知道列印伺服器盒上的 **TCP** 連接埠。如果印表機連接到列印伺服器盒且電源開啟，此 **TCP** 連接埠通常可在列印伺服器盒電源開啟一段時間之後，以 `nmap` 套件的 `nmap` 公用程式決定。例如，`nmap IP-address` 會傳送列印伺服器盒的以下輸出：

Port	State	Service
23/tcp	open	telnet
80/tcp	open	http
515/tcp	open	printer
631/tcp	open	cups
9100/tcp	open	jetdirect

此輸出表示連接到列印伺服器盒的印表機可透過連接埠 9100 上的 **TCP** 插槽定址。根據預設，`nmap` 僅會檢查幾個 `/usr/share/nmap/nmap-services` 中所列出一般熟知的連接埠。若要檢查所有可能的連接埠，請使用指令 `nmap -p from_port-to_port IP-address`。這可能會花費一些時間。如需詳細資訊，請參閱 `nmap` 的線上文件。

輸入以下指令

```
echo -en "\rHello\r\f" | netcat -w 1 IP-address port  
cat file | netcat -w 1 IP-address port
```

將字元字串或檔案直接傳送到對應連接埠以測試印表機是否可在此連接埠上定址。

11.8.5 列印成品損毀而無錯誤訊息

對列印系統而言，在 CUPS 後端完成資料至接收者 (印表機) 的資料傳輸時，列印工作便完成。如果接收者的進一步處理失敗 (例如，如果印表機無法列印印表機特定資料)，列印系統並不知道。如果印表機無法列印印表機特定資料，請選取更適用於印表機的不同 PPD 檔案。

11.8.6 停用佇列

如果到接收者的資料傳輸在數次嘗試之後完全失敗，CUPS 後端 (如 USB 或 socket) 會向列印系統報告錯誤 (向 cupsd)。後端會決定在報告資料傳輸失敗之前，是否要繼續嘗試以及要嘗試幾次是合理的。因為進一步的嘗試可能徒勞無功，cupsd 會停用對應佇列的列印。排除問題的起因之後，系統管理員必須以指令 `cupsenable` 重新啟動列印。

11.8.7 CUPS 瀏覽：刪除列印工作

如果 CUPS 網路伺服器透過瀏覽向用戶端主機廣播它的佇列，而且在用戶端主機上有適合的本地 cupsd 在作用中，用戶端 cupsd 會從應用程式接收列印工作，並將它們轉送給伺服器上的 cupsd。當 cupsd 接收列印工作時，會被指定新的工作號碼。因此，用戶端主機上的工作號碼和伺服器上的工作號碼不同。因為列印工作通常會立刻轉送，所以無法以用戶端主機上的工作號碼來刪除，因為用戶端 cupsd 一旦將列印工作轉送給伺服器 cupsd，便認為列印工作已完成。

若要刪除伺服器上的列印工作，假使伺服器尚未完成列印工作 (即尚未將工作完全傳送至印表機)，請使用 `lpstat -h cups.example.com -o` 這類指令來決定伺服器上的工作編號。使用此工作號碼，伺服器上的列印工作便可刪除：

```
cancel -h cups.example.com queue-jobnumber
```

11.8.8 損毀的列印工作與資料傳輸錯誤

如果您在列印程序中將印表機電源關閉再打開，或是關機再重新啟動電腦，列印工作仍然在佇列中，而且會繼續列印。必須以 `cancel` 將損毀的列印工作從佇列中移除。

如果列印工作損毀，或是主機和印表機之間的通訊發生問題，印表機會印出數頁含有不明字元的紙張，因為它無法正確地處理資料。若要處理這個問題，請遵循下列步驟：

- 1 若要停止列印，請從噴墨印表機取出所有紙張，或是打開雷射印表機的紙匣。高品質的印表機會有按鈕可取消目前的列印成品。
- 2 列印工作可能仍在佇列中，因為只有將工作完全傳送到印表機之後，才會移除。使用 `lpstat -o` 或 `lpstat -h cups.example.com -o` 檢查目前列印中的佇列。使用 `cancel queue-jobnumber` 或 `cancel -h cups.example.com queue-jobnumber` 刪除列印工作。
- 3 即使列印工作已從佇列刪除，部份資料可能仍會傳送到印表機。請檢查對應佇列的 CUPS 後端程序是否仍在執行中，並將它終止。例如，對於连接到並列埠的印表機，可使用指令 `fuser -k /dev/lp0` 來終止所有仍在存取印表機的程序 (更精確的說，就是並列埠)。
- 4 將印表機關閉一段時間以完全重設印表機。然後裝入紙張並開啟印表機電源。

11.8.9 除錯 CUPS 列印系統

使用以下標準程序以找出 CUPS 列印系統中的問題：

- 1 設定 `/etc/cups/cupsd.conf` 中的 `LogLevel debug`。
- 2 停止 `cupsd`。
- 3 移除 `/var/log/cups/error_log*` 以避免必須搜尋很大的記錄檔。
- 4 啟動 `cupsd`。
- 5 重覆造成問題的動作。
- 6 檢查 `/var/log/cups/error_log*` 中的訊息以辨識問題的起因。

11.8.10 如需更多資訊

許多特定問題的解決方法在「Novell 知識庫」(<http://support.novell.com/>) 中都有說明。請搜尋文字 CUPS 找到相關文章。

使用 udev 進行動態核心設備管理

12

核心可以新增或移除執行中系統中幾乎所有的設備。設備狀態(已插入或移除設備)中的變更必須傳播至使用者空間。在插入和探查設備時，必須立刻設定設備。特定設備的使用者必須收到任何狀態變更的通知。udev 會提供必要的基礎結構以便動態維護設備 `/dev` 目錄中的節點檔案和符號連結。udev 規則會提供將外部工具插入核心設備事件處理的方法。這項工具可用來自定 udev 設備處理，例如，新增要執行的特定程序檔來作為核心設備處理的一部份，或是在設備處理過程中要求並輸入其他資料來進行分析。

12.1 `/dev` 目錄

`/dev` 中的設備節點可用來存取對應的核心設備。透過 udev，`/dev` 目錄會反映核心的目前狀態。每個核心設備都有一個對應的設備檔案。如果設備與系統的連接中斷，該設備節點就會遭到移除。

`/dev` 目錄內容會保存在暫存檔系統中，而且所有檔案都會在每次系統開機時重新建立。特別經過手動建立或變更的檔案在重新開機後都不會存在。無論可存放於 `/lib/udev/devices` 目錄的對應核心設備狀態為何，靜態檔案和目錄都必須存在於 `/dev` 目錄中。在系統啟動時，該目錄內容將複製到 `/dev` 目錄，並具備與 `/lib/udev/devices` 中檔案相同的擁有權和許可權。

12.2 核心 uevent 和 udev

sysfs 檔案系統會輸出必要的設備資訊。每個核心已偵測和啟始化的設備，都會建立包含其設備名稱的目錄。其中會包含設備特定的屬性內容。

每次新增或移除設備時，核心都會傳送 uevent 以通知 udev 發生變更。udev 精靈會在啟動時讀取和分析一次 `/etc/udev/rules.d/*.rules` 檔案的所有指定規則，並將其保存在記憶體中。如果變更、新增或移除了規則檔案，精靈可以使用指令 `udevadm control reload_rules` 重新載入所有規則的記憶體內部表示。此動作也可以在執行 `/etc/init.d/boot.udev reload` 時完成。如需有關 udev 規則及其語法的詳細資訊，請參閱第 12.6 節「透過 udev 規則影響核心設備事件處理」[126頁]。

每個收到的事件都將與提供的規則集合進行比對。這些規則可新增或變更事件環境識別碼、要求要建立設備節點的特定名稱、新增指向該節點的符號連結，或是新增要在設備節點建立後執行的程式。驅動程式核心 uevents 會從核心網路連結插槽接收。

12.3 驅動程式、核心模組和設備

核心匯流排驅動程式會查探設備。核心會為每個偵測到的設備建立內部設備結構，而驅動程式核心會向 udev 精靈傳送 uevent。匯流排設備會以特殊格式的 ID 識別本身，表明其為何種設備。通常這些 ID 會包含廠商和產品 ID，以及其他子系統特定值。每個匯流排都會指定自己的 ID 配置，即所謂的 MODALIAS。核心會接收這些設備資訊、組織 MODALIAS ID 字串，並隨事件傳送該字串。例如，USB 滑鼠的 ID 字串將如下所示：

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

每個設備驅動程式都包含有設備可處理的已知別名清單。清單會包含在核心模組檔案本身。depmod 程式會讀取 ID 清單，並且為目前所有可用模組在核心的 `/lib/modules` 目錄中建立 `modules.alias` 檔案。透過此基礎結構，模組載入方式就會像在每次出現帶有 MODALIAS 識別碼的事件時呼叫 `modprobe` 一樣容易。如果是呼叫 `modprobe $MODALIAS`，此次呼叫就會比對設備的已組織設備別名和模組指定別名。如果有找到符合項目，該模組就可載入。這些動作都是由 udev 觸發，而且是自動發生。

12.4 開機和初始設備設定

在執行 `udev` 精靈之前於開機程序期間發生的設備事件都會遺失，這是因為處理這些事件的基礎結構是執行於根檔案系統中，而在該階段進行時無法使用。為彌補這一損失，核心在 `sysfs` 檔案系統中之每部設備的設備目錄中都提供了一份 `uevent` 檔案。使用 `add` 寫入該檔案，核心便可重新傳送與開機期間所遺失的相同事件。負責 `/sys` 中所有 `uevent` 檔案的簡易迴圈，可以再次觸發所有事件，建立設備節點並執行設備設定。

例如，開機期間出現的 USB 滑鼠可能無法由早期的開機邏輯啟始化，這是因為當時尚無法使用驅動程式。設備探查事件遺失，而且無法找到設備的核心模組。`udev` 只需在可以使用根目錄檔案系統之後要求核心提供所有設備事件，這樣 USB 滑鼠設備的事件就可再次執行，因此並不需要手動搜尋可能連結的設備。現在，它會在已裝載根目錄檔案系統中找到核心模組，並讓 USB 滑鼠完成啟始化。

從使用者空間的角度，執行期間的設備冷插拔 (`ColdPlug`) 順序和設備探查並沒有明顯的不同。這兩種情況都會使用相同規則來進行比對，而且會執行相同的設定程式。

12.5 監控執行中的 `udev` 精靈

`udevadm monitor` 程式可用來視覺化驅動程式核心事件以及 `udev` 事件程序的時間。

```
UEVENT[1185238505.276660] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UDEV   [1185238505.279198] add    /devices/pci0000:00/0000:00:1d.2/usb3/3-1
(usb)
UEVENT[1185238505.279527] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UDEV   [1185238505.285573] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UDEV   [1185238505.305026] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10 (input)
UEVENT[1185238505.305442] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
UEVENT[1185238505.306440] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV   [1185238505.325384] add
```

```
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/event4 (input)
UDEV [1185238505.342257] add
/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10/mouse2 (input)
```

UEVENT 行會顯示核心已透過網路連結傳送的事件。UDEV 行會顯示已完成的 udev 事件處理常式。列印時間是百萬分之一秒。介於 UEVENT 和 UDEV 之間的時間是指 udev 處理此事件所耗費的時間，或者是 udev 精靈延遲執行以便此事件能與執行中相關事件同步的時間。例如，硬碟分割區的事件永遠會等待主要磁碟設備事件完成，因為分割區事件可能與主要磁碟事件向硬碟查詢的資料有關。

udevadm monitor --env 會顯示完整的事件環境：

```
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw
```

udev 也會將訊息傳送到 syslog。控制哪個訊息要傳送到 syslog 的預設 syslog 優先順序，會在 udev 組態檔 /etc/udev/udev.conf 中指定。執行精靈的記錄優先程度可使用 udevadm control log_priority=level/number 來變更。

12.6 透過 udev 規則影響核心設備事件處理

udev 規則可以比對核心新增至事件本身的任何內容，或者任何由核心輸出到 sysfs 的資訊。規則也可向外部程式要求其他資訊。每個事件都會與所有指定規則進行比對。所有規則都位於 /etc/udev/rules.d 目錄中。

規則檔案中的每一行都包含至少一個金鑰值組合。金鑰類型共有兩種，包括比對和指定金鑰。當所有比對金鑰都與指定值相符時就會套用規則，而該指定值就會指定給指定金鑰。相符的規則可以指定設備節點名稱、新增指向該節點的符號連結，或是執行指定程式成為事件處理的一部份。如果找不到任何符合規

則，就會使用預設的設備節點名稱來建立設備節點。如需有關規則語法和為比對或輸入資料所提供之金鑰的詳細資訊，請參閱 `udev` 線上文件。以下範例規則提供了對 `udev` 規則語法的基本介紹。所有範例規則均來自 `/etc/udev/rules.d/50-udev-default.rules` 之下的 `udev` 預設規則集。

範例 12.1 範例 `udev` 規則

```
# console
KERNEL=="console", MODE="0600", OPTIONS="last_rule"

# serial devices
KERNEL=="ttyUSB*", ATTRS{product}=="[Pp]alm*Handheld*", SYMLINK+="pilot"

# printer
SUBSYSTEM=="usb", KERNEL=="lp*", NAME="usb%k", SYMLINK+="usb%k", GROUP="lp"

# kernel firmware loader
SUBSYSTEM=="firmware", ACTION=="add", RUN+="firmware.sh"
```

主控台規則包含三個金鑰：一個比對金鑰 (`KERNEL`) 與兩個指派金鑰 (`MODE`、`OPTIONS`)。 `KERNEL` 比對規則用於搜尋類型為主控台的所有項目的設備清單。只有完全符合的項目才有效，才會觸發此規則讓其執行。 `MODE` 金鑰用於將特殊權限指派給設備節點，在此案例中，僅此設備的擁有者才會被指派讀取與寫入權限。 `OPTIONS` 金鑰用於將此規則做為要套用至所有此類型設備的最後一條規則。符合此特殊設備類型的任何後續規則都不會生效。

序列設備規則在 `50-udev-default.rules` 中雖已不再可用，但仍值得瞭解一下。它包含兩個比對金鑰 (`KERNEL` 和 `ATTRS`) 與一個指派金鑰 (`SYMLINK`)。 `KERNEL` 金鑰用於搜尋類型為 `ttyUSB` 的所有設備。使用 `*` 萬用字元時，此金鑰可比對多部此類設備。第二個比對金鑰 `ATTRS` 用於檢查 `sysfs` 中針對 `ttyUSB` 設備的 `product` 屬性檔案是否包含特定字串。指派金鑰 (`SYMLINK`) 用於觸發將符號連結新增至 `/dev/pilot` 下的此設備。此金鑰中使用的運算子 (`+=`) 將告知 `udev` 額外執行此動作，即使先前或之後的規則會新增其他符號連結也是如此。由於此規則包含兩個比對金鑰，因此僅當兩個條件均滿足時才適用。

印表機規則可處理 `USB` 印表機，它包含兩個比對金鑰，必須同時套用這兩個金鑰才能套用整個規則 (`SUBSYSTEM` 與 `KERNEL`)。三個指派金鑰用於命名此設備類型 (`NAME`)、建立符號設備連結 (`SYMLINK`) 以及分組此設備類型的成員 (`GROUP`)。在 `KERNEL` 金鑰中使用 `*` 萬用字元可使其符合多部 `lp` 印表機設備。可以在 `NAME` 與 `SYMLINK` 金鑰中使用替代項，透過內部設備名稱延伸這些字串。例如，第一部 `lp USB` 印表機的符號連結會讀取 `/dev/usb/lp0`。

核心韌體載入程式規則可讓 **udev** 在執行時期透過外部輔助程式程序檔載入其他韌體。**SUBSYSTEM** 比對金鑰可搜尋韌體子系統。**ACTION** 金鑰可檢查是否已新增任何屬於韌體子系統的設備。**RUN+=** 金鑰可觸發執行 `firmware.sh` 程序檔以查找要載入的韌體。

某些一般特性適用於所有規則：

- 每條規則都包含一個或多個以逗號分隔的金鑰值對。
- 金鑰的操作由運算子決定。**udev** 規則支援多種不同的運算子。
- 每個指定值必須括在引號中。
- 規則檔案中的每一行都表示一條規則。如果某規則的長度超出一行，請將 \ 加入到不同行中，就如同在外圍程序語法中一樣。
- **udev** 規則支援符合 `*`、`?` 與 `[]` 模式的外圍程序式模式。
- **udev** 規則支援替代項。

12.6.1 在 **udev** 規則中使用運算子

建立金鑰時，您可以根據要建立的金鑰類型從多個不同的運算子中進行選擇。比對金鑰通常僅用於尋找符合或明顯不符合搜尋值的值。比對金鑰可包含以下運算子：

==

比較是否相等。如果金鑰包含搜尋模式，則所有符合此模式的結果均有效。

!=

比較是否不相等。如果金鑰包含搜尋模式，則所有符合此模式的結果均有效。

指派金鑰可使用以下運算子：

=

將某個值指派給金鑰。如果金鑰先前包含值清單，則此金鑰將重設並僅指派單一值。

`+=`

將某個值新增至包含項目清單的金鑰。

`:=`

指派最終值。不允許後續規則再做任何變更。

12.6.2 在 udev 規則中使用替代項

udev 規則支援使用佔位符與替代項。使用方式與任何其他程序檔中的方式類似。

udev 規則可使用以下替代項：

`%r, $root`

依預設為設備目錄 `/dev`。

`%p, $devpath`

DEVPATH 的值。

`%k, $kernel`

KERNEL 的值或內部設備名稱。

`%n, $number`

設備編號。

`%N, $temnode`

設備檔案的暫存名稱。

`%M, $major`

設備的主要編號。

`%m, $minor`

設備的次要編號。

`%s{attribute}, $attr{屬性}`

sysfs 屬性的值 (由屬性指定)。

`%E{變數}, $attr{變數}`

環境變數的值 (由變數指定)。

`%c, $result`
PROGRAM 的輸出。

`%%`
% 字元。

`$$`
\$ 字元。

12.6.3 使用 udev 比對金鑰

比對金鑰描述在可以套用 udev 規則之前必須滿足的條件。以下為可用的比對金鑰：

ACTION
事件動作的名稱，例如新增或移除設備時的 `add` 或 `remove`。

DEVPATH
事件設備的設備路徑，例如 `DEVPATH=/bus/pci/drivers/ipw3945`，用於搜尋與 `ipw3945` 驅動程式相關的所有事件。

KERNEL
事件設備的內部 (核心) 名稱。

SUBSYSTEM
事件設備的子系統，例如 `SUBSYSTEM=usb`，適用於與 USB 設備相關的所有事件。

ATTR{ 檔案名稱 }
事件設備的 `sysfs` 屬性。例如，若要比對 `vendor` 屬性檔案名稱中包含的字串，可以使用 `ATTR{vendor}=="On[sS]tream"`。

KERNELS
可讓 udev 向上搜尋符合設備名稱的設備路徑。

SUBSYSTEMS
可讓 udev 向上搜尋符合設備子系統名稱的設備路徑。

DRIVERS

可讓 **udev** 向上搜尋符合設備驅動程式名稱的設備路徑。

ATTR{ 檔案名稱 }

可讓 **udev** 向上搜尋符合 **sysfs** 屬性值的設備路徑。

ENV{ 金鑰 }

環境變數的值，例如 ENV{ ID_BUS }="ieee1394，用於搜尋與 FireWire 匯流排 ID 相關的所有事件。

PROGRAM

可讓 **udev** 執行外部程式。若要成功執行，程式必須以離開碼零返回。RESULT 金鑰可使用程式的輸出 (列印至 **stdout**)。

RESULT

比對上次 PROGRAM 呼叫的輸出字串。即可將此金鑰包含於相同規則中 (如 PROGRAM 金鑰)，也可含於後續規則中。

12.6.4 使用 **udev** 指派金鑰

與上述比對金鑰不同，指派金鑰不會說明必須滿足的條件，而是將值、名稱和動作指派給透過 **udev** 維護的設備節點。

NAME

要建立之設備節點的名稱。如果規則已設定了節點名稱，則將忽略適用於此節點的所有其他含 NAME 金鑰的規則。

SYMLINK

與要建立的節點相關聯之符號連結的名稱。可以為多項比對規則新增符號連結以便使用設備節點進行建立。您還可以使用空格字元分隔符號連結名稱，從而為一項規則中的一個節點指定多個符號連結。

OWNER, GROUP, MODE

新設備節點的權限。在此處指定的值將覆寫已編譯的任何項目。

ATTR{ 金鑰 }

指定要寫入事件設備之 **sysfs** 屬性中的值。如果使用運算子 ==，也會使用此金鑰來比對 **sysfs** 屬性的值。

ENV{ 金鑰 }

告知 **udev** 將某變數輸出到環境。如果使用運算子 `==`，也會使用此金鑰來比對環境變數。

RUN

告知 **udev** 將某程式新增至要為此設備執行的程式清單。對極短任務套用此項時要格外小心，避免封鎖此設備的其他事件。

LABEL

在 `GOTO` 可以跳轉之處新增一個標籤。

GOTO

通知 **udev** 跳過多項規則並繼續使用含有由 `GOTO` 金鑰所參考之標籤的規則。

IMPORT{ 類型 }

將變數載入事件環境，如外部程式的輸出。**udev** 可輸入多種不同類型的變數。如果未指定任何類型，**udev** 會根據檔案權限的可執行位元嘗試自行決定類型。

- `program` 可告知 **udev** 執行外部程式並輸入其輸出。
- `file` 可告知 **udev** 輸入文字檔。
- `parent` 可告知 **udev** 輸入父代設備中儲存的金鑰。

WAIT_FOR_SYSFS

告知 **udev** 等待要為特定設備建立的指定 `sysfs` 檔案，例如，

`WAIT_FOR_SYSFS="ioerr_cnt"` 將通知 **udev** 在 `ioerr_cnt` 檔案建立完成前需一直等待。

OPTIONS

`OPTION` 金鑰可以包含多個值：

- `last_rule` 告知 **udev** 忽略所有後續規則。
- `ignore_device` 告知 **udev** 完全忽略此事件。
- `ignore_remove` 告知 **udev** 忽略針對該設備的所有後續移除事件。

- `all_partitions` 告知 `udev` 為區塊設備上的所有可用分割區建立設備節點。

12.7 永久設備命名

動態設備目錄和 `udev` 規則基礎結構可以為所有磁碟設備提供固定名稱，無論其辨識順序或設定使用的連接為何。核心所建立的每個適當區塊設備，都會採用針對特定匯流排、磁碟類型或檔案系統所設計的工具進行檢查。`udev` 會根據動態核心指定設備節點名稱，維護指向該設備的永久符號連結類別：

```
/dev/disk
|-- by-id
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B -> ../../sda
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part1 -> ../../sda1
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part6 -> ../../sda6
|   |-- scsi-SATA_HTS726060M9AT00_MRH453M4HWHG7B-part7 -> ../../sda7
|   |-- usb-Generic_STORAGE_DEVICE_02773 -> ../../sdd
|   `-- usb-Generic_STORAGE_DEVICE_02773-part1 -> ../../sdd1
|-- by-label
|   |-- Photos -> ../../sdd1
|   |-- SUSE10 -> ../../sda7
|   `-- devel -> ../../sda6
|-- by-path
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0 -> ../../sda
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part1 -> ../../sda1
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part6 -> ../../sda6
|   |-- pci-0000:00:1f.2-scsi-0:0:0:0-part7 -> ../../sda7
|   |-- pci-0000:00:1f.2-scsi-1:0:0:0 -> ../../sr0
|   |-- usb-02773:0:0:2 -> ../../sdd
|   |-- usb-02773:0:0:2-part1 -> ../../sdd1
`-- by-uuid
    |-- 159a47a4-e6e6-40be-a757-a629991479ae -> ../../sda7
    |-- 3e999973-00c9-4917-9442-b7633bd95b9e -> ../../sda6
    `-- 4210-8F8C -> ../../sdd1
```

12.8 udev 使用的檔案

`/sys/*`

由 Linux 核心提供的虛擬檔案系統，可輸出所有目前已知設備。`udev` 用此資訊在 `/dev` 中建立設備節點。

`/dev/*`

動態建立的設備節點和開機時從 `/lib/udev/devices/*` 複製的靜態內容。

下列檔案和目錄包含了 `udev` 基礎結構的重要元件：

`/etc/udev/udev.conf`

主要 `udev` 組態檔。

`/etc/udev/rules.d/*`

`udev` 事件符合規則。

`/lib/udev/devices/*`

靜態 `/dev` 內容。

`/lib/udev/*`

從 `udev` 規則中呼叫的協助程式。

12.9 如需更多資訊

如需關於 `udev` 基礎結構的詳細資訊，請參閱下列 `man` 頁面：

`udev`

關於 `udev`、金鑰、規則和其他重要組態問題的一般資訊。

`udevadm`

`udevadm` 可用於控制 `udev` 的執行時期行為、請求核心事件、管理事件佇列以及提供簡單的除錯機制。

`udev`

關於 `udev` 事件管理精靈的資訊。

X Window System

X Window System (X11) 是 UNIX 中既成現實標準的圖形化使用者介面。X 採網路結構，可讓應用程式在一個主機上啟動而在透過任何種類的網路 (LAN 或國際網路) 連接的其他主機上顯示。本章說明 X Window System 環境的安裝與最佳化，並提供有關在 SUSE® Linux Enterprise Desktop 中使用字型的背景資訊。

13.1 手動設定 X Window System

依照預設，X Window System 是以 SaX2 介面設定的，如第 6.2 節「Setting Up Graphics Card and Monitor」(第 6 章 *Setting Up Hardware Components with YaST*, ↑[部署指南](#)) 中所述。但也可手動編輯組態檔進行設定。

警告: 錯誤的 X 組態可能會損壞您的硬體

進行 X Window System 組態時請務必小心。在完成組態之前，絕對不可以啟動 X Window System。錯誤設定的系統會對硬體造成無法挽回的損害 (特別是對固定頻率的監視器)。本書與 SUSE Linux Enterprise Desktop 的創作者對於產生的任何損害概不負責。本資訊是仔細研究的結果，但不保證提及的所有方法都正確以及不會損害您的硬體。

指令 `sax2` 可建立 `/etc/X11/xorg.conf` 檔案。這是 X Window System 的主要組態檔。請在此處找出和您的圖形卡、滑鼠以及監視器相關的所有設定。

重要: 使用 X -configure

若之前嘗試 SUSE Linux Enterprise Desktop 的 SaX2 失敗，則請使用 x -configure 設定您的 X 設定。若您的設定包含專屬的二進位檔驅動程式，則 X -configure 無法工作。

下列段落會描述組態檔 `/etc/X11/xorg.conf` 的結構。它由多個段落組成，每一個都會處理組態的某個層面。每一個段落以關鍵字 `Section` `<designation>` 開始，並以 `EndSection` 結束。下列轉換會套用到所有部份：

```
Section "designation"
    entry 1
    entry 2
    entry n
EndSection
```

可用的段落類型列出在 [表格 13.1 「/etc/X11/xorg.conf 中的段落」](#) [136頁] 中。

表格 13.1 */etc/X11/xorg.conf 中的段落*

類型	代表意義
Files	用於字型與 RGB 顏色表的路徑。
ServerFlags	伺服器行為的一般切換器。
Module	伺服器應載入的模組清單
InputDevice	輸入設備，例如，鍵盤和特殊輸入設備 (觸控板、搖桿等) 都是在這個段落設定。這個段落含有 Driver 的重要參數以及定義 Protocol 與 Device 的選項。您每個連接到電腦的設備通常有一個 InputDevice 段落。
Monitor	所用的伺服器。此段落的重要元素為：Identifier (稍後的 Screen 定義中將會參考它)、重新整理速率 VertRefresh 以及同步頻率限制 (HorizSync 和 VertRefresh)。設定值以 MHz、kHz 和 Hz 提供。通常，伺服器會拒絕和監視器規格不對應的模式行。這樣可防止不小心將太高的頻率送往顯示器。

類型	代表意義
Modes	<p>特定螢幕解析度的模式行參數。這些參數可以由 <code>SaX2</code> 依據使用者給定的值計算，通常不需要變更。如果您要連接固定頻率監視器，現在可以手動變更。在 <code>/usr/share/doc/howto/en/html/XFree86-Video-Timings-HOWTO</code> 目錄 (howtoenh 套件中提供) 中的 HOWTO 檔案中，可以找到各個數值之意義的詳細資料。若要手動計算 VESA 模式，可以使用 <code>cvt</code> 工具。例如，若要計算解析度為 1680x1050、頻率為 60Hz 之顯示器的模式行，請使用指令 <code>cvt 1680 1050 60</code>。</p>
Device	<p>特定圖形卡。它以本身的描述名稱表示。本節中可用的選項主要取決於所用的驅動程式。例如，如果使用 <code>i810</code> 驅動程式，可以在手冊頁 <code>man 4 i810</code> 中找到有關可用選項的詳細資訊。</p>
Screen	<p>這個段落和 <code>Monitor</code> 以及 <code>Device</code> 一起構成 X.Org 所有必要的設定。在 <code>Display</code> 子段落，請指定虛擬螢幕的大小 (<code>Virtual</code>)、<code>Viewport</code>，以及該螢幕使用的 <code>Modes</code>。</p> <p>請注意，某些驅動程式要求所有使用的組態都必須存在於 <code>Display</code> 區段的某個位置。例如，如果您使用筆記型電腦並想要使用比內部 LCD 大的外部顯示器，則可能需要在 <code>Modes</code> 行末尾新增比內部 LCD 支援的解析度更高的解析度。</p>
ServerLayout	<p>單螢幕顯示或多螢幕顯示組態的配置。這個段落連結輸入設備 <code>InputDevice</code> 和顯示設備 <code>Screen</code>。</p>
DRI	<p>提供資訊給「直接算圖基礎結構 (DRI)」。</p>

`Monitor`、`Device`，與 `Screen` 將在以下詳細說明。如需其他段落的進一步資訊，請參閱 **X.Org** 和 `xorg.conf` 的手冊頁。

`xorg.conf` 中可能有多個不同的 `Monitor` 和 `Device` 段落。也可能有多個 `Screen` 段落。`ServerLayout` 段落決定使用這些段落中的哪些段落。

13.1.1 Screen 段落

Screen 段落是由 monitor 和 device 段落組成，已決定要使用的解析度和色彩深度。Screen 段落可能如同 [範例 13.1 「/etc/X11/xorg.conf 檔的螢幕段落」](#) [138頁] 一般。

範例 13.1 /etc/X11/xorg.conf 檔的螢幕段落

```
Section "Screen"❶
    DefaultDepth 16❷
    SubSection "Display"❸
        Depth 16❹
        Modes "1152x864" "1024x768" "800x600"❺
        Virtual 1152x864❺
    EndSubSection
    SubSection "Display"
        Depth 24
        Modes "1280x1024"
    EndSubSection
    SubSection "Display"
        Depth 32
        Modes "640x480"
    EndSubSection
    SubSection "Display"
        Depth 8
        Modes "1280x1024"
    EndSubSection
    Device "Device[0]"
    Identifier "Screen[0]"❺
    Monitor "Monitor[0]"
EndSection
```

- ❶ Section 決定段落類型，在此情況下為 Screen。
- ❷ DefaultDepth 決定若無指定色彩深度的話，預設使用的色彩深度。
- ❸ 會為每種色彩深度指定不同的 Display 子段落。
- ❹ Depth 決定與該組 Display 設定搭配使用的色彩深度。可能的值為 8、15、16、24 與 32，雖然並不是每個值都受所有 X 伺服器模組或解析度支援。
- ❺ Modes 段落包括可能螢幕解析度的清單。X 伺服器將由左而右檢查這個清單。對於每一個解析度，X 伺服器會在 Modes 段落搜尋適合的 Modeline。Modeline 同時由顯示器與圖形卡的功能決定。Monitor 設定值將決定 Modeline 的結果。

第一個找到的解析度是 Default mode。按 **Ctrl + Alt + +** (數字鍵盤)，可以切換到清單右邊的下一個解析度。按 **Ctrl + Alt + -** (數字鍵盤) 可切換到上一個解析度。您可以在 X 執行中變更解析度。

- ⑥ Display 子段落最後一行的 Depth 16 指虛擬螢幕大小。虛擬螢幕可能的最大大小由圖形卡上安裝的記憶體數量以及所要的色彩深度決定，而不是顯示器的最大解析度。如果省略此行，虛擬解析度便是實體解析度。因為目前的圖形卡都附有大量的視訊記憶體，您可以建立非常大的虛擬桌面。不過，如果將大部份的視訊記憶體用於虛擬桌面，將可能無法使用 3D 功能。例如，如果卡上有 16MB 的視訊 RAM，虛擬螢幕最大可使用 4096x4096 像素，8 位元色彩深度。不過，不建議將所有記憶體用於虛擬螢幕，特別是加速卡，因為卡的記憶體還要用於各種字型與圖形的快取。
- ⑦ Identifier 行 (此處為 Screen[0]) 對這個段落提供一個定義的名稱，以便對下列 ServerLayout 段落提供唯一性參照。Device 和 Monitor 行指定屬於這個定義的圖形卡和顯示器。它們透過本身對應的名稱或識別碼和 Device 以及 Monitor 段落連結。以下詳細討論這些段落。

13.1.2 Device 段落

Device 段落描述特定的圖形卡。xorg.conf 中的設備項目數沒有限制，但每一個設的名稱要使用關鍵字 Identifier (識別碼) 來區別。若您安裝多張圖形卡，段落將只依序編號。第一個稱為 Device[0]，第二個稱為 Device[1]，以下類推。以下檔案顯示摘自一台使用 Matrox Millennium PCI 圖形卡的電腦 (如 SaX2 所設定) 之 Device 區段的一段文字：

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"❶
    Driver         "mga"❷
    Identifier     "Device[0]"
    VendorName     "Matrox"
    Option         "sw_cursor"
EndSection
```

- ❶ BusID 表示要在其中安裝圖形卡的 PCI 或 AGP 插槽。這個值和指令 `lspci` 顯示的 ID 相符。X 伺服器需要十進位格式的詳細資料，但 `lspci` 以十六進位格式顯示這些值。SaX2 會自動偵測 BusID 的值。
- ❷ SaX2 會自動設定 Driver 的值，並指定您的圖形卡要使用的驅動程式。如果是 Matrox Millennium 圖形卡，驅動程式模組稱為 `mga`。X 伺服器會在定義於 `drivers` 子目錄的 Files 段落中的 `ModulePath` 中搜尋。在標準

安裝中，此為 `/usr/lib/xorg/modules/drivers` 目錄或適用於 64 位元作業系統目錄的 `/usr/lib 64/xorg/modules/drivers` 目錄。名稱會附加 `_drv.o`，因此，如果是 `mga` 驅動程式，將載入驅動動程式檔案 `mga_drv.o`。

X 伺服器或驅動程式的行為可以透過附加的選項來操作。Device 段落中的選項 `sw_cursor` 便是一個範例。它可以停用硬體滑鼠游標並描述使用軟體的滑鼠指標。驅動程式模組不同，可用的選項各有不同，可以在 `/usr/share/doc/套件名稱` 目錄內之驅動程式模組的描述檔案中找到。通常情況下，也可以在手冊頁 (`man xorg.conf`、`man 4 <驅動程式模組>` 和 `man 4 chips`) 中找到有效的選項。

如果圖形卡有多個視訊連接器，可將此卡的各個不同設備設定為單個檢視窗。請使用 `SaX2` 以此方式設定圖形介面。

13.1.3 Monitor 和 Modes 段落

和 Device 段落類似，Monitor 和 Modes 段落分別描述一個顯示器。組態檔 `/etc/X11/xorg.conf` 可以含有沒有個數限制的 Monitor 段落。每個 Monitor 段落都會參考帶有 `UseModes` 行 (若可用) 的 Modes 段落。如果沒有可用於 Monitor 段落的 Modes 段落，X 伺服器將從一般同步值計算適當的值。ServerLayout 段落指定相關的 Monitor 段落。

顯示器定義應該由有經驗的使用者進行設定。模式行是 Monitor 段落的重要組成部份。模式行可設定對應解析度的水平與垂直計時。顯示器內容，特別是允許的頻率，儲存在 Monitor 區段中。可以使用公用程式 `cvt` 產生標準 VESA 模式。如需詳細資訊，請參閱 `cvt` 手冊頁 (`man cvt`)。

警告

除非您深入瞭解監視器與圖形卡功能，否則請勿變更模式行，因為這樣做可能會嚴重損壞您的監視器。

想自行開發顯示器描述的人員，應非常熟悉 `/usr/share/X11/doc` 文件的內容。安裝 `xorg-x11-doc` 套件以尋找 PDF 與 HTML 頁面。

在今天，需要手動指定模式行的的情況非常少見。如果您使用先進的多頻顯示器，依規則，X 伺服器可以透過 DDC 直接從顯示器讀取允許頻率以及最佳解析

度，如 SaX2 組態區段中所述。如果因某種原因而不適用，請使用 X 伺服器內附的一種 VESA 模式。此模式對大多數圖形卡與顯示器組合都起作用。

13.2 安裝與設定字型

在 SUSE Linux Enterprise Desktop 上安裝其他字型非常簡單。只要將字型複製到位於 X11 字型路徑中的任何目錄即可 (請參閱第 13.2.1 節「X11 核心字型」[142頁])。若要啟用字型，安裝目錄應為 `/etc/fonts/fonts.conf` 中所設定目錄的子目錄 (請參閱第 13.2.2 節「Xft」[143頁])，或以 `/etc/fonts/suse-font-dirs.conf` 包含於此檔案中。

以下內容摘自 `/etc/fonts/fonts.conf`。此檔案為適用於大多數組態的標準組態檔案。它還定義了包含目錄 `/etc/fonts/conf.d`。在此目錄中，所有以兩位數值開頭的檔案或符號連結將由 `fontconfig` 載入。關於此功能的詳細說明，請參閱 `/etc/fonts/conf.d/README`。

```
<!-- Font directory list -->
<dir>/usr/share/fonts</dir>
<dir>/usr/X11R6/lib/X11/fonts</dir>
<dir>/opt/kde3/share/fonts</dir>
<dir>/usr/local/share/fonts</dir>
<dir>~/.fonts</dir>
<include ignore_missing="yes">conf.d</include>
```

`/etc/fonts/suse-font-dirs.conf` 是自動產生的，以將包含於 (多半是協力廠商) 應用程式，如 OpenOffice.org、Java 或 Adobe Acrobat Reader 中的字型拉進來。某些 `/etc/fonts/suse-font-dirs.conf` 中的典型項目看起來像這樣：

```
<dir>/usr/lib64/ooo-2.0/share/fonts</dir>
<dir>/usr/lib/jvm/java-1_4_2-sun-1.4.2.11/jre/lib/fonts</dir>
<dir>/usr/lib64/jvm/java-1.5.0-sun-1.5.0_07/jre/lib/fonts</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font</dir>
<dir>/usr/X11R6/lib/Acrobat7/Resource/Font/PFM</dir>
```

若要在整個系統上安裝其他字型，請以 `root` 身份手動將字型檔案複製到合適的目錄，如 `/usr/share/fonts/truetype`。此外，這個作業可以透過「KDE 控制中心」中的 KDE 字型安裝程式來執行。結果完全相同。

您也可以建立符號連結，來替代複製實際字型。例如，如果您擁有裝載 Windows 分割區上的授權字型並且想要使用它們，便可能需要這樣做。接下來，請執行 `SuSEconfig --module fonts`。

`SuSEconfig --module fonts` 會執行指令碼 `/usr/sbin/fonts-config` 來處理字型組態。關於此指令碼的詳細資訊，請參閱其手冊頁 (`man fonts-config`)。

點陣字型、TrueType 與 OpenType 字型以及 Type1 (PostScript) 字型的程序完全相同。所有這些字型類型可安裝到任何目錄。

X.Org 包含兩種完全不同的字型系統：舊的「*X11 核心字型系統*」與新設計的「*Xft 與 fontconfig*」系統。以下數節簡短描述這兩種系統。

13.2.1 X11 核心字型

目前，X11 核心字型系統不僅支援點陣字型，也支援縮放字型如 Type1 字型、TrueType 與 OpenType 字型。縮放字型僅只支援不需要消除鋸齒及子像素 (subpixel) 處理的字型，含有更多語言的文字符號的縮放字型所需的載入時間也越長。也支援 Unicode 字型，但是處理速度較慢且需要更多的記憶體。

X11 核心字型系統具有某些先天性的弱點。該字型系統已過時，無法再以有效的方法擴充。保留這種字型是為了回溯相容，但最好儘可能使用更先進的 Xft 和 fontconfig 系統。

為了能夠進行作業，X 伺服器必須知道它有哪些字型可用，以及系統中的哪些地方可以找到這些字型。這是由 `FontPath` 變數負責處理，該變數包含所有有效系統字型目錄的路徑。每個目錄中的 `fonts.dir` 檔案，可以列出目錄中可用的字型有哪些。`FontPath` 是 X 伺服器在啟動時產生的。它會在 `/etc/X11/xorg.conf` 組態檔案的 `FontPath` 項目中，搜尋一個有效的 `fonts.dir` 檔案。這些項目會在 `Files` 區段中找到。使用 `xset q` 顯示實際的 `FontPath`。也可以使用 `xset` 在執行時變更此路徑。若要新增其他路徑，請使用 `xset+fp <path>`。若要移除不要的路徑，請使用 `xset-fp <path>`。

如果 X 伺服器已經啟動，裝載目錄中新安裝的字型可以透過指令 `xsetfp rehash` 來設成可用。這個指令以 `SuSEconfig--module fonts` 執行。由於指令 `xset` 必須存取執行中的 X 伺服器，因此只有在 `SuSEconfig--module fonts` 是從可存取之執行中 X 伺服器的外圍程序啟動時才有效。實現此目的的

最簡單方法是輸入 `su` 和 `root` 密碼，採用 `root` 權限。`su` 可以將啟動 X 伺服器之使用者的存取權傳送到 `root` 外圍程序。要檢查字型是否安裝正確以及是否可以透過 X11 核心字型系統來使用，請使用 `xlsfonts` 指令來列出所有可用字型。

根據預設，SUSE Linux Enterprise Desktop 使用 UTF-8 地區設定。因此，最好使用 Unicode 字型 (在 `xlsfonts` 的輸出中，字型名稱的結尾為 `iso10646-1`)。`xlsfonts | grep iso10646-1` 可以列出所有可用的 Unicode 字型。SUSE Linux Enterprise Desktop 隨附的 Unicode 字型絕大部分都至少具有歐洲語言所需的文字符號 (舊編碼方式為 `iso-8859-*`)。

13.2.2 Xft

從一開始，Xft 的程式設計人員便保證，會支援平滑美觀的可縮放字型。如果使用 Xft，字型將由使用字型的應用程式處理，而不是如 X11 核心字型系統那樣由 X 伺服器處理。在這種方式，個別的應用程式需要存取實際字型檔並完全控制文字符號的處理方式。多種語言文字的正确顯示基礎便是由此構成。直接存取字型檔對於在列印中內嵌字型，來保證列印出的外觀和螢幕輸出一致非常有用。

在 SUSE Linux Enterprise Desktop 中，KDE 和 GNOME 這兩個桌面環境、Mozilla 以及其他許多應用程式預設已使用 Xft。越來越多的應用程式已經從舊 X11 核心字型系統改用 Xft。

Xft 使用 `fontconfig` 程式庫來尋找字型以及操作字型的處理方式。`fontconfig` 的內容由全域組態檔案 `/etc/fonts/fonts.conf` 控制。特殊組態應新增至 `/etc/fonts/local.conf` 與使用者特定的組態檔案 `~/.fonts.conf`。這些 `fontconfig` 組態檔每一個都必須以下列開頭

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

並以下列結束

```
</fontconfig>
```

要新增字型搜尋目錄，請附加下列行：

```
<dir>/usr/local/share/fonts/</dir>
```

不過，通常並不需要這樣做。依預設，使用者專屬目錄 `~/.fonts` 已經輸入到 `/etc/fonts/fonts.conf` 中。因此，安裝額外的字型時，只要將它們複製到 `~/.fonts` 即可。

您還可以插入操作字型外觀的規則。例如，輸入

```
<match target="font">
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

可以關閉所有字型的消除鋸齒效果，

```
<match target="font">
  <test name="family">
    <string>Luxi Mono</string>
    <string>Luxi Sans</string>
  </test>
  <edit name="antialias" mode="assign">
    <bool>false</bool>
  </edit>
</match>
```

可以關閉特定字型的消除鋸齒效果。

依預設，大部份的應用程式都是使用字型名稱 `sans-serif` (或相等的 `sans`)、`serif` 或 `monospace`。這些都不是真實的字型，而是可以解析為適當字型的別名，視語言設定而定。

使用者很容易在 `~/.fonts.conf` 新增規則來將這些別名解析為屬意的字型：

```
<alias>
  <family>sans-serif</family>
  <prefer>
    <family>FreeSans</family>
  </prefer>
</alias>
<alias>
  <family>serif</family>
  <prefer>
    <family>FreeSerif</family>
  </prefer>
</alias>
<alias>
  <family>monospace</family>
  <prefer>
    <family>FreeMono</family>
```

```
</prefer>
</alias>
```

因為幾乎所有的應用程式都預設使用這些別名，所以幾乎整個系統會受到影響。這樣，您幾乎可以簡單地隨處使用喜歡的字型而不必針對個別應用程式修改字型設定。

請使用指令 `fc-list` 來找出已安裝且可用的字型。例如，指令 `fc-list` 指令可傳回所有字型的清單。要找出含有希伯來文 (`:lang=he`) 所有文字符號的可用縮放字型 (`:scalable=true`)、這些字型的名稱 (`family`)、樣式 (`style`)、粗細 (`weight`) 以及包含這些字型的檔案名稱，請輸入下列指令：

```
fc-list ":lang=he:scalable=true" family style weight
```

這個指令的輸出看起來如下：

```
Lucida Sans:style=Demibold:weight=200
DejaVu Sans:style=Bold Oblique:weight=200
Lucida Sans Typewriter:style=Bold:weight=200
FreeSerif:style=Bold,polkrepko:weight=200
FreeSerif:style=Italic,ležeče:weight=80
FreeSans:style=Medium,navadno:weight=80
DejaVu Sans:style=Oblique:weight=80
FreeSans:style=Oblique,ležeče:weight=80
```

可以使用 `fc-list` 來查詢的重要參數：

表格 13.2 *fc-list* 的參數

參數	意義及可用值
<code>family</code>	字型系列的名稱，例如，FreeSans。
<code>foundry</code> (製造商)	字型的製造商，例如，urw。
<code>style</code> (樣式)	字型樣式，例如，Medium (中)、Regular (一般)、Bold (粗體)、Italic (斜體) 或是 Heavy (濃)。
<code>lang</code>	字型支援的語言，例如，de 表示德文、ja 表示日文、zh-TW 表示繁體中文、zh-CN 表示簡體中文。

參數	意義及可用值
<code>weight</code>	字型粗細，例如，80 表示一般，或是 200 表示粗體。
<code>slant</code> （斜度）	通常，斜度 0 表示沒有斜度，100 表示斜體。
<code>file</code>	字型檔案的名稱。
<code>outline</code> （外框）	<code>true</code> （真）表示外框字型，或是 <code>false</code> （偽）表示其他字型。
<code>scalable</code> （縮放）	<code>true</code> （真）表示縮放字型，或是 <code>false</code> （偽）表示其他字型。
<code>bitmap</code> （點陣）	<code>true</code> （真）表示點陣字型，或是 <code>false</code> （偽）表示其他字型。
<code>pixelsize</code> （點大小）	以點表示的字型大小。這個選項和 <code>fc-list</code> 一起使用時只對點陣字型有意義。

13.3 如需更多資訊

請安裝 `xorg-x11-doc` 和 `howtoenh` 套件來取得有關 X11 的詳盡資訊。可在專案的首頁 <http://www.x.org> 上找到關於 X11 開發的更多資訊。

手冊頁中詳細描述了 `xorg-x11-driver-video` 套件隨附的很多驅動程式。例如，如果使用 `radeon` 驅動程式，可以在 `man 4 radeon` 中找到有關此驅動程式的詳細資訊。

有關協力廠商驅動程式的資訊可在 `/usr/share/doc/packages/<套件名稱>` 中找到。例如，安裝套件後，`x11-video-nvidiaG01` 的文件就位於 `/usr/share/doc/packages/x11-video-nvidiaG01` 中。

使用 FUSE 存取檔案系統

FUSE 是使用者空間中的檔案系統 (*file system in userspace*) 的縮寫。這表示您可以以非特權使用者的身份設定並裝載檔案系統。一般情況下，您必須是 `root` 才能執行此任務。FUSE 自身就是一個核心模組。將 FUSE 與外掛程式結合便能延伸其功能，幾乎可存取所有檔案系統，如遠端 SSH 連線、ISO 影像及其他

14.1 設定 FUSE

您必須先安裝套件 `fuse` 才能使用 FUSE。您需要不同套件中的其他外掛程式，具體取決於要使用的檔案系統。使用 YaST 以 `fuse` 或 `file system` 為關鍵字搜尋這些套件。

一般而言，您無需設定 FUSE 即可直接使用。但建議您建立可將所有定點組合於其中的目錄。例如，可以建立目錄 `~/mounts` 並在該處插入不同檔案系統的子目錄。

14.2 裝載 NTFS 分割區

NTFS，即新技術檔案系統 (*New Technology File System*)，是 Windows NT、2000、XEP、Vista 等多種 Windows 版本的預設檔案系統。它取代了 FAT 檔案系統。若要以一般使用者身份裝載 Windows 分割區，請執行下列步驟：

- 1 切換為 `root` 身份，然後安裝套件 `ntfs-3g`。

- 2 建立目錄 `/media/windows`。
- 3 確定所需的 Windows 分割區。使用 YaST 並啟動磁碟分割程式模組，以檢視屬於 Windows 的分割區，但不要做任何變更。或者也可以切換為 `root` 身份，然後執行 `/sbin/fdisk -l`。尋找分割區類型為 HPFS/NTFS 的分割區。
- 4 在讀寫模式下裝載分割區。使用相應的 Windows 分割區取代佔位符 `DEVICE`：

```
ntfs-3g /dev/DEVICE /media/windows
```

如果要在唯讀模式下使用 Windows 分割區，請附加 `-o ro`：

```
ntfs-3g /dev/DEVICE /media/windows -o ro
```

指令 `ntfs-3g` 使用目前的使用者 ID (UID) 與群組 ID (GID) 裝載指定設備。若要對其他使用者設定寫入權限，請使用指令 `id USER` 以取得 UID 與 GID 的輸出。使用下列指令進行設定：

```
id tux
uid=1000(tux) gid=100(users) Gruppen=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE /media/windows -o uid=1000,gid=100
```

可在線上文件中找到其他選項。

若要卸載資源，請使用：

```
fusermount -u /media/windows
```

14.3 使用 SSHFS 裝載遠端檔案系統

SSH，即安全外圍程序網路通訊協定，可用於透過安全通道在兩台電腦間交換資料。要透過 FUSE 建立 SSH 連線，請執行下列步驟：

- 1 安裝套件 `sshfs`。
- 2 建立要從中存取遠端電腦的目錄。建議使用 `~/mounts/HOST`。使用遠端電腦的名稱取代 `HOST`。
- 3 裝載遠端檔案系統：

```
sshfs USER:HOST ~/mounts/HOST
```


使用相應的值取代 *USER* 與 *HOST*。

- 4 輸入您用於存取遠端電腦的密碼。

14.4 裝載 ISO 檔案系統

要檢視 ISO 影像，可以使用 `fuseiso` 套件進行裝載：

- 1 安裝套件 `fuseiso`。
- 2 建立目錄 `~/mounts/iso`。
- 3 裝載 ISO 影像：

```
fuseiso ISO_IMAGE ~/mounts/iso
```

您只能讀取 ISO 影像中的內容，無法寫回內容。

14.5 可用的 FUSE 外掛程式

FUSE 的功能會隨外掛程式而變化。下表列出了常見的外掛程式。

表格 14.1 可用的 FUSE 外掛程式

<code>fuseiso</code>	裝載包含 ISO9660 檔案系統的 CD-ROM 影像
<code>ntfs-3g</code>	裝載 NTFS 卷冊 (包含讀寫支援)
<code>sshfs</code>	基於 SSH 檔案傳輸通訊協定的檔案系統用戶端
<code>wdfs</code>	裝載 WebDAV 檔案系統

14.6 如需更多資訊

有關詳細資訊，請參閱 FUSE 的首頁 <http://fuse.sourceforge.net>。

第 III. 行動電腦 部分

Linux 的行動計算功能

人們多半會將行動計算的概念與筆記型電腦、PDA 和行動電話及其彼此間資料的交換聯想在一起。行動硬體元件 (如外接硬碟、隨身碟或數位相機) 可以連接到筆記型電腦或桌上電腦系統。行動計算環境需要許多軟體元件，而且有些應用程式是專為行動用途量身訂製的。

15.1 筆記型電腦

筆記型電腦的硬體與一般桌上電腦系統不同。這是因為像可交換性、佔用的空間和省電之類準則的相關屬性。行動硬體的製造廠商開發了一些標準介面，例如 PCMCIA (國際個人電腦記憶卡協會)、Mini PCI 和 Mini PCIe，可以使用它們來擴充筆記型電腦的硬體。這些標準涵蓋了記憶卡、網路介面卡、ISDN 和數據卡，以及外接硬碟。

提示: SUSE Linux Enterprise Desktop 和 Tablet PC

SUSE Linux Enterprise Desktop 也支援 Tablet PC。Tablet PC 隨附的觸控板/數位板可讓您使用數位筆甚至是指尖直接在螢幕上編輯資料，而無需使用滑鼠和鍵盤。它們的安裝與設定方式與所有其他系統非常相似。如需安裝和設定 Tablet PC 的詳細資料，請參閱第 17 章使用 *Tablet PC* [173頁]。

15.1.1 省電

製造筆記型電腦時包含最有效使用電源系統元件，造就了它們不必靠電力系統即能使用的適用性。它們在省電的貢獻上至少與作業系統的貢獻一樣重要。對於可影響筆記型電腦電源的使用以及影響使用電池電源之操作時間的各種方法，SUSE® Linux Enterprise Desktop 均可提供支援。下列清單以遞減排序陳述對省電的貢獻：

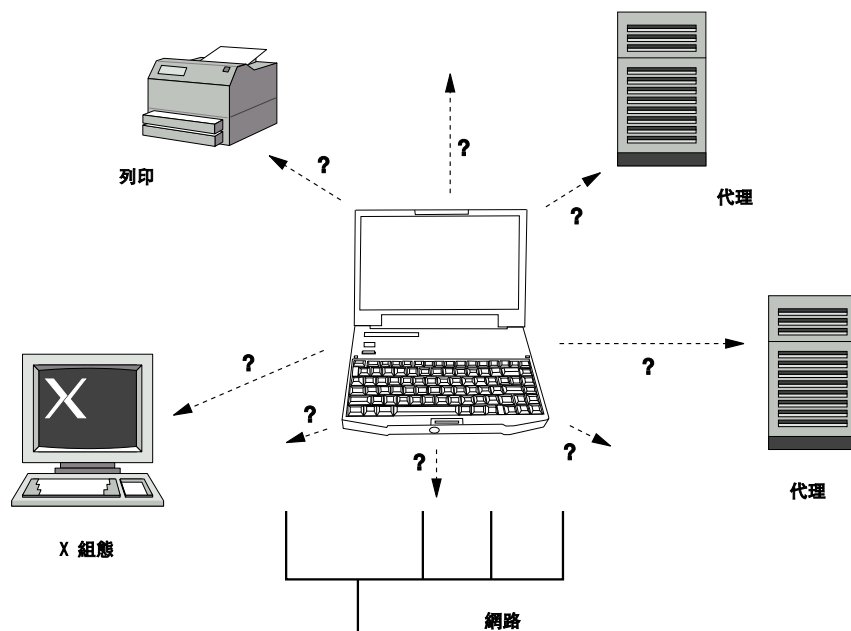
- 調節 CPU 速度.
- 暫停時關閉顯示器亮度.
- 手動調整顯示器亮度.
- 將未使用的 HotPlug 配件 (USB CD-ROM、外接滑鼠、未使用的 PCMCIA 卡、WLAN 等) 解除連接。
- 閒置時降低硬碟轉速.

如需桌面特定電源管理的詳細資訊，請參閱第 1.15 節「Controlling Your Desktop's Power Management」(第 1 章 *Getting Started with the GNOME Desktop*, ↑*GNOME 使用者指南*) 中關於如何使用 GNOME 電源管理程式的說明。如需 KDE 電源管理 Applet 的詳細資訊，請參閱第 8 章 *Controlling Your Desktop's Power Management with KPowerSave* (↑*KDE 使用者指南*)。

15.1.2 與變動作業環境的整合

您的系統在用於行動計算時，常需與變動的作業環境搭配。許多服務與環境習習相關，基礎用戶端必須重新設定。SUSE Linux Enterprise Desktop 可以為您處理這項任務。

圖形 15.1 將行動電腦整合到現有環境中



筆記型電腦在小型家用網路及公司網路之間來回交替使用的情形，影響的服務包括：

網路

這包括指定 IP 位址、名稱解析、網際網路連接、以及與其他網路的連接。

列印

視網路而定，必須出現可用印表機及可用列印伺服器目前的資料庫。

電子郵件和代理

列印時，對應伺服器的清單必須是最新的。

X (圖形環境)

如果您的筆記型電腦暫時連接到視訊設備或外接螢幕，需要使用不同的顯示器組態。

SUSE Linux Enterprise Desktop 提供了幾種將筆記型電腦整合到現有作業環境的方法：

NetworkManager

NetworkManager 是特地為筆記型電腦的行動網路量身訂做的。它讓您能夠輕易地在不同網路環境之間或不同網路類型之間 (例如無線區域網路和乙太網路) 自動切換。NetworkManager 支援無線 LAN 中的 WEP 與 WPA-PSK 加密。亦支援撥接連接 (使用 smpppd)。兩種桌面環境 (GNOME 和 KDE) 都包含 NetworkManager 的前端工具。如需有關桌面 applet 的更多資訊，請參閱第 22.4 節「使用 KDE NetworkManager 工具集」[260頁]與第 22.5 節「使用 GNOME NetworkManager Applet」[260頁]。

表格 15.1 NetworkManager 的使用案例

我的電腦...	使用 NetworkManager
是筆記型電腦	是
有時候會連接到不同網路	是
提供網路服務 (例如 DNS 或 DHCP)	否
只使用靜態 IP 位址	否

在不應由 NetworkManager 處理網路組態時，請使用 YaST 工具來設定網路連線。

SCPM

SCPM (系統組態設定檔管理, system configuration profile management) 允許儲存系統的任意組態狀態到一種稱為「設定檔」(profile) 的「快照」(snapshot)。設定檔可以針對不同的情況而建立。當系統於變更的環境下 (家用網路、公司網路) 操作時，這種設定檔很有用。能夠直接在設定檔之間切換。若要在您的系統上安裝並執行 SCPM，請安裝套件 kscpm，將設定檔選擇工具 KDE applet 新增到面板，使用 YaST 設定檔管理模組啟用 SCPM，然後設定無需輸入 root 密碼即可切換設定檔的使用者。決定要在系統重新開機之後保留設定檔變更，還是應在關機時將其丟棄。確定所有資源群組

(即網路和印表機等服務)都在作用中。繼續使用透過設定檔選擇工具啟動的 SUMF (SCPM Unified Management Front-End) 工具建立實際的設定檔。為要在此系統中使用的所有不同安裝建立設定檔。切換設定檔有兩種方式：在正在執行的系統中使用設定檔選擇工具 applet 來進行，或者在系統開機時使用 F3 鍵。切換設定檔時，SCPM 會自動將系統組態調整到所選設定檔中配置的新環境。

SLP

服務位置通訊協定 (SLP) 簡化了筆記型電腦到現有網路的連線。沒有 SLP，筆記型電腦的管理員通常需要熟知網路中的可用服務。SLP 廣播特定類型服務的可用性到本地網路中的所有用戶端。支援 SLP 的應用程式可以處理由 SLP 分派出來的資訊，而且能夠自動設定。SLP 甚至能夠用來安裝系統，省略搜尋適用安裝來源的時間。如需有關 SLP 的詳細資訊，請參閱[第 20 章網路中的 SLP 服務](#) [247頁]。

15.1.3 軟體選項

行動用途的許多特殊任務領域是由專屬軟體所負責的：系統監控 (特別是電池電量)、資料同步化，以及與週邊和網際網路的無線通訊。以下幾節涵蓋了 SUSE Linux Enterprise Desktop 為每項任務提供的最重要的應用程式。

系統監控

以下為 SUSE Linux Enterprise Desktop 提供的兩種 KDE 系統監控工具：

KPowersave

KPowersave 是在控制台中顯示可充電電池狀態的 applet。圖示會調整，表示電源供應的類型。使用 AC 電源時，會顯示一個小型插頭圖示。使用電池時，圖示會變更為電池。要求 root 密碼後，會開啟電源管理的 YaST 模組的對應功能表。這樣可以設定不同電源的系統行為。

KSysguard

KSysguard 是獨立的應用程式，可將系統中所有可測量的參數匯集到一個監控環境中。KSysguard 具有 ACPI (電池狀態)、CPU 負載、網路、磁碟分割及記憶體使用率的監控程式。它還能監看及顯示所有系統程序。可自定所收集資料的顯示及過濾。透過網路可以同時監控各種資料頁的不同系統參數，或在並列埠中收集各個機器的資料。KSysguard 也能在沒有 KDE 環境的機器上做為精靈來執行。如需有關此程式的詳細資訊，請參閱其整合的說明功能或 SUSE 說明頁。

在 GNOME 桌面中，使用 GNOME 電源管理程式偏好設定和系統監視器。

同步化資料

在沒有連接網路的行動機器與公司中連接網路的工作站之間切換時，必須讓所有個體間已處理的資料保持同步。這些資料可能包括電子郵件資料夾、目錄以及個別檔案，在出差時需要和公司同時保持最新。這兩種情況中的解決方案如下：

同步化電子郵件

使用 IMAP 帳戶在公司網路中儲存您的電子郵件。然後使用任何已啟用 IMAP 的中斷連接電子郵件用戶端 (如 *GNOME 使用者指南* (↑*GNOME 使用者指南*) 與 *KDE 使用者指南* (↑*KDE 使用者指南*) 中所述的 Mozilla Thunderbird Mail、Evolution 或 KMail)，從工作站存取電子郵件。電子郵件用戶端需要設定，如此「傳送郵件」會永遠存取相同的資料夾。這樣可確保完成同步化程序後，所有訊息及其狀態資訊都能使用。使用郵件用戶端中執行的 SMTP 伺服器來傳送訊息，代替全系統 MTA postfix 或 sendmail，來接收有關未傳送郵件的可靠回應。

同步化檔案與目錄

有數種公用程式適合用來同步化筆記型電腦與工作站之間的資料。

無線通訊

筆記型電腦不僅能夠使用電纜連接家用或公司網路，還可以透過無線方式連接其他電腦、周邊設備、行動電話或 PDA。Linux 支援三種類型的無線通訊：

WLAN

涵蓋最廣泛無線技術的 WLAN，是唯一適合操作大型，有時甚至是極端分散網路的模式。個別機器可以彼此連接，形成一個獨立的無線網路或存取網際網路。稱為存取點的設備，是做為啟用 WLAN 設備的基礎工作站，而且扮演存取網際網路的中介角色。行動使用者可以在存取點之間切換，端視所在位置及哪個存取點提供最佳連接而定。與行動電話的通訊方式類似，供 WLAN 使用者使用的大型網路，不用將它們連結到特定位置才能存取。如需有關 WLAN 的詳細資訊，請參閱第 19.1 節「無線區域網路」[237頁]。

藍芽

藍芽在所有無線技術中的應用範圍最廣泛。它可以用於電腦 (筆記型電腦) 與 PDA 或行動電話之間的通訊，像 IrDA 的功能一樣。也可以用來連接可

視範圍內的各個電腦。藍芽也應用在連接無線系統元件，像鍵盤或滑鼠。不過，此技術的範圍還不足以連接遠端系統與網路。WLAN 是穿透實體障礙物 (如牆) 進行通訊的選用技術。

IrDA

IrDA 是最短距離的無線技術。兩邊的通訊方必須位於彼此可見的距離內。無法克服如牆之類的障礙物。IrDA 的一個應用方式是從筆記型電腦傳輸檔案到行動電話。可使用 IrDA 來涵蓋筆記型電腦到行動電話的短距離。長距離傳送檔案到收件者是由行動網路處理。IrDA 的另一種應用是在公司內以無線方式傳輸列印工作。

15.1.4 資料安全性

理想而言，您應使用多種方式保護筆記型電腦上的資料，不受未授權的存取。可以從下列三大面向來採取適當的安全性措施：

防止竊取

不論在什麼時候，都要避免機器遭到竊取。各種保全工具 (例如鎖鏈) 都可以在零售店中買到。

增強式驗證

除了透過登入和密碼進行標準驗證之外，還使用生物驗證。SUSE Linux Enterprise Desktop 支援指紋驗證。如需詳細資訊，請參閱第 3 章 *Using the Fingerprint Reader* (↑*Security Guide*)。

保全系統上的資料

重要資料在傳輸時不僅要加密，在硬碟上也要加密。這樣在竊取的情況中可確保其安全。如需使用 SUSE Linux Enterprise Desktop 建立加密分割區的資訊，請參閱第 12 章 *Encrypting Partitions and Files* (↑*Security Guide*)。另一種情況是在使用 YaST 新增使用者時建立加密的主目錄。

重要：資料安全性和暫停寫入到磁碟

暫停寫入到磁碟時，不會解除裝載加密分割區。因此，任何人只要偷到硬體並且將硬碟復原，即可使用分割區上的資料。

網路安全性

應該保護任何資料的傳輸，無論其方式為何。有關 Linux 及網路的一般安全性議題，請參閱第 1 章 *Security and Confidentiality* (↑*Security Guide*)。與無線網路相關的安全性措施，請參閱第 19 章 *無線通訊* [237 頁]。

15.2 行動硬體

SUSE Linux Enterprise Desktop 能夠自動偵測 Firewire (IEEE 1394) 或 USB 上的行動儲存設備。行動儲存設備這個詞，包含了任何種類的 FireWire 或 USB 硬碟、USB 隨身碟或數位相機。只要這些設備一透過對應介面與系統連接，即可自動偵測與設定。GNOME 和 KDE 的檔案管理員都提供靈活的行動硬體項目處理方式。若要安全卸載所有這些媒體，請使用檔案管理員 KDE 或 GNOME 的「安全移除」(KDE) 或「卸載卷冊」(GNOME) 功能。關於透過桌面處理抽取式媒體的詳細資訊，請參閱 *GNOME 使用者指南* (↑*GNOME 使用者指南*) 與 *KDE 使用者指南* (↑*KDE 使用者指南*)。

外接硬碟 (USB 和 FireWire)

系統正確辨識外接硬碟後，其圖示就會出現在檔案管理員中。按一下圖示，會顯示設備的內容。在此可以建立、編輯或刪除資料夾及檔案。若要重新命名硬碟的系統指定名稱，請在圖示上按一下滑鼠右鍵所開啟的功能表中，選取對應的功能表項目。此名稱的變更，僅限顯示於檔案管理員中。因此，裝載於 `/media` 中設備的描述符號不會受影響。

USB 隨身碟

這些設備由系統處理，如外接硬碟一樣。同樣可以在檔案管理員中重新命名項目。

數位相機 (USB 和 FireWire)

由系統辨識的數位相機，在檔案管理員概觀中也是顯示為外接硬碟機。KDE 允許讀取及存取 URL `camera:/` 上的圖片。然後便可以使用 digiKam 或 f-spot 來處理影像。針對進階相片處理，請使用 The GIMP。關於 digiKam、f-spot 與 The GIMP 的簡介，請參閱第 24 章 *Managing Your Digital Image Collection* (↑*應用程式指南*)、第 25 章 *Managing Your Digital Image Collection with F-Spot* (↑*應用程式指南*) 與第 23 章 *Manipulating Graphics with The GIMP* (↑*應用程式指南*)。

15.3 行動電話和 PDA

桌上電腦或筆記型電腦可以透過藍芽或 IrDA 進行通訊。有些型號支援兩種通訊協定，而有些則僅支援其中一種。兩種通訊協定的使用範圍及相關的延伸說明文件已經在**章節「無線通訊」** [158頁]中提及。行動電話通訊協定的組態，於手冊中均有說明。

Evolution 與 Kontact 均支援在 Palm, Inc. 生產的手持裝置之間進行同步化作業。開始與設備連接後，不論使用哪種方式，都可輕鬆透過精靈的協助加以使用。設定 Palm Pilot 的支援後，必須判斷應該同步化哪些類型的資料 (通訊錄、約會等)。如需詳細資訊，請參閱 *GNOME 使用者指南* (↑*GNOME 使用者指南*) 與 *KDE 使用者指南* (↑*KDE 使用者指南*)。

程式 opensync 提供了更全面的同步化解決方案 (對於不同的設備，請參閱套件 libopensync、msynctool 及相應的外掛程式)。

15.4 如需更多資訊

有關行動設備與 Linux 所有問題的參考重點，請參閱<http://tuxmobil.org/>。該網站的各個小節，能夠處理筆記型電腦、PDA、行動電話及其他行動硬體的硬體及軟體方面。

<http://tuxmobil.org/>的類似方法由<http://www.linux-on-laptops.com/>提供。在此可以找到有關筆記型電腦和掌上型設備的資訊。

關於筆記型電腦相關議題，SUSE 提供了專屬的郵件清單 (使用德文)。請參閱<http://lists.opensuse.org/opensuse-mobile-de/>。在此清單上，使用者與開發人員討論了 SUSE Linux Enterprise Desktop 行動計算的所有觀念。以英文張貼的文章會收到回應，但是大部分的歸檔資訊僅有德文版本。使用<http://lists.opensuse.org/opensuse-mobile/>可獲取英文張貼的文章。

<http://en.opensuse.org/OpenSync> 中提供了關於 OpenSync 的資訊。

電源管理

電源管理對筆記型電腦十分重要，對其他系統也很有用。ACPI (進階組態和電源介面)可以在所有新式電腦(筆記型電腦、桌上型電腦和伺服器)上使用。電源管理技術需要配備合適的硬體與 BIOS 常式。大多數筆記型電腦和許多新式的桌上型電腦及伺服器都符合這些需求。此技術還可以控制 CPU 頻率比例，這有助於省電及降低噪音。

16.1 省電功能

省電功能不僅對於筆記型電腦的行動用途很重要，對於桌上型系統也很重要。主要功能以及在 ACPI 中的用途包括：

待命

不支援。

暫停 (記憶體)

此模式會將整個系統狀態寫入 RAM 中。接著，除了 RAM 以外，整個系統都會進入睡眠狀態。在此狀態中，電腦所使用的電源極少。此狀態的好處是可以在幾秒內將工作復原到暫停之前的狀態，而不用開機或重新啟動應用程式。此功能等同於 ACPI 狀態 S3。對此狀態的支援仍在開發中，因此支援程度多半因硬體而異。

睡眠 (暫停磁碟)

在此操作模式，會將整個系統狀態寫入硬碟，然後關閉系統。至少要有與 RAM 一樣大的交換分割區，才能寫入所有作用中資料。要從此狀況重新啟用需耗時 30 到 90 秒。還原時會回到暫停前的狀態。有些製造商會為此模式

提供有用的混合功能，例如 IBM Thinkpad 中的 RediSafe。等同的 ACPI 狀態為 S4。在 Linux 中，暫停寫入到磁碟是由獨立於 ACPI 之外的核心常式來執行。

電池監視器

ACPI 會檢查電池充電狀態並提供相關資訊。此外，ACPI 會在電力到達某個關鍵狀態時，協調要執行的動作。

自動關閉電源

關機後，電腦會關閉電源。此功能很重要，尤其是在電池用盡前所執行的自動關機。

處理器速度控制

與 CPU 連結時有三種方式可節省電源：頻率和電壓比例（也稱為 PowerNow! 或 Speedstep）、節流，以及使處理器暫休（C 狀態）。依據電腦的操作模式，也可以合併這些操作方法。

16.2 ACPI

ACPI (進階組態與電源介面) 可讓作業系統設定和控制個別的硬體元件。ACPI 可取代 PnP 與 APM。它能提供一些資訊，包括電池、變電器、溫度、風扇以及「關閉蓋子」或「電池電力不足」等系統事件。

BIOS 會提供一些表格，內含關於個別元件與硬體的存取方法等資訊。作業系統會使用這些資訊來執行任務，像是指定中斷或啟用和停用元件。因為作業系統會執行儲存於 BIOS 中的指令，所以 BIOS 實行會決定其功能。ACPI 能偵測和載入的表格在 `/var/log/boot.msg` 中可以找到。請參閱第 16.2.3 節「疑難排解」[166 頁]，以取得更多有關 ACPI 問題疑難排解的資訊。

16.2.1 控制 CPU 效能

CPU 有三種省電方法。依據電腦的操作模式的不同，這些方法可合併使用。省電也表示能降低系統溫度，減低風扇的使用頻率。

頻率和電壓比例

PowerNow! 與 Speedstep 是 AMD 與 Intel 分別針對此技術所使用的實作。不過，此技術也套用於其他製造商的處理器中。CPU 的時脈頻率及其核心電壓會同時降低，產生高於線性的省電效能。也就是當頻率減半(效能減半)，

使用的電量卻能遠低於原本的一半。此技術與ACPI無關。CPU 頻率比例的執行有兩種主要的方式—利用核心本身或利用使用者空間應用程式。因此，有不同的核心調節器，這可在 `/sys/devices/system/cpu/cpu*/cpufreq/` 下設定。

使用者空間調節器

如果設定使用者空間調節器，核心會將 CPU 頻率比例的控制權交給使用者空間應用程式 (通常是一個精靈)。在 SUSE Linux Enterprise Desktop 套裝軟體中，此精靈為 powersaved 套件。使用此執行方式時，CPU 頻率會根據目前的系統負載來調整。預設狀態下會使用核心執行方式之一。但是，對某些硬體，或特定的處理器或驅動程式而言，使用者空間執行仍是唯一可行的解決方案。

要求式調節器

這是動態 CPU 頻率規則的核心執行方式，應該適用於大部分系統。只要系統負載一過高，就立即提高 CPU 頻率。系統負載降低，則會隨之降低。

保守調節器

此調節器類似要求式調節器，但所用的規則比較保守。系統負載過高的情況必須持續一段時間後，才會提高 CPU 頻率。

省電調節器

CPU 頻率固定設為可能的最小值。

效能調節器

CPU 頻率固定設為可能的最大值。

調節時脈頻率

此技術會忽略部分 CPU 的時脈訊號脈衝。到達 25% 調節時，會省略四分之一脈衝，到達 87.5% 時則每八次脈衝僅有一次會到達處理器。不過，節省用電量稍低於線性。通常調節功能僅在無此頻率比例時使用，或是為了最大化省電效果時使用。此外，此技術必須使用特定程序來進行控制。系統介面是 `/proc/acpi/processor/*/throttling`。

使處理器進入睡眠

作業系統會在沒事可做時使處理器進入睡眠。在此情況中，作業系統會傳送 `halt` 指令給 CPU。一共有三種狀態：C1、C2 和 C3。在最節省的 C3 狀態中，連處理器快取與主記憶體間的同步化也會暫停。因此，僅能在沒有任何設備透過 Bus master 活動來修改主記憶體內容時可以套用此狀態。有些驅動

程式會禁止使用 C3。目前的狀態會顯示在 `/proc/acpi/processor/*/power` 中。

頻率比例及調節只在處理器忙碌時使用，因為在處理器閒置時，一定會套用最節省的 C 狀態。如果 CPU 正忙碌，頻率比例是建議的省電方法。通常處理器僅有部份的工作負載。在此情況中，可以使用較低的頻率。通常，最佳方法是使用核心要求式調節器或精靈 (例如 `powersaved`) 來控制動態頻率比例。對電池的操作而言，靜態設定為較低頻率比較好，也可以用在您想降低電腦溫度或減低噪音時。

調節應做最後手段使用，例如，在高度系統負載下仍要延伸電池操作時間時。不過在調節過多時，有些系統無法運作順暢。此外，當 CPU 要做的事不多時，調節 CPU 是無意義的動作。

16.2.2 ACPI 工具

ACPI 公用程式包含僅顯示電池充電等級與溫度等資訊的工具 (`acpi`、`klaptopdaemon` 及 等等。) 、協助在 `/proc/acpi` 中存取結構或協助監控變更 (`akpi`、`acpiw`、`gtkacpiw`) 的工具，以及在 BIOS 中編輯 ACPI 表格的工具 (`pmtools` 套件)。

16.2.3 疑難排解

共有兩種不同類型的問題。一方面是核心的 ACPI 程式碼包含無法及時偵測到的錯誤。在這種情況中，將會有可供下載的解決方案。不過通常問題是因 BIOS 而起。有時，會刻意在 BIOS 中整合與 ACPI 規格不符的技術，以避免在其他常見作業系統中的 ACPI 實作錯誤。會在黑名單中將那些在 ACPI 實行中有重大錯誤的硬體元件記錄下來，以避免 Linux 核心對這些元件使用 ACPI。

發生問題時要做的第一件事是更新 BIOS。如果電腦未能開機，下列中的某一個開機參數也許有幫助：

`pci=noacpi`

不使用 ACPI 來設定 PCI 設備。

`acpi=ht`

僅執行一個簡單的資源組態。不將 ACPI 用於其他目的。

acpi=off
關閉 ACPI。

警告: 未使用 ACPI 的開機問題

有些較新的機器 (尤其是 SMP 系統及 AMD64 系統) 需透過 ACPI 以正確設定硬體。關閉這些機器的 ACPI 會發生隨之而來的問題。

有時，透過 USB 或 FireWire 連接的硬體會另機器混淆。如果機器拒絕開機，則拔除所有不需要的硬體插頭，並再試一次。

開機後，可使用 `dmesg | grep -2i acpi` 指令來監控系統的開機訊息 (或所有訊息，因為也可能是 ACPI 以外的因素構成問題)。如果是在分析 ACPI 表格時發生問題，則最重要的 DSDT 表格可用改良版本來替換。在此情況中，會忽略 BIOS 的錯誤 DSDT。程序在 [第 16.4 節「疑難排解」](#) [169頁] 中描述。

在核心組態中，有個啟用 ACPI 除錯訊息的切換。如果已編譯並安裝好一個具有 ACPI 除錯能力的核心，則專家將能取得詳細資訊支援，以便搜尋錯誤。

如果您曾遇到 BIOS 問題或硬體問題，建議您聯絡製造商。尤其是哪些一直未提供 Linux 支援的製造商，更應該出面解決這些問題。唯有讓製造商得知他們有不少使用 Linux 的客戶，他們才會嚴肅地處理這些問題。

如需更多資訊

- <http://www.cpqlinux.com/acpi-howto.html> (詳細的 ACPI HOWTO，內含 DSDT 修補程式)
- <http://www.intel.com/technology/iapc/acpi/index.htm> (進階組態與電源介面)
- <http://www.lesswatts.org/projects/acpi/> (Sourceforge 的 ACPI4Linux 計劃)
- <http://www.poupinou.org/acpi/> (Bruno Ducrot 的 DSDT 修補程式)

16.3 硬碟的休眠

在 Linux 中，可在不需使用硬碟時，讓硬碟完全進入睡眠狀態，或是讓硬碟以更省電、更安靜的方式來運作。在目前的筆記型電腦中，您不用手動關閉硬碟，因為它們會在不用的時候自動進入省電操作模式。不過，如果您想最大化省電效果，可嘗試下列幾種方法。

`hdparm` 應用程式能修改各種硬碟設定。`-y` 選項能立即將硬碟切換到待命模式。`-Y` 能讓她進入睡眠。`hdparm -s x` 則會讓硬碟閒置一段時期後關閉。如下所示取代 `x`：0 會停用此機制，使得硬碟持續執行。1 到 240 的值將乘以 5 秒。241 到 251 的值則是以 30 分鐘為一個單位，依序從 30 分鐘的閒置到 11 倍的 330 分鐘的閒置。

可以使用 `-B` 選項來控制硬碟內部的省電選項。可從 0 到 255 中選取一個值，以最大化省電效果或最大化電力輸出。其結果視硬碟用途而定，難以評估。如果要減少硬碟噪音，請使用 `-M` 選項。從 128 到 254 中選取一個值，以決定要安靜或快速。

通常，要讓硬碟進入睡眠不是件容易的事。在 Linux 中，會有多個程序寫入硬碟中，因而重複喚醒硬碟。因此，有必要去瞭解 Linux 如何處理那些要寫入硬碟的資料。首先，會將所有資料在 RAM 中做緩衝處理。`pdflush` 精靈會監控此緩衝區。當資料到達特定的時間限制，或當緩衝區已填滿至某一程度時，會將緩衝區的內容注入硬碟。緩衝區的大小則動態地由記憶體地的大小及系統負載來決定。依預設，`pdflush` 會設成較短的間隔，以最大化資料的完整性。`pdflush` 每 5 秒檢查一次緩衝區，並將資料寫入硬碟。以下變數較為重要：

```
/proc/sys/vm/dirty_writeback_centisecs
```

包含 `pdflush` 線串喚醒前的延遲時間 (以百分之一秒計)。

```
/proc/sys/vm/dirty_expire_centisecs
```

定義最遲應在其後將改動分頁寫出的時間框架。預設為 3000，即 30 秒。

```
/proc/sys/vm/dirty_background_ratio
```

`pdflush` 開始寫入改動分頁之前改動分頁的最大百分比。預設為 5%。

```
/proc/sys/vm/dirty_ratio
```

改動分頁超過此總記憶體的百分比時，系統會強制執执行程序以在其時間片段內將改動寫入緩衝區，而不是繼續寫入改動。

警告: 損害資料完整性

變更 `pdflush` 精靈的設定會損害資料的完整性。

除了這些程序之外，`ReiserFS` 與 `Ext3` 等記錄檔案系統不經由 `pdflush` 即將中繼資料寫入硬碟，也會使得硬碟無法減速。為了避免這類情形，正在開發適用於行動設備的核心延伸程式。請參閱 `/usr/src/linux/Documentation/laptop-mode.txt` 以取得詳細資訊。

另一個重要因素在於啟動程式的行為方式。例如，好的編輯器會定期為修改中的檔案，將隱藏備份檔寫入硬碟，因而喚醒硬碟。停用這類功能可能會傷害資料的完整性。

與此相關，`postfix` 郵件精靈會使用 `POSTFIX_LAPTOP` 變數。如果將此變數設為 `yes`，`postfix` 會減少存取硬碟的頻率。

在 `SUSE Linux Enterprise Desktop` 中，這些技術是由 `laptop-mode-tools` 控制的。

16.4 疑難排解

所有錯誤訊息和警示會記錄在檔案 `/var/log/messages` 中。如果您找不到所需資訊，請使用 `/etc/sysconfig/powersave/common` 檔案中的 `DEBUG` 提高 `powersave` 訊息的資料詳細程度。將變數值增加到 7 或甚至 15，並重新啟動精靈。`/var/log/messages` 中更詳細的錯誤訊息可協助您找出錯誤。以下幾節涵蓋了關於省電與不同睡眠模式的最常見問題。

16.4.1 以硬體支援啟用 **ACPI**，但沒有作用

如果您遭遇到 **ACPI** 的問題，請使用指令 `dmesg|grep -i acpi` 來搜尋 **ACPI** 特定訊息的 `dmesg` 輸出。必須更新 **BIOS** 以解決問題。請到您筆記型電腦製造商的首頁，尋找更新的 **BIOS** 版本並安裝。請詢問製造商以符合最新 **ACPI** 規格。在 **BIOS** 更新後，如果錯誤持續發生，請繼續以更新的 **DSDT** 取代您 **BIOS** 中的錯誤 **DSDT** 表格：

- 1 從 <http://acpi.sourceforge.net/dsdt/index.php> 下載您系統的 DSDT。檢查檔案是否已解壓縮，並以所示副檔名 .aml (ACPI 機器語言) 編譯。如果是此狀況，請繼續步驟 3。
- 2 如果下載的表格副檔名為 .asl (ACPI 原始語言)，請以 `iasl` (pmtools 套件) 編譯。輸入 `iasl -sa file.asl` 指令。
- 3 將檔案 DSDT.aml 複製到任何位置 (建議 /etc/DSDT.aml)。編輯 /etc/sysconfig/kernel 並將路徑與 DSDT 檔案搭配。啟動 mkinitrd (套件 mkinitrd)。只要您安裝核心並使用 mkinitrd 來建立 initrd，當系統啟動時，修改的 DSDT 便會整合並載入。

16.4.2 CPU 頻率沒有作用

請參考核心來源 (kernel-source) 以查看是否支援您的處理器。您需要特殊核心模組或模組選項以啟用 CPU 頻率控制。此資訊可在 /usr/src/linux/Documentation/cpu-freq/* 中找到。

16.4.3 暫停和待命沒有作用

由於 DSDT 實做 (BIOS) 的錯誤，ACPI 系統在暫停或待命時可能會有問題。若這樣的話，請更新 BIOS。

當系統嘗試卸載錯誤模組時，系統會停止或不觸發暫停事件。如果您不卸載模組或停止服務 (導致無法成功暫停)，也可能發生此狀況。在兩種情況下，都請嘗試辨識無法啟動休眠模式的錯誤模組。記錄檔案 /var/log/pm-suspend.log 包含了有關目前的狀況及錯誤之所在的詳細資訊。修改 /usr/lib/pm-utils/defaults 中的 SUSPEND_MODULES 變數，在暫停或待機之前先卸載有問題的模組。

請參閱 <http://www.opensuse.org/Pm-utils> 與 <http://www.opensuse.org/S2ram>，以獲取更多關於如何修改暫停與恢復程序的詳細資訊。

16.5 如需更多資訊

- <http://www.opensuse.org/S2ram> — 如何使暫停寫入到 RAM 運作
- <http://www.opensuse.org/Pm-utils> — 如何修改一般暫停架構

使用 Tablet PC

SUSE® Linux Enterprise Desktop 具有對 Tablet PC 的支援。在下列資訊中，瞭解如何安裝並設定 Tablet PC，探索一些可接受數位筆輸入的有用的 Linux* 應用程式。

支援下列 Tablet PC：

- 具有一系列 Wacom 設備 (例如 ACER TM C30x 系列、Fujitsu Lifebook T 系列 (T30xx/T40xx/T50xx)、Gateway C-140X/E-295C、HP Compaq TC1100/TC4200/TC4400、2710p/2730p、IBM/Lenovo X41t/X61t、LG LT20、Motion M1200/M1400、OQO 02、Panasonic Toughbook CF-18、Toshiba Portege/Tecra M 系列以及 Satellite R15/R20) 的 Tablet PC。
- 具有 Wacom USB 設備 (例如 ASUS R1E/R1F、Gateway C-120X/E-155C、HP Pavilion tx2000/tx2100/tx2500 系列) 的 Tablet PC。
- 具有 FinePoint 設備 (例如 Gateway C210X/M280E/CX2724、HP Compaq TC1000) 的 Tablet PC。
- 具有觸控式螢幕設備 (例如 Asus R2H、Clevo TN120R、Fujitsu Siemens Computers P-Series、LG C1、Samsung Q1/Q1-Ultra) 的 Tablet PC。

安裝 Tablet PC 套件並正確設定數位裝置後，可以使用筆 (亦稱手寫筆) 的輸入來執行下列動作或應用程式：

- 登入 KDM 或 GDM
- 解除 KDE 和 GNOME 桌面螢幕的鎖定
- 可由其他指標設備 (例如滑鼠或觸控板) 觸發的動作，例如，在螢幕上移動游標；啟用應用程式；關閉、移動視窗以及調整視窗大小；切換視窗焦點；拖放物件
- 在 X Window System 的應用程式中使用筆勢辨識
- 使用 GIMP 繪圖
- 使用 Jarnal、Xournal 等應用程式做記筆記或素描，或者使用 Dasher 編輯大量文字

注：安裝所需的鍵盤或滑鼠

在安裝 SUSE Linux Enterprise Desktop 期間，不能將筆用做輸入設備。如果您的 Tablet PC 沒有內建的鍵盤或觸控板，請將外部鍵盤或滑鼠連接至 Tablet PC，以安裝系統。

17.1 安裝 Tablet PC 套件

Tablet PC 所需的套件包含在 TabletPC 安裝模式中 — 如果安裝時選取了該選項，以下套件應已安裝在您的系統中：

- `cellwriter`：字元手寫輸入面板
- `jarnal`：基於 Java 的筆記應用程式
- `wacom-kmp(-default)`：具有 USB Wacom 設備之 Tablet PC 的核心驅動程式
- `xournal`：用於記筆記和素描的應用程式
- `xstroke`：適用於 X Window System 的筆勢辨識程式

- `xvkbd`: 適用於 X Window System 的虛擬鍵盤
- `x11-input-fujitsu`: 適用於 Fujitsu P-Series 手寫板的 X 輸入模組
- `x11-input-evtouch`: 具有觸控式螢幕之某些 Tablet PC 的 X 輸入模組
- `x11-input-wacom`: 適用於 Wacom 手寫板的 X 輸入模組
- `x11-input-wacom-tools`: 供 Wacom 手寫板使用的組態、診斷和文件庫

如果未安裝這些套件，則需要透過指令行手動安裝所需套件，或選取 TabletPC 模式在 YaST 中進行安裝。

17.2 設定手寫板設備

您可在安裝期間透過變更「**圖形卡**」選項在「**硬體組態**」螢幕中設定 Tablet PC (不包括具有觸控式螢幕的 Tablet PC)。或者，您可在安裝結束後的任意時間設定 (內部或外部) 手寫板設備。

- 1 透過指令行或按 **Alt + F2** 並輸入 `sax2` 來啟動 SaX2。
- 2 如果使用 Wacom 或 Finepoint 設備，請按一下「**手寫板**」以顯示「**手寫板內容**」。

如果您使用的是具有觸控式螢幕的 Tablet PC，則請按一下「**觸控式螢幕**」。

- 3 從右側的清單中選取「*TabletPC*」做為廠商，並選取手寫板名稱，然後核取「**啟用此手寫板**」。

如果未列出您的機器並且您確定擁有 Wacom 設備，請選取「*Wacom ISDV4 Tablet PC (SERIAL)*」或「*Wacom ISDV4 Tablet PC (USB)*」。

- 4 切換至「**電子筆**」索引標籤並確定已啟用以下選項：「**新增筆**」與「**新增橡皮擦**」。如果您的 Tablet PC 具有觸控式螢幕，還請啟用「**新增觸控**」。
- 5 按一下「**確定**」儲存變更。

完成 X Window System 組態設定後，請透過登出以重新啟動 X 伺服器。或者，請離開使用者介面並在虛擬主控台中執行 `init 3 && init 5`。

設定手寫板設備後，即可使用筆 (或者，依據您的 Tablet PC、您的指紋) 用做輸入設備。

17.3 使用虛擬鍵盤

若要登入 KDE 或 GNOME 桌面或解除螢幕鎖定，可以按普通方式或透過虛擬鍵盤 `xvkbd` (顯示於登入欄位下方) 輸入使用者名稱和密碼。若要設定鍵盤或存取整合的說明，請按一下左下角的「`xvkbd`」欄位，以開啓 `xvkbd` 主功能表。

如果輸入不可見 (或者未傳送到所需視窗)，請在 `xvkbd` 中按一下「焦點」鍵以重新導向至焦點，然後按一下將取得鍵盤事件的視窗。

圖形 17.1 `xvkbd` 虛擬鍵盤



如果登入後要使用 `xvkbd`，請從主功能表啟動，或者在外圍程序中使用 `xvkbd`。

17.4 旋轉顯示器

請使用 `KRandRTray` (KDE) 或 `gnome-display-properties` (GNOME) 以手動方式即時旋轉顯示畫面或調整顯示畫面大小。`KRandRTray` 與 `gnome-display-properties` 都是用於 X 伺服器之 `RANDR` 延伸的 Applet。

請從主功能表啟動 `KRandRTray` 或 `gnome-display-properties`，也可以輸入 `krandrtray` 或 `gnome-display-properties` 從外圍程序啟動 Applet。啟動各自的 Applet 後，Applet 圖示通常會新增至系統匣。如果 `gnome-display-`

properties 圖示未自動出現在系統匣中，請確定「顯示器解析度設定」對話方塊中的「在面板中顯示畫面」已啟用。

若要使用 KRandRTray 旋轉顯示畫面，請在圖示上按一下滑鼠右鍵並選取「設定顯示」。從組態對話方塊中選取所需的方向。

若要使用 gnome-display-properties 旋轉顯示畫面，請在圖示上按一下滑鼠右鍵並選取所需的方向。您的顯示器會立即旋轉到新的方向。圖形手寫板的方向也會隨之發生變化，因此它仍可以正確解譯出筆的移動。

如果在變更桌面方向時出現問題，如需詳細資訊，請參閱第 17.7 節「疑難排解」[181 頁]。

17.5 使用筆勢辨識

SUSE Linux Enterprise Desktop 包括適用於筆勢辨識的 CellWriter 和 xstroke。兩個應用程式均接受將筆或其他指向設備產生的筆劃當做 X Window System 上應用程式的輸入。

17.5.1 使用 CellWriter

透過 CellWriter，您可以將字元寫入儲存格的網格 — 寫入後立即會以字元為依據進行辨識。完成寫入後，可以將該輸入傳送至目前焦點所在的應用程式。使用 CellWriter 進行筆勢辨識之前，需要訓練應用程式以辨識您的手寫：您需要訓練特定按鍵對應中的每個字元 (未經訓練的字元不會啟用，因此無法使用)。

過程 17.1 訓練 CellWriter

- 1 從主功能表啟動 CellWriter 或在指令行輸入 `cellWriter`。首次啟動時，CellWriter 會自動以訓練模式啟動。在訓練模式中，將顯示目前所選按鍵對應的字元集。
- 2 將要用於字元的筆勢輸入到相關的字元儲存格中。首次輸入後，背景會變為白色，而字元本身則顯示為淺灰色。多次重複此筆勢直到字元顏色變為黑色。未經訓練的字元顯示在淺灰或棕色背景上 (視桌面色彩配置而定)。
- 3 重複此步驟直到針對所有所需字元完成 CellWriter 訓練。

- 4 若要為另一種語言訓練 CellWriter，請按一下「設定」按鈕並從「語言」索引標籤選取語言。「關閉」組態對話方塊。按一下「訓練」按鈕並在「CellWriter」視窗之右下角的下拉式方塊中選取按鍵對應。現在重複新按鍵對應的訓練。
- 5 完成按鍵對應訓練後，按一下「訓練」按鈕以切換至一般模式。

在一般模式下，CellWriter 視窗會顯示要在其中輸入筆勢的一對空白儲存格。按一下「Enter」按鈕後，字元才會傳送至其他應用程式，所以在將字元做為輸入之前您可以進行更正或刪除。信任度低的已辨識字元將以反白顯示。若要更正輸入，請在儲存格上按一下滑鼠右鍵，隨即會出現內容功能表。若要刪除字元，請使用筆之橡皮擦，或使用滑鼠中鍵清除儲存格。完成在 CellWriter 中的輸入後，請按一下應用程式視窗定義將接收輸入的應用程式。然後按一下「Enter」將輸入傳送至應用程式。

圖形 17.2 用 CellWriter 進行筆勢辨識



按一下「按鍵」按鈕，將取得可取代手寫辨識的虛擬鍵盤。

若要隱藏 CellWriter，請關閉 CellWriter 視窗。應用程式即會在系統匣中以一個圖示顯示。若要再次顯示輸入視窗，請按一下系統匣中的圖示。

17.5.2 使用 Xstroke

透過 xstroke，您可以將筆或其他指向設備產生的筆劃當做 X Window System 上應用程式的輸入。xstroke 字母系統是一種單筆劃字母系統，類似於 Graffiti* 字母系統。啟動後，xstroke 會將輸入傳送到目前聚焦的視窗。

- 1 可以從主功能表啟動 xstroke，或者在外圍程序中使用 xstroke 來啟動。這會在您的系統匣上新增一個鉛筆圖示。

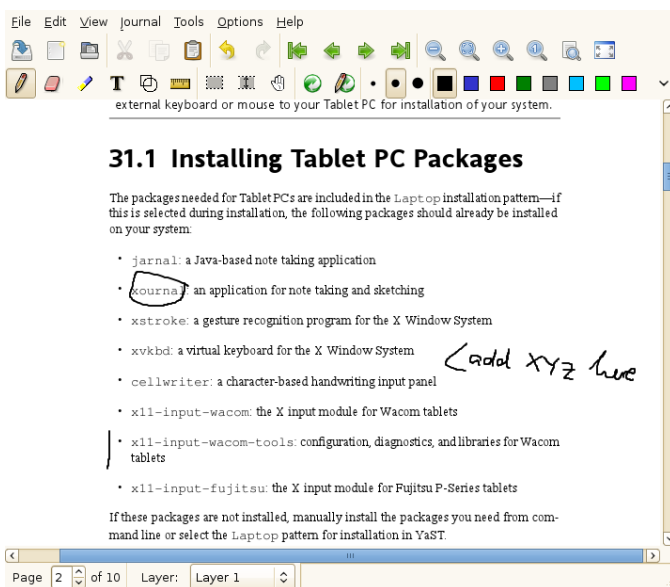
- 2 啟動您要使用筆為其建立文字輸入的應用程式 (例如終端機視窗、文字編輯器或 OpenOffice.org Writer)。
- 3 若要啟用筆劃辨識模式，按一下鉛筆圖示即可。
- 4 使用筆或其他指標設備執行圖形手寫板上的某些筆勢。xstroke 可以擷取筆勢，並將其轉換成文字，顯示在焦點所在的應用程式視窗中。
- 5 若要將焦點切換到不同的視窗，請用筆在所需視窗上按一下並按住片刻 (或使用桌面控制中心內定義的鍵盤捷徑)。
- 6 若要停用筆勢辨識模式，請再次按一下鉛筆圖示。

17.6 使用筆來做筆記和素描

若要用筆繪圖，可以使用專業的圖形編輯器 (例如 GIMP)，或者嘗試下列記筆記應用程式之一：Xournal 或 Jarnal。憑藉 Xournal 和 Jarnal，您可以用筆來做筆記、繪圖或者為 PDF 檔案加入備註。做為適用於多種平台之 Java 為基礎的應用程式，Jarnal 還提供基本的協作功能。若需更多資訊，請參考 <http://www.dklevine.com/general/software/tcl000/jarnal-net.htm>。在儲存內容時，Jarnal 會以歸檔格式 (*.jaj) 儲存資料，該歸檔中另含一個 SVG 格式的檔案。

可以從主功能表啟動 Jarnal 或 Xournal，或者在外圍程序中輸入 jarnal 或 xournal 來啟動。例如，要在 Xournal 中為 PDF 檔案加入備註，可以選取「檔案」>「註釋 PDF」，然後從檔案系統中開啓 PDF 檔案。使用筆或其他指向設備註釋 PDF 檔案，然後選取「檔案」>「列印到 PDF」以儲存變更。

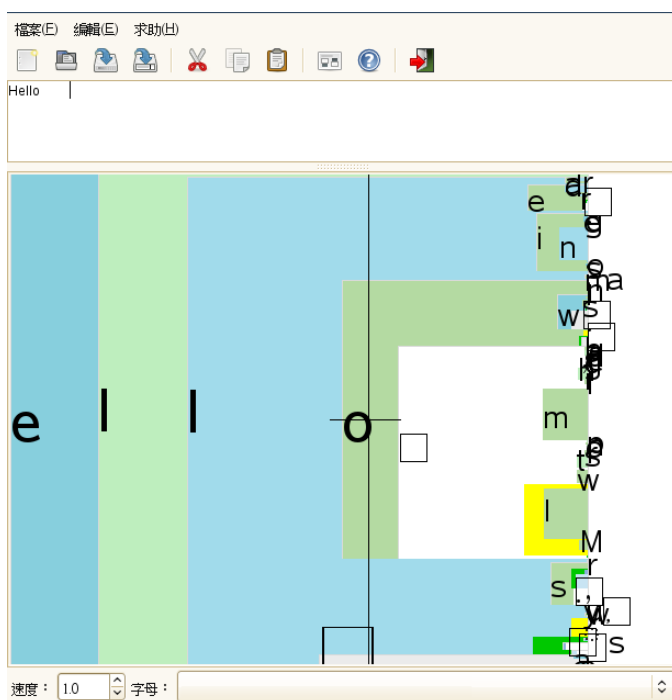
圖形 17.3 使用 *Xournal* 註釋 PDF



Dasher 是另一個有用的應用程式。它專門用於鍵盤輸入不適用或不可用的場合。接受少量的訓練後，僅憑一支筆 (或其他輸入設備 — 甚至是視線追蹤器所驅動的設備) 即可快速輸入大量的文字。

可以從主功能表啟動 Dasher，也可以在外圍程序中使用 `dasher` 來啟動。將筆向一個方向移動，應用程式將開始在右側縮放成字母。隨後，根據十字型游標中心產生的字母生成或聯想文字，並在視窗的上方輸出。若要停止或開始書寫，請用筆按一下顯示器。在視察底部修改成字速度。

圖形 17.4 使用 Dasher 編輯文字



Dasher 的概念適用於許多語言。如需詳細資訊，請造訪 Dasher 網站，其中提供了綜合性的文件、展示和訓練文字。其網址為 <http://www.inference.phy.cam.ac.uk/dasher/>

17.7 疑難排解

登入畫面上不顯示虛擬鍵盤

有時，登入畫面上不顯示虛擬鍵盤。若要解決此問題，請重新啟動 X 伺服器，方法是按 **Ctrl + Alt + <—**，或者在 Tablet PC 上按相應的鍵 (如果您使用的是不帶整合鍵盤的平板型號)。如果虛擬鍵盤仍未顯示，請將外部鍵盤連接到平板型號 Tablet PC，再使用硬體鍵盤登入。

Wacom 圖形手寫板的方向不會變更

在外圍程序內部使用 `xrandr` 指令即可變更顯示器的方向。輸入 `xrandr --help` 可檢視可用選項。若要同步變更圖形手寫板的方向，則需要依照下面的說明修改指令：

- 對於正常方向 (0° 旋轉)：

```
xrandr --output LVDS ---rotate normal && xsetwacom set "Mouse[7]" Rotate NONE
```

- 對於 90° 旋轉 (順時針縱向)：

```
xrandr --output LVDS ---rotate right && xsetwacom set "Mouse[7]" Rotate CW
```

- 對於 180° 旋轉 (橫向)：

```
xrandr --output LVDS --rotate inverted && xsetwacom set "Mouse[7]" Rotate HALF
```

- 對於 270° 旋轉 (逆時針縱向)：

```
xrandr --output LVDS --rotate left && xsetwacom set "Mouse[7]" Rotate CCW
```

請注意，上面的指令依 `/etc/X11/xorg.conf` 組態檔中的內容而定。如果已按照第 17.2 節「設定手寫板設備」[175頁] 中所述使用 `SaX2` 設定了設備，指令應按照其所寫內容起作用。如果您在 `xorg.conf` 中手動變更了手寫板筆輸入設備的「識別碼」，請將 `"Mouse[7]"` 用新的「識別碼」取代。如果您的 Wacom 設備支援觸摸 (可用手指在手寫板上移動游標)，則還需要旋轉觸控式設備。

17.8 如需更多資訊

此處提到的一些應用程式未提供配套的線上說明，但您可以在安裝系統的 `/usr/share/doc/package/packageName` 中或以下網站中，找到關於其用法與組態的一些有用資訊：

- 如需 Xournal 手冊，請參考 <http://xournal.sourceforge.net/manual.html>
- <http://www.dklevine.com/general/software/tc1000/jarnal.htm#documentation> 上提供了 Jarnal 文件
- <http://davesource.com/Projects/xstroke/xstroke.txt> 上提供了 xstroke 手冊頁
- Linux Wacom 網站 <http://linuxwacom.sourceforge.net/index.php/howto/x11> 上提供了關於設定 X 的 HOWTO 資訊
- 從 <http://www.inference.phy.cam.ac.uk/dasher/> 可以找到詳細介紹 Dasher 專案之資訊的網站
- 如需有關 CellWriter 的詳細資訊和文件，請造訪 <http://risujin.org/cellwriter/>
- 如需有關 gnome-display-properties 的資訊，請造訪 <http://en.opensuse.org/GNOME/Multiscreen>。

第 IV. 服務 部分

基本網路

Linux 提供所有必要的網路工具及功能，以整合到所有類型的網路結構。自定的 Linux 通訊協定、TCP/IP，具有各種服務與特殊功能，將會在這裏討論。使用網路卡、數據機或其他設備進行網路存取可以使用 YaST 來設定。也可使用手動方式來設定組態。本章僅討論基本機制及相關的網路組態檔案。

Linux 及其他 Unix 作業系統使用 TCP/IP 通訊協定。它不是單一網路通訊協定，而是能夠提供各種服務的網路通訊協定家族的一員。**表格 18.1 「TCP/IP 通訊協定家族中的數種通訊協定」** [187頁] 中列示的通訊協定，提供透過 TCP/IP 在兩個機器之間交換資料的用途。由 TCP/IP、全球網路所結合而成的網路，也稱為「網際網路」。

RFC 代表 *要求建議 (Request for Comments)*。RFC 是描述作業系統及其應用程式的各種網際網路通訊協定和執行程序的文件。RFC 文件描述網際網路通訊協定的設定。若要擴展您對於任何通訊協定的知識，請參閱適當的 RFC 文件。

<http://www.ietf.org/rfc.html> 上提供了這些文件。

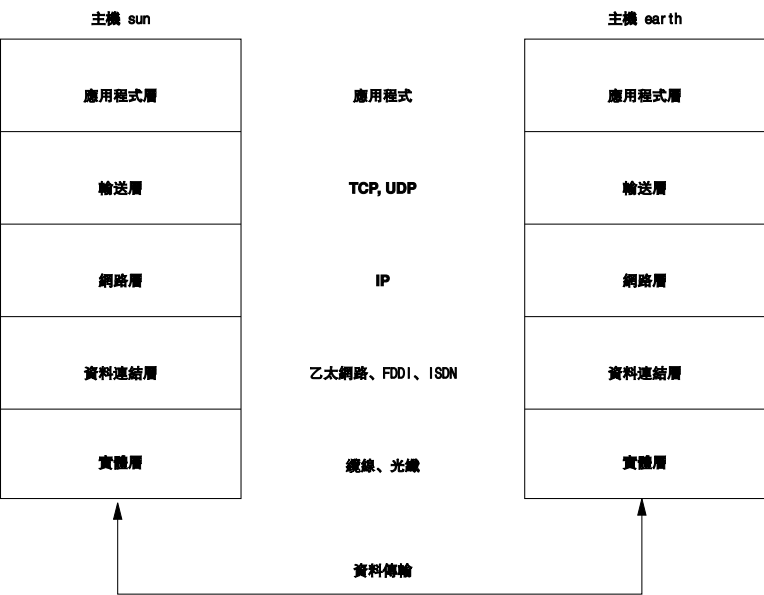
表格 18.1 TCP/IP 通訊協定家族中的數種通訊協定

協定	描述
TCP	傳輸控制通訊協定：連線導向的安全通訊協定。傳輸的資料首先由應用程式當做資料流傳送出去，然後再由作業系統轉換為適當格式。資料送達目的地主機的相關應用程式時，使用的仍是最初傳送的原始資料流格式。TCP 判斷在傳輸期間是否遺失了資料，或者資料的順序是否混亂。只要是資料順序很重要的地方，就會執行 TCP。

協定	描述
UDP	使用者資料包通訊協定：無連線、不安全的通訊協定。要傳送的資料以應用程式產生的封包形式加以傳送。不會保證資料抵達接收者時的順序，而且可能會發生資料遺失的情況。UDP 適用以記錄為導向的應用程式。它的特點是延遲時間比 TCP 短。
ICMP	網際網路控制訊息通訊協定：基本上，這不是適用一般使用者的通訊協定，而是發佈錯誤報表的特殊控制通訊協定，能夠控制參與 TCP/IP 資料傳送之機器的行為。此外，它還提供特殊的回音模式，可以使用 ping 程式檢視。
IGMP	網際網路群組管理通訊協定：此通訊協定在實做 IP 多點廣播時控制機器行為。

如 **圖形 18.1 「TCP/IP 的簡化層模型」** [189頁] 中所顯示，資料交換發生在不同層。實際的網路層是透過 IP (網際網路通訊協定，Internet Protocol) 進行不安全的資料傳輸。在 IP 的上方，TCP (傳輸控制通訊協定，Transmission Control Protocol) 可以保證資料傳輸某種程度的安全性。IP 層是由基本的硬體相依通訊協定所支援，例如乙太網路。

圖形 18.1 TCP/IP 的簡化層模型

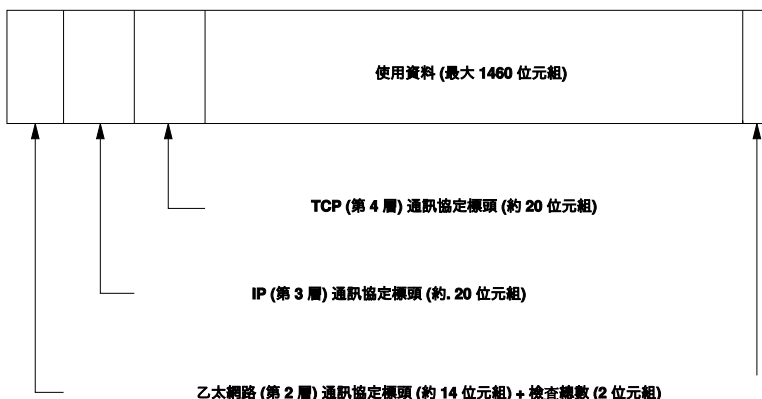


圖表提供每層的一或兩個範例。層的順序是依據**抽象階層 (abstraction level)**。最低層非常靠近硬體。不過，最上層對硬體而言幾乎是完全抽象的。每層都有自己的特殊功能。這些特殊功能通常隱含於其描述中。資料連結及實體層代表使用的實體網路 (如乙太網路)。

幾乎所有的硬體通訊協定都是採用封包導向模式。要傳輸的資料是裝在**封包**中，因為無法一次傳送它。**TCP/IP** 封包的大小上限約為**64KB**。封包一般而言較小，因為可能受限於網路硬體的關係。乙太網路上資料封包的最大上限約為**1500** 個位元組。在乙太網路上傳送資料時，**TCP/IP** 封包的大小受限於此數量。如果傳送更多資料，則需要由作業系統傳送更多資料封包。

因為每層有自己指定的功能，關於每層的其他資訊必須儲存於資料封包中。這些資訊放在封包的「**標頭**」中。每層皆在產生的封包前端附加小的資料區塊，稱為通訊協定標頭。在乙太網路纜線上傳送的**TCP/IP** 資料封包範例，可參閱在**圖形 18.2「TCP/IP 乙太網路封包」** [190頁] 中的說明。proofsum 位於封包結尾，不在開頭處。這樣可幫助網路硬體簡化程序。

圖形 18.2 TCP/IP 乙太網路封包



當應用程式在網路上傳送資料時，資料會經過每一層，除實體層外，全部在 Linux 核心執行。每層都負責準備資料使其能夠傳送到下一層。最底層最後要負責傳送資料。接收到資料時則反轉執行整個程序。就如同洋蔥的層級一般，在每層中，會從已傳輸的資料上移除通訊協定標頭。最後，傳輸層負責讓目的地的應用程式可以使用資料。以這種方式，每層僅直接與上下兩層通訊。對於應用程式而言，無論資料是透過 100 MBit/s FDDI 網路或 56-Kbit/s 數據線進行傳輸，都沒有關係。同樣地，對於資料線而言，只要封包的格式正確，無論傳送的是哪種類型的資料也是無關的。

18.1 IP 位址與路由

在此節中的討論僅限於 IPv4 網路。如需有關 IPv6 通訊協定 (IPv4 的後繼者) 的資訊，請參閱第 18.2 節「IPv6—下一代的網際網路」[193頁]。

18.1.1 IP 位址

網際網路上的每台電腦都有唯一的 32 位元位址。這些 32 位元 (或 4 位元組) 一般所寫入的格式，如 範例 18.1 「寫入 IP 位址」[190頁] 中的第二列所述。

範例 18.1 寫入 IP 位址

```
IP Address (binary): 11000000 10101000 00000000 00010100
IP Address (decimal): 192. 168. 0. 20
```

採用十進位格式，四位元組以十進位數字系統撰寫，以句號分隔。IP 位址是指定給主機或網路介面。其他地方無法使用。此規則有例外狀況，但是與下文中並無關聯。

IP 位址中的點表示階層系統。直到 1990 年代，IP 位址仍嚴格地以類別加以分類。然而，此系統已證明太過死板而已停止採用。現在，則是使用無類別路由 (classless routing)，即 CIDR (無類別網域間路由，classless interdomain routing)。

18.1.2 網路遮罩與路由

網路遮罩是用來定義子網路的位址範圍。如果兩台主機位於相同的子網路遮罩，他們可以直接相互連結，如果他們不在同一個子網路遮罩，則需要閘道位址，以處理子網路和其他網路的通訊。若要檢查兩個 IP 位址是否位於同一子網路，只要使用網路遮罩「AND」兩個位址。如果結果相同，兩個 IP 位址位於同一個網路。如果不同，遠端的 IP 位址，即為遠端介面，只能透過閘道來通訊。

若要瞭解網路遮罩如何作用，請參閱範例 18.2「連結 IP 位址到網路遮罩」[191頁]。網路遮罩由 32 位元組成，可辨認 IP 位址屬於哪個網路。這些位元為 1 標示 IP 位址中的對應位元，即表示為同屬一個網路。所有位元為 0 標示位元在子網路內。這表示愈多位元為 1，子網路就愈小。因為網路遮罩永遠由多個連續的 1 組成，也可以計算網路遮罩內的位元數。範例 18.2「連結 IP 位址到網路遮罩」[191頁]中，第一個 24 位元的網路也可寫成 192.168.0.0/24。

範例 18.2 連結 IP 位址到網路遮罩

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
-----
Result of the link:        11000000 10101000 00000000 00000000
In the decimal system:      192.      168.      0.      0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
-----
Result of the link:        11010101 10111111 00001111 00000000
In the decimal system:      213.      95.      15.      0
```

舉另外一個例子：使用相同乙太網路纜線連接的所有機器，通常位於同一個子網路中，而且可以直接存取。即使以交換器或橋接器實際分配子網路時，仍然可以直接連接這些主機。

位於本地子網路外的 IP 位址只能在設定目標網路的閘道時，才能與本地通訊。在大部分的狀況下，只能有一個閘道來處理所有對外的通訊。但是，您也可以為不同的子網路，設定多個閘道。

如果已經設定閘道，所有的外部 IP 封包會傳送到適當的閘道。然後此閘道會試圖以同樣方式傳送封包--主機對主機—直到連結到目標主機或封包 TTL (持續時間) 過期。

表格 18.2 特定位址

位址類型	描述
基本的網路位址	這是網路遮罩「及」網路中的任何位址，如Result下的 範例 18.2「連結 IP 位址到網路遮罩」 [191頁] 所顯示。此位址不能指定給任何主機。
廣播位址	基本來說，即為「存取此子網路的所有主機」。若要產生此位址，網路遮罩會以二進位格式反轉，連結到具有邏輯 OR 的基本網路位址。因此以上範例會得到 192.168.0.255。此位址無法指派給任何主機。
本地主機	位址 127.0.0.1 是指定到每個主機上的「迴路設備」(loopback device)。使用此位址可以設定到您自己機器的連接。

因為 IP 位址在全世界必須是唯一的，您不能只選取隨機位址。如果要設立私人 IP 結構的網路，有三種位址網域可以使用。這些將無法從其他網際網路取得連結，因為他們無法透過網路傳送。這些位址網域在 RFC 1597 指定並列於 **表格 18.3「私人 IP 位址網域」** [192頁] 中。

表格 18.3 私人 IP 位址網域

網路/網路遮罩	網域
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x – 172.31.x.x
192.168.0.0/255.255.0.0	192.168.x.x

18.2 IPv6—下一代的網際網路

由於全球資訊網 (World Wide Web, WWW) 的出現，過去十五年內，在網際網路上透過 TCP/IP 進行通訊的電腦數目暴增。自從 CERN 的 Tim Berners-Lee (<http://public.web.cern.ch>) 於 1990 年發明 WWW 以來，網際網路主機的數量從幾千台成長為幾百萬台。

如前面所述，IPv4 位址僅由 32 個位元組成。而且，損失了一些 IP 位址——由於組織網路的方式，使得這些 IP 位址無法使用。您的子網路中可用的位址數目是位元數的平方減 2。例如，子網路有 2 個、6 個或 14 個位址可用。例如，如果要連接 128 個主機到網際網路，則子網路需要 256 個 IP 位址，但是其中只有 254 個可用，因為子網路結構本身需要用掉兩個 IP 位址：廣播與基本網路位址。

在目前的 IPv4 通訊協定之下，DHCP 或 NAT (網路位址轉譯，Network Address Translation) 是典型的機制，可用來避免位址可能不足的問題。搭配保持私人和公用位址空間分開的方式，能夠減輕短少的情形。其中產生的問題是在於其組態，設定麻煩且難於維護。若要在 IPv4 網路中設定主機，需要一些位址項目，如主機自己的 IP 位址、子網路遮罩、閘道位址，可能還需要名稱伺服器位址。您必須知道所有這些項目，且無法從其他地方取得。

透過 IPv6，位址短少及繁複組態的情形應該都成為過去式了。以下小節說明更多 IPv6 改善的部分及它帶來的好處，還有關於從舊通訊協定轉移到新通訊協定的資訊。

18.2.1 優點

新通訊協定帶來的最重要、最顯而易見的改善，是能夠大量擴充可用的位址空間。IPv6 位址是由 128 個位元值組成，而不是傳統的 32 位元。這樣提供了數以千兆的 IP 位址。

然而，IPv6 位址不僅是在長度方面與之前的位址不同；這些位置的內部結構也不同，可能包含有關系統及其所屬網路的更明確資訊。有關 IPv6 的詳細資訊，可以在 **第 18.2.2 節「定址類型與結構」** [195頁] 中找到。

以下是新通訊協定一些其他優勢的清單：

自動設定

IPv6 讓網路能夠「隨插即用」(plug and play)，表示新設定的系統不需經過任何手動設定，即可整合到(區域)網路。新主機使用其自動設定組態機制，

從鄰近的路由器上可用的資訊取得自己的位址，依賴的是稱為「網路芳鄰探查」(Neighbor Discovery, ND)的通訊協定。這個方法不需要管理員的介入，而且不需要維護分配位址的中央伺服器，這是 IPv4 的另一個優勢，因為自動位址分配需要 DHCP 伺服器。

機動性

IPv6 能夠同時將數個位址指定給一個網路介面。這樣可讓使用者輕鬆存取多個網路，有時可與行動電話服務公司提供的國際漫遊服務相比：當您帶您的行動電話出國時，一到對應區域，電話就會自動登入外國服務，因此仍可用同一個號碼聯絡到您，且您也可和本國一樣撥打電話。

安全通訊

使用 IPv4，網路安全性是附加的功能。IPv6 包括 IPSec 為其中一個核心功能，允許系統在安全的通道上進行通訊，避免網際網路上的外人竊聽。

反向相容性

實際上，不可能一次將整個網際網路從 IPv4 切換到 IPv6。因此，很重要的是，兩個通訊協定不僅要能夠共存於網際網路上，也得要能夠共存於一個系統中。這是藉由相容的位址 (IPv4 位址可以輕易轉譯為 IPv6 位址) 以及使用一些通道來確保共存。請參閱第 18.2.3 節「IPv4 與 IPv6 的共存」[198頁]。另外，系統可以仰賴「雙重堆疊 IP」(Dual Stack IP) 技術，同時支援這兩種通訊協定，這表示系統有兩個完全分開的網路堆疊，如此一來，兩種通訊協定版本不會相互干擾。

透過多重廣播自定量身訂做的服務

利用 IPv4，有些服務 (如 SMB) 需要廣播它們的封包到區域網路上的所有主機。IPv6 以更精細的方法，透過「多重廣播」(multicasting)—將一些主機定位為群組的一部分，讓伺服器定址主機 (這與透過「廣播」(broadcasting) 定位所有主機，或透過「單點廣播」(unicasting) 個別定址每個主機的方式不同)。定址為群組的主機，取決於具體的應用程式。例如，有些預先定義的群組可以定址所有名稱伺服器 (「所有名稱伺服器多重廣播群組」) 或所有路由器 (「所有路由器多重廣播群組」)。

18.2.2 定址類型與結構

如上述，目前 IP 通訊協定有兩個重要缺失：IP 位址的短缺越來越嚴重，以及設定網路與管理輪遞表成為複雜而繁重的任務。IPv6 透過擴充位址空間到 128 個位元解決了第一個問題。第二個問題的解決方式則是引入階層位址結構，結合配置網路位址的複雜技術，以及 *multihoming* (指定數個位址到一個設備，可以存取數個網路)。

使用 IPv6 時，瞭解三種不同類型的位址是很有用的：

單點廣播 (Unicast)

這類位址恰好與一個網路介面有關聯。這類位址的封包僅傳送到一個目的地。因此，單點廣播位址用來傳送封包到區域網路或網際網路上的個別主機。

多重廣播 (Multicast)

這類位址與一組網路介面有關聯。這類位址的封包會傳送到屬於該組的所有目的地。多重廣播位址主要由特定網路服務使用，可直接與特定主機群組通訊。

任點廣播 (Anycast)

這類位址與一組介面有關聯。這類位址的封包會根據基礎路由通訊協定的原則，傳送到最靠近傳送者的群組成員。使用任點廣播位址，讓主機更易於找出在指定網路區域中提供特定服務的伺服器。相同類型的所有伺服器擁有一樣的任點廣播位址。只要主機要求服務，它會從最靠近位置的伺服器接收回覆，由路由通訊協定決定。如果此伺服器因為某種原因失敗，通訊協定會自動選取第二個最靠近的伺服器，或是選取第三個伺服器，依此類推。

IPv6 位址由八個四位數欄位組成，每個都代表 16 個位元，以十六進位標記法寫入。這些位址以冒號(:)分隔。指定欄位內的任何前導零位元組可以刪除，但是欄位內或尾端的零不能刪除。另一個慣例是多於四個連續的零位元組可能會摺疊成兩個冒號。然而，每個位址只允許一個::。這類的簡略的標記法，顯示於**範例 18.3 「範例 IPv6 位址」** [195頁] 中，其中三行都是代表相同的位址。

範例 18.3 範例 IPv6 位址

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                               : 10 : 1000 : 1a4
```


IPv6位址的每個部分都有定義的功能。第一個位元組形成字首，指定位址類型。中間的部分是位址的網路部分，但是可能不會使用。位址的尾端形成主機部分。透過 IPv6，在位址尾端的斜線後表示字首的長度，可以定義網路遮罩。位址如 **範例 18.4「指定字首長度的 IPv6 位址」** [196頁] 中所示，包含的資訊是形成位址網路部分的前 64 個位元以及形成其主機部分的最後 64 個位元。換句話說，64 表示網路遮罩從左邊開始填入 64 個 1 位元值。就像 IPv4 一樣，IP 位址使用 AND 結合網路遮罩的值，判斷主機是否位於相同的子網路或另一個子網路。

範例 18.4 指定字首長度的 IPv6 位址

fe80::10:1000:1a4/64

IPv6 知道關於數個字首的預定類型。有一些顯示在 **表格 18.4「各種 IPv6 字首」** [196頁] 中。

表格 18.4 各種 IPv6 字首

字首 (十六進位)	定義
00	IPv4 位址與透過 IPv6 的 IPv4 相容位址。這些位址用來維護與 IPv4 的相容性。其使用仍然要求路由器能夠轉譯 IPv6 封包為 IPv4 封包。數個特殊的位址，如迴路設備的位址，也有此字首。
2 或 3 做為第一個數字	可彙總的全域單點廣播位址。在 IPv4 的情形中，可以指定介面形成部分的特定子網路。目前有下列位址空間：2001::/16 (產品品質位址空間) 與 2002::/16 (6to4 位址空間)。
fe80::/10	連結本地位址。具有這種字首的位址不應該傳送，因此僅能從相同的子網路內連接。
fec0::/10	本地網站位址。這些位址可以傳送，但是僅能在所屬組織的網路內傳送。事實上，它們等同於目前私人網路位址空間的 IPv6 (例如，10.x.x.x)。
ff	這些都是多重廣播位址。

單點廣播位址由三個基本元件組成：

公用拓撲 (Public Topology)

第一個部分 (也包含上述的其中一種字首) 用來透過公用網際網路傳送封包。它包含了有關提供網際網路存取的公司或機構資訊。

網站拓撲 (Site Topology)

第二個部分包含有關傳送封包的目的地子網路的傳送資訊。

介面識別碼 (Interface ID)

第三個部分識別傳送封包的介面。它也允許 MAC 形成部分的位址。這個前提是 MAC 在全球是唯一的，由硬體製造商在設備中編碼固定識別碼，可相當程度地簡化組態程序。事實上，結合前 64 個位址位元形成 EUI-64 記號，加上從 MAC 取得的最後 48 個位元，而其餘的 24 個位元則包含有關記號類型的特殊資訊。如此一來，就可以指定 EUI-64 記號給沒有 MAC 的介面，例如以 PPP 或 ISDN 為基礎的介面。

在此基本結構的最上層，IPv6 會分辨五種不同類型的單點廣播位址：

:: (未指定的)

第一次啟始介面時，如果還未用其他方法判定位址時，主機會使用此位址做為其來源位址。

:::1 (迴路)

迴路設備的位址。

IPv4 相容位址

IPv6 位址是由 IPv4 位址以及由 96 個零位元組成的字首形成的。這類相容性位址用於通道 (請參閱第 18.2.3 節「IPv4 與 IPv6 的共存」[198頁])，允許 IPv4 與 IPv6 主機在純 IPv4 環境中彼此通訊。

對應到 IPv6 的 IPv4 位址

這類位址以 IPv6 標記法指定純 IPv4 位址。

本地位址

有兩種位址類型用於本地：

連結本地

這類位址僅能用於本地子網路。具有此類型來源或目標位址的封包不應該傳送到網際網路或其他子網路。這些位址包含特殊字首 (fe80::/10) 以及網路卡的介面識別碼，加上由空位元組所組成的中間部分。自動設定組態以便與屬於相同子網路中的其他主機通訊時，會使用這類位址。

網站本地

具有這種位址的封包可以傳送到其他子網路，但是不能到更寬廣的網際網路，而必須保留在組織自己的網路內。這類位址用於內部網路，而且等同於 IPv4 所定義的私人位址空間。它們包含特殊字首 (fec0::/10)、介面識別碼，以及指定子網路識別碼的 16 位元欄位。同樣地，其他則是填入空位元組。

因為引入了 IPv6 這種全新的功能，所以每個網路介面通常會取得數個 IP 位址，其優點是可透過相同介面存取數個網路。其中一個網路可以設定完全自動化(使用 MAC 和已知的字首)，只要一啟用 IPv6(使用連結本地位址) 即可連接區域網路上的所有主機。利用形成位址部分的 MAC，全球使用的任何 IP 位址都成為唯一的。位址的唯一變數部分，是指定網站拓撲和公用拓撲，該部分視主機目前正在操作的實際網路而定。

如果主機要在不同的網路之間往返，至少需要兩個位址。其中一個，即主位址，不僅包含了介面識別碼，也包含了其通常所屬之主網路(及其對應字首)的識別碼。主位址是靜態位址，因此它通常不會變更。儘管如此，預定要送到行動主機的所有封包，還是可以傳送到主位址，無論是在主網路或其他外部網路中操作。這可藉由 IPv6 全新功能來達成，如「無狀態自動設定」與「網路芳鄰探查」。除了其主位址外，行動主機也取得一或多個其他的位址，這些位址屬於漫遊的外部網路。這些外部網路稱為 *care-of* 位址。主網路具有封包在外部漫遊時轉寄預定要送到主機的設備。在 IPv6 環境中，這個任務是由主代辦執行的，它會取得所有預定要送到主位址的封包，透過通道轉送它們。另一方面，預定送到 *care-of* 位址的封包會直接傳送到行動主機，不會特別繞行。

18.2.3 IPv4 與 IPv6 的共存

連接網際網路的所有主機從 IPv4 轉移到 IPv6 是一種漸進程序。這兩種通訊協定某些時候會共存。在一個系統上共存，可保證執行兩種通訊協定的「**雙重堆疊**」。但是仍出現一些問題，就是使用 IPv6 的主機如何與 IPv4 主機通訊，以及由 IPv4 結構主導的目前網路如何傳輸 IPv6 封包。最佳的解決方案是提供通道及相容性位址(請參閱第 18.2.2 節「**定址類型與結構**」[195頁])。

IPv6 主機或多或少孤立於(全球) IPv4 網路間，可透過通道通訊：IPv6 封包會被包成 IPv4 封包，在 IPv4 網路中移動。兩個 IPv4 主機之間的連接，稱為「**通道**」。若要完成這個目的，封包必須包含 IPv6 目的地位址(或對應字首)以及通道接收端上遠端主機的 IPv4 位址。基本通道可以根據主機管理員之間的協議「**手動**」設定；這也稱為「**靜態通道**」。

不過，靜態通道的組態及維護通常需要密集勞力，才能使用它們應付每天的通訊需求。因此，IPv6 提供三種不同的「動態通道」方法：

6over4

IPv6 封包會自動封裝成 IPv4 封包，透過能夠多重廣播的 IPv4 網路進行傳送。IPv6 的訣竅是將整個網路(網際網路)視為一個大型的區域網路(LAN)。如此即能自動判定 IPv4 通道的接收端。然而，這個方法不能適當的延伸，而且也因 IP 多重廣播目前在網際網路上並不普遍的事實而受到阻礙。所以，它僅能為啟用多重廣播的小型公司或機構的網路提供解決方案。這個方法的規格詳述於 RFC 2529。

6to4

利用此方法，IPv4 位址會自動從 IPv6 位址產生，使得隔離的 IPv6 主機能夠在 IPv4 網路上通訊。不過，有關這些隔離的 IPv6 主機及網際網路之間的通訊，目前已出現一些問題。該方法詳述於 RFC 3056。

IPv6 通道代理

這個方法仰賴提供 IPv6 主機專屬通道的特殊伺服器。詳述於 RFC 3053。

18.2.4 設定 IPv6

若要設定 IPv6，通常不需要在個別工作站中做任何變更。IPv6 預設會開啟這個選項。您可在章節「Network Configuration」(第 4 章 *Installation with YaST*, ↑*部署指南*)所述的網路設定步驟中的安裝期間停用。若要在安裝的系統上停用或啟用 IPv6，請使用 YaST 「網路設定」模組。在「全域選項」索引標籤上，根據需要核取或取消核取「啟用 IPv6」選項。若要手動啟用 IPv6，請以 root 身份輸入 `modprobe ipv6`。

由於 IPv6 的自動組態概念，網路卡會在連結本地網路中指定一個位址。工作站通常不會進行路由表格管理。工作站可使用「路由器通告通訊協定」，向網路路由器查詢應使用的前置號碼和閘道。可使用 `radvd` 程式來設定 IPv6 路由器。此程式會通知工作站該 IPv6 位址應使用的前置號碼和路由器。或者，也可使用 `zebra/quagga` 自動設定位址和路由的組態。

請參閱 `ifcfg-tunnel (5)` 的手冊頁，瞭解如何使用 `/etc/sysconfig/network` 檔案來設定不同類型的通道。

18.2.5 如需更多資訊

上述綜覽沒有完整地涵蓋 IPv6 主題。如需更深入的探討這種新的通訊協定，請參閱以下線上文件和書籍：

<http://www.ipv6.org/>
所有有關 IPv6 的入門資訊。

<http://www.ipv6day.org>
啟動您 IPv6 網路所需的所有資訊。

<http://www.ipv6-to-standard.org/>
啟用 IPv6 產品的清單。

<http://www.bieringer.de/linux/IPv6/>
在此處可找到 Linux IPv6-HOWTO 和許多與此主題相關的連結。

RFC 2640
有關 IPv6 的基本 RFC。

IPv6 Essentials
描述此主題所有重要面向的書籍，《*IPv6 Essentials*》由 Silivia Hagen 所著 (ISBN 0-596-00125-8)。

18.3 名稱解析

DNS 協助指定 IP 位址給一或多個名稱以及指定名稱給 IP 位址。在 Linux 中，這種轉換通常是由已知為 bind 的特殊類型軟體執行的。處理這個轉換的機器稱為「名稱伺服器」(name server)。名稱組成階層系統，其中每個名稱元件以點分隔。但是，名稱階層與上述的 IP 位址階層無關。

考慮使用完整名稱，如 `jupiter.example.com`，採用 `hostname.domain` 格式。完整名稱，也就是完整網域名稱 (*Fully Qualified Domain Name, FQDN*)，是由主機名稱和網域名稱 (`example.com`) 組成的。後者也包含了「最上層網域」(top level domain)或 TLD (`com`)。

TLD 指定因為過去的緣故變得相當混淆。習慣上，美國使用三個字母的網域名稱。全世界的其他國家，則是使用兩個字母的 ISO 國際代碼為標準。除此之外，

2000 年引入了較長的 TLD，代表特定活動範圍 (例如，`.info`、`.name`、`.museum`)。

在早期的網際網路 (1990 年前)，是使用檔案 `/etc/hosts` 儲存網際網路上所有機器的代表名稱。這種方式，對於連接到網際網路、快速增長的電腦數量層面而言，很快就證實是不切實際的。基於此因素，又開發出分散式的資料庫，以廣泛分散的方式來儲存主機名稱。這種資料庫與名稱伺服器類似，沒有有關網際網路上所有主機的立即可用資料，但是可以分散要求到其他名稱伺服器。

階層的最上層是由「*root* 名稱伺服器」(root name server)所使用。這些 root 名稱伺服器管理最上層網域，且由「網路資訊中心」(Network Information Center, NIC) 負責管理。每個 root 名稱伺服器知道負責指定最上層網域的名稱伺服器。有關最上層網域 NIC 的資訊可從 <http://www.internic.net> 取得。

DNS 的功能不只是解析主機名稱。名稱伺服器也知道哪個主機，即「郵件交換器」(*Mail Exchanger, MX*)，負責接收該網域的電子郵件。

若要让您的機器能夠解析 IP 位址，它必須知道至少一個名稱伺服器及其 IP 位址。透過 YaST 的幫助可以輕鬆指定這類名稱伺服器。如果您使用數據機撥接連接，完全不需要手動設定名稱伺服器。撥接通訊協定在連接建立時會提供名稱伺服器位址。

whois 通訊協定與 DNS 密切相關。利用此程式，可快速找出伺服器負責哪個指定網域。

注: MDNS 和 `.local` 網域名稱

`.local` 最上層網域將被解析程式視為連結本地網域。DNS 要求將做為多路廣播 DNS 要求予以傳送，而非通常的 DNS 要求。如果已在名稱伺服器組態中使用了 `.local` 網域，則必須在 `/etc/host.conf` 中關閉此選項。同時，請檢視 `host.conf` 手冊頁。

如果要在安裝期間關閉 MDNS，請使用 `nomdns=1` 做為開機參數。

如需有關多路廣播 DNS 的詳細資訊，請參閱 <http://www.multicastdns.org>。

18.4 使用 YaST 手動設定網路連線

Linux 可支援多種網路類型。大多數使用不同的設備名稱和組態檔，會分佈在檔案系統的不同位置。要更瞭解手動網路組態的綜覽，請參閱[第 18.6 節「手動設定網路連線」](#) [221頁]。

在預設會啟用 NetworkManager 的 SUSE Linux Enterprise Desktop 中，所有網路卡都已設定好。如果未啟用 NetworkManager，只會自動設定連接 (連接了網路纜線) 的第一個介面。可隨時在安裝的系統上設定其他的硬體。以下幾節將描述 SUSE Linux Enterprise Desktop 支援之所有網路連線類型的網路組態。

18.4.1 使用 YaST 設定網路卡

若要在 YaST 中設定有線或無線網路卡，請選取「[網路設備](#)」>「[網路設定](#)」。啟動模組後，YaST 將顯示「[網路設定](#)」對話方塊，其中包含四個索引標籤：「[全域選項](#)」、「[綜覽](#)」、「[主機名稱/DNS](#)」和「[路由](#)」。

使用「[全域選項](#)」索引標籤可設定一般網路選項，例如 NetworkManager、IPv6 和一般 DHCP 選項的使用方式。若需要更多的資訊，請參閱[章節「設定全域網路選項的組態」](#) [203頁]。

「[綜覽](#)」索引標籤包含有關已安裝網路介面與組態的資訊。此處會列出所有正確偵測到之網路卡的名稱。在此對話方塊中，您可以手動設定新網路卡、移除或變更其組態。若要手動設定未自動偵測到的卡，請參閱[章節「設定未偵測到的網路卡」](#) [209頁]。若要變更已設定卡的組態，請參閱[章節「變更網路卡組態」](#) [204頁]。

使用「[主機名稱/DNS](#)」索引標籤可設定機器的主機名稱並指定要使用的伺服器。若需要更多的資訊，請參閱[章節「設定主機名稱和 DNS」](#) [210頁]。

「[路由](#)」索引標籤用於設定路由組態。如需相關資訊，請參閱[章節「設定路由」](#) [211頁]。

圖形 18.3 設定網路組態



設定全域網路選項的組態

使用 YaST 「網路設定」模組的「全域選項」索引標籤，可以設定重要的全域網路選項，例如 NetworkManager、IPv6 和一般 DHCP 用戶端選項的使用方式。這些設定適用於所有網路介面。

在「網路設定方法」中，選擇管理網路連線的方式。若希望 NetworkManager 桌面 applet 管理所有介面的連線，請選擇「使用者以 *NetworkManager* 控制」。此選項最適合用於在多個有線和無線網路之間進行切換。如果您執行的不是桌面環境 (GNOME 或 KDE)，或者您的電腦是 Xen 伺服器、虛擬系統或者會在網路中提供 DHCP 或 DNS 等網路服務，請使用「使用 *ifup* 的傳統方法」。如果使用 NetworkManager，則應該使用 `nm-applet` 設定網路選項，「網路設定」模組的「綜覽」、「主機名稱/DNS」與「路由」索引標籤都處於停用狀態。如需 NetworkManager 的詳細資訊，請參閱第 22 章使用 *NetworkManager* [257 頁]。

在「*IPv6 通訊協定設定*」中，選擇是否要使用 *IPv6* 通訊協定。可以同時使用 *IPv6* 和 *IPv4*。預設會啟用 *IPv6*。但是，在不使用 *IPv6* 通訊協定的網路中，停用 *IPv6* 通訊協定時回應較快。若要停用 *IPv6*，請取消核取「*啟用IPv6*」選項。這會停用 *IPv6* 核心模組的自動載入功能。重新開機後會套用此設定。

在「*DHCP 用戶端選項*」中，設定 *DHCP* 用戶端的選項。若希望 *DHCP* 用戶端要求伺服器一律廣播其回應，請核取「*申請廣播回應*」。如果您的機器需要在不同的網路之間移動，可能會需要核取該項。在一個網路中，每個 *DHCP* 用戶端的「*DHCP 用戶端識別碼*」均不能相同。若將其留為空白，則預設會使用網路介面的硬體位址。但是，如果您要使用同一個網路介面執行多個虛擬機，因此會使用同一個硬體位址，則請在此處指定不限格式的唯一識別碼。

「*要傳送的主機名稱*」指定當 *dhcpcd* 將訊息傳送至 *DHCP* 伺服器時，主機名稱選項欄位使用的字串。有些 *DHCP* 伺服器會根據此主機名稱 (動態 *DNS*) 更新名稱伺服器區域 (正向或反向記錄)。此外，有些 *DHCP* 伺服器需要「*要傳送的主機名稱*」選項欄位包含用戶端傳送之 *DHCP* 訊息中的特定字串。保留「*自動*」將傳送目前的主機名稱 (即 */etc/HOSTNAME* 中定義的主機名稱)。將選項欄位留為空白則不會傳送主機名稱。如果您不希望根據 *DHCP* 的資訊變更預設路由，請取消核取「*透過DHCP 變更預設路由*」。

變更網路卡組態

若要變更網路卡組態，請在 *YaST* 的「*網路設定*」>「*綜覽*」中偵測到的網路卡清單中選取網路卡，然後按一下「*編輯*」。「*網路卡設定*」對話方塊隨即出現，您可以使用「*一般*」、「*位址*」和「*硬體*」索引標籤調整網路卡組態。如需關於無線網路卡組態的更多資訊，請參閱第 19.1.2 節「*使用 YaST 進行設定*」[240頁]。

設定 IP 位址

在「*網路卡設定*」對話方塊的「*位址*」索引標籤中，可以設定網路卡的 *IP* 位址或確定其 *IP* 位址的方式。系統支援 *IPv4* 和 *IPv6* 兩種位址。您可以為網路卡設定「*無IP 位址*」(適用於 *bonding* 設備)、「*靜態指定的IP 位址*」(*IPv4* 或 *IPv6*)，也可以透過「*DHCP*」與/或「*Zeroconf*」為其指定「*動態位址*」。

若要使用「*動態位址*」，請選擇是使用「*僅限DHCP 版本4*」(適用於 *IPv4*)、「*僅限DHCP 版本6*」(適用於 *IPv6*)，還是「*DHCP 版本4 與6*」。

若情況適合，系統會將安裝時第一個可用的已連結網路卡自動設定為使用透過 DHCP 設定的自動位址。在預設會啟用 NetworkManager 的 SUSE Linux Enterprise Desktop 中，所有網路卡都已設定好。

若您用的是 DSL 連線而非 ISP (網際網路服務提供者) 指定的靜態 IP，還應該使用 DHCP。若您決定使用 DHCP，請在 YaST 網路卡組態模組中開啟「**網路設定**」對話方塊，於「**全域選項**」索引標籤的「**DHCP 用戶端選項**」中設定詳細資料。在「**申請廣播回應**」中指定 DHCP 用戶端是否應要求伺服器一律廣播其回應。如果您的機器是需要不同網路之間移動的行動用戶端，則可能需要核取此選項。若您將虛擬主機設定為透過同一個介面與不同的主機進行通訊，則需要使用「**DHCP 用戶端識別碼**」來分辨它們。

DHCP 對於用戶端組態是不錯的選擇，但不適用於伺服器組態。若要設定靜態 IP 位址，請如下執行：

- 1 在 YaST 網路卡組態模組的「**綜覽**」索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「**編輯**」。
- 2 在「**位址**」索引標籤中，選擇「**靜態指定的 IP 位址**」。
- 3 輸入「**IP 位址**」。使用 IPv4 和 IPv6 位址都可以。在「**子網路遮罩**」中輸入網路遮罩。若使用 IPv6 位址，請以 /64 格式使用「**子網路遮罩**」做為字首長度。

您還可以選擇為此位址輸入完全合法的「**主機名稱**」，它將會寫入 /etc/hosts 組態檔案中。

- 4 按一下「**下一步**」。
- 5 若要啟用組態，請按一下「**確定**」。

若使用靜態位址，系統將不會自動設定名稱伺服器和預設閘道。若要設定名稱伺服器，請依**章節「設定主機名稱和 DNS」** [210頁]中的說明進行。若要設定閘道，請依**章節「設定路由」** [211頁]中的說明進行。

設定別名

一張網路卡可擁有多 IP 位址，稱為別名。若要設定您網路卡的別名，請如下執行：

- 1 在 YaST 網路卡組態模組的「綜覽」索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「編輯」。
- 2 在「位址」>「其他位址」索引標籤中，按一下「新增」。
- 3 輸入「別名」、「IP 位址」和「網路遮罩」。別名中不要包含介面名稱。
- 4 按一下「確定」。
- 5 按一下「下一步」。
- 6 若要啟用組態，請按一下「確定」。

變更設備名稱和 Udev 規則

可以在網路卡正在使用中時變更它的設備名稱。也可以決定網路卡是否應由 udev 透過其硬體 (MAC) 位址或透過匯流排 ID 識別。後者更適合大型伺服器，可以方便熱交換網路卡。若要使用 YaST 設定這些選項，請執行下列步驟：

- 1 在 YaST 「網路設定」模組的「綜覽」索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「編輯」。
- 2 移至「硬體」索引標籤。目前的設備名稱顯示在「Udev 規則」中。按一下「變更」。
- 3 選擇 udev 是應透過網路卡的「MAC 位址」還是透過「匯流排 ID」來識別網路卡。網路卡目前的 MAC 位址和匯流排 ID 將顯示在對話方塊中。
- 4 若要變更設備名稱，請選取「變更設備名稱」選項，然後編輯名稱。
- 5 按一下「確定」，然後按「下一步」。
- 6 若要啟用組態，請按一下「確定」。

變更網路卡核心驅動程式

有些網路卡可能有多個核心驅動程式可供使用。如果網路卡已設定，則 YaST 可讓您從可用的適用驅動程式清單中選取要使用的核心驅動程式。還可為核心驅動程式指定選項。若要使用 YaST 設定這些選項，請執行下列步驟：

- 1 在 YaST 網路設定模組的「綜覽」索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「編輯」。
- 2 移至「硬體」索引標籤。
- 3 在「模組名稱」中選取要使用的核心驅動程式。在「選項」中以選項=值的格式輸入選定驅動程式的選項。若要使用多個選項，應以空格將其隔開。
- 4 按一下「確定」，然後按「下一步」。
- 5 若要啟用組態，請按一下「確定」。

啟動網路設備

若使用 ifup 的傳統方法，便可設定您的設備在開機時、連接纜線時、偵測到網路卡時啟動，或以手動方式啟動，或者永不啟動。若要變更設備啟動，請執行下列步驟：

- 1 在 YaST 的「網路設備」>「網路設定」中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「編輯」。
- 2 在「一般」索引標籤中，從「設備啟用」中選擇希望的項目。

選擇「開機時」可在系統開機時啟動設備。若使用「電纜連線」，系統會監控介面，探查是否有實體的連線。若使用「Hotplug」，只要介面可用就會將其設定。它與「開機時」選項類似，唯一的區別在於如果開機時介面不存在，則不會發生錯誤。選擇「手動」可以透過 ifup 或 KInternet 手動控制介面。選取「永不」則根本不會啟動設備。「在 NFSroot 時」與「開機時」類似，區別是無法使用指令 `rcnetwork stop` 將介面關機。若您使用的是 nfs 或 iscsi 根檔案系統，請使用此選項。

- 3 按一下「下一步」。
- 4 若要啟用組態，請按一下「確定」。

一般只有系統管理員可以啟動和取消啟動網路介面。若您希望所有使用者都能夠透過 KInternet 啟動此介面，請選取「允許非 root 使用者透過 Kinternet 控制設備」。

設定最大傳送單位大小

您可以設定介面的最大傳送單位 (MTU)。MTU 指允許的最大封包大小 (以位元組計)。MTU 越高，頻寬效率就越高。但是，大型封包可能會將慢速介面阻擋一段時間，這會加劇後續封包的延遲。

- 1 在 YaST 的「網路設備」>「網路設定」中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「編輯」。
- 2 在「一般」索引標籤中，從「設定 MTU」清單中選取所需的項目。
- 3 按一下「下一步」。
- 4 若要啟用組態，請按一下「確定」。

設定防火牆

您不需輸入第 9.4.1 節「Configuring the Firewall with YaST」(第 9 章 *Masquerading and Firewalls*, ↑*Security Guide*) 所說明的詳細防火牆資訊，僅需在設備設定時判斷基本防火牆設定即可。請執行下列步驟：

- 1 開啟 YaST 的「網路設備」>「網路設定」模組。在「綜覽」索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下「編輯」。
- 2 進入「網路設定」對話方塊的「一般」索引標籤。
- 3 決定您要為介面指派的防火牆區域。可用的選項如下：

停用防火牆

只有當防火牆已停用以及防火牆根本未執行時，此選項才可用。唯有您的機器位於受外部防火牆保護的更大網路中時，才能使用此選項。

自動指定區域

只有當防火牆已啟用時，此選項才可用。防火牆會執行，並且介面會自動指定至防火牆區域。包含關鍵字 `any` 的區域或外部區域將會用於這類介面。

內部區域 (未保護)

防火牆會執行，但不強制執行任何保護此介面的規則。如果您的機器位於受外部防火牆保護的較大網路中，請使用此選項。如果機器具有多個網路介面，此選項也適用於連接到內部網路的介面。

廢除區域

廢除區域是內部網路與(有潛在風險的)網際網路之前的另一道防線。從內部網路與網際網路都可連接到指派至此區域的主機，但主機無法連存取內部網路。

外部區域

防火牆執行於此介面上，且會全面保護其抵禦其他危險網路流量的攻擊。此為預設選項。

4 按「下一步」。

5 按一下「確定」以啟用組態。

設定未偵測到的網路卡

您的網路卡有可能會無法正確偵測到。若發生此情況，該卡就不會出現在偵測到的網路卡清單中。若您確定您的系統具備網路卡的驅動程式，可手動設定。也可設定特殊的網路設備類型，如橋接、bond、TUN 或 TAP。若要設定未偵測到的網路卡或特殊設備，請執行下列步驟：

- 1 在 YaST 的「網路設備」>「網路設定」>「綜覽」對話方塊中，按一下「新增」。
- 2 在「硬體」對話方塊中，從可用的選項中設定介面的「設備類型」和「組態名稱」。如果網路卡是 PCMCIA 或 USB 設備，請啟用個別的核取方塊並使用「下一步」來結束對話方塊。否則，您可以根據需要定義要用於網路卡的核心「模組名稱」以及卡的「選項」。
- 3 按一下「下一步」。
- 4 在「一般」、「位址」和「硬體」索引標籤中設定介面的所有必要選項，例如 IP 位址、設備啟動或防火牆區域。如需組態選項的詳細資訊，請參閱[章節「變更網路卡組態」](#) [204頁]。

- 5 若您介面設備類型選擇「無線」，請在下一個對話設定無線連接。關於無線設備組態的詳細資料位於第 19.1 節「無線區域網路」[237頁]。
- 6 按一下「下一步」。
- 7 若要啟用新網路組態，請按一下「確定」。

設定主機名稱和 DNS

若您在安裝期間未變更網路組態，且已有有線網路卡可用，則系統會自動為您的電腦產生主機名稱並啟動 DHCP。同時也會自動產生您主機要整合至網路環境所需的的名稱服務資訊。若網路位址設定使用 DHCP，則網域名稱伺服器清單會自動填入適當的資料。若您希望使用靜態設定，請手動設定數值。

若要變更您電腦的名稱並調整名稱伺服器搜尋清單，請如下執行：

- 1 在 YaST 的「網路設備」模組中，移至「網路設定」>「主機名稱/DNS」索引標籤。
- 2 輸入「主機名稱」並根據需要輸入「網域名稱」。如果機器是郵件伺服器，網域就格外重要。請注意，主機名稱是全域的，會套用於所有已設定的網路介面。

若您要使用 DHCP 獲取 IP 位址，則電腦的主機名稱由 DHCP 自動設定。若要連接到其他網路，最好停用此選項，因為其他網路可能會指定其他主機名稱，而在執行時期變更主機名稱會混淆圖形桌面。若要停止使用 DHCP 獲取 IP 位址，請取消核取「透過 DHCP 變更主機名稱」。

若使用 DHCP 獲取 IP 位址，主機名稱預設會寫入 `/etc/hosts` 並會解析為 127.0.0.2 IP 位址。若要停用此功能，請取消核取「將主機名稱寫入 `/etc/hosts`」，但請注意，未連接至網路時將無法解析您的主機名稱。

- 3 在「修改 DNS 組態」中，選取修改 DNS 組態 (名稱伺服器、搜尋清單、`/etc/resolv.conf` 檔案的內容) 的方式。

若選取「使用預設規則」選項，則組態由 `netconfig` 程序檔來處理，這樣會將靜態定義的資料 (使用 YaST 或在組態檔案中) 與從 DHCP 用戶端或 NetworkManager 動態取得的資料合併。此預設規則能夠處理大部分的情況。

若選取「*僅手動*」選項，則無法使用 `netconfig` 修改 `/etc/resolv.conf` 檔案。但是可以手動編輯此檔案。

若選取「*自定規則*」選項，則應指定定義合併規則的「*自定規則*」字串。該字串包含要視為設定有效來源的介面名稱清單(以逗號分隔)。您也可以不提供完整介面名稱，而是使用基本萬用字元比對多個介面。例如，`eth*ppp?` 將先找到所有 `eth`，然後找到所有 `ppp0-ppp9` 介面。以下為指定如何套用 `/etc/sysconfig/network/config` 檔案中所定義之靜態設定的兩個特定規則值：

`STATIC`

靜態設定必須與動態設定合併在一起。

`STATIC_FALLBACK`

僅當沒有動態設定可用時，才使用靜態設定。

如需詳細資訊，請參閱 `man 8 netconfig`。

- 4 輸入「*名稱伺服器*」並填寫「*網域搜尋*」清單。名稱伺服器必須透過 IP 位址而非主機名稱指定，例如 `192.168.1.116`。在「*網域搜尋*」索引標籤中指定的名稱就是用於解析沒有指定網域之主機名稱的網域名稱。如果使用多個「*網域搜尋*」，請以逗號或空格將其隔開。
- 5 若要啟用組態，請按一下「*確定*」。

設定路由

若要讓您的電腦與其他電腦和其他網路通訊，必須提供路由資訊，以讓網路流量採取正確的路徑。若使用 `DHCP`，會自動提供此資訊。若使用靜態設定，必須手動新增此資料。

- 1 在 YaST 中，移至「*網路設定*」>「*路由*」。
- 2 輸入「*預設開道*」的 IP 位址。預設開道會比對每個可能的目的地，但是如果任何其他符合所需位址的項目已經存在，就會使用該項目而非預設路由。
- 3 可以在「*路由表*」中輸入更多項目。輸入「*目的地*」網路 IP 位址、「*開道*」IP 位址和「*網路遮罩*」。選取要透過其將流量路由至所定義網路的

「設備」(減號表示任意設備)。若要省略這些值的任何一個，請使用減號「-」。若要在表中輸入預設閘道，請在「目的地」欄位中使用「default」。

注

若使用了多個預設路由，則可以指定權值選項確定哪個路由的優先程度較高。要指定權值選項，請在「選項」中輸入 `- metric 數值`。權值最高的路由做為預設路由。如果該網路設備已解除連接，將會移除其路由，使用下一個路由。不過，目前的核心不會在靜態路由中使用權值，只有 `multipathd` 等路由精靈會使用。

4 如果系統是路由器，請在「網路設定」中啟用「*IP 轉遞*」選項。

5 若要啟用組態，請按一下「確定」。

18.4.2 數據機

在「YaST 控制中心」中，您可透過「網路設備」>「數據機」存取數據機組態。如果無法自動偵測到您的數據機，請移至「數據機設備」索引標籤，按一下「新增」開啟對話方塊以手動組態。在「數據機設備」中輸入數據機要連接的介面。

提示: CDMA 和 GPRS 數據機

使用 YaST 「數據機」模組來設定支援的 CDMA 和 GPRS 數據機，這與您設定一般數據機使用的方式一樣。

圖形 18.4 數據機組態

 **數據機參數**

輸入所有數據機終端值。 「數據機裝置」指定數據機所連接的埠。 ttyS0、ttyS1 等是指序列埠。 [更多](#)

數據機設備(U):

撥號前置號碼 (如果需要)(X):

撥號模式
☒ 按鍵式撥號(K)
☐ 轉盤式撥號(P)

特殊設定
☒ 開啟喇叭(S)
☒ 偵測撥號聲音(E)

如果您使用的是專用交換機(PBX)，可能需要輸入撥號前置號碼。通常為0。請參閱 PBX 隨附的說明。另外，請選擇是否使用按鍵式或轉盤式撥號、是否應該打開喇叭以及數據機偵測撥號音前是否應等待。如果數據機連接到分機，就不能啟用最後一個選項。

在「詳細資料」下，設定傳輸速率和數據機的啟始字串。只有在無法自動偵測您的數據機或是數據機需要特殊設定來傳輸資料時，才能變更這些設定。這些設定主要適用於 ISDN 終端機介面卡。按一下「確定」來結束此對話方塊。要將數據機控制權委託給不具備 root 許可權的一般使用者，請啟用「允許非 root 使用者透過 Kinternet 控制設備」。使用此方式，使用者即可啟用和停用介面，而不需要管理員的許可。在「撥號前置號碼正規表示式」下，指定一個正規表示式。KInternet 中的「撥號前置號碼」(一般使用者均可修改)，必須符合此正規表示式。如果此欄位為空白，則使用者必須有管理員許可才能設定不同的「撥號前置號碼」。

在下一個對話方塊中選取ISP。如果要從國內的ISP預先定義清單中選擇，請選取「國家」。或者，可按一下「新增」來開啟一個對話方塊，您可以在其中輸入您的ISP資料。這包括撥接連接名稱、ISP名稱以及您的ISP提供的登入名稱和密碼。啟用「永遠詢問密碼」，以提示您在每次連接時輸入密碼。

在最後一個對話方塊中，可以指定其他的連接選項：

視需要撥號

如果您啟用了「依需求撥號」，請至少設定一個名稱伺服器。因為許多程式需要定期從網際網路獲取資料，所以請僅在網路網際連線費用較低的情況下使用此功能。

連接時修改 DNS

此選項是依照預設啟用的，每次您連接到網際網路時，即會更新名稱伺服器位址。

自動取回 DNS

如果提供者沒有在連接後傳輸其網域名稱伺服器，則應停用此選項並手動輸入DNS資料。

自動重新連接

若啟用此選項，則會在連線失敗後自動重新建立連線。

忽略提示

此選項會停止偵測來自撥號伺服器的所有提示。如果建立的連線太慢或完全沒有作用，請嘗試此選項。

外部防火牆介面

選取此選項可啟動防火牆並將介面設為外部。這樣就可讓您在連接網際網路時免遭外部的攻擊。

閒置逾時 (秒)

使用此選項來指定網路靜止一段時間後即自動中斷數據機的連接。

IP 詳細資料

這會開啟位址組態對話方塊。如果您的ISP沒有為您的主機指定動態IP位址，請停用「動態IP位址」然後輸入您主機的本地IP位址和遠端IP位址。請向您的ISP詢問此資訊。讓「預設路由」保持為啟用狀態並選取「確定」來結束此對話方塊。

選取「**下一步**」回到原先的顯示數據機組態摘要的對話方塊。按一下「**確定**」關閉此對話方塊。

18.4.3 ISDN

使用此模組來為您的系統設定一個或多個 ISDN 卡。如果 YaST 未偵測到您的 ISDN 卡，請在「*ISDN 設備*」索引標籤中按一下「**新增**」，然後手動選取您的卡。可以有多個介面，但多個 ISP 只能設定一個介面。在接下來的對話方塊中，設定網路卡正常運作所需的 ISDN 選項。

圖形 18.5 ISDN 組態

contr0 的 ISDN 低階組態
使用「開機時」，會在系統開機時載入驅動程式。若使用「手動」，驅動程式必須以 rcisdn start 指令開始。只有 root 使用... [更改](#)

ISDN 卡資訊
廠商: Abocom/Magitek
ISDN 卡: 2BD1

驅動程式(Δ):
HiSax driver

ISDN 協定
☒ Euro-ISDN (EDSS1)(E)
☐ 1TR6(G)
☐ 租用專線(L)
☐ NI1(I)

國家(C): 德國
代碼(D): +49

區域號碼(Δ):
撥號前置號碼(D):

☒ 啟動 ISDN 記錄(I)

啟用服務(D):
開機時

[說明](#) [取消](#) [上一步](#) [確定](#)

在下一個對話方塊中(如 圖形 18.5 「ISDN 組態」 [215頁] 中所示)，選取要使用的通訊協定。預設為「*Euro-ISDN (EDSS1)*」，但如果是較舊或較大型的交換機，則選取「*1TR6*」。如果您是在美國，請選取「*NI1*」。在相關欄位中選取您的國家。其對應的國家代碼會顯示在旁邊的欄位。最後，提供您的「**區域號碼**」和「**撥號前置號碼**」(如有需要)。若不想記錄所有的 ISDN 流量，請取消核取「**啟動 ISDN 記錄**」選項。

「啟動設備」可定義 ISDN 介面的啟動方式：選取「開機時」會在系統每次開機時啟動 ISDN 驅動程式。「手動」要求您以 root 身份使用 `rcisdn start` 指令來載入 ISDN 驅動程式。「熱插」用於 PCMCIA 或 USB 設備，會在插入設備後載入驅動程式。完成上述設定之後，請選取「確定」。

在下一個對話方塊中，指定您 ISDN 卡的介面類型，並將 ISP 新增至現有的介面。介面可能為 SyncPPP 或 RawIP 類型，但大多數 ISP 是在 SyncPPP 模式中操作，其說明如下。

圖形 18.6 ISDN 介面組態



視您的特殊設定而定，在「我的電話號碼」中必須輸入的號碼會有所不同：

ISDN 卡直接連至電話插孔

標準 ISDN 連接會提供 3 組電話號碼 (稱為多重用戶號碼或 MSN)。如果訂閱者要求更多，最多可以到 10，所有的 MSN 都必須在此輸入，但不需輸入區域碼。如果您輸入錯誤的號碼，您的電話操作員會自動回復指定給您的 ISDN 連接的第一組 MSN。

連接至專用交換機的 ISDN 卡

此外，視安裝的設備而定，組態可能會有所不同：

1. 小型的專用交換機 (PBX) 大多使用 Euro-ISDN (EDSS1) 通訊協定來撥接內線電話。這些電話交換機有一個內部 S0 匯流排並在連接設備上使用內部號碼。

使用其中一組內部號碼做為您的 MSN。您至少可使用一組電話交換機的 MSN，這些 MSN 已啟用可直接對外撥號。如果無法使用，請嘗試撥 0。如需詳細資訊，請參閱您的電話交換機隨附的文件。

2. 較大型的商用電話交換機通常使用 1TR6 通訊協定來撥接內線電話。其 MSN 稱為 EAZ，通常可對應直撥號碼。如果要在 Linux 中設定組態，請輸入 EAZ 的最後一碼即可。最後一步是試著撥從 1 到 9 的每個數字。

如果想在下一個收費單位開始之前終止連接，可啟用「*ChargeHUP*」。不過，要記得並不是每個 ISP 都適用。您也可以勾選對應的選項來啟用通道合併 (多重連結 PPP)。最後，您可以選取「外部防火牆介面」和「重新啟動防火牆」來啟用連結的防火牆。要讓沒有管理員許可權的一般使用者能夠啟動或取消啟動介面，請選取「允許非 *root* 使用者透過 *Kinternet* 控制設備」。

「詳細資料」會開啟一個對話方塊，用來執行更複雜的連線配置，與一般的家庭使用者不相關。選取「確定」以結束「詳細資料」對話方塊。

下一個對話方塊中可進行 IP 位址設定。如果您的提供者沒有給您一個靜態 IP，請選取「動態 IP 位址」。或者，可根據您 ISP 的規格，在提供的欄位中輸入您主機的本地 IP 位址和遠端 IP 位址。如果該介面應做為網際網路的預設路由，請選取「預設路由」。每個主機只能設定一個介面做為預設路由。選取「下一步」來結束此對話方塊。

下列對話方塊中可讓您設定您的國家和選取 ISP。清單中所列的 ISP 只有撥號計費提供者。如果您的 ISP 不在清單上，請選取「新增」。會開啟「提供者參數」對話方塊，您可在其中輸入您 ISP 的所有詳細資料。輸入電話號碼時，在數字之間不能有空白或逗號。最後，請輸入您的 ISP 所提供的登入名稱和密碼。完成後，請選取「下一步」。

如果要在獨立的工作站上使用「視要求撥號」，請指定名稱伺服器 (DNS 伺服器)。大多數 ISP 均支援動態 DNS，即在您每次連接時 ISP 都會送出一個名稱伺服器的 IP 位址。不過，如果您使用的是單一工作站，您仍需提供一個替代位址，例如 192.168.22.99。如果您的 ISP 不支援動態 DNS，請指定 ISP 的名稱伺服器 IP 位址。如有需要，可為連接指定一個時限，即為未使用網路的時間

(以秒為單位)，之後即會自動終止連接。使用「**下一步**」來確認您的設定。YaST 會顯示已設定介面的摘要。若要啟用這些設定，請選取「**確定**」。

18.4.4 纜線數據機

在有些國家/地區，透過電視纜線來存取網際網路的現象相當普遍。有線電視用戶通常會有一部數據機，一邊連接到電視纜線的輸出端子，另一邊(使用10Base-TG 雙絞纜線)連接到電腦網路卡。接著纜線數據機會以固定的 IP 位址提供專用的網際網路連線。

視您的 ISP 所提供的說明而定，設定網路卡時可選取「**動態位址**」或「**靜態指定的IP 位址**」。現在大多數提供者都使用 DHCP。靜態 IP 位址通常是特殊商用帳戶的一部分。

有關纜線數據機組態的進一步資訊，請參閱「支援資料庫」文章中的相關主題，網址為 http://en.opensuse.org/SDB:Setting_Up_an_Internet_Connection_via_Cable_Modem_with_SuSE_Linux_8.0_or_Higher。

18.4.5 DSL

如果要設定您的 DSL 設備，請從 YaST 「**網路卡設備**」區段選取「**DSL**」模組。此 YaST 模組含有許多對話方塊，可根據下列其中一種通訊協定在其中設定 DSL 連接的參數：

- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz 網路卡)
- 點對點通道通訊協定 (PPTP)—奧地利

在「**DSL 組態綜覽**」對話方塊的「**DSL 設備**」索引標籤中，提供有已安裝 DSL 設備的清單。若要變更某個 DSL 設備的組態，請在清單中將其選定，然後按一下「**編輯**」。按一下「**新增**」可以手動設定新的 DSL 設備。

PPPoE 或 PPTP 類型的 DSL 連接組態，會要求對應的網路卡必須正確設定。如果您尚未完成此步驟，請先選取「**設定網路卡**」來進行設定(請參閱 [第 18.4.1 節「使用 YaST 設定網路卡」](#) [202頁])。若是 DSL 連結，將會自動指定位址而不

是透過 DHCP 指定，因此您不可啟用「動態位址」選項。而是應該輸入介面的靜態虛擬位址，例如 192.168.22.1。在「子網路遮罩」中，輸入 255.255.255.0。如果您要設定獨立的工作站，則應保留「預設閘道」為空白。

提示

「IP 位址」和「子網路遮罩」中的值只是預留位置。只用來啟始化網路卡，而不代表 DSL 連接等等。

在第一個 DSL 組態對話方塊 (請參閱圖形 18.7 「DSL 組態」 [219頁]) 中，選取「PPP 模式」和DSL數據機要連接的「乙太網路卡」(大多為eth0)。然後使用「啟動設備」來指定是否要在開機過程中建立 DSL 連結。按一下「允許非 root 使用者透過 KInternet 控制設備」，以授權不具備 root 許可權的一般使用者透過 KInternet 啟動或取消啟動介面。

在下一個對話方塊中選取國家/地區，並從該國家/地區的多個 ISP 營運商中進行選擇。後續的 DSL 組態對話方塊詳細資料會視目前已設定的選項而定，因此下列段落只提供簡單的介紹。如需可用選項的詳細資訊，請參閱對話方塊中的詳細說明。

圖形 18.7 DSL 組態

DSL 組態

在此處可以設定 DSL 連接最重要的設定。首先，請選擇 PPP 模式。這可以是 PPP over Ethernet (PPPoE) 、 PPP over ATM (PPPoATM) 、 CAPI for ADSL 或 Poin... 更多

DSL 連接設定

PPP 模式(M):

PPP over Ethernet

PPP 模式相關設定

VPI/VCI(V)

乙太網路卡(E)

3c905C-TX/TX-M [Tornado]

網路卡 - DHCP 位址

變更裝置(C)

何種網路名稱 IP 位址(I)

10.0.0.138

設定網路卡(C)

啟用網路(D):

手動

☒ 允許非 root 使用者經由 KInternet 進行裝置控制(H)

說明

取消(C)

上一步(B)

下一步(N)

如果要在獨立的工作站上使用「視要求撥號」，請指定名稱伺服器 (DNS 伺服器)。大部份 ISP 均支援動態 DNS，即為每次連接時，名稱伺服器送出的 IP 位址。不過，如果是單一工作站，請提供一個替代位址，例如 192.168.22.99。如果您的 ISP 不支援動態 DNS，請輸入您 ISP 提供的名稱伺服器 IP 位址。

「閒置逾時 (秒)」定義網路靜止一段時間後即自動中斷連接。合理的逾時秒數為 60 和 300 秒。如果停用「視需要撥號」，將逾時值設為零將有助於避免自動掛斷。

T-DSL 的組態與 DSL 的設定非常相似。只要選取「*T-Online*」做為您的提供者，YaST 就會開啟 T-DSL 組態對話方塊。在此對話方塊中，提供一些 T-DSL 要求的其他資訊—線路 ID、T-Online 號碼、使用者代碼和您的密碼。在您加入 T-DSL 後，應該會收到這些資訊。

18.5 NetworkManager

NetworkManager 是行動工作站理想的解決方案。有了 NetworkManager，您在移動中，就不必耽心網路介面的組態設定，也不必切換網路。NetworkManager 能夠自動連接到已知的 WLAN 網路。如果您有兩種或更多種連接方式，它還能夠連到速度較快的連接。

不過，NetworkManager 解決方案並不適用於所有情況，因此您仍然可以在管理網路連線 (ifup) 的傳統方法與 NetworkManager 之間進行選擇。如果要使用 NetworkManager 管理網路連線，請依第 22.2 節「啟用 NetworkManager」[258 頁] 中所述啟用 YaST 網路設定模組中的 NetworkManager，然後使用 NetworkManager 設定網路連線。如需使用案例的清單以及如何設定與使用 NetworkManager 的詳細說明，請參閱第 22 章使用 NetworkManager [257 頁]。

ifup 與 NetworkManager 之間的區別包括：

root 權限

如果您使用 NetworkManager 設定網路，就可以隨時使用 Applet 在您的桌面環境上輕鬆切換、停止或啟動網路連線。NetworkManager 也讓您不需要 root 權限就可以變更和設定無線網路卡連接。因此，NetworkManager 是行動工作站理想的解決方案。

傳統使用 ifup 設定組態時，雖然也有一些方法可以在有或沒有使用者互動 (例如由使用者管理的設備) 的情況下切換、停止或啟動連接，但是一定要有

`root` 權限才能變更或設定網路設備。這對行動計算設備而言，往往會造成問題，因為不可能預先設定好所有可能的連線。

網路連線的類型

不論是傳統組態還是 NetworkManager，都能夠使用 DHCP 和靜態組態來處理與無線網路 (WEP、WPA-PSK 和 WPA-Enterprise 存取)、撥號和有線網路的網路連線。它們也支援透過 VPN 的連接。

NetworkManager 會嘗試使用最好的連接，讓您的電腦隨時保持連接狀態。如果網路纜線突然中斷，它會嘗試重新連接。它可以從您的無線連線清單中找到訊號最強的網路，並自動用它來連接。如果要用 `ifup` 達到相同的功能，必須執行很多組態工作。

18.6 手動設定網路連線

網路軟體的手動組態應該永遠是最後的替代方案。建議使用 YaST。不過，這個有關網路組態的背景資訊也可協助您使用 YaST。

檔核心偵測到網路卡並建立對應的網路介面時，它會根據探查到設備或載入核心模組的順序，為該設備指定名稱。只有在非常簡單或嚴密控制的硬體環境中，才可以預測預設核心設備的名稱。允許在執行時期新增或移除硬體或支援自動設定設備組態的系統，無法預測核心在重新開機時指定的固定網路設備名稱。

但是，所有系統組態工具均依賴於永久的介面名稱。`udev` 解決了此問題。`udev` 永久網路產生器 (`/etc/udev/rules.d/75-persistent-net-generator.rules`) 會產生符合硬體的規則 (預設使用硬體位址)，並且為硬體指定一個永久唯一的介面。網路介面的 `udev` 資料庫儲存在檔案 `/etc/udev/rules.d/70-persistent-net.rules` 中。該檔案中的每一行描述一個網路介面並指定其永久名稱。系統管理員可以透過編輯 `NAME=""` 項目來變更指定的名稱。您也可以使用 YaST 修改永久規則。

表格 18.5 「手動網路組態程序檔」 [222頁] 總結了與網路組態相關的最重要的程序檔。

表格 18.5 手動網路組態程序檔

指令	函數
<code>if{up,down,status}</code>	<code>if*</code> 程序檔會開啟、停止網路介面或傳回指定介面的狀態。詳細資訊請參閱 <code>ifup</code> 的手冊頁。
<code>rcnetwork</code>	使用 <code>rcnetwork</code> 程序檔可以啟動、停止或重新啟動所有網路介面或指定的網路介面。 <code>rcnetwork stop</code> 用於停止網路介面， <code>rcnetwork start</code> 用於啟動網路介面， <code>rcnetwork restart</code> 用於重新啟動網路介面。若只需停止、啟動或重新啟動一個介面，請使用該命令，並在後面指定介面名稱，例如 <code>rcnetwork restart eth0</code> 。 <code>rcnetwork status</code> 指令顯示介面的狀態、IP 位址以及 DHCP 用戶端是否正在執行。使用 <code>rcnetwork stop-all-dhcp-clients</code> 和 <code>rcnetwork restart-all-dhcp-clients</code> 可以停止或重新啟動在網路介面上執行的 DHCP 用戶端。

如需 `udev` 和永久設備名稱的詳細資訊，請參閱第 12 章 [使用 `udev` 進行動態核心設備管理](#) [123 頁]。

18.6.1 組態檔案

本節提供網路組態檔的綜覽，並說明其用途和使用的格式。

`/etc/sysconfig/network/ifcfg-*`

這些檔案包含網路介面的組態。其中包含啟動模式和 IP 位址等資訊。可以使用的參數請參閱 `ifup` 的手冊頁。此外，如果一般設定僅用於一個介面，則檔案 `dhcp`、`wireless` 和 `config` 中的所有變數都可以用於 `ifcfg-*` 檔案。

/etc/sysconfig/network/{config、dhcp、wireless}

檔案 config 包含 ifup、ifdown、和 ifstatus 行為的一般設定；dhcp 則包含無線區域網路介面卡之 DHCP 和 wireless 的設定。所有三個組態檔中的變數都已被註解。/etc/sysconfig/network/config 中的某些變數也會在 ifcfg-* 檔案中用到，在這些檔案中會以更高的優先程度進行處理。/etc/sysconfig/network/ifcfg.template 檔案列出了可在永久介面中指定的變數。但是，/etc/sysconfig/network/config 的大多數變數都是全域變數，無法在 ifcfg 檔案中覆寫。例如，NETWORKMANAGER 或 NETCONFIG_* 變數就是全域變數。

/etc/sysconfig/network/{routes,ifroute-*}

TCP/IP 封包的靜態路由在此決定。您可以將各種系統任務所需的靜態路由都輸入 /etc/sysconfig/network/routes 檔案：主機的路由、透過閘道前往主機的路由，以及網路的路由。對於需要個別路由的介面，請定義額外的組態檔案：/etc/sysconfig/network/ifroute-*。以介面的名稱取代 *。在路由組態檔中的項目看起來就像這樣：

# Destination	Dummy/Gateway	Netmask	Device
#			
127.0.0.0	0.0.0.0	255.255.255.0	lo
204.127.235.0	0.0.0.0	255.255.255.0	eth0
default	204.127.235.41	0.0.0.0	eth0
207.68.156.51	207.68.145.45	255.255.255.255	eth1
192.168.0.0	207.68.156.51	255.255.0.0	eth1

路由的目的地是在第一個資料欄。這個資料欄可能包含網路或主機的 IP 位址，這是指「可到達的」名稱伺服器、完整合格的網路或主機名稱。

第二個資料欄包含預設的閘道或是可以存取主機或網路的閘道。第三個資料欄包含在閘道後面的網路或主機的網路遮罩。例如，遮罩為 255.255.255.255，供在閘道後面的主機使用。

第四欄只與連接至本地主機的網路相關，例如迴路、乙太網路、ISDN、PPP 以及虛擬設備。必須在這裏輸入設備名稱。

第五欄(可選)可指定路由的類型。不需要的欄位必須包含 - 減號，以確保解析程式可正確解譯指令。如需詳細資訊，請參閱 routes(5) 線上文件。

/etc/resolv.conf

主機所屬的網域指定於此檔案(關鍵字 `search`)；另外也會列出要存取的名稱伺服器位址的狀態(關鍵字 `nameserver`)。可以在檔案中指定多個網域名稱。解析不完整的名稱時，會嘗試附加個別 `search` 項目產生一個名稱。可以在多行中指定多個伺服器名稱，每個名稱均以 `nameserver` 開頭。注解前面標有 `#` 符號。**範例 18.5** 「`/etc/resolv.conf`」 [224頁] 顯示了 `/etc/resolv.conf` 的格式。

不過，您不可手動編輯 `/etc/resolv.conf`。它是由 `netconfig` 程序檔產生的。若不想使用 `YaST` 來定義靜態 DNS 組態，請在 `/etc/sysconfig/network/config` 檔案中手動編輯適當的變數：

`NETCONFIG_DNS_STATIC_SEARCHLIST` (用於查詢主機名稱的 DNS 網域名稱清單)、`NETCONFIG_DNS_STATIC_SERVERS` (用於查詢主機名稱的名稱伺服器 IP 位址清單)、`NETCONFIG_DNS_FORWARDER` (定義必須設定的 DNS 轉遞者的名稱)。若要使用 `netconfig` 停用 DNS 組態，請設定 `NETCONFIG_DNS_POLICY=''`。如需 `netconfig` 的詳細資訊，請參閱 `man 8 netconfig`。

範例 18.5 `/etc/resolv.conf`

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

/sbin/netconfig

`netconfig` 是一個模組化工具，用於管理其他網路組態設定。它根據預先定義的規則，將靜態定義的設定與自動組態機制 (如 `dhcp` 或 `ppp`) 提供的設定進行合併。透過呼叫負責修改組態檔案和重新啟動服務或類似動作的 `netconfig` 模組，將必要的變更套用至系統。

`netconfig` 可以辨識三個主要動作。`dhcp` 或 `ppp` 等精靈使用 `netconfig modify` 與 `netconfig remove` 指令提供或移除 `netconfig` 的設定。使用者僅可使用 `netconfig update` 指令：

modify

`netconfig modify` 指令會修改目前的介面和服務特定的動態設定，並更新網路組態。`Netconfig` 從標準輸入或從以 `--lease-file` 檔案名稱選項指定的檔案讀取設定，並將它們儲存於內部，直到系統重新開機或執行下一個修改或移除動作為止。系統會覆寫同一個介面與服務組合的現有設定。介面由 `-i` 介面名稱參數來指定。服務由 `-s` 服務名稱參數來指定。

remove

`netconfig remove` 指令會移除修改動作為指定介面和服務組合提供的動態設定，並更新網路組態。介面由 `-i` 介面名稱參數來指定。服務由 `-s` 服務名稱參數來指定。

update

`netconfig update` 指令會使用目前的設定更新網路組態。變更規則或靜態組態時，此指令非常實用。

`netconfig` 規則和靜態組態設定可透過手動方式定義，或使用 YaST 在 `/etc/sysconfig/network/config` 檔案中定義。自動組態工具 (如 `dhcp` 或 `ppp`) 提供的動態組態設定透過這些工具經由 `netconfig modify` 和 `netconfig remove` 動作直接傳送。`NetworkManager` 也可以使用 `netconfig modify` 與 `netconfig remove` 動作。`NetworkManager` 啟用時，`netconfig` (在 `auto` 規則模式中) 只會使用 `NetworkManager` 設定，忽略使用傳統 `ifup` 方法設定的任何其他介面中的設定。如果 `NetworkManager` 未提供任何設定，則使用靜態設定做為備用設定。系統不支援同時使用 `NetworkManager` 與傳統 `ifup` 方法。

如需 `netconfig` 的詳細資訊，請參閱 `man 8 netconfig`。

/etc/hosts

在此檔中 (請參閱 [範例 18.6「/etc/hosts」](#) [225頁])，IP 位址是指定給主機名稱。如果沒有執行任何名稱伺服器，將使用此 IP 連接設定的所有主機將列示於此。對於每個主機，分別在檔案中輸入一行包含 IP 位址、完全合法的主機名稱及主機名稱的項目。IP 位址必須在行的開頭，然後以空格和定位點分隔這些項目。註解的前面永遠是 `#` 符號。

範例 18.6 /etc/hosts

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

/etc/networks

在此檔中，網路名稱會轉換為網路位址。格式與 `hosts` 檔案格式相似，但是網路名稱在位址前。請參閱範例 18.7 「`/etc/networks`」 [226頁]。

範例 18.7 `/etc/networks`

```
loopback      127.0.0.0
localnet      192.168.0.0
```

/etc/host.conf

名稱解析，即透過解析程式庫 (Resolver Library) 翻譯主機及網路名稱，是由此檔案控制的。該檔案僅用於與 `libc4` 或 `libc5` 連結的程式。對於目前的 `glibc` 程式，請參閱 `/etc/nsswitch.conf` 中的設定。每個參數必須永遠是獨立一行。註解的前面是 `#` 符號。表格 18.6 「`/etc/host.conf` 的參數」 [226頁] 顯示出可用的參數。`/etc/host.conf` 範例是顯示在 範例 18.8 「`/etc/host.conf`」 [227頁]。

表格 18.6 `/etc/host.conf` 的參數

<code>order hosts, bind</code>	指定名稱解析時服務的存取順序。可用的引數有 (以空格或逗號分隔): <code>hosts</code> : 搜尋 <code>/etc/hosts</code> 檔案 <code>bind</code> : 存取名稱伺服器 <code>nis</code> : 使用 NIS
<code>multi on/off</code>	定義在 <code>/etc/hosts</code> 中所輸入的主機是否可以有多個 IP 位址。
<code>nospoof on spoofalert on/off</code>	這些參數會影響名稱伺服器 <i>spoofing</i> ，但並不會對網路組態產生任何影響。
<code>trim domainname</code>	指定的網域名稱在主機名稱解析後會與主機名稱分隔 (只要主機名稱包括網域名稱)。只有在本地網域分離出

來的名稱位於 `/etc/hosts` 檔案，但是仍然使用附加的網域名稱進行辨識時，這個選項才有用。

範例 18.8 `/etc/host.conf`

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

`/etc/nsswitch.conf`

GNU C Library 2.0 的介紹伴隨名稱服務切換 (NSS, Name Service Switch) 的介紹。詳細資訊請參閱 `nsswitch.conf` (5) 一文和 *GNU C Library 參考手冊*。

查詢的順序定義於檔案 `/etc/nsswitch.conf`。`nsswitch.conf` 範例是顯示在 範例 18.9 「`/etc/nsswitch.conf`」[227頁]。註解從 `#` 符號開始。在此範例中，`hosts` 資料庫下的項目表示要求是透過 DNS 傳送到 `/etc/hosts` (files)。

範例 18.9 `/etc/nsswitch.conf`

```
passwd:      compat
group:       compat

hosts:       files dns
networks:    files dns

services:    db files
protocols:   db files

netgroup:    files
automount:   files nis
```

NSS 上可用的「資料庫」列示於 表格 18.7 「透過 `/etc/nsswitch.conf` 的可用資料庫」[228頁]。此外，將來應該還有 `automount`、`bootparams`、`netmasks`、和 `publickey`。NSS 資料庫的組態選項將列於 表格 18.8 「NSS「資料庫」的組態選項」[228頁]。

表格 18.7 透過 `/etc/nsswitch.conf` 的可用資料庫

aliases	sendmail 所執行的郵件別名；請參閱 man 5 aliases。
ethers	乙太網路位址。
group	適用於 getgrent 所使用的使用者群組。請參閱 group 的 man 頁面。
hosts	gethostbyname 及類似功能所使用的主機名稱與 IP 位址。
netgroup	在網路中有效的主機與使用者清單，以利控制存取權限，請參閱 netgroup(5) 一文。
networks	getnetent 所使用的網路名稱與位址。
passwd	getpwent 所使用的使用者密碼；請參閱 passwd(5) 一文。
protocols	getprotoen 所使用的網路通訊協定；請參閱 protocols(5) 一文。
rpc	getrpcbyname 及類似功能所使用的遠端程序呼叫名稱與位址。
services	getservent 使用的網路服務。
shadow	getspnam 所使用的使用者遮蔽密碼；請參閱 shadow(5) 一文。

表格 18.8 NSS「資料庫」的組態選項

files	直接存取檔案，例如 <code>/etc/aliases</code>
db	透過資料庫存取

<code>nis</code> 、 <code>nisplus</code>	NIS，請參閱第 4 章 <i>Using NIS</i> (↑ <i>Security Guide</i>)
<code>dns</code>	只能做為 <code>hosts</code> 與 <code>networks</code> 的延伸
<code>compat</code>	只能做為 <code>passwd</code> 、 <code>shadow</code> 以及 <code>group</code> 的延伸

/etc/nscd.conf

此檔案用來設定 `nscd` (名稱服務快取精靈)。請參閱 `nscd(8)` 與 `nscd.conf(5)`。依預設，`passwd` 與 `groups` 的系統項目是由 `nscd` 快取。這對於目錄服務 (如 NIS 和 LDAP) 的效能而言是很重要的，否則每次存取名稱或群組時都需要使用網路連線。預設是不會快取 `hosts`，因為 `nscd` 快取主機的機制會造成本地系統無法信任轉寄以及反向查詢檢查。不要要求 `nscd` 快取名稱，而是設定快取 DNS 伺服器。

如果啟用 `passwd` 的快取，通常需要 15 秒，才能辨識新增的本地使用者。使用指令 `rcnscd restart` 重新啟動 `nscd`，縮短這段等待時間。

/etc/HOSTNAME

這是沒有附加網域名稱的主機名稱。機器開機時數個程序檔會讀取該檔案。它可以只包含一行，其中設定了主機名稱。

18.6.2 測試與組態

將組態寫入您的組態檔案之前，可先進行測試。若要設定測試組態，請使用 `ip` 指令。若要測試連接，請使用 `ping` 指令。同時也可使用較舊的組態工具，如 `ifconfig` 和 `route`。

`ip`、`ifconfig` 和 `route` 等指令會以不儲存組態檔案的方式直接變更網路組態。除非您將組態輸入正確的組態檔案，否則重新開機之後網路組態的變更就會遺失。

以 ip 設定網路介面

`ip` 這個工具可顯示設定路由、網路設備、原則路由和通道。它是用來取代 `ifconfig` 和 `route` 等舊工具用的。

`ip` 是非常複雜的工具。其一般語法是 `ip options object command`。您可使用下列物件：

`link`

此物件代表網路設備。

`address`

此物件代表設備的 IP 位址。

`neighbour`

此物件代表 ARP 或 NDISC 快取項目。

`route`

此物件代表路由表格項目。

`rule`

此物件代表路由原則資料庫中的規則。

`maddress`

此物件代表多重廣播位址。

`mroute`

此物件代表多重廣播路由快取項目。

`tunnel`

此物件表示 IP 上的通道。

若未提供指令，會使用預設指令，通常是 `list`。

您可使用 `ip link set device_name command` 指令變更設備狀態。例如，若要停用設備 `eth0`，請輸入 `ip link set eth0 down`。若要重新啟用，請使用 `ip link set eth0 up`。

啟用設備之後，就可加以設定。若要設定 IP 位址，請使用 `ip addr add ip_address + dev device_name`。例如，若要將介面 `eth0` 的 IP 位址以標

準廣播 (選項 `brd`) 設定為 `192.168.12.154/30`，請輸入 `ip addr add 192.168.12.154/30 brd + dev eth0`。

若要具備作用中連接，必須設定預設閘道。若要為您的系統設定閘道，請輸入 `ip route add 閘道 IP 位址`。若要轉換某個 IP 位址，請使用 `nat:ip route add nat ip_address via other_ip_address`。

若要顯示所有設備，請使用 `ip link ls`。若只希望顯示運作中介面，請使用 `ip link ls up`。若要列印設備的介面統計值，請輸入 `ip -s link ls device_name`。若要檢是您設備的位址，請輸入 `ip addr`。在 `ip addr` 的輸出中同時也可找到您設備的 MAC 位址相關資訊。若要顯示所有路由，請使用 `ip route show`。

如需使用 `ip` 的詳細資訊，請輸入 `ip help` 或參閱 `ip(8)` 線上文件。`help` 選項也適用於所有 `ip` 物件。例如，假設您要閱讀 `ip addr` 的說明，請輸入 `ip addr help`。請在 `/usr/share/doc/packages/iproute2/ip-cref.pdf` 中尋找 `ip` 的手冊。

以 ping 測試連接

`ping` 指令是測試 TCP/IP 連接運作的標準工具。其使用 ICMP 通訊協定，將小型資料封包 `ECHO_REQUEST` 傳送至目的地主機，要求立即回應。如果有作用，`ping` 會顯示訊息，指示網路連結基本上是正常的。

`ping` 所做的不僅止於測試兩台電腦之間的連接狀態，它還可以提供某些有關於連接品質的基本資訊。您可在 **範例 18.10「指令 ping 的輸出」** [232頁] 中看到 `ping` 輸出的一些範例。倒數第二行包含已傳送封包數、遺失封包數、執行 `ping` 總時間等資訊。

對於目的地，可使用主機名稱或 IP 位址，例如 `ping example.com` 或 `ping 192.168.3.100`。程式會持續傳送封包，直到您按下 `Ctrl + C` 為止。

若您只需要檢查連接功能性，您可以 `-c` 選項限定封包數目。例如，若要將 `ping` 限制於三個封包，請輸入 `ping -c 3 example.com`。

範例 18.10 指令 ping 的輸出

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

兩個封包之間的預設間隔為一秒。若要變更間隔，ping 提供了選項 `-i`。例如，若要將 ping 間隔增加到十秒，請輸入 `ping -i 10 example.com`。

在具備多網路設備的系統中，透過特定介面位址傳送 ping 非常實用。若要實現此功能，請使用 `-I` 選項並提供所選設備的名稱，例如 `ping -I wlan1 example.com`。

如需使用 ping 的選項與詳細資訊，請輸入 `ping -h` 或參閱 `ping (8)` 線上文件。

以 ifconfig 設定網路

`ifconfig` 是傳統網路設定工具。與 `ip` 相反，此指令只能用於介面組態。若您希望設定路由，請使用 `route`。

注: ifconfig 和 ip

`ifconfig` 這個程式已過時。請改用 `ip`。

無疑的，`ifconfig` 會顯示目前作用中介面的狀態。如同您在 範例 18.11 「`ifconfig` 指令的輸出」[233頁]中所見，`ifconfig` 具有排列整齊而詳盡的輸出。輸出第一行亦包含您設備的 MAC 位址、HWaddr 數值等資訊。

範例 18.11 *ifconfig* 指令的輸出

```
eth0      Link encap:Ethernet  HWaddr 00:08:74:98:ED:51
          inet6 addr: fe80::208:74ff:fe98:ed51/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:634735 errors:0 dropped:0 overruns:4 frame:0
          TX packets:154779 errors:0 dropped:0 overruns:0 carrier:1
          collisions:0 txqueuelen:1000
          RX bytes:162531992 (155.0 Mb)  TX bytes:49575995 (47.2 Mb)
          Interrupt:11 Base address:0xec80

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:8559 errors:0 dropped:0 overruns:0 frame:0
          TX packets:8559 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:533234 (520.7 Kb)  TX bytes:533234 (520.7 Kb)

wlan1     Link encap:Ethernet  HWaddr 00:0E:2E:52:3B:1D
          inet addr:192.168.2.4  Bcast:192.168.2.255  Mask:255.255.255.0
          inet6 addr: fe80::20e:2eff:fe52:3b1d/64 Scope:Link
          UP BROADCAST NOTRAILERS RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:50828 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43770 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:45978185 (43.8 Mb)  TX bytes:7526693 (7.1 MB)
```

如需使用 *ifconfig* 的選項與詳細資訊，請輸入 *ifconfig -h* 或參閱 *ifconfig* (8) 線上文件。

以 **route** 設定路由

route 是操作 IP 路由表格的程式。您可使用此指令檢是您的路由組態，並新增或移除路由。

注: **route** 與 **ip**

route 這個程式已過時。請改用 *ip*。

若您需要快速又易於理解的路由組態資訊以判別路由問題，*route* 是特別實用的工具。若要檢視您目前的路由組態，請以 *root* 身份輸入 *route -n*。

範例 18.12 `route -n` 指令的輸出

```
route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags   MSS Window  irtt Iface
10.20.0.0        *                255.255.248.0    U        0 0          0 eth0
link-local        *                255.255.0.0      U        0 0          0 eth0
loopback          *                255.0.0.0        U        0 0          0 lo
default           styx.exam.com    0.0.0.0          UG       0 0          0 eth0
```

如需使用 `route` 的選項與詳細資訊，請輸入 `route -h` 或參閱 `route` (8) 線上文件。

18.6.3 啟動程序檔

除了上述的組態檔之外，還有在機器開機時載入網路程式的各種程序檔。只要系統切換到其中一個 *multiuser runlevels*，就會啟動這些程序檔。在 [表格 18.9「網路程式的一些啟動程序檔」](#) [234頁] 中提供了一些程序檔的說明。

表格 18.9 網路程式的一些啟動程序檔

<code>/etc/init.d/network</code>	這個程序檔可處理網路介面的組態。如果未啟動網路服務，則不會實作任何網路介面。
<code>/etc/init.d/xinetd</code>	啟動 <code>xinetd</code> 。 <code>xinetd</code> 可以用來讓伺服器服務能夠在系統上使用。例如，只要開啟 FTP 連接，它即可啟動 <code>vsftpd</code> 。
<code>/etc/init.d/portmap</code>	啟動 RPC 伺服器所需的連接埠對應程式，如 NFS 伺服器。
<code>/etc/init.d/nfssserver</code>	啟動 NFS 伺服器。
<code>/etc/init.d/postfix</code>	控制後置程序。
<code>/etc/init.d/ypserv</code>	啟動 NIS 伺服器。
<code>/etc/init.d/ypbind</code>	啟動 NIS 用戶端。

18.7 smpppd 做為撥號助理

部份家庭使用者沒有連接到網際網路的專線。而是使用撥號連接。視撥號方法 (ISDN 或 DSL) 而定，連接是由 `ippd` 或 `pppd` 來控制。基本上，連接所需做的事就是正確地啟動這些程式。

如果您有單一速率的連接，不會產生撥號連接的其他成本，請直接啟動個別的精靈。使用 KDE Applet 或指令行介面以控制撥號連接。如果網際網路閘道不是您所使用的主機，您可能需要透過網路主機來控制撥號連接。

這裏將會需要 `smpppd`。它會為輔助程式提供一致的介面並以兩個方向運作。首先，它會撰寫所需的 `pppd` 或 `ippd` 程式，然後控制其撥號內容。其次，它會提供各個提供者給使用者程式，並傳輸關於連接目前狀態的資訊。因為 `smpppd` 也可以透過網路控制，所以它適合從私人子網路的工作站中，控制連至網際網路的撥號連接。

18.7.1 設定 smpppd

`smpppd` 所提供的連接會自動由 YaST 設定。實際的撥號程式 `KInternet` 與 `cinternet` 也是預先設定的。手動設定只需設定 `smpppd` 的其他功能，例如遠端控制。

`smpppd` 的組態檔為 `/etc/smpppd.conf`。根據預設，它不會啟用遠端控制。此組態檔最重要的選項為：

`open-inet-socket = yes/no`

若要透過網路控制 `smpppd`，此選項必須設定為 `yes`。`smpppd` 所傾聽的埠為 3185。如果此參數設為 `yes`，則也應該設定 `bind-address`、`host-range` 以及 `password` 參數。

`bind-address = ip address`

如果主機具有數個 IP 位址，請使用此參數以決定那個 IP 位址 `smpppd` 應該接受連接。預設為監聽所有位址。

`host-range = 最小 ip 最大 ip`

`host-range` 參數會定義網路範圍。在此範圍中的 IP 位址之主機擁有 `smpppd` 的存取權。所有不在此範圍中的主機都將拒絕存取。

`password = password`

透過指定密碼，將用戶端限制為授權的主機。因為這是純文字的密碼，您不應該高估它所提供的安全性。如果沒有指定密碼，則所有的用戶端都將允許存取 `smpppd`。

`slp-register = yes/no`

使用此參數，`smpppd` 服務就可以透過 SLP 在網路中宣告。

有關 `smpppd` 的詳細資訊，請參閱 `smpppd(8)` 和 `smpppd.conf(5)`。

18.7.2 設定 KInternet 與 cinternet 以供遠端使用

KInternet 與 cinternet 可用於控制本地或遠端 `smpppd`。cinternet 是圖形 KInternet 的指令行對應部分。若要準備這些公用程式以供遠端 `smpppd` 使用，請手動編輯 `/etc/smpppd-c.conf` 組態檔或使用 KInternet。此檔案只使用四個選項：

`sites = list of sites`

在此，可看出要搜尋 `smpppd` 的前端。這些前端將會依這裏所指定的順序來測試選項。local 選項可命令建立與本地 `smpppd` 的連線。gateway 選項可指向閘道上的 `smpppd`。config-file 指示應與 `/etc/smpppd-c.conf` 檔案中之 `server` 與 `port` 選項所指定的 `smpppd` 建立連線。slp 可命令前端連接至透過 SLP 找到的 `smpppd`。

`server = server`

此處指定 `smpppd` 所執行的主機。

`port = 連接埠`

此處指定執行 `smpppd` 所使用的連接埠。

`password = password`

插入為 `smpppd` 所選取的密碼。

如果 `smpppd` 為作用中，您現在可以嘗試存取它，例如，使用 cinternet `--verbose --interface-list`。如果現在碰到任何困難，請參閱 `smpppd-c.conf(5)` 與 `cineternet(8)`。

無線通訊

您可以藉由使用 Linux 系統，與其它電腦、手機或週邊設備進行各種通訊。WLAN (無線區域網路) 可用來建立筆記型電腦網路。藍芽可用來連接個別的系統元件 (滑鼠、鍵盤)、週邊設備、手機、PDA 及個人電腦。IrDA 大多用來與 PDA 或手機進行通訊。通用行動通訊系統 (Universal Mobile Telecommunications System, UMTS, 也稱為 3G) 可提供數種多媒體服務，例如瀏覽 Web 或傳送和接收訊息。本章節將介紹這些技術及其組態設定。

19.1 無線區域網路

無線區域網路已成為行動運算世界中不可或缺的一環！如今，大多數的筆記型電腦都有內建的 WLAN 卡。WLAN 卡所使用的無線通訊 802.11 標準是由 IEEE 組織所制定。此標準最初用於最大傳輸率 2 MBit/s。其間已增加許多新的標準來提高資料傳輸率。這些補充項目定義調變、傳輸輸出及傳輸率等詳細資訊 (請參閱 [表格 19.1 「WLAN 標準綜覽」](#) [237 頁])。此外，很多公司都實作具有專利權或草稿功能的硬體。

表格 19.1 WLAN 標準綜覽

名稱	頻段 (GHz)	最大傳輸率 (MBit/s)	記事
802.11 舊	2.4	2	過時的；實際上無法取得終端設備
802.11a	5	54	較少干涉

名稱	頻段 (GHz)	最大傳輸率 (MBit/s)	記事
802.11b	2.4	11	較不普遍
28.29oz	2.4	54	常見，與 11b 向後相容
802.11n draft	2.4 與/或 5	300	通用

SUSE® Linux Enterprise Desktop 不支援 802.11 舊版網路卡。支援使用 802.11a、802.11b、802.11g 與 802.11n draft 的大多數網路卡。新卡通常符合 802.11n draft 標準，但也有使用 802.11g 的卡。

19.1.1 函數

使用無線網路時，您可以用各種不同的技術和組態來確保快速、高品質的安全連接。不同的作業類型適合不同的設定方式。選擇正確的驗證方法相當困難。可用的加密方法會有不同的優缺點。

基本上，無線網路可分為管理網路和臨機操作網路。管理網路中有一個管理元件，即存取點。在此模式(又稱為基礎結構模式)之下，網路上所有 WLAN 工作站的連接都會通過該存取點，此存取點亦可連接至乙太網路。臨機操作網路中沒有存取點。由於工作站之間可直接進行通訊，因此臨機操作網路通常比受管理網路要快。不過，臨機操作網路中的傳輸範圍及參與工作站的數目相當有限。也不支援 WPA 驗證。因此，通常使用存取點。WLAN 卡甚至可做為存取點。某些網路卡支援此項功能。

驗證

無線網路比有線網路更容易受到攔截和危害，因此各項標準均包含驗證和加密方式。在較早版本的 IEEE 802.11 標準中，可在 WEP 條款下找到這些項目的說明。然而，WEP 已證實不夠安全(請參閱[章節「安全性」](#) [244頁])，WLAN 業者(組成「*Wi-Fi 聯盟*」)已定義一項新的安全標準，稱為 WPA，用來提高 WEP 的安全性。更新的 IEEE 802.11i 標準(又稱為 WPA2，WPA 建立於 802.11i 草擬版本)包含 WPA 及其他驗證和加密方式。

目前管理網路時會使用各種不同的驗證機制，以便確定只有獲得授權的工作站才可進行連接：

開啟

開放的系統，即不需要驗證的系統。任何工作站均可加入網路。但可使用 WEP 加密 (請參閱 [章節「加密」](#) [240頁])。

共用金鑰 (根據 IEEE 802.11)

此程序使用 WEP 金鑰進行驗證。不過，並不建議採用此程序，因為它使 WEP 金鑰更容易受到攻擊。攻擊者只需要截聽工作站與存取點間的通訊達足夠的時間就行了。在驗證過程中，雙方交換相同的資訊，先是以加密的形式，然後是以未加密形式。這樣就可以利用適合的工具重新建構金鑰。由於此方式是以 WEP 金鑰來驗證和加密，因此無法提高網路的安全性。擁有正確 WEP 金鑰的工作站可以驗證、加密及解密。缺乏金鑰的工作站則無法解密已收到的封包。因此，不論是否需要驗證其身份，該工作站均無法進行通訊。

WPA-PSK (根據 IEEE 802.1x)

WPA-PSK (PSK 即 Pre-Shared Key (預先共用金鑰) 的縮寫) 的作用方式與共用金鑰程序相似。所有連接工作站及存取點都要有相同的金鑰。此金鑰長度為 256 位元，且通常以密碼片語的方式輸入。本系統不需要如 WPA-EAP 一樣複雜的金鑰管理，並且更適合個人使用。因此，WPA-PSK 有時稱為 WPA「家用」。

WPA-EAP (根據 IEEE 802.1x)

WPA-EAP 實際上並非驗證系統，而是用來傳送驗證資訊的協定。WPA-EAP 用來保護企業中的無線網路。在私人網路中幾乎很少用到。因此，WPA-EAP 有時稱為 WPA「企業」。

WPA-EAP 需要 Radius 伺服器才能驗證使用者。EAP 提供三種不同方式連接與驗證伺服器：TLS(輸送層安全性)、TTLS(通道傳輸層安全性)，與 PEAP(保護擴展驗證協議)。在 Nutshell 中，這些選項的運作方式如下：

EAP-TLS

TLS 驗證仰賴伺服器和用戶端相互交換憑證。首先，伺服器會向評估它的用戶端提供其憑證。如果用戶端認為該憑證有效，就會接著向伺服器提供其憑證。雖然 TLS 是安全的，它仍需要網路中的工作憑證管理基礎結構。這種基礎結構在私有網路中很難找到。

EAP-TTLS 和 PEAP

TTLS 和 PEAP 都是兩階段的協定。第一個階段建立安全性連線，而第二個階段則交換用戶端驗證資料。它們需要的憑證管理負荷 (如果有的話) 遠低於 TLS。

加密

可使用各種不同的加密方式，防止未經授權者讀取無線網路中交換的資料封包，或進入網路：

WEP (定義於 IEEE 802.11 中)

此標準使用 RC4 加密演算法，最初的金鑰長度為 40 位元，後來增加為 104 位元。視 24 位元的啟始向量是否包含其中而定，其長度通常為 64 位元或 128 位元。然而此標準具有某些弱點。此系統所產生的金鑰可能受到攻擊。儘管如此，使用 WEP 仍然比完全不加密的網路來得好。

某些廠商已實作了非標準「動態 WEP」。其功能與 WEP 完全相同並具有同樣的弱點，唯一的區別在於金鑰會透過金鑰管理服務進行定期變更。

TKIP (定義於 WPA/IEEE 802.11i 中)

此金鑰管理協定定義於 WPA 標準中，使用與 WEP 相同的加密演算法，其弱點則均已消除。因為每個資料封包都有一個新的金鑰，所以攻擊這些金鑰等於白費力氣。TKIP 與 WPA-PSK 必須搭配使用。

CCMP (定義於 IEEE 802.11i 中)

CCMP 說明金鑰管理。通常與 WPA-EAP 搭配使用，但也可配合 WPA-PSK 使用。根據 AES 的規定所進行的加密，比 WEP 標準下的 RC4 加密更安全。

19.1.2 使用 YaST 進行設定

若要設定無線網路卡，請在 YaST 控制中心選取「[網路設備](#)」>「[網路設定](#)」。此時將開啟「網路設定」對話方塊，從中可以設定一般網路設定。如需有關一般網路組態的詳細資訊，請參閱[第 18.4 節「使用 YaST 手動設定網路連線」](#) [202 頁]。系統偵測到的所有網路卡會在「[綜覽](#)」索引標籤中列出。

從清單中選擇無線網路卡並按一下「[編輯](#)」以開啟「網路卡設定」對話方塊。設定在「[位址](#)」索引標籤中使用動態還是靜態 IP 位址。您還可以調整「[一般](#)」和「[硬體](#)」設定(如「[設備啟用](#)」或「[防火牆區域](#)」)以及驅動程式設定。大多數情況下，無需變更預先設定的值。

按「[下一步](#)」前進至無線網路卡特定組態對話方塊。如果您使用的是 NetworkManager (如需詳細資訊，請參閱[第 18.5 節「NetworkManager」](#) [220 頁])，則無需調整無線設備設定，因為這些設定會由 NetworkManager 依需求設定—請

繼續按「下一步」與「是」完成組態設定。如果您僅在特定無線網路中使用電腦，請在此處進行 WLAN 操作的基本設定。

圖形 19.1 YaST：設定無線網路卡



The image shows the 'Wireless Network Card Configuration' window in YaST. The window title is '無線網路卡組態'. Below the title is a subtitle: '可在此處設定無線網路適配器的最重要設定。『操作模式』取決於網路拓撲。模式可以是『臨時操作』（沒有存取點的對等網路）、... 其他。'

The main configuration area is titled '無線裝置設定'. It contains the following fields and controls:

- '操作模式(P):' (Operation Mode): A dropdown menu with '管理的' (Managed) selected.
- '網路名稱 (ESSID)(I):' (Network Name (ESSID)): A text input field with a '掃描網路' (Scan Network) button next to it.
- '驗證模式(A):' (Authentication Mode): A dropdown menu with 'WEP - 開放' (WEP - Open) selected.
- '金鑰輸入類型' (Key Input Type): Radio buttons for '通訊密碼(E)' (WPA-PSK) (selected), 'ASCII(A)' (WPA-EAP), and '16 進位(H)' (WPA-PSK).
- '加密金鑰(E):' (Encryption Key): A text input field.
- '專家設定(E)' (Expert Settings) and 'WEP 金鑰(W)' (WEP Key) buttons.

At the bottom of the window are four buttons: '說明' (Help), '中止(B)' (Cancel), '上一步(B)' (Previous), and '下一步(N)' (Next).

操作模式

工作站可用三種模式來連接 WLAN。適合的模式根據所通訊的網路而有所不同：「*Ad-hoc*」（無存取點的對等網路）、「*Managed*」（由存取點管理的網路），或「*Master*」（您的網路卡當作存取點使用）。若要使用任何 WPA-PSK 或 WPA-EAP 模式，就必須將作業模式設為「受管理」。

網路名稱 (ESSID)

在無線網路中，所有工作站都要有相同的 ESSID 才能互相通訊。在未指定任何項的情況下，網路卡可能會自動選取一個存取點，該存取點可能不是您想要使用的。使用「掃描網路」可獲得可用無線網路的清單。

驗證模式

為網路選取適合的驗證方法：「沒有加密」、「WEP 開放」、「WEP 共用金鑰」、「WPA-EAP」或「WPA-PSK」。如果您選取 WPA 驗證，則必須設定網路名稱 (ESSID)。

金鑰輸入類型

WEP 與 WPA-PSK 驗證方法需要輸入金鑰。必須以「密碼片語」、「ASCII」字串或「16 進位」字串形式輸入金鑰。

WEP 金鑰

可以在此處輸入預設金鑰，也可以按一下「WEP 金鑰」進入進階金鑰組態對話方塊。設定金鑰長度為「128 位元」或「64 位元」。預設值為「128 位元」。在對話方塊底下的清單中，最多可指定四個金鑰，讓您的工作站用來加密。按「設定為預設值」將其中之一設為預設金鑰。除非您對此做變更，否則 YaST 都會使用第一個輸入金鑰為預設金鑰。如果標準金鑰被刪除了，則必須手動標記其他金鑰為預設金鑰。按一下「編輯」來修改現有的清單項目或建立新金鑰。在此例中，快顯視窗會提示您選擇一個輸入類型（「密碼片語」、「ASCII」或「16 進位」）。如果您選取了「密碼片語」，請輸入一個單字或字元，會據此及先前指定的長度建立金鑰。「ASCII」必須輸入 5 個字元以建立 64 位元金鑰；輸入 13 個字元以建立 128 位元金鑰。「16 進位」則必須輸入 10 個字元以建立 64 位元金鑰，或 26 個字元以在 16 進位表示法中建立 128 位元金鑰。

WPA-PSK

若要輸入一個 WPA-PSK 金鑰，請選取「密碼片語」輸入法或「16 進位」。在「密碼片語」模式下，必須輸入 8 至 63 個字元。在「16 進位」模式下，必須輸入 64 個字元。

進階設定

此按鈕會開啟一個對話方塊，說明 WLAN 連接的組態設定細節。通常無需變更預先設定的設定。

通道

應只有「臨機操作」和「主要」模式會用到 WLAN 工作站的工作通道規格。在「管理」模式下，網路卡會自動搜尋可用的通道以連接存取點。在「臨機操作」模式中，必須從提供的通道 (11 至 14 個，取決於您所在的國家) 中選取其中之一，讓您的工作站可與其他工作站通訊。在「主要」模式下，必須決定一個通道，讓您的網路卡可以用它來提供存取點功能。此選項的預設值為「自動」。

位元率

視您網路的效能而定，您可以設定點對點間特定的傳輸位元率。在預設值為「自動」的情況下，系統會試著選擇使用最高的資料傳輸率。有些 WLAN 卡不支援位元率的設定。

存取點

在擁有多個存取點的環境中，只要指定 MAC 位址即可預選其中一個存取點。

使用電源管理

當您出外時，可使用省電技術，使您的電池操作時間達到最久。如需電源管理的詳細資訊，請參閱**第 16 章 電源管理** [163頁]。使用電源管理可能會影響連線品質，增加網路延遲時間。

按「下一步」完成設定。如果已選擇 WPA-EAP 驗證，則需要另一個組態步驟才能在 WLAN 中部署工作站。輸入網路管理員給您的身份證明。若為 TLS，請提供「識別」、「用戶端憑證」、「用戶端金鑰」以及「伺服器憑證」。TTLS 和 PEAP 需要「識別」和「密碼」。「伺服器憑證」和「匿名識別」是選擇性項目。YaST 可搜尋 /etc/cert 下的任何憑證。因此，請將指定給您的憑證儲存到此位置，並將對這些檔案的存取權限限制為 0600 (擁有者讀取和寫入)。按一下「詳細資料」來進入進階驗證對話方塊，以便設定 WPA-EAP。選擇 EAP-TTLS 或 EAP-PEAP 通訊之第二階段的驗證方法。如果您在上一個對話方塊中選取 TTLS，請選擇 any、MD5、GTC、CHAP、PAP、MSCHAPv1 或 MSCHAPv2。若您選取 PEAP，請選擇 any、MD5、GTC 或 MSCHAPv2。如果您無法使用自動決定的設定，請使用「**PEAP 版本**」來強制使用特定 PEAP 執行方式。

重要：無線網路的安全性

記得使用支援的驗證和加密方式，以保護您的網路資料傳輸。第三者可在未加密的 WLAN 連接上截取所有的網路資料。即使是不嚴密的加密 (WEP) 也比不加密來得好。如需詳細資訊，請參閱**章節「加密」** [240頁]和**章節「安全性」** [244頁]。

19.1.3 公用程式

wireless-tools 套件包含允許設定無線區域網路特定參數並取得統計資料的公用程式。如需相關資訊，請參閱http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html。

19.1.4 設定 WLAN 的秘訣與技巧

這些秘訣可協助您調整 WLAN 的速度、穩定性及安全性。

穩定性及速度

無線網路的效能及可靠性，要看連接工作站是否能從其他工作站收到清楚的訊號。牆壁之類的障礙物會大大減弱訊號強度。訊號強度愈弱，傳輸速度愈慢。指令行 (連結品質 欄位) 上的 `iwconfig` 公用程式、`NetworkManager` 或 `KNetworkManager`，可檢查運作時的訊號強度。若您的訊強度出現問題，試著將您的設備安裝在其他地方，或調整您存取點的天線方向。許多 PCMCIA WLAN 卡都有輔助天線，可大幅提高接收度。由廠商指定的速率 (例如 54 MBit/s) 為一額定值，代表推定的最大值。實際上，最大資料產生量還不到該值的一半。

安全性

如果您要建立一個無線網路，請記得在缺乏安全措施的情況下，任何在傳輸範圍內的人都可輕易地進入您的網路。因此，應確定啟用加密方式。所有 WLAN 卡和存取點都支援 WEP 加密。雖然不是安全無虞，但仍足以阻礙可能的攻擊。WEP 通常足夠個人使用。WPA-PSK 則是更佳的選擇，不過較舊型的存取點或路由器並未在其 WLAN 功能中使用 PA-PSK。有些設備只要透過防火牆的更新，即可執行 WPA。此外，儘管 Linux 支援絕大多數硬體元件上的 WPA，但是某些驅動程式不支援 WPA。如果沒有 WPA，則 WEP 仍然比完全不加密來得好。對於需要進階安全性的企業來說，只有在執行 WPA 的情況下才可操作無線網路。

19.1.5 疑難排解

如果您的 WLAN 卡沒有回應，檢查看看您是否已下載所需的韌體。如需詳細資訊，請參閱 `/usr/share/doc/packages/wireless-tools/README.firmware`。

多重網路設備

現代的筆記型電腦通常具備一張網路卡和一張 WLAN 卡。如果您以 DHCP 來設定這兩者 (自動指定位址)，則可能會出現名稱解析和預設閘道的問題。如果您可以偵測到路由器，卻無法瀏覽網際網路，表示已出現此問題。支援資料庫提供本主題的文章，網址是：http://en.opensuse.org/SDB:Name_Resolution_Does_Not_Work_with_Several_Concurrent_DHCP_Clients。

Prism2 網路卡的問題

有許多驅動程式可用於裝有 Prism2 晶片的設備。各種不同的網路卡多多少少都可以與不同的驅動程式配合運作。使用這些卡時，WPA 只能配合 hostap 驅動程式來運作。如果這些網路卡無法順利運作，或完全無法運作，或者您想使用 WPA，請參閱 `/usr/share/doc/packages/wireless-tools/README.prism2`。

19.1.6 如需更多資訊

Linux「無線工具」的開發者 Jean Tourrilhes 在其網頁中提供大量有關無線網路的有用資訊。請參閱http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Wireless.html。

網路中的 SLP 服務

服務位置通訊協定 (SLP) 是開發用來簡化區域網路內的網路用戶端組態。若要設定網路用戶端 (包含所有必要的服務)，一般而言管理員需要對於網路中可用的伺服器有詳細的瞭解。SLP 可讓區域網路中的所有用戶端，都知道選定服務的可用性。支援 SLP 的應用程式可以使用散佈的資訊並可自動設定。

SUSE® Linux Enterprise Desktop 支援使用 SLP 所提供的安裝來源進行安裝，並包含許多具有 SLP 整合支援的系統服務。YaST 和 Konqueror 都包含 SLP 適當的前端介面。您可以使用 SLP 以提供主要的功能給網路上的用戶端，例如系統上的安裝伺服器、檔案伺服器或是列印伺服器。

重要: SUSE Linux Enterprise Desktop 中的 SLP 支援

提供 SLP 支援的服務包含 cupsd、rsyncd、ypserv、openldap2、ksysguardd、saned、kdm、vnc、login、smpppd、rpasswd、postfix 以及 sshd (透過 fish)。

20.1 安裝

依預設僅會安裝一個 SLP 用戶端與若干 slptool。若要透過 SLP 提供服務，請安裝 openslp-server 套件。若要安裝套件，請啟動 YaST，再選取「軟體」>「軟體管理」。接著請選擇「過濾器」>「模式」，再按一下「其他伺服器」。選取 openslp-server。請確認安裝所需套件，完成此安裝程序。

20.2 啟用 SLP

您系統必須執行 `slpd`，才能提供 SLP 服務。如果機器僅做為用戶端執行並且不提供服務，則無需執行 `slpd`。就像大部分在 SUSE Linux Enterprise Desktop 中的系統服務一樣，`slpd` 精靈是利用獨立的 `init` 程序檔來控制的。安裝完成後，依預設此精靈將處於非啟動狀態。若要暫時啟動，請以 `root` 身份執行 `rcslpd start` 或執行 `rcslpd stop` 將其停止。請以 `restart` 或 `status` 執行重新啟動或狀態檢查。若 `slpd` 在開機後應始終處於啟用狀態，請在「YaST「系統」>「系統服務 (Runlevel)」中啟用 `slpd`，或以 `root` 身份執行 `insserv slpd` 指令。這包括要在開機時啟動之服務集中的 `slpd`。

20.3 SUSE Linux Enterprise Desktop 中的 SLP 前端

若要尋找在您網路中透過 SLP 提供的服務，請使用 SLP 前端。SUSE Linux Enterprise Desktop 包含多種前端：

slptool

`slptool` 是簡易的指令行程式，可以在網路中宣告 SLP 查詢或宣告專用的服務。`slptool --help` 可以列出所有可用的選項與功能。您也可以從處理 SLP 資訊的程序檔呼叫 `slptool`。例如，若要尋找在目前網路中自我宣告的所有網路時間伺服器，請執行以下指令：

```
slptool findsrvs service:ntp
```

YaST

在 YaST 中也可使用 SLP 瀏覽器。不過，在 YaST 控制中心中不可使用此瀏覽器。若要啟動此 YaST 模組，請以 `root` 使用者身份執行 `yast2 slp`。按一下使用者介面左側的其他協定，以獲取有關相應服務的詳細資訊。

20.4 透過 SLP 提供服務

許多 SUSE Linux Enterprise Desktop 中的應用程式透過使用 `libslp` 程式庫已經具有整合的 SLP 支援。如果尚未使用 SLP 支援來編譯服務，請使用下列方式之一來透過 SLP 進行編譯：

使用 `/etc/slp.reg.d` 的靜態註冊

針對每個新的服務建立個別的註冊檔。下列是註冊掃描器服務的檔案範例：

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

在此檔案中最重要的一行為 *service URL*，它是以 `service:` 開始。這包含服務類型 (`scanner.sane`) 以及位址 (可在其中找到伺服器可用的服務)。`$HOSTNAME` 會以完整的主機名稱自動取代。接著可以找到相關服務的 TCP 埠名稱，它們之間是以冒號分隔。然後輸入服務應該出現的語言以及註冊期間的秒數。這些都應該使用逗號與服務 URL 分隔。在 0 與 65535 之間設定註冊期間的值。0 會防止註冊。65535 會移除所有的限制。

註冊檔另外還包含 `watch-port-tcp` 與 `description` 兩個變數。`watch-port-tcp` 將 SLP 服務宣告連結至由 `slpd` 檢查到的服務狀態，即相關服務是否啟用。第二個變數是對顯示在適當瀏覽器中的服務，提供更為精確的描述。

使用 `/etc/slp.reg` 的靜態註冊

此方法與使用 `/etc/slp.reg.d` 的程序的唯一區別在於所有服務都會在中心檔案內分組。

以 `slptool` 動態註冊

如果要動態註冊服務而無需組態檔案，請使用 `slptool` 指令行公用程式。這一公用程式還可用於取消註冊現有服務提供而無需重新啟動 `slpd`。

20.5 如需更多資訊

下列來源提供關於 SLP 的進一步資訊：

RFC 2608、2609、2610

RFC 2608 一般會處理 SLP 的定義。RFC 2609 會處理更加詳細的服務 URL 語法，而 RFC 2610 則會透過 SLP 處理 DHCP。

<http://www.openslp.org/>

OpenSLP 計劃的首頁。

`/usr/share/doc/packages/openslp`

此目錄包含所有關於 SLP 的可用文件，包括 `README.SuSE` (含有 SUSE Linux Enterprise Desktop 詳細資訊)、RFC 以及兩個介紹性的 HTML 文件。想要使用 SLP 功能的程式設計師可以在 `openslp-devel` 套件隨附的《*程式設計師指南*》中找到詳細資訊。

使用 NTP 進行時間同步化

NTP (網路時間協定) 機制是一種協定，用於同步化網路上的系統時間。首先，機器可以從提供可靠時間來源的伺服器取得時間。其次，機器本身在網路中可以做為其他電腦的時間來源。這個目標是雙重的，即維護絕對正確的時間，並同步化網路內所有機器的系統時間。

維護精準的系統時間對於許多情況都非重要。內建的硬體 (BIOS) 時鐘通常無法符合像是資料庫等應用程式的需求。手動校正系統時間有可能會造成嚴重的問題，因為，例如時間倒退將可能造成重要應用程式無法正常運作。在網路中，通常需要同步所有機器中的系統時間，而手動調整時間的做法並不可取。ntp 提供了一種用於解決這種問題的機制。它會透過網路中可靠的時間伺服器來持續調整系統時間。它可以進一步管理本地參考的時鐘，例如收音機控制的時鐘。

注

若要透過 Active Directory 啟用時間同步，請遵照 [Joining an AD Domain](#) (↑*Security Guide*) 中的說明。

21.1 使用 YaST 設定 NTP 用戶端

ntp 會預先設定為使用本地電腦時鐘做為時間參考。不過，使用 (BIOS) 時鐘，僅供萬一沒有更精確的時間來源時備用。YaST 使用了 NTP 用戶端的組態。對於為執行的系統，請使用快速組態或進階組態。對於受防火牆保護的系統，進階組態可以開啟 SuSEfirewall2 中的必要連接埠。

21.1.1 進階的 NTP 用戶端組態

您可以手動設定 NTP 用戶端，也可以透過 DHCP 自動獲得網路中可用的 NTP 伺服器清單。如果選擇「透過 DHCP 設定 NTP 精靈」，則無法使用以下介紹的手動選項。

用戶端要查詢的伺服器以及其他時間來源會列在「一般設定」索引標籤的下半部。修改清單時，可依需要使用「新增」、「編輯」以及「刪除」。「顯示記錄」可用來檢視用戶端的記錄檔。

按一下「新增」以新增時間資訊的新來源。在下列對話方塊中，選取進行時間同步化的來源類型。可用的選項如下：

伺服器

另一個對話方塊可讓您選取 NTP 伺服器。在系統開機時，請啟用「用以啟始同步化」來觸發伺服器與用戶端之間的時間資訊同步化。「選項」可讓您指定 ntpd 的其他選項。

使用「存取控制選項」，您可以限制遠端電腦可以使用電腦上執行的精靈來執行的動作。僅當核取「安全性設定」索引標籤上的「將 NTP 服務限制為僅限已設定的伺服器」後，此欄位才處於啟用狀態。這些選項對應於 `/etc/ntp.conf` 中的 `restrict` 子句。例如，`nomodify notrap noquery` 不允許伺服器修改電腦的 NTP 設定，並且不允許使用 NTP 精靈的設陷設備（遠端事件登入功能）。建議對超出您控制範圍（例如在網際網路上）的伺服器使用這些限制。

如需詳細資訊，請參閱 `/usr/share/doc/packages/ntp-doc` (ntp-doc 套件的一部份)。

點

點 (peer)，是指一台建立了對稱關係的機器：它可同時做為時間伺服器與用戶端。若要在相同的網路中使用點而非伺服器，請輸入系統的位址。其餘的對話方塊與「伺服器」對話方塊相同。

收音機時鐘

若要在系統中使用收音機時鐘來進行時間同步化，請在此對話方塊中輸入時鐘類型、單位編號、設備名稱以及其他選項。按一下「驅動程式校正」，即可微調驅動程式。`/usr/share/doc/packages/ntp-doc/refclock.html` 中提供了關於本地無線電時鐘作業的詳細資訊。

外寄廣播

時間資訊與查詢也可以透過網路中的廣播傳輸。請在此對話方塊中輸入廣播所應傳送至的位址。除非您已經有類似收音機控制時鐘的可靠時間來源，否則請勿啟用廣播。

內送廣播

如果您想要讓用戶端透過廣播接收其資訊，請在這些欄位中輸入應該接受的個別封包位址。

圖形 21.1 進階 NTP 組態：安全性設定



在「安全性設定」索引標籤中，決定 `ntpd` 是否應於 `chroot jail` 中啟動。依照預設，會啟用「在 *Chroot Jail* 中執行 *NTP 精靈*」。由於此選項可以防止攻擊者損毀整個系統，因此在 `ntpd` 遭受攻擊時，會有較高的安全性。

「將 *NTP 服務* 限制為僅限已設定的伺服器」可增加系統安全性，方法是禁止遠端電腦檢視並修改電腦的 *NTP* 設定，並且禁止使用用於遠端事件登入的設陷設備。啟用此選項後，這些限制會套用至所有遠端電腦，除非您覆寫「一般設定」

索引標籤的時間來源清單中個別電腦的存取控制選項。對於所有其他遠端電腦，僅允許查詢本地時間。

如果 `SuSEfirewall2` 在作用中 (預設情況下)，請啟用「在防火牆中開啟埠」。如果您讓連接埠保持為關閉，就不可能對時間伺服器建立連接。

21.2 手動設定網路中的 ntp

在網路上使用時間伺服器的最簡單方式就是設定伺服器參數。例如，如果可以從網路存取名為 `ntp.example.com` 的時間伺服器，那麼，請新增以下行，將此伺服器的名稱新增到 `/etc/ntp.conf` 檔案：

```
server ntp.example.com
```

若要新增更多時間伺服器，請以關鍵字伺服器插入其他行。在以 `rcntpd start` 指令啟始化 `ntpd` 後，系統大約會花費一個小時穩定時間，並建立累積記錄檔案以校正本地電腦時鐘。使用累積記錄檔案，就可以在電腦一開機後立即計算硬體時鐘的系統錯誤。它會立即使用校正，使系統時間具有更高的穩定性。

有兩種方法可以將 NTP 機制做為用戶端：首先，用戶端可固定在每段間隔時間後向已知伺服器查詢時間。隨著用戶端的增加，此方法可能造成伺服器的高負載。其次，用戶端可以等待網路中的廣播時間伺服器所送出的 NTP 廣播。此方法具有伺服器品質未知的缺點，而且伺服器送出錯誤的資訊可能造成嚴重的問題。

如果時間是經由廣播取得，就不需要伺服器名稱。在這樣的情形下，請在 `/etc/ntp.conf` 組態檔中輸入 `broadcastclient`。若要完全使用一或多個已知的時間伺服器，請輸入以 `servers` 開頭的名稱。

21.3 設定本地參考時鐘

軟體套件 `ntp` 包含用於連接本地參考時鐘的驅動程式。`ntp-doc` 套件的 `/usr/share/doc/packages/ntp-doc/refclock.html` 檔案中提供了受支援時鐘的清單。每個驅動程式都與數字關聯。在 `ntp` 中，實際組態工作是透過虛擬 IP 位址來執行。把時鐘當成在網路中一樣，將它輸入 `/etc/ntp.conf` 檔案中。因此，會指定特殊的 IP 位址，`127.127.t.u` 給它們。在此，`t` 代表時

鐘的類型並可決定使用哪一個驅動程式，而 u 是代表單位，可決定使用哪一個介面。

一般而言，個別設備都具有描述組態細節的特殊參數。檔案 `/usr/share/doc/packages/ntp-doc/drivers/driverNN.html` (其中的 NN 為驅動程式的編號) 提供了關於特定類型之時鐘的資訊。例如，「type 8」時鐘 (透過序列介面的收音機時鐘) 需要其他可以更精確地指定時鐘的模式。例如，Conrad DCF77 接收器模組具有模式 5。若要使用此時鐘做為偏好的參考，請指定 `prefer` 關鍵字。Conrad DCF77 接收器模組的完整 `server` 行如下所示：

```
server 127.127.8.0 mode 5 prefer
```

其他的時鐘也使用相同的模式。安裝 `ntp-doc` 套件後，便可在 `/usr/share/doc/packages/ntp-doc/` 目錄中找到 `ntp` 的文件。檔案 `/usr/share/doc/packages/ntp-doc/refclock.html` 提供了指向描述驅動程式參數之驅動程式頁面的連結。

使用 NetworkManager

NetworkManager 是筆記型電腦與其他可攜式電腦的理想解決方案。有了 NetworkManager，您在移動中，就不必耽心網路介面的組態設定，也不必考慮如何在有線網路與無線網路之間進行切換。NetworkManager 能夠自動連接到已知的無線網路。還可以管理多個平行的網路連線，然後預設會使用最快的連線。此外，您還可以手動在可用網路之間切換，並使用系統匣中的 Applet 或工具集管理網路連線。

對於筆記型電腦，NetworkManager 預設處於啟用狀態。但是您也可以使用 YaST 隨時啟用或停用它，如第 22.2 節「啟用 NetworkManager」[258頁] 中所述。

22.1 NetworkManager 的使用案例

NetworkManager 提供了複雜且直觀的使用者介面，利用該介面使用者可以輕鬆切換網路環境。但是 NetworkManager 解決方案不適用於下列情況：

- 您的電腦會為網路的其他電腦提供網路服務，例如，它是 DHCP 或 DNS 伺服器。
- 您的電腦為 Xen 伺服器，或您的系統是 Xen 之中的虛擬系統。
- 您要使用 SCPM 來健行網路組態管理。如果要同時使用 SCPM 和 NetworkManager，請在 SCPM 組態中停用網路資源。

22.2 啟用 NetworkManager

如果要使用 NetworkManager 管理網路連線，請啟用「YaST 網路設定」模組中的 NetworkManager。若要啟用 NetworkManager，請執行下列步驟：

- 1 執行 YaST，然後前往「網路設備」>「網路設定」。
- 2 此時將開啟「網路設定」對話方塊。前往「全域選項」索引標籤。
- 3 在「網路設定方法」欄位中，啟用「用 NetworkManager 控制使用者」。
- 4 按一下「完成」。
- 5 選擇用於管理網路連線的方法後，請透過 DHCP 或靜態 IP 位址使用自動組態來設定您的網路卡，或請設定您的數據機(對於撥號連線，請使用「網路設備」>「數據機」)。若要設定內部或 USB ISDN 數據機，請選取「網路設備」>「ISDN」。若要設定內部或 USB DSL 數據機，請選取「網路設備」>「DSL」。

如需使用 YaST 設定網路組態的詳細資訊，請參閱第 18.4 節「使用 YaST 手動設定網路連線」[202頁] 和第 19.1 節「無線區域網路」[237頁]。

啟用 NetworkManager 後，請使用 NetworkManager 設定網路連線，如第 22.3 節「設定網路連線」[258頁] 中所述。

如果要停用 NetworkManager 並以傳統方式控制網路，請選擇「網路設定方法」欄位中的「使用 *ifup* 的傳統方法」選項。

22.3 設定網路連線

在 YaST 中啟用 NetworkManager 後，請透過「GNOME 控制中心」或 KDE 4 的「個人設定」中的可用對話方塊設定網路連線。如果使用 GNOME，請從主功能表啟動「GNOME 控制中心」，然後選取「系統」>「網路組態」以開啟「網路組態」對話方塊。如果使用 KDE，請從主功能表按一下「設定桌面」啟動「個人設定」，然後選取「進階」>「網路設定」以開啟「網路設定」對話方塊。

圖形 22.1 GNOME 網路組態對話方塊



或者，您也可以透過系統匣中的 NetworkManager Applet/工具集啟動組態對話方塊，方法是按一下「設定」(KDE) 或在 GNOME Applet 上按一下滑鼠右鍵並選取「編輯連線」。

GNOME 與 KDE 4 組態對話方塊會顯示針對所有類型網路連線 (例如有線、無線、行動寬頻、DSL 以及 VPN 連線) 的索引標籤。NetworkManager 還支援連接至 802.1X 受保護的網路。

若要新增連線，請按一下要使用之連線類型的索引標籤，然後按一下「新增」。輸入「連線名稱」以及連線詳細資料。如果每個連線類型可使用多個實體設備 (例如，您的機器配有兩個乙太網路卡或兩個無線介面卡)，請指定設備的「MAC 位址」(硬體位址)，以將連線關聯到此設備。按一下「確定」或「套用」確認您的設定。按一下 NetworkManager Applet 或工具集後，隨即可顯示可用網路清單，新設定的網路連線也將出現在該清單中。

注：隱藏的網路

若要連接至「隱藏」網路 (一種不廣播其服務的網路)，則您必須知道該網路的延伸服務組織別碼 (ESSID)，因為無法自動偵測出該識別碼。在此情況下，請視需要輸入 ESSID 與加密參數。

編輯每個連線時，您還可以定義 NetworkManager 是否應自動使用此連線 (啟用「自動連線」) 或對整個系統使用此連線 (啟用「所有使用者都可以使用」)。任何使用者登入之前，此類系統連線可由所有使用者共享，並且在 NetworkManager 啟動後即可使用。要建立並編輯系統連線，需要 root 許可權。

22.4 使用 KDE NetworkManager 工具集

在 KDE 4 中，用於控制 NetworkManager 的 KNetworkManager Applet 已由 NetworkManager 工具集替代。工具集是一些小型應用程式，可以整合至桌面或面板。如果已為 NetworkManager 控制設定了網路，則工具集通常會於桌面環境啟動時自動啟動，並且在系統匣中顯示為一個圖示。

NetworkManager 工具集以圖示顯示目前網路狀態，並使用通知報告發生的變更。使用工具集可以設定新的網路連線、手動選取其他網路連線、停用無線網路或視需要一併切換至離線模式。圖示的外觀視目前網路連線的類型和狀態而定。將滑鼠游標停留在圖示上方即可查看有關該連線的詳細資料。

NetworkManager 將連線分為受信任和不受信任兩種。受信任的連線是您明確選取過的任何網路，除此以外的連線都屬於不受信任。在連線圖示上按一下滑鼠右鍵即可顯示您過去曾經至少用過一次之連線的清單。功能表中已標記的就是目前使用的連接。

按一下任意連線 Applet 可隨時選擇其他網路連線。這個選擇的優先程度高於自動選取的網路。只要選擇的網路處於可用的狀態，就會一直使用該網路，這表示即使插入網路纜線，也不會自動切換成有線網路連線。

22.5 使用 GNOME NetworkManager Applet

在 GNOME 中，可以使用 GNOME NetworkManager Applet 控制 NetworkManager。如果已為 NetworkManager 控制設定了網路，則 Applet 通常會於桌面環境啟動時自動啟動，並會在系統匣中顯示一個圖示。

如果系統匣未顯示 GNOME NetworkManager Applet，則 Applet 可能未啟動。按 Alt + F2 並輸入 nm-applet 可手動啟動。

22.5.1 連接至有線網路

如果電腦已藉由網路纜線連接至現有網路，請使用 NetworkManager applet 來選擇網路連接。

- 1 在 applet 圖示上按一下滑鼠左鍵，以顯示可用網路的功能表。功能表中已選取的就是目前使用的連接。
- 2 若要切換到其他網路，請先在清單中選擇該網路。
- 3 若要關閉所有網路連接 (無論是有線還是無線)，請在 applet 圖示上按一下滑鼠右鍵，然後取消勾選「啟用網路」。

22.5.2 連接至無線網路

可用的可見無線網路均列在「無線網路」下的 GNOME NetworkManager Applet 功能表中。每個網路的訊號強度也會顯示在功能表中。加密的無線網路以盾牌圖示標明。

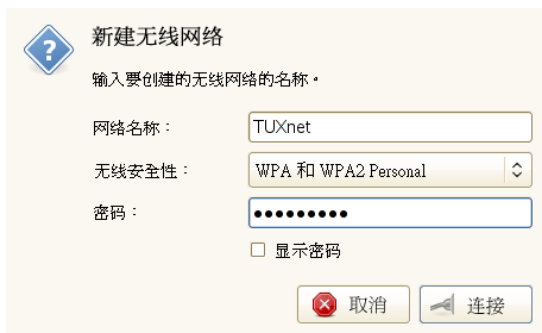
過程 22.1 連接至無線網路

- 1 若要連接至某個無線網路，請按一下 applet 圖示，然後從可用無線網路清單中選擇一項。
- 2 如果該網路已加密，則會開啓一個對話方塊。請選擇網路使用的「無線安全性」類型，然後輸入相應的「密碼」。
- 3 若要連接到因不廣播服務設置識別碼 (ESSID) 而無法自動偵測到的網路，請按一下 NetworkManager 圖示，選擇「連接到其他無線網路」。
- 4 在開啟的對話方塊中，輸入 ESSID，必要時設定加密參數。
- 5 若要停用無線網路，請在 Applet 圖示上按一下滑鼠右鍵，取消勾選「啟用無線」。在飛機上或者在其他任何不允許使用無線網路的環境，此方法非常有用。

22.5.3 將無線網路卡設定成存取點

如果您的無線網路卡支援存取點模式，您可以使用 NetworkManager 來設定組態。

- 1 按一下「[建立新的無線網路](#)」。



- 2 新增網路名稱，並在「[無線安全性](#)」對話方塊中設定加密。

重要: 不保護無線網路會帶來安全風險

如果將「[無線安全性](#)」設定為「無」，則所有人都可以連接到您的網路、重複使用您的連線並攔截您的網路連線。若要限制存取點的存取，並保全您的連線，請使用加密。您可以在各種 WEP 與基於 WPA 加密之間加以選擇。如果不確定哪一種技術最適合您，請參閱[章節「驗證」](#) [238頁]。

22.6 NetworkManager 與 VPN

NetworkManager 支援多項虛擬私人網路 (VPN) 技術：

- NovellVPN—套件 NetworkManager-novellvpn
- OpenVPN—套件 NetworkManager-openvpn
- vpnc (Cisco)—套件 NetworkManager-vpnc

- PPTP (點對點通道通訊協定)—套件 `NetworkManager-pptp`

若要透過 `NetworkManager` 使用 VPN，請先安裝相應的 VPN 套件。每項 VPN 技術需要兩個套件：其中一個為上述套件之一 (提供對 `NetworkManager` 的一般支援)，另一個為針對您 Applet 的個別桌面特定的套件。

對於 KDE，請選擇下列選項之一：

- 針對 `KNetworkManager`—套件的 `NovellVPN` 支援
`NetworkManager-novellvpn-kde4`
- 針對 `KNetworkManager`—套件的 `OpenVPN` 支援
`NetworkManager-openvpn-kde4`
- 針對 `KNetworkManager`—套件的 `vpnc` (Cisco) 支援
`NetworkManager-vpnc-kde4`

針對 KDE 的 PPTP 支援目前尚不可用，但即將可用。

對於 GNOME，請選擇下列選項之一：

- 針對 GNOME `NetworkManager` Applet—套件的 `NovellVPN` 支援
`NetworkManager-novellvpn-gnome`
- 針對 GNOME `NetworkManager` Applet—套件的 `OpenVPN` 支援
`NetworkManager-openvpn-gnome`
- 針對 GNOME `NetworkManager` Applet—套件的 `vpnc` (Cisco) 支援
`NetworkManager-vpnc-gnome`
- 針對 GNOME `NetworkManager` Applet—套件的 PPTP (點對點通道通訊協定) 支援
`NetworkManager-pptp-gnome`

安裝完套件後，請依第 22.3 節「設定網路連線」[258頁]中所述設定 VPN 連線。

22.7 NetworkManager 和安全性

`NetworkManager` 將無線連線分為受信任和不受信任兩種。受信任的連線是過去您明確選取過的任何網路，除此以外的連線都屬於不受信任。受信任的連線以

存取點的名稱和 MAC 位址來識別。使用 MAC 位址可確保別的存取點不能使用受信任連線的名稱。

NetworkManager 會定期掃描是否有可用的無線網路。如果找到多個受信任的網路，便自動選取最近使用的那個網路。如果所有網路都不受信任，NetworkManager 會等待您做出選擇。

如果加密設定變更，但名稱和 MAC 位址未變，NetworkManager 會嘗試進行連接，但它會先要求您確認新的加密設定並提供所有更新，例如新的金鑰。

NetworkManager 可識別兩種類型的連線：使用者連線與系統連線。使用者連線是第一位使用者登入時 NetworkManager 可使用的連線，要求使用者提供所有必要的身份證明。使用者登出後，連線即會斷開，並從 NetworkManager 中移除。任何使用者登入之前，定義為系統連線的連線可由所有使用者共享，並且在 NetworkManager 啟動後即可使用。對於系統連線，必須在連線建立時提供所有身份證明。此類系統連線可用於自動連接要求驗證的網路。如需有關如何設定使用者或系統連線的資訊，請參閱第 22.3 節「設定網路連線」[258頁]。

如果從使用無線連線切換成離線模式，NetworkManager 會將 ESSID 設為空白，以確保該網路卡斷開連線。

22.7.1 儲存密碼與身份證明

如果您不希望每次連接加密網路時都要重新輸入身份證明，則可以使用桌面特定的工具 (GNOME 金鑰圈管理員或 KWalletManager) 將加密的身份證明儲存在磁碟中，並以主密碼加以保護。如需 GNOME 金鑰圈管理員的詳細資訊，請參閱第 2.4.3 節「Managing Keyrings」(第 2 章 *Customizing Your Settings*, ↑GNOME 使用者指南)。

NetworkManager 也可從憑證儲存區取回安全連線 (例如，加密的有線、無線或 VPN 連線) 的憑證。若需更多資訊，請參考第 13 章 *Certificate Store* (↑*Security Guide*)。

另一種方法是使用 Novell CASA 的 Single Sign-on 功能。Single Sign-on 是一種存取控制方法。藉由這一方法，使用者在經過一次驗證後，即可存取多個軟體系統的資源。如果您的系統已設定了 Novell CASA，則 NetworkManager 將不會要求提供附加密碼來解除 GNOME 金鑰圈管理員的鎖定。當使用者登入到桌面時，金鑰圈會自動解除鎖定。如需關於 Novell CASA 的詳細資訊，請參閱

<http://developer.novell.com/wiki/index.php/Special:Downloads/casa>。

22.8 常見問題解答

以下提供了有關使用 NetworkManager 設定特殊網路選項的常見問題。

如何將連線關聯到特定設備？

依預設，NetworkManager 中的連線是特定於設備類型的：它們適用於具有相同類型的所有實體設備。如果每個連線類型可使用多個實體設備 (例如，您的機器配有兩個乙太網路卡)，則您可明確指定特定設備的硬體位址 (或 MAC 位址)，以將連線關聯到該類設備。

使用 applet/widget 中的「連線資訊」，或使用指令行工具 (如 nm-tool 或 ifconfig) 的輸出，來查詢設備的 MAC 位址。然後啟動用於設定網路連線的對話方塊，方法是：在 GNOME 控制中心內，使用「系統」>「網路組態」，或者在 KDE 4 內，透過「個人設定」中的「進階」>「網路設定」。選擇要修改的連線，然後按一下「編輯」。在「有線」或「無線」索引標籤中輸入設備的「MAC 位址」，並按一下「確定」確認您的變更。

在偵測到多個存取點具有相同 ESSID 的情況下，如何指定特定的存取點？

如果有具有不同無線頻段 (a/b/g/n) 的多個存取點可用，系統預設會自動選擇訊號最強的存取點。要置換此存取點，請在設定無線連線時使用「BSSID」欄位。

基本服務組識別碼 (BSSID) 可唯一識別每個基本服務組。在基礎結構基本服務組中，BSSID 是無線存取點的 MAC 位址。在獨立的 (臨機操作) 基本服務組中，BSSID 是由 46 位元隨機數字產生的本地管理的 MAC 位址。

啟動用於設定網路連線的對話方塊，方法是：在 GNOME 控制中心內，使用「系統」>「網路組態」，或者在 KDE 4 內，透過「個人設定」中的「進階」>「網路設定」。選擇要修改的無線連線，然後按一下「編輯」。在「無線」索引標籤中，輸入 BSSID。

如何與其他電腦共享網路連線？

主要設備 (連接至網際網路的設備) 不需要任何特殊組態。不過，您需要以如下方式設定連接至本地 Hub 或機器的設備：

1. 啟動用於設定網路連線的對話方塊，方法是：在 GNOME 控制中心內，使用「系統」>「網路組態」，或者在 KDE 4 內，透過「個人設定」中的「進階」>「網路設定」。選擇要修改的連線，然後按一下「編輯」。切換到「IPv4 設定」索引標籤。從「方法」下拉式清單中，選擇「與其他電腦共享」。這樣可讓 IP 流量轉遞並會執行設備上的 DHCP 伺服器。確認您在 NetworkManager 中所做的變更。
2. 由於 DHCP 伺服器使用連接埠 67，請確定防火牆未封鎖此連接埠：機器共享連線時，啟動 YaST 並選取「安全性與使用者」>「防火牆」。切換到「允許的服務」類別。如果「DHCP 伺服器」尚未顯示為「允許的服務」，則從「要允許的服務」中選取「DHCP 伺服器」，然後按一下「新增」。確認您在 YaST 中所做的變更。

如何使用動態位址 (DHCP、PPP 與 VPN) 提供靜態 DNS 資訊？

如果 DHCP 伺服器提供的 DNS 資訊 (與/或路由) 無效，則可將其置換。啟動用於設定網路連線的對話方塊，方法是：在 GNOME 控制中心內，使用「系統」>「網路組態」，或者在 KDE 4 內，透過「個人設定」中的「進階」>「網路設定」。選擇要修改的連線，然後按一下「編輯」。切換到「IPv4 設定」索引標籤，然後從「方法」下拉式清單中，選擇「僅限自動位址 (DHCP)」。在「DNS 伺服器」與「搜尋網域」欄位中輸入 DNS 資訊。按一下「路由」新增其他路由或置換自動路由。確認您的變更。

如何在使用者登入之前將 NetworkManager 連接到受密碼保護的網路？

定義用於此目的的系統連線。若需更多資訊，請參考 [第 22.7 節「NetworkManager 和安全性」](#) [263頁]。

22.9 疑難排解

可能發生連線問題。與 NetworkManager 有關的一些常見問題包括 Applet 未啟動、缺少 VPN 選項和使用 SCPM 的問題。解析和預防這些問題的方法會視使用的工具而定。

NetworkManager 桌面 Applet/工具集不啟動

如果已為 NetworkManager 控制設定了網路，則 GNOME NetworkManager Applet 或 KDE NetworkManager 工具集應自動啟動。如果 Applet/工具集不啟動，請檢查 YaST 中的 NetworkManager 是否啟用，如 [第 22.2 節「啟用 NetworkManager」](#) [258頁] 中所述。然後確定針對桌面環境的相應套件是否

也已安裝。如果使用的是 KDE 4，則套件為 `NetworkManager-kde4`。針對 GNOME 使用者的套件為 `NetworkManager-gnome`。

如果安裝了 GNOME 桌面 Applet 但由於某些原因未執行 (例如不小心結束)，請使用指令 `nm-applet` 手動啟動。

如果 KDE 4 系統匣未顯示網路連線的任何圖示 (可能是由於在 YaST 中使用 NetworkManager 從靜態網路組態切換到了使用者控制的組態)，請按照以下方法將 NetworkManager 工具集新增至面板：在面板中的空白修補程式上按一下滑鼠右鍵，然後選取「**面板選項**」>「**新增工具集**」。(如果桌面物件目前處於鎖定狀態，可能需要先按一下「**解除鎖定工具集**」才能新增物件。) 請在顯示的對話方塊中選取「*NetworkManager*」並按一下「**新增工具集**」。

NetworkManager Applet/工具集不包含 VPN 選項

針對 NetworkManager 的支援、Applet 與適用於 NetworkManager 的 VPN 均在各自獨立套件中發佈。如果您的 NetworkManager Applet/工具集不包含 VPN 選項，請檢查是否安裝了具有支援 VPN 技術之 NetworkManager 的套件。若需要更多的資訊，請參閱第 22.6 節「**NetworkManager 與 VPN**」[262頁]。

SCPM 不切換網路組態

您可能是將 SCPM 搭配 NetworkManager 一起使用。NetworkManager 目前無法搭配 SCPM 設定檔一起使用。當 SCPM 設定檔也會變更網路設定時，請勿一起使用 NetworkManager 和 SCPM。如果要同時使用 SCPM 和 NetworkManager，請在 SCPM 組態中停用網路資源。

無可用的網路連線

如果已正確設定網路連線，且網路連線的所有其他元件 (路由器等) 也均已設定並正在執行中，有時需要重新啟動電腦的網路介面。若要執行此操作，請以 root 身份登入指令行並執行 `rcnetwork restart`。

22.10 如需更多资讯

您可以在下列網站和目錄中找到更多有關 NetworkManager 的資訊：

- <http://www.gnome.org/projects/NetworkManager/>—NetworkManager 專案頁面

- 如需有關 KDE NetworkManager 工具集的詳細資訊，請參閱 <http://userbase.kde.org/KNetworkManager>。
- 同時也請參閱下列目錄中有關 NetworkManager、GNOME NetworkManager Applet 與 KDE NetworkManager 工具集的最新資訊：`/usr/share/doc/packages/NetworkManager/`、`/usr/share/doc/packages/NetworkManager-kde4/` 以及 `/usr/share/doc/packages/NetworkManager-gnome/`

Samba

使用 Samba 可以將 Unix 機器設定為 Mac OS X、Windows 以及 OS/2 機器的檔案與列印伺服器。Samba 已經是一個開發至完全成熟且相當複雜的產品。使用 YaST、SWAT (一種 Web 介面)，或透過手動編輯組態檔的方式設定 Samba。

23.1 術語

下列為 Samba 文件和 YaST 模組中常用的詞彙。

SMB 通訊協定

Samba 使用基於 NetBIOS 服務的 SMB (伺服器訊息區塊) 通訊協定。由於 Microsoft 發行了此通訊協定，因此其他的軟體製造商可以建立連接至 Microsoft 網域網路的連線。使用 Samba，SMB 通訊協定就可以在 TCP/IP 通訊協定上運作，因此 TCP/IP 通訊協定必須安裝在所有的用戶端上。

CIFS 通訊協定

CIFS (一般網際網路檔案系統) 通訊協定是 Samba 所支援的另一種通訊協定。CIFS 定義用於網路上的標準遠端檔案系統存取通訊協定，讓使用者群組可以透過網路分工合作和共享文件。

NetBIOS

NetBIOS 是用來在提供名稱服務的機器之間進行通訊的軟體介面 (API)。它允許連接至網路的機器保留自己的名稱。在保留後，就可以使用名稱來定址這些機器。在此沒有檢查名稱的中央程序。在網路上的任何機器都可以保留它所需的任何數量名稱，只要這些名稱尚未使用。可以針對不同的網路結構實作 NetBIOS 介面。有一個與網路硬體一起緊密運作的執行程序，稱為

NetBEUI，不過這通常稱為 NetBIOS。與 NetBIOS 一起執行的網路通訊協定是 Novell 的 IPX (經由 TCP/IP 的 NetBIOS) 與 TCP/IP。

經由 TCP/IP 所傳送的 NetBIOS 名稱，與 `/etc/hosts` 中所使用的名稱，或由 DNS 所定義的名稱完全不相同。NetBIOS 使用自己完全獨立的命名慣例。不過，一般建議使用與 DNS 主機名稱相對應的名稱，如此可使管理較為輕鬆，或者可本地使用 DNS。Samba 預設是使用此對應名稱。

Samba 伺服器

Samba 伺服器可為用戶端提供 SMB/CIFS 服務和 NetBIOS over IP 命名服務。對於 Linux，Samba 伺服器有三種精靈可用：smnd (用於 SMB/CIFS 服務)、nmbd (用於命名服務) 與 winbind (用於驗證)。

Samba 用戶端

Samba 用戶端是透過 SMB 通訊協定使用 Samba 伺服器所提供之 Samba 服務的系統。所有一般的作業系統 (例如 Mac OS X、Windows 以及 OS/2) 都支援 SMB 通訊協定。TCP/IP 通訊協定必須安裝在所有的電腦上。Samba 提供適用於不同 UNIX 類別的用戶端。就 Linux 而言，有一個 SMB 的核心模組，允許在 Linux 系統層級上整合 SMB 資源。您不必為 Samba 用戶端執行任何精靈。

共享

SMB 伺服器透過共享方式向用戶端提供資源。共享是指印表機和位在伺服器上的目錄及其子目錄。它是利用名稱來輸出，並且可藉由其名稱來存取。共享名稱可以設成任何名稱——它並不需是輸出目錄的名稱。也會指定一個名稱給印表機。用戶端可以透過其名稱存取印表機。

DC

領域控制器 (DC) 是處理網域中帳戶的伺服器。進行資料複製時，可在一個網域中使用其他領域控制器。

23.2 設定 Samba 伺服器

如需設定 Samba 伺服器，請參閱 SUSE Linux Enterprise Server 文件。

23.3 設定用戶端

用戶端只能透過 TCP/IP 存取 Samba 伺服器。NetBEUI 與透過 IPX 的 NetBIOS 無法與 Samba 一起使用。

23.3.1 使用 YaST 設定 Samba 用戶端

設定 Samba 用戶端以存取 Samba 伺服器上的資源 (檔案或印表機)。在「網路服務」>「Windows 網域成員」對話方塊中輸入網域或工作群組。如果您啟用「Linux 驗證也使用 SMB 資訊」，使用者驗證將會在 Samba 伺服器上執行。在完成所有的設定後，按一下「完成」以完成組態。

23.4 做為登入伺服器的 Samba

在以 Windows 用戶端為主的網路中，通常會建議使用者只註冊一個有效的帳戶與密碼。在以 Windows 為基礎的網路中，這個任務是由主要網域控制器 (PDC) 來處理。您可使用設定為 PDC 的 Windows NT 伺服器，但此任務亦可透過 Samba 伺服器的協助完成。在 `smb.conf` 的 `[global]` 區段中必須編輯的項目如 [範例 23.1「在 `smb.conf` 中的全域區段」](#) [271 頁] 所示。

範例 23.1 在 `smb.conf` 中的全域區段

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

如果使用加密密碼進行驗證，則 Samba 伺服器必須能夠處理這些密碼。在 `[global]` 區段中的 `encrypt passwords = yes` 項目可以啟用此功能 (加上 Samba 版本 3，現在這個是預設值)。除此之外，必須準備符合 Windows 加密格式的使用者帳戶與密碼。請使用 `smbpasswd -a name` 指令來執行此動作。使用下列指令為電腦建立網域帳戶 (Windows 網域概念所需)：

範例 23.2 設定機器帳戶

```
useradd hostname\$$
smbpasswd -a -m hostname
```

使用 `useradd` 指令，就會加上貨幣符號。當使用 `-m` 參數時，`smbpasswd` 指令就會自動插入這個符號。加備註的組態範例 (`/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`) 包含自動化此任務的設定值。

範例 23.3 機器帳戶的自動化設定

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \  
-s /bin/false %m$
```

若要確定 Samba 可以正確地執行此程序檔，請選擇具有所需管理員權限的 Samba 使用者。若要這麼做，請選取一個使用者，並將它加入 `ntadmin` 群組。在此之後，就可以使用下列指令指定 `Domain Admin` 狀態給所有屬於此 Linux 群組的使用者：

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

如需關於此主題的詳細資訊，請參閱 `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` 中第 12 章的「Samba HOWTO 文件集」。

23.5 如需更多資訊

可在數位文件中取得更多有關 Samba 的資訊。如果已安裝 Samba 文件，若要取得更多的線上文件與範例，請在指令行輸入 `apropossamba` 以顯示一些手冊頁或直接瀏覽 `/usr/share/doc/packages/samba` 目錄。在 `examples` 子目錄中有加備註的範例組態 (`smb.conf.SuSE`)。

Samba 團隊所提供的「Samba HOWTO 文件集」，包含疑難排解一節。除此之外，文件的第五部份提供檢查組態的逐步指南。安裝 `samba-doc` 套件後，您可以在 `/usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf` 中找到 Samba HOWTO 文件集。

另請參閱 <http://en.opensuse.org/Samba> 中 openSUSE wiki 的 Samba 頁面。

使用 NFS 共享檔案系統

透過網路配送和共享檔案系統在公司環境中極為常見。NFS 是一個成熟的系統，也可以與黃頁通訊協定 NIS 搭配使用。如需可與 LDAP 搭配使用並且也可 Kerberos 化的更安全通訊協定，請勾選 NFSv4。

與 NIS 一起使用 NFS 會讓網路對使用者而言透明化。可以使用 NFS 透過網路配送任意檔案系統。如果安裝妥當，使用者將會發現，不論他們目前使用的是哪個終端機，他們始終處於同一個環境中。

24.1 安裝必要軟體

您不需安裝其他軟體就能將您的主機設定為 NFS 用戶端。設定 NFS 用戶端時所需的套件都已預設安裝。

24.2 以 YaST 輸入檔案系統

被授權的使用者可從 NFS 伺服器裝載 NFS 目錄到自己的檔案目錄樹。可以使用 YaST 模組「*NFS 用戶端*」實現此目的。按一下「*新增*」並輸入 NFS 伺服器的主機名稱、要輸入的目錄和在本地裝載此目錄的定點。按一下第一個對話方塊中的「*完成*」後，變更就會生效。

在「*NFS 設定*」索引標籤中按一下「*在防火牆中開啟埠*」以開啟防火牆，可從遠端電腦存取服務。防火牆的狀態顯示於核取方塊旁。使用 NFSv4 時，請確定

啟用「啟用 *NFSv4*」核取方塊，並且「*NFSv4* 網域名稱」中包含的值與 *NFSv4* 伺服器使用的值相同。預設網域為 `localdomain`。

請按一下「完成」儲存變更。請參閱圖形 24.1 「使用 YaST 的 *NFS* 用戶端組態」[274頁]。

組態將會寫入 `/etc/fstab` 中，並會裝載指定的檔案系統。當您稍後啟動 YaST 組態用戶端時，它也會從這個檔案讀取現有組態。

圖形 24.1 使用 YaST 的 *NFS* 用戶端組態



24.3 手動輸入檔案系統

您也可以從 *NFS* 伺服器手動輸入檔案系統。先決條件是執行中的 **RPC** 埠對應程式，以 `root` 身份輸入 `rpcbind start` 便可啟動。一旦符合先決條件，就以下列方式使用 `mount` 指令，讓遠端輸入的檔案系統就能像本地硬碟一樣地在檔案系統內進行裝載：

```
mount host:remote-path local-path
```

例如，如果要輸入來自機器 `nfs.example.com` 的使用者目錄，請使用以下指令：

```
mount nfs.example.com:/home /home
```

24.3.1 使用自動裝載服務

除了一般本地設備的裝載之外，`autofs` 精靈也可以用來裝載遠端檔案系統。若要這麼做，請將下列項目加入您的 `/etc/auto.master` 檔案：

```
/nfsmounts /etc/auto.nfs
```

如果 `auto.nfs` 檔案能適當完成，`/nfsmounts` 目錄此後就可做為用戶端上所有 NFS 裝載作業的根部。`auto.nfs` 這個名稱的選擇是以方便為考量，您可以自行選擇任何名稱。在選取的檔案中 (若不存在，則請您建立) 加入所有 NFS 裝載作業的項目，如以下範例所示：

```
localdata -fstype=nfs server1:/data  
nfs4mount -fstype=nfs4 server2:/
```

請以 `rcautofs start` 啟用設定。在此範例中，`/data` 伺服器 1 的 `/nfsmounts/localdata` 目錄會裝載 NFS，而伺服器 2 的 `/nfsmounts/nfs4mount` 會裝載 NFSv4。

如果 `/etc/auto.master` 檔案在 `autofs` 服務的執行過程中接受編輯，則自動裝載器必須重新啟動才能使變更生效。請以 `rcautofs restart` 執行此操作。

24.3.2 手動編輯 `/etc/fstab`

`/etc/fstab` 中典型的 NFSv3 裝載項目如下：

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

您也可以手動將 NFSv4 的裝載加入 `/etc/fstab` 檔案。對於這些裝載項目，請在第三欄中使用 `nfs4` (而非使用 `nfs`)，並確定遠端檔案系統在第一欄中的 `nfs.example.com:` 之後以 `/` 提供。`/etc/fstab` 中的 NFSv4 裝載範例如下：

```
nfs.example.com:/ /local/pathv4 nfs4 rw,noauto 0 0
```

`noauto` 選項可防止在啟動時自動裝載檔案系統。如果要手動裝載各自的檔案系統，則可以縮短用於裝載的指令，並且僅需提供如下所示的定點：

```
mount /local/path
```

請注意，如果未輸入 `noauto` 選項，系統的啟始化程序檔會在啟動時處理這些檔案系統的裝載。

24.4 NFS 搭配使用 Kerberos

若要為 NFS 使用 Kerberos 驗證，則 GSS 安全性必須啟用。若要這麼做，請在 YaST 起始對話方塊中選取「啟用 GSS 安全性」。請注意，必須有一個工作中的 Kerberos 伺服器才能使用此功能。YaST 不會設定伺服器，而只是使用所提供的功能。如果您要使用 Kerberos 驗證與 YaST 組態，在執行 NFS 組態之前至少需完成以下步驟：

- 1 請確定伺服器和用戶端位於相同的 Kerberos 網域中。這表示它們會存取相同的 KDC (金鑰發佈中心) 伺服器，並會共享它們的 `krb5.keytab` 檔案 (所有機器上的預設位置都是 `/etc/krb5.keytab`)。
- 2 請以 `rcgssd start` 來啟動用戶端上的 `gssd` 服務。

如需有關設定已獲監督 (kerberized) 之 NFS 的詳細資訊，請參閱第 24.5 節「如需更多資訊」[276頁]中的連結。

24.5 如需更多資訊

除了 `exports`、`nfs` 和 `mount` 的線上文件以外，`/usr/share/doc/packages/nfs-tls/README` 中也提供了有關設定 NFS 伺服器和用戶端的資訊。線上文件位於以下網頁文件中：

- 如需詳細的線上技術文件，請造訪 SourceForge [<http://nfs.sourceforge.net/>]。
- 如需設定已監督之 NFS 的指示，請參閱 NFS 第 4 版開放原始碼實作參考 [<http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html>]。

- 如果您有任何關於 NFSv4 的問題，請參閱 Linux NFSv4 常見問題 [<http://www.citi.umich.edu/projects/nfsv4/linux/faq/>]的 FAQ。

檔案同步化

今日有許多人使用數台電腦—在家使用一台、在辦公室使用一台或數台電腦，而且可能在外面還使用筆記型電腦或 PDA。許多檔案都需要各存一份在所有這些電腦上。您可能希望能在每一部電腦上工作、修改檔案，之後還能讓所有的電腦都擁有最新的資料。

25.1 可用的資料同步化軟體

對於永久以快速網路連結的電腦而言，資料同步化不是問題。在此狀況下，使用 NFS 之類的網路檔案系統，並將檔案儲存在伺服器上，所有主機就可以透過網路來存取相同的資料。但如果網路連線品質很差或不是永久的，就無法使用此方法。當您出門在外使用筆記型電腦時，所有需要的檔案複本都必須在本地硬碟上。不過之後需要將修改過的檔案同步化。當您在某一台電腦上修改檔案時，請確定也更新了所有其他電腦上的同一檔案。至於一些零散的複本，則可以使用 scp 或 rsync 手動更新它。然而，如果有許多檔案需要處理，則該程序有可能變得很複雜，而且需要更小心才能避免類似以舊檔案覆寫新檔案的錯誤。

警告：資料遺失的風險

在您開始使用同步化系統管理資料之前，應該先好好的認識要使用的程式並測試其功能。對於重要檔案而言，備份是不可或缺的動作。

手動同步化資料非常耗時，而且是一種容易發生錯誤的任務，要避免這些缺點，可以使用一種以多種方法將此工作自動化的程式。下列摘要讓您概略瞭解這些程式的運作方式和使用方法。如果您打算使用它們，請詳閱程式文件。

25.1.1 CVS

CVS 主要是用來管理程式來源版本，它使得檔案複本可以保留在多台電腦上。因此，它也適用於資料同步化。CVS 負責維護伺服器上的中央儲存庫，檔案及檔案的變更都儲存在此。在本地執行的變更會交付至儲存庫，並且可以利用更新從其他電腦擷取。兩個程序都必須由使用者起始。

CVS 對於錯誤有非常大的彈性與包容性，所以可以應付多台電腦同時進行變更的情況。變更會合併，而且如果在相同行發生變更，就會報告衝突。衝突發生時，資料庫會維持一致的狀態。只有在用戶端主機上才能看到衝突，並加以解決。

25.1.2 rsync

當不需要版本控制，但是大型目錄結構需要透過緩慢的網路連線進行同步化時，rsync 工具針對僅傳送檔案中的變更可提供已開發成熟的機制。這不只包含文字檔，也包含二進位檔。為了偵測檔案之間的不同，rsync 會將檔案分為區塊並計算它們的檢查總數。

偵測變更將需要付出相當大的成本。要同步化的系統應該要具有相當的硬體配備，才能使用 rsync。RAM 尤其重要。

25.2 選取程式時所要考慮的決定性因素

在決定要使用哪個程式時，必須考慮一些重要的因素。

25.2.1 用戶端對伺服器與點對點

配送資料時常使用的模式有兩種。第一個模式是，所有的用戶端都以中央伺服器為準，將其檔案同步化。伺服器至少必須偶爾可以讓所有的用戶端存取。CVS 使用此模式。

另一種可能性就是，讓網路上所有主機都以點對點的方式將彼此間的資料同步化。rsync 實際作用於用戶端模式，但任何用戶端都可以當作伺服器使用。

25.2.2 可攜式

在許多其他的作業系統上也可以使用 CVS 以及 rsync，包含各種 Unix 與 Windows 系統。

25.2.3 互動式與自動化

在 CVS 中，資料同步化是由使用者以手動方式啟動。這讓使用者對於要同步化的資料進行良好的控制，並可輕鬆地處理衝突。然而，如果同步化間隔太長，就比較可能發生衝突。

25.2.4 衝突：發生與解決

即使有數個人員同時在某個大型的程式專案上一起工作，在 CVS 中發生衝突的機率還是相當地少。這是因為文件是在個別的行列上進行合併。當發生衝突時，只有一個用戶端會受到影響。CVS 中的衝突通常都可以輕易解決。

rsync 中則無衝突處理功能。使用者必須小心不要覆寫檔案，並手動解決所有可能的衝突。基於安全著想，可以另外使用 RCS 這一類的版本設定系統。

25.2.5 選取和新增檔案

在 CVS 中，必須分別使用 `cvs` 或 `add` 指令，明確地新增目錄與檔案。這讓使用者對於要同步化的檔案擁有更大的控制權。另一方面，新檔案時常會被忽略，特別是在處理大量的檔案而忽略了 `cvs` 以及 `update` 輸出中的問號時。

25.2.6 歷程

CVS 還有另一項功能，那就是可以重新建構舊的檔案版本。每個變更都可以插入簡短的編輯符號，而且之後可以根據其內容與符號輕易地追蹤檔案的發展。這對論文與程式文字而言，是一種很珍貴的助力。

25.2.7 資料量與硬碟需求

所有相關主機的硬碟都需要有足夠的可用空間來儲存所有分散式的資料。CVS 在伺服器上還需要額外的空間，供儲存庫資料庫使用。檔案歷程記錄也會儲存在伺服器上，因此需要更多的空間。當文字格式的檔案變更時，只需儲存修改過的那幾行。每當變更檔案時，二進位檔案就會需要與該檔案大小相同的額外空間。

25.2.8 GUI

有經驗的使用者通常會從指令行執行 CVS。然而，在 *cervisia* 之類的 Linux 系統中，以及 *wincvs* 之類的其他作業系統中都有圖形使用者介面。許多開發工具 (像是 *kdevelop*) 以及文字編輯器 (像是 *Emacs*) 都支援 CVS。使用這些前端程式的話，衝突的解決方案通常會更容易執行。

25.2.9 使用者親切性

rsync 較易於使用且適合新進人員。CVS 某種程度上較難操作。使用者必須瞭解儲存庫與本地資料之間的互動。對資料的變更應該先在本機與儲存庫合併。這是使用 *cvs* 或 *update* 指令來執行。接著必須使用 *cvs* 或 *commit* 指令將資料傳送回儲存庫。只要瞭解此程序，新進人員就可輕鬆使用 CVS。

25.2.10 防止攻擊的安全性

在傳輸期間，應該保護資料以防攔截和竄改。CVS 與 *rsync* 都可以透過 *ssh* (安全的外圍程式) 來使用，提供安全性以防護此類攻擊。應該避免透過 *rsh* (遠端外圍程式) 執行 CVS。同樣地也不建議在不安全的網路中使用 *pserver* 機制存取 CVS。

25.2.11 針對資料遺失的防護

開發人員使用 CVS 來管理程式專案已經有很長的一段時間，而且極為穩定。因為開發的歷程記錄皆已儲存，所以 CVS 甚至提供保護，防止某些使用者錯誤發生，例如不小心刪除檔案。

表格 25.1 檔案同步化工具的功能：-- = 很差，- = 差或無法使用，o = 中等，+ = 良好，++ = 優異，x = 可用

	CVS	rsync
主/從	C-S	C-S
可攜式	Lin、Un*x、Win	Lin、Un*x、Win
互動	x	x
速度	o	+
衝突	++	o
檔案 Sel。	Sel./file, dir.	目錄
歷程	x	-
硬碟空間	--	o
GUI	o	-
困難度	o	+
攻擊	+(ssh)	+(ssh)
資料損失	++	+

25.3 CVS 簡介

如果經常編輯某些個別檔案，並且以檔案格式儲存，例如 ASCII 文字或程式來源文字，就非常適合使用 CVS 進行同步化。您可以使用 CVS 同步化其他格式的資料，例如 JPEG 檔案，但是會造成大量的資料，因為檔案的所有變體都會永久儲存在 CVS 伺服器上。在這種情形下，CVS 的大部份功能都將無法使用。只有在所有工作站都可以存取同一伺服器時，才能使用 CVS 同步化檔案。

25.3.1 設定 CVS 伺服器

server 是所有有效檔案所在的主機，這包含所有檔案的最新版本。任何靜態工作站都可做為伺服器。可能的話，CVS 儲存庫的資料應該包含在定期備份中。

當設定 CVS 伺服器時，透過 SSH 授予使用者該伺服器的存取權可能是不錯的方式。如果使用者是以 *tux* 的身份登入伺服器，而且 CVS 軟體既安裝在伺服器上也安裝在用戶端上，則必須在用戶端上設定下列環境變數：

```
CVS_RSH=ssh CVSROOT=tux@server:/serverdir
```

`cvsinit` 指令可用來從用戶端啟始 CVS 伺服器。這只需要做一次。

最後，必須為同步化指定名稱。在用戶端上選取或建立一個目錄，專門存放 CVS 所管理的檔案 (目錄也可以是空的)。該目錄的名稱也就是同步化的名稱。在此範例中，該目錄稱為 *synchome*。變更至此目錄，然後輸入下列指令將同步化名稱設為 *synchome*：

```
cvs import synchome tux wilber
```

許多 CVS 指令需要註解。因此，CVS 會啟動編輯器 (如果沒有定義編輯器，則會啟動環境變數 `$EDITOR` 或 `vi` 中所定義的編輯器。) 在指令行先輸入註解，就可以避免編輯器的呼叫，如下列範例所示：

```
cvs import -m 'this is a test' synchome tux wilber
```

25.3.2 使用 CVS

現在可以使用 `cvsco synchome` 從所有主機取出同步化儲存庫。這會在用戶端上建立新的子目錄 *synchome*。若要將變更交付至伺服器，請換至 *synchome* 目錄 (或其中一個子目錄)，然後輸入 `cvscommit`。

根據預設，所有的檔案 (包括子目錄) 都會交付給伺服器。若只要交付個別的檔案或目錄，請在 `cvscommit file1 directory1` 中指定它們。在將它們交付至伺服器前，必須使用像是 `cvsadd file1 directory1` 之類的指令，將

新檔案與目錄新增至儲存庫。接著，使用 `cvscscommit file1 directory1` 交付新增的檔案與目錄。

您更換至另一個工作站時，如果先前的工作階段尚未取出該工作站的同步化儲存庫，請現在取出。

使用 `cvsupupdate` 啟動與伺服器的同步化。依照 `cvsupupdate file1 directory1` 中的個別檔案或目錄來更新。若要查看目前檔案與伺服器上所存版本的差異，請使用 `cvsdiff` 或 `cvsdiff file1 directory1` 指令。使用 `cvcs-nq update` 來查看哪些檔案會受到更新的影響。

以下是更新期間所顯示的一些狀態符號：

U

已更新本地版本。這將會影響伺服器所提供及在本地系統上所遺失的全部檔案。

M

已修改本地版本。如果在伺服器上有一些變更，可以在本地複本中將差異合併。

P

本地版本已使用伺服器版本修補。

C

本地檔案與儲存庫中目前的版本衝突。

?

這個檔案並不存在於 CVS。

M 狀態是指在本地修改過的檔案。或是將本地複本交付至伺服器，或是移除本地檔案後再執行一次更新。在此例中，會從伺服器擷取遺失的檔案。如果您交付本地已修改的檔案，而該檔案是在同一行中變更並交付，則有可能造成衝突，這是以 C 表示。

如果發生這種狀況，請檢視檔案中的衝突記號 (>> 與 <<)，並在兩個版本之間選擇一個。這有可能是相當麻煩的工作，您可能會決定捨棄變更、刪除本地檔案，然後輸入 `cvsup`，以便從伺服器擷取目前的版本。

25.4 rsync 簡介

當有大量的資料需要定期傳輸，但是並無太多變更時，rsync 就非常有用。例如，當建立備份時就非常適用。另一個應用程式則將焦點放在建置伺服器。這些伺服器儲存了網頁伺服器的完整目錄樹狀結構，會定期鏡像處理至 DMZ 中的網頁伺服器。

25.4.1 組態與作業

rsync 可以使用兩個模式來操作。它是用來歸檔或複製資料。若要完成此動作，在目標系統上只需要像是 ssh 的遠端外圍程序。然而，rsync 也可以像 daemon 一樣，用來為網路提供目錄。

rsync 的基本作業模式並不需要任何特殊的組態。rsync 直接允許將完整的目錄鏡像處理至另一個系統。例如，下列指令可在名為 sun 的備份伺服器上建立 tux 主目錄的備份：

```
rsync -baz -e ssh /home/tux/ tux@sun:backup
```

下列指令可用來還原目錄：

```
rsync -az -e ssh tux@sun:backup /home/tux/
```

到此為止，其處理方式與一般的複製工具 (例如 scp) 的處理沒有太大的差別。

rsync 應該在「rsync」模式中操作，才能充分使用其所有的功能。執行方法是在其中一個系統上啟動 rsyncd 精靈。在 /etc/rsyncd.conf 檔案中設定它。例如，如果要讓 /srv/ftp 目錄可供 rsync 使用，請使用下列組態：

```
gid = nobody
uid = nobody
read only = true
use chroot = no
transfer logging = true
log format = %h %o %f %l %b
log file = /var/log/rsyncd.log
```

```
[FTP]
```

```
path = /srv/ftp
comment = An Example
```

接著使用 rcrsyncdstart 來啟動 rsyncd。在開機程序期間也可以自動啟動 rsyncd。設定的方式有兩種，一是在 YaST 所提供的 runlevel 編輯器中啟用此服

務，另一是手動輸入 `insservrsyncd` 指令。`rsyncd` 也可以由 `xinetd` 啟動。然而，只有對很少使用 `rsyncd` 的伺服器才建議這麼做。

本範例也建立了傾聽所有連接的記錄檔。這個檔案是儲存在 `/var/log/rsyncd.log`。

這樣才可以從用戶端系統測試傳輸。使用下列指令來執行此動作：

```
rsync -avz sun::FTP
```

此指令會列出伺服器上 `/srv/ftp` 目錄中所有存在的檔案。這個要求也會記錄在 `/var/log/rsyncd.log` 記錄檔中。若要啟動實際的傳輸，請提供目標目錄。請使用 `.` 來代表目前的目錄。例如：

```
rsync -avz sun::FTP .
```

依照預設，在使用 `rsync` 同步化時不會刪除檔案。如果要強制刪除檔案，就必須加上額外的選項 `--delete`。若要確保不會刪除較新的檔案，則可以改用 `--update` 選項。任何產生的衝突都必須手動解決。

25.5 如需更多資訊

CVS

可在首頁 <http://www.cvshome.org> 找到有關 CVS 的重要資訊。

rsync

關於 `rsync` 的重要資訊，請參閱手冊頁面。指令為 `manrsync` 和 `manrsyncd.conf`。如需關於 `rsync` 作業原則的技術參考資料，可在 `/usr/share/doc/packages/rsync/tech_report.ps` 中找到。您可以在專案網站 <http://rsync.samba.org/> 上找到關於 `rsync` 的最新消息。

