

SUSE Linux

參考

www.novell.com

10.0

09/13/2005



參考

List of Authors: Jörg Arndt, Stefan Behlert, Frank Bodammer, James Branam, Volker Buzek, Klara Cihlarova, Stefan Dirsch, Olaf Donjak, Roman Drahtmüller, Thorsten Dubiel, Torsten Duwe, Thomas Fehr, Stefan Fent, Werner Fink, Kurt Garloff, Joachim Gleißner, Carsten Groß, Andreas Grünbacher, Berthold Gunreben, Franz Hassels, Andreas Jaeger, Jana Jaeger, Klaus Kämpf, Andi Kleen, Hubert Mantel, Lars Marowsky-Bree, Chris Mason, Johannes Meixner, Lars Müller, Matthias Nagorni, Anas Nashif, Siegfried Olschner, Edith Parzefall, Peter Pöml, Thomas Renninger, Hannes Reinecke, Thomas Rölz, Heiko Rommel, Marcus Schäfer, Thomas Schraitle, Klaus Singvogel, Hendrik Vogelsang, Klaus G. Wagner, Rebecca Walter, Christian Zoz

本出版品是 Novell Inc. 的智慧財產。

內容可以部份或全部重製，條件是每一個複本上都要可以清楚看到版權標籤。

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，SUSE LINUX GmbH、作者或譯者都不需對任何錯誤或造成的結果負責。

本手冊中引用的很多軟體和硬體說明，皆為註冊商標。所有商業名稱皆受版權限制，而且可能是註冊商標。SUSE LINUX GmbH 十分注重製造商的拼法。本手冊出現的產品名稱和商標(帶有或不帶有特定標記)皆受商標和交易保護法管轄，因此受到版權的限制。

您若有任何建議和意見，請提供給 documentation@suse.de。

內容目錄

關於本指南	xv
部 I 進階部署狀況	19
1 遠端安裝	21
1.1 遠端安裝的安裝方式	21
1.2 安裝保存安裝來源的伺服器	29
1.3 準備啟動目標系統	37
1.4 啟動要進行安裝的目標系統	46
1.5 監控安裝程序	50
2 進階磁碟安裝	55
2.1 SCSI 設備的永久性設備名稱	55
2.2 LVM 組態	56
2.3 軟體 RAID 組態	62
部 II 網際網路	69
3 網頁瀏覽器 Konqueror	71
3.1 標籤式瀏覽	72
3.2 儲存網頁與圖形	73
3.3 網際網路關鍵字	73
3.4 書籤	74
3.5 Java 與 JavaScript	74
3.6 若需詳細資訊	75

4	Firefox	77
4.1	瀏覽網站	77
4.2	尋找資訊	79
4.3	管理書籤	79
4.4	使用下載管理員	81
4.5	自訂 Firefox	81
4.6	從 Firefox 中列印	84
4.7	若需詳細資訊	84
5	Linphone — Linux 桌面的 VoIP	85
5.1	設定 Linphone	85
5.2	測試 Linphone	89
5.3	進行通話	90
5.4	接聽通話	91
5.5	使用通訊錄	91
5.6	疑難排解	92
5.7	詞彙	93
5.8	若需詳細資訊	94
6	使用 KGpg 加密	95
6.1	產生新的金鑰組合	95
6.2	匯出公共金鑰	97
6.3	匯入金鑰	97
6.4	金鑰伺服器對話方塊	98
6.5	文字與檔案加密	101
6.6	若需詳細資訊	102
部 III	多媒體	103
7	Linux 音效	105
7.1	混音器	105
7.2	多媒體播放器	110
7.3	CD：播放與轉檔 (Ripping)	114
7.4	使用 Audacity 進行硬碟錄音	118
7.5	直接錄音與播放 WAV 檔案	121
8	電視、視訊、廣播與網路攝影機	123
8.1	使用 motv 觀看電視	123
8.2	視訊文字支援	125
8.3	網路攝影機與 motv	126

8.4	nxtvepg — 個人電腦上的電視雜誌	126
8.5	以 xawtv4 觀賞數位視訊廣播	128
9	K3b -- 燒錄 CD 或 DVD	131
9.1	建立資料光碟	131
9.2	建立音樂光碟	134
9.3	複製 CD 或 DVD	135
9.4	燒錄 ISO 映像	136
9.5	建立多作業的 CD 或 DVD	136
9.6	若需詳細資訊	137
部 IV	辦公室	139
10	OpenOffice.org Office 套裝軟體	141
10.1	與其他辦公室應用程式的相容性	142
10.2	使用 Writer 進行文字處理	143
10.3	介紹 Calc	146
10.4	介紹 Impress	146
10.5	介紹 Base	146
10.6	若需詳細資訊	147
11	Evolution: 電子郵件與行事曆程式	149
11.1	從其他郵件程式輸入電子郵件	149
11.2	Evolution 綜覽	150
11.3	郵件	151
11.4	連絡人	155
11.5	行事曆	156
11.6	與掌上型裝置同步化資料	158
11.7	用於 GroupWise 使用者的 Evolution	158
11.8	若需詳細資訊	159
12	Kontact: 電子郵件與行事曆程式	161
12.1	從其他郵件程式輸入電子郵件	161
12.2	Kontact 概觀	162
12.3	郵件	163
12.4	連絡人	167
12.5	行事曆	169
12.6	與掌上型裝置同步化資料	171
12.7	GroupWise 使用者的 Kontact	171
12.8	若需詳細資訊	173

13 使用 KPilot 同步化掌上型電腦	175
13.1 KPilot 所使用的介質	176
13.2 設定掌上型電腦的連線	177
13.3 設定 KAddressBook 介質	178
13.4 管理待辦項目與事件	178
13.5 使用 KPilot	179
 14 使用 Beagle	 183
14.1 索引資料	184
14.2 搜尋資料	185
 部 V 圖形	 189
 15 數位相機與 Linux	 191
15.1 連接至相機	191
15.2 存取相機	192
15.3 使用 Konqueror	192
15.4 使用 Digikam	193
15.5 使用 f-spot	200
15.6 若需詳細資訊	207
 16 Kooka--掃描應用程式	 209
16.1 預覽	210
16.2 完成掃描	211
16.3 功能表	212
16.4 藝廊	213
16.5 光學字元辨識	214
 17 使用 GIMP 處理繪圖	 215
17.1 繪圖格式	215
17.2 啟動 GIMP	216
17.3 GIMP 入門	218
17.4 儲存影像	219
17.5 列印影像	220
17.6 若需詳細資訊	221

部 VI 機動性	223
18 Linux 的行動計算功能	225
18.1 筆記型電腦	225
18.2 行動硬體	230
18.3 行動電話和 PDA	231
18.4 若需詳細資訊	232
19 PCMCIA	233
19.1 硬體	233
19.2 軟體	234
20 系統組態設定檔管理	235
20.1 詞彙	235
20.2 使用 YaST 設定檔管理員	236
20.3 使用指令行設定 SCPM	239
20.4 使用 Profile Chooser Applet	242
20.5 疑難排解	242
20.6 在系統開機時選取設定檔	243
20.7 若需詳細資訊	243
21 電源管理	245
21.1 省電功能	245
21.2 APM	247
21.3 ACPI	248
21.4 硬碟的休眠	254
21.5 powersave 套件	255
21.6 YaST 電源管理模組	262
22 無線通訊	267
22.1 無線區域網路	267
22.2 藍芽	276
22.3 紅外線資料傳輸	285
部 VII 管理	289
23 Linux 的安全性	291
23.1 偽裝與防火牆	291

23.2	SSH: 安全網路作業	301
23.3	加密分割區和檔案	305
23.4	安全性與機密性	308
24	Linux 存取控制清單	319
24.1	ACL 的優點	319
24.2	定義	320
24.3	處理 ACL	320
24.4	應用程式的 ACL 支援	327
24.5	若需詳細資訊	328
25	系統監視公用程式	329
25.1	開啟檔案的清單: lsof	329
25.2	使用者存取檔案: fuser	331
25.3	檔案內容: stat	331
25.4	USB 設備: lsusb	332
25.5	SCSI 設備的相關資訊: scsiinfo	332
25.6	處理程序: top	333
25.7	程序清單: ps	334
25.8	程序樹: pstree	335
25.9	功能介紹: w	336
25.10	記憶體的使用: free	336
25.11	核心環狀緩衝區: dmesg	337
25.12	檔案系統及其使用: mount、df 和 du	337
25.13	/proc 檔案系統	338
25.14	vmstat、iostat 及 mpstat	340
25.15	procinfo	340
25.16	PCI 資源: lspci	342
25.17	執行程式的系統呼叫: strace	343
25.18	一個程式執行階段的程式庫呼叫: ltrace	344
25.19	指定需要的程式庫: ldd	344
25.20	其它有關 ELF 二進位的資訊	345
25.21	程序間的通訊: ipcs	345
25.22	使用 time 進行時間管理	346
部 VIII	系統	347
26	64 位元系統環境的 32 位元和 64 位元應用程式	349
26.1	執行期間支援	349
26.2	軟體開發	350
26.3	Biarch 平台的軟體編譯	350

26.4	核心規格	351
27	使用外圍程序	353
27.1	使用指令行上的 Bash	353
27.2	使用者和存取許可權	363
27.3	重要的 Linux 指令	368
27.4	vi 編輯器	379
28	啟動及設定 Linux 系統	385
28.1	Linux 開機程序	385
28.2	init 程序	388
28.3	透過 /etc/sysconfig 設定系統	396
29	開機載入器	399
29.1	開機管理	400
29.2	選取開機載入器	400
29.3	使用 GRUB 開機	400
29.4	使用 YaST 設定開機載入器	410
29.5	解除安裝 Linux 開機載入器	414
29.6	建立開機 CD	415
29.7	圖形化 GRUB 畫面	416
29.8	疑難排解	416
29.9	若需詳細資訊	418
30	SUSE Linux 的特殊功能	419
30.1	特殊軟體套件的資訊	419
30.2	虛擬主控台	426
30.3	鍵盤配置	426
30.4	語言與國家專用的設定	427
31	印表機操作	431
31.1	列印系統的工作流程	432
31.2	連線印表機的方法和通訊協定	433
31.3	安裝軟體	433
31.4	設定印表機	434
31.5	應用程式的組態	439
31.6	SUSE Linux 的特殊功能	440
31.7	疑難排解	445

32 熱插拔系統	451
32.1 設備和介面	452
32.2 熱插拔事件	453
32.3 熱插拔設備組態	453
32.4 自動化模組載入	454
32.5 開機程序檔冷插拔	455
32.6 錯誤分析	455
 33 使用 udev 的動態裝置節點	 457
33.1 建立規則	458
33.2 保留字元替換	458
33.3 關鍵字樣式比對	459
33.4 關鍵字選擇	459
33.5 大容量儲存裝置的一致名稱	460
 34 Linux 的檔案系統	 463
34.1 詞彙	463
34.2 Linux 的主要檔案系統	464
34.3 其他支援的檔案系統	469
34.4 Linux 的大型檔案支援	470
34.5 若需詳細資訊	472
 35 X Window System	 473
35.1 使用 SaX2 來設定 X11	473
35.2 X 組態最佳化	475
35.3 安裝與設定字型	480
35.4 OpenGL—3D 組態	485
 36 使用 PAM 驗證	 489
36.1 PAM 組態檔的結構	489
36.2 sshd 的 PAM 組態	491
36.3 PAM 模組的組態	493
36.4 若需詳細資訊	495
 37 使用 Xen 的圖形化系統	 497
37.1 Xen 安裝	498
37.2 領域安裝	499
37.3 設定 Xen 外來領域	502
37.4 啟動和控制 Xen 領域	502

37.5	若需詳細資訊	504
部 IX 服務		505
38 基本網路		507
38.1	IP 位址與路由	510
38.2	IPv6 --下一代的網際網路	513
38.3	名稱解析	520
38.4	使用 YaST 手動設定網路連線	521
38.5	手動設定網路連線	531
38.6	smpppd 做為撥號助理	540
39 網路中的 SLP 服務		543
39.1	註冊自己的服務	543
39.2	在 SUSE Linux 中的 SLP 前端	544
39.3	啟動 SLP	545
39.4	若需詳細資訊	545
40 領域名稱系統		547
40.1	DNS 概論	547
40.2	使用 YaST 進行組態	547
40.3	啟動名稱伺服器 BIND	555
40.4	組態檔 /etc/named.conf	556
40.5	區域檔案	560
40.6	區域資料的動態更新	564
40.7	安全交易	564
40.8	DNS 安全性	565
40.9	若需詳細資訊	566
41 使用 NIS		567
41.1	使用 YaST 設定 NIS 伺服器	567
41.2	設定 NIS 用戶端	572
42 使用 NFS 共用檔案系統		575
42.1	以 YaST 輸入檔案系統	575
42.2	手動輸入檔案系統	576
42.3	以 YaST 輸出檔案系統	576
42.4	手動輸出檔案系統	578

43	DHCP	581
43.1	使用 YaST 設定 DHCP 伺服器	582
43.2	DHCP 軟體套件	585
43.3	DHCP 伺服器 dhcpcd	585
43.4	若需詳細資訊	589
44	使用 xntp 進行時間同步化	591
44.1	使用 YaST 設定 NTP 用戶端	591
44.2	設定網路中的 xntp	594
44.3	設定本機參考時鐘	595
45	LDAP — 一種目錄服務	597
45.1	LDAP 與 NIS 的比較	599
45.2	LDAP 目錄樹的結構	599
45.3	使用 slapd.conf 來設定伺服器	602
45.4	LDAP 目錄中的資料處理	607
45.5	YaST LDAP 用戶端	610
45.6	在 YaST 中設定 LDAP 使用者和群組	617
45.7	若需詳細資訊	618
46	Apache 網頁伺服器	621
46.1	序言和詞彙	621
46.2	安裝	623
46.3	組態	629
46.4	虛擬主機	644
46.5	Apache 模組	649
46.6	安全性	660
46.7	疑難排解	661
46.8	若需詳細資訊	662
47	檔案同步化	665
47.1	可用的資料同步化軟體	665
47.2	選取程式時所要考慮的決定性因素	667
47.3	Unison 的介紹	671
47.4	CVS 簡介	673
47.5	Subversion 簡介	675
47.6	rsync 簡介	678
47.7	mailsync 簡介	680

48 Samba	683
48.1 設定伺服器	685
48.2 做為登入伺服器的 Samba	689
48.3 使用 YaST 設定 Samba 伺服器	690
48.4 設定用戶端	692
48.5 最佳化	693

關於本指南

本手冊提供 SUSE Linux 的一般常識。主要是針對系統管理員和擁有基本系統管理常識的家庭使用者使用撰寫。本手冊提供每天所需使用應用程式並提供進階安裝和組態過程的詳細說明。

進階部署狀況

學習如何在複雜的環境部署 SUSE Linux。

網際網路、多媒體、Office 和圖形

取得家庭使用者最需要的應用程式的概覽。

機動性

取得 SUSE Linux 行動運算的簡介，以及瞭解如何設定無線運算、有效管理和設定檔管理的不同選項。

管理

學習如何安全保護 SUSE Linux、處理檔案系統存取控制並了解 Linux 管理員必須知道的重要公用程式。

系統

取得 Linux 系統元件的說明並了解如何相互運作。

服務

學習如何設定不同的網路和 SUSE Linux 的檔案服務。

1 回饋

我們想了解您對有關本手冊和其他本產品的文件的想法和建議。請使用線上文件每頁下方的使用備註功能，或至 <http://www.novell.com/documentation/feedback.html> 並在此輸入您的想法。

2 其他文件

本 SUSE Linux 產品也提供其他手冊，您可以在線上 <http://www.novell.com/documentation/> 或在已安裝系統的 `/usr/share/doc/manual/` 路徑取得。

啟動

本指南包含初次使用 SUSE Linux 的步驟。您可以在 <http://www.novell.com/documentation/suse10/> 取得本文件的線上版本。

《Novell AppArmor Powered by Immunix 1.2 Installation and QuickStart Guide》

本手冊大略說明 AppArmor 產品的初次安裝步驟。您可以在 <http://www.novell.com/documentation/apparmor/> 取得本文件的線上版本。

《Novell AppArmor Powered by Immunix 1.2 Administration Guide》

本指南包含在您的環境使用 AppArmor 的詳細資訊。您可以在 <http://www.novell.com/documentation/apparmor/> 取得本文件的線上版本。

3 文件慣例

本手冊使用下列印刷慣例：

- `/etc/passwd`：檔名和目錄名稱
- 保留字元：以實際的值來取代保留字元
- PATH：環境變數 PATH
- `ls`、`--help`：指令、選項與參數
- 使用者：使用者或群組
- `[Alt]`、`[Alt] + [F1]`：要按的鍵或鍵組合
- 檔案、檔案 → 另存新檔：功能表項目、按鈕
- 跳舞的企鵝 (參閱企鵝章節)：這是前往其他書中章節的參照。

4 聲明

在許多人自發的努力之下，全球的 Linux 開發人員促進了 Linux 的發展。我們十分感謝他們的努力，沒有他們，這個版本不會存在。不僅如此，我們還要謝謝 Frank Zappa 與 Pawar。當然還要特別感謝 Linus Torvalds。

祝您使用愉快！

您的 SUSE 團隊

部 I. 進階部署狀況

遠端安裝

SUSE Linux 有數種不同的安裝方法。您可以依照 章 1, *使用 YaST 安裝* (↑啟動) 中的說明，從 CD 或 DVD 進行一般安裝，或者您也可以選擇使用不同的網路架構方式來安裝 SUSE Linux，或者採用完全自動安裝的方式來進行安裝。

以下兩個簡要的檢查清單將介紹每個安裝方式：第一個清單會列出每個方法的先決條件；另一個清單則是以圖解方式介紹基本安裝程序。接著會介紹用於每個安裝方式的所有技術的詳細資訊。

注

在以下章節中，會將要安裝新 SUSE Linux 的系統稱做 *目標系統* 或 *安裝目標*。而 *安裝來源* 這個名詞則是用來表示所有安裝資料的來源，其中包括實體媒體 (例如 CD 和 DVD) 以及您網路中發佈安裝資料的網路伺服器。

1.1 遠端安裝的安裝方式

此章節將介紹最常用於遠端安裝的安裝方式。不論使用那個方式，都必須仔細檢查先決條件清單，並依照該安裝方式中說明的程序進行安裝。如果需要某個特定步驟的詳細說明，請連接至每個步驟中提供的連結以取得說明。

重要

X Window 系統的組態不屬於任何遠端安裝程序中的一個部分。完成安裝後，請以 `root` 身分登入目標系統、輸入 `init 3`，接著啟動 `SaX2` 並依照 [節 35.1, "使用 SaX2 來設定 X11" \[473\]](#) 中描述的說明設定圖形硬體。

1.1.1 透過 VNC 執行簡易遠端安裝 — 靜態網路組態

此安裝類型仍需要某個程度的實體存取權限，以便存取並啟動目標系統來進行安裝。安裝本身完全受控於遠端工作站，遠端工作站會使用 VNC 連線至安裝程式來進行安裝。且在 [章 1, 使用 YaST 安裝 \(↑啟動\)](#) 中執行手動安裝時，使用者必須與安裝程式進行互動。

使用此安裝類型時，請確定已符合以下需求：

- 遠端安裝來源：有網路連線的 NFS、HTTP、FTP 或 SMB
- 有網路連線的目標系統
- 使用網路連線和 VNC 檢視器軟體或已啟用 Java 的瀏覽器 (Firefox、Konqueror、Internet Explorer 或 Opera) 來控制系統
- 用來啟動目標系統的實體開機媒體 (CD 或 DVD)
- 已指定用於安裝來源和控制系統的有效靜態 IP 位址
- 已將有效的靜態 IP 位置指定到目標系統

若要執行此安裝類型，請按照下列步驟進行：

- 1 依照 [節 1.2, "安裝保存安裝來源的伺服器" \[29\]](#) 中的說明安裝安裝來源。
- 2 使用 SYSE Linux 媒體套件的第一張 CD 或 DVD 啟動目標系統。
- 3 出現目標系統的開機畫面時，請使用開機選項提示來設定適當的 VNC 選項和安裝來源的位址。如需詳細說明，請參閱 [節 1.4, "啟動要進行安裝的目標系統" \[46\]](#)。

目標系統會啟動到以文字模式為基礎的環境，並提供網路位址和顯示編號，而 VNC 檢視器應用程式或瀏覽器必須使用這些資訊才可在圖形安裝環境下進行安裝。VNC 安裝會透過 OpenSLP 自我宣告，您可以使用 Konqueror 在 `service: //` 或 `slp: //` 模式中找到 VNC 安裝。

- 4 接著，請在控制工作站上開啟 VNC 檢視應用程式或網頁瀏覽器，並依照節 1.5.1, "安裝 VNC" [50] 中的說明連線到目標系統。
- 5 依照 章 1, 使用 YaST 安裝 (↑啟動) 中的說明執行安裝。

目標系統重新開機後，您必須重新連線至目標系統，以完成安裝的最後一個部分。

- 6 完成安裝。

1.1.2 透過 VNC 執行簡易遠端安裝 — 透過 DHCP 的動態網路組態

此安裝類型仍需要某個程度的實體存取權限，以便存取並啟動目標系統來進行安裝。請在 DHCP 定義網路組態。安裝本身完全受控於遠端工作站，遠端工作站會使用 VNC 連線至安裝程式，但使用者仍必須與安裝程式進行互動，以實際設定組態。

使用此安裝類型時，請確定已符合以下需求：

- 遠端安裝來源：有網路連線的 NFS、HTTP、FTP 或 SMB
- 有網路連線的目標系統
- 使用網路連線和 VNC 檢視器軟體或已啟用 Java 的瀏覽器 (Firefox、Konqueror、Internet Explorer 或 Opera) 來控制系統
- 用來啟動目標系統的實體開機媒體 (CD、DVD、自訂的開機磁片)
- 執行提供 IP 位址的 DHCP 伺服器

若要執行此安裝類型，請按照下列步驟進行：

- 1 依照 節 1.2, "安裝保存安裝來源的伺服器" [29] 中的說明安裝安裝來源。選擇 NFS、HTTP 或 FTP 網路伺服器。若要使用 SMB 安裝來源，請參閱 節 1.2.5, "管理 SMB 安裝來源" [36]。
- 2 使用 SYSE Linux 媒體套件的第一張 CD 或 DVD 啟動目標系統。
- 3 出現目標系統的開機畫面時，請使用開機選項提示來設定適當的 VNC 選項和安裝來源的位址。如需詳細說明，請參閱 節 1.4, "啟動要進行安裝的目標系統" [46]。

目標系統會啟動到以文字模式為基礎的環境，並提供網路位址和顯示編號，而 VNC 檢視器應用程式或瀏覽器必須使用這些資訊才可在圖形安裝環境下進行安裝。VNC 安裝會透過 OpenSLP 自我宣告，您可以使用 Konqueror 在 service: // 或 slp: // 模式中找到 VNC 安裝。
- 4 接著，請在控制工作站上開啟 VNC 檢視應用程式或網頁瀏覽器，並依照 節 1.5.1, "安裝 VNC" [50] 中的說明連線到目標系統。
- 5 依照 章 1, 使用 YaST 安裝 (↑啟動) 中的說明執行安裝。

目標系統重新開機後，您必須重新連線至目標系統，以完成安裝的最後一個部分。
- 6 完成安裝。

1.1.3 透過 VNC 執行遠端安裝 — PXE 開機和網路喚醒功能

此安裝類型為完全自動安裝。目標機器將經由遠端啟動。使用者只需在實際安裝時才需要與安裝程式進行互動。此方法適用於跨網站部署。

若要執行此安裝類型，請確定已符合以下需求：

- 遠端安裝來源：有網路連線的 NFS、HTTP、FTP 或 SMB
- TFTP 伺服器
- 執行用於您網路的 DHCP 伺服器

- 目標系統可以使用 PXE 開機、網路和網路喚醒功能，並可連線至網路
- 使用網路連線和 VNC 檢視器軟體或已啟用 Java 的瀏覽器 (Firefox、Konqueror、Internet Explorer 或 Opera) 來控制系統

若要執行此安裝類型，請按照下列步驟進行：

- 1 依照 [節 1.2, "安裝保存安裝來源的伺服器" \[29\]](#) 中的說明安裝安裝來源。選擇一個 NFS、HTTP、FTP 網路伺服器，或依照 [節 1.2.5, "管理 SMB 安裝來源" \[36\]](#) 的說明設定 SMB 安裝來源。
- 2 安裝 TFTP 伺服器以保存目標系統可用來開機的開機影像。這會在 [節 1.3.2, "設定 TFTP 伺服器" \[39\]](#) 中說明。
- 3 安裝 DHCP 伺服器以提供所有機器 IP 位址，並在目標系統顯示 TFTP 伺服器的位置。這會在 [節 1.3.1, "設定 DHCP 伺服器" \[37\]](#) 中說明。
- 4 準備用於 PXE 開機的目標系統。如需進一步詳細說明，請參閱 [節 1.3.5, "準備用於 PXE 開機的目標系統" \[45\]](#)。
- 5 使用網路喚醒功能啟動目標系統的開機程序。這會在 [節 1.3.7, "網路喚醒功能" \[45\]](#) 中說明。
- 6 接著，請在控制工作站上開啟 VNC 檢視應用程式或網頁瀏覽器，並依照 [節 1.5.1, "安裝 VNC" \[50\]](#) 中的說明連線到目標系統。
- 7 依照 [章 1, 使用 YaST 安裝 \(↑啟動\)](#) 中的說明執行安裝。

目標系統重新開機後，您必須重新連線至目標系統，以完成安裝的最後一個部分。

- 8 完成安裝。

1.1.4 透過 SSH 執行簡易遠端安裝 — 靜態網路組態

此安裝類型仍需要某個程度的實體存取權限，以便存取並啟動目標系統並決定安裝目標的 IP 位址。安裝本身完全受控於遠端工作站，遠端工作站會使用 SSH

連線至安裝程式。且如章 1, 使用 *YaST* 安裝(↑啟動)中所述, 執行一般安裝時, 使用者必須與安裝程式進行互動。

使用此安裝類型時, 請確定已符合以下需求:

- 遠端安裝來源: 有網路連線的 NFS、HTTP、FTP 或 SMB
- 有網路連線的目標系統
- 使用網路連線和 VNC 檢視器軟體或已啟用 Java 的瀏覽器 (Firefox、Konqueror、Internet Explorer 或 Opera) 來控制系統
- 用於目標系統的實體開機媒體 (CD、DVD、自訂的開機磁片)
- 已指定用於安裝來源和控制系統的有效靜態 IP 位址
- 已將有效的靜態 IP 位置指定到目標系統

若要執行此安裝類型, 請按照下列步驟進行:

- 1 依照 [節 1.2, "安裝保存安裝來源的伺服器" \[29\]](#) 中的說明安裝安裝來源。
- 2 使用 SYSE Linux 媒體套件的第一張 CD 或 DVD 啟動目標系統。
- 3 出現目標系統的開機畫面時, 請使用開機選項提示來設定適當的網路連線參數和安裝來源位址, 並啟用 SSH。如需詳細說明, 請參閱 [節 1.4.3, "使用自訂開機選項" \[48\]](#)。

目標系統會啟動到以文字模式為基礎的環境, 並提供網路位址和顯示編號, 而所有 SSH 用戶端必須使用這些資訊才可在圖形安裝環境下進行安裝。

- 4 接著, 請在控制工作站上開啟終端機視窗, 並依照 ["連線到安裝程式" \[52\]](#) 中的說明連線到目標系統。
- 5 依照章 1, 使用 *YaST* 安裝(↑啟動)中的說明執行安裝。

目標系統重新開機後, 您必須重新連線至目標系統, 以完成安裝的最後一個部分。

- 6 完成安裝。

1.1.5 透過 SSH 執行簡易遠端安裝 — 透過 DHCP 的動態網路組態

此安裝類型仍需要某個程度的實體存取權限，以便存取並啟動目標系統並決定安裝目標的 IP 位址。安裝本身完全受控於遠端工作站，遠端工作站會使用 VNC 連線至安裝程式，但使用者仍必須與安裝程式進行互動，以實際設定組態。

使用此安裝類型時，請確定已符合以下需求：

- 遠端安裝來源：有網路連線的 NFS、HTTP、FTP 或 SMB
- 有網路連線的目標系統
- 使用網路連線和 VNC 檢視器軟體或已啟用 Java 的瀏覽器 (Firefox、Konqueror、Internet Explorer 或 Opera) 來控制系統
- 用來啟動目標系統的實體開機媒體 (CD 或 DVD)
- 執行提供 IP 位址的 DHCP 伺服器

若要執行此安裝類型，請按照下列步驟進行：

- 1 依照 [節 1.2, "安裝保存安裝來源的伺服器" \[29\]](#) 中的說明安裝安裝來源。選擇 NFS、HTTP 或 FTP 網路伺服器。若要使用 SMB 安裝來源，請參閱 [節 1.2.5, "管理 SMB 安裝來源" \[36\]](#)。
- 2 使用 SYSE Linux 媒體套件的第一張 CD 或 DVD 啟動目標系統。
- 3 出現目標系統的開機畫面時，請使用開機選項提示來輸入適當的網路連線參數和安裝來源位置，並啟用 SSH。請參閱 [節 1.4.3, "使用自訂開機選項" \[48\]](#)，以取得使用這些參數的詳細說明。

目標系統會啟動到以文字模式為基礎的環境，並提供網路位址，而所有 SSH 用戶端必須使用這些資訊才可在圖形安裝環境下進行安裝。
- 4 接著，請在控制工作站上開啟終端機視窗，並依照 ["連線到安裝程式" \[52\]](#) 中的說明連線到目標系統。
- 5 依照 [章 1, 使用 YaST 安裝 \(↑啟動\)](#) 中的說明執行安裝。

目標系統重新開機後，您必須重新連線至目標系統，以完成安裝的最後一個部分。

6 完成安裝。

1.1.6 透過 SSH 執行遠端安裝 — PXE 開機和網路喚醒功能

此安裝類型為完全自動安裝。目標機器將經由遠端啟動。

若要執行此安裝類型，請確定已符合以下需求：

- 遠端安裝來源：有網路連線的 NFS、HTTP、FTP 或 SMB
- TFTP 伺服器
- 執行用於您網路的 DHCP 伺服器，以提供靜態 IP 給需要安裝的主機。
- 目標系統可以使用 PXE 開機、網路和網路喚醒功能，並可連線至網路
- 有網路連線功能和 SSH 用戶端軟體的控制軟體

若要執行此安裝類型，請按照下列步驟進行：

- 1 依照 [節 1.2, "安裝保存安裝來源的伺服器" \[29\]](#) 中的說明安裝安裝來源。選擇 NFS、HTTP 或 FTP 網路伺服器。如需 SMB 安裝來源的組態資訊，請參閱 [節 1.2.5, "管理 SMB 安裝來源" \[36\]](#)。
- 2 安裝 TFTP 伺服器以保存目標系統可用來開機的開機影像。這會在 [節 1.3.2, "設定 TFTP 伺服器" \[39\]](#) 中說明。
- 3 安裝 DHCP 伺服器以提供所有機器 IP 位址，並在目標系統顯示 TFTP 伺服器的位置。這會在 [節 1.3.1, "設定 DHCP 伺服器" \[37\]](#) 中說明。
- 4 準備用於 PXE 開機的目標系統。如需進一步詳細說明，請參閱 [節 1.3.5, "準備用於 PXE 開機的目標系統" \[45\]](#)。
- 5 使用網路喚醒功能啟動目標系統的開機程序。這會在 [節 1.3.7, "網路喚醒功能" \[45\]](#) 中說明。

- 6 接著，請在控制工作站上啟動 VNC 用戶端，並依照 [節 1.5.2, "安裝 SSH" \[52\]](#) 中的說明連線到目標系統。
- 7 依照 章 1, *使用 YaST 安裝* (↑啟動) 中的說明執行安裝。

目標系統重新開機後，您必須重新連線至目標系統，以完成安裝的最後一個部分。
- 8 完成安裝。

1.2 安裝保存安裝來源的伺服器

根據做為 SUSE Linux 網路安裝來源機器上所執行的作業系統的不同，有數個伺服器組態選項可供使用。設定安裝伺服器最簡單的方法就是使用 SUSE LINUX Enterprise Server 9 或 SUSE Linux 9.3 (或更新版本) 上的 YaST。若使用其他版本的 SUSE LINUX Enterprise Server 或 SUSE Linux，則請手動設定安裝來源。

提示

您甚至可以將安裝 Microsoft Windows 的機器做為您 Linux 部署的安裝伺服器。如需詳細資料，請參閱 [節 1.2.5, "管理 SMB 安裝來源" \[36\]](#)。

1.2.1 使用 YaST 設定安裝伺服器

YaST 會提供圖形工具，方便您建立網路安裝來源。它支援 HTTP、FTP 和 NFS 網路安裝伺服器。

- 1 請以 root 身分登入要做為安裝伺服器的機器。
- 2 啟動 YaST → 其他 → 安裝伺服器。
- 3 選取伺服器類型 (HTTP、FTP 或 NFS)。

之後，每次啟動系統時都會自動啟動選取的伺服器服務。如果您的系統上已在執行選取的服務類型，且您要手動設定該伺服器，請選取 *不要設定任何網路服務* 選項以停用伺服器服務的自動組態功能。在這兩種情況下，都必須定義安裝資料可用於伺服器上的目錄。

4 設定所需的伺服器類型。

這個步驟與伺服器服務的自動組態功能相關。若您停用自動組態功能，則請略過此步驟。為可找到安裝資料所在的 FTP 或 HTTP 伺服器根目錄定義別名。之後，您即可經由 `ftp://Server-IP/Alias/Name` (FTP) 或 `http://Server-IP/Alias/Name` (HTTP) 找到安裝來源。*Name* 代表安裝來源的名稱，可按照以下步驟定義。如果您在上一個步驟中選取 NFS，那麼請定義萬用字元和匯出選項。您可以經由 `nfs://Server-IP/Name` 存取 NFS 伺服器。您可以在 [章 42, 使用 NFS 共用檔案系統 \[575\]](#) 中找到 NFS 和匯出選項的詳細說明。

5 設定安裝來源。

將安裝媒體複製到目的地前，請先定義安裝來源的名稱 (最好是以產品縮寫和版本做為名稱，以方便記憶)。YaST 允許提供媒體的 ISO 影像，而不需使用安裝 CD 的副本。如果您要製作 ISO 影像，請啟動相關的核取方塊並指定本地存放 ISP 檔案的目錄路徑。根據使用此安裝伺服器要發佈產品的不同，您可能還需要使用包含附加程式或 Service Pack 的 CD，以完整安裝產品。如果您啟動 *提示放入其他 CD* 選項，YaST 會自動提醒您提供這些媒體。若要透過 OpenSLP 宣告網路中的安裝伺服器，請啟動適當的選項。

提示

但是，請考慮您的網路設定是否支援該選項，再決定是否要透過 OpenSLP 選告您的安裝來源。如此才可以確保每台目標機器皆可進入該網路安裝路徑。使用 SLP 開機選項啟動的目標系統將會尋找網路安裝來源，且您不需進一步設定任何組態。如需此選項的詳細資訊，請參閱 [節 1.4, "啟動要進行安裝的目標系統" \[46\]](#)。

6 上傳安裝資料。

設定安裝伺服器最長的步驟就是複製實際安裝 CD。請依照 YaST 要求的順序插入媒體，接著等待複製程式結束。來源複製完成後，請選取 *完成已返回現有資訊來源* 的概觀頁面並關閉組態。

現在，您已完成安裝伺服器的設定，並可開始提供服務。之後每當啟動系統時，將會自動啟動該伺服器。您將不需再進行任何操作。如果您一開始就使用 YaST 停用選定網路服務的自動組態功能，那麼您只需正確手動設定和啟動此服務即可。

若要停用安裝來源，請在概觀頁面中選取 **變更**，以開啟可用安裝來源的清單。請選擇要移除的項目，接著選取 **刪除**。此刪除動作只會影響到要停用的伺服器服務。安裝資料本身還是會保留在選定的目錄中。但是，您還是可以手動移除資料。

如果您的安裝伺服器應要提供一個以上的產品版本，那麼請啟動 YaST 安裝伺服器模組，並在現有安裝來源的概觀頁面中選取 **設定** 來設定新的安裝來源。

1.2.2 手動設定 NFS 安裝來源

基本上，設定 NFS 安裝來源只需兩個步驟。第一個步驟是，建立保存安裝資料的目錄架構，並將安裝媒體複製到此架構中。第二個步驟是，將保存安裝資料的目錄匯出至網路。

若要建立保存安裝資料的目錄，請執行下列步驟：

- 1 登入為 **root** 使用者。
- 2 建立一個之後要保存所有安裝資料的目錄，並將資料放入該目錄。例如：

```
mkdir install/product/productversion
cd install/product/productversion
```

將會以產品名稱的縮寫取代 *product*，在此案例中，取代字串為 **SUSE Linux**；*productversion* 則會被包含產品名稱和版本的字串所取代。

- 3 請在插入媒體套件中的每張 CD 時執行以下指令：

- a 將安裝 CD 中的所有內容複製到安裝伺服器目錄：

```
cp -a /media/path_to_your_CD-ROM_drive .
```

您的 CD 或 DVD 光碟機所在位置的實際路徑將會取代 *path_to_your_CD-ROM_drive*。根據您系統所使用光碟機類型的不同，可能會是 *cdrom*、*cdrecorder*、*dvd* 或 *dvdrecorder*。

- b 重新命名 CD 編號的目錄：

```
mv path_to_your_CD-ROM_drive CDx
```

CD 的實際編號將會取代 *x*。

若要使用 YaST 透過 NFS 匯出安裝來源，請執行下列步驟：

- 1 登入為 root 使用者。
- 2 啟動 YaST → 網路服務 → NFS 伺服器。
- 3 選取 啟動 NFS 伺服器 和 開啟防火牆中的連接埠，接著按一下 下一步。
- 4 選取 新增目錄，並輸入要保存安裝資料的目錄路徑。在此案例中，路徑為 `/productversion`。
- 5 選取 新增主機，並輸入安裝資料要匯出的目標機器主機名稱。此處除了可以指定主機名稱外，您還可以使用萬用字元、網路位址範圍或只要指定您網路的網域名稱即可。您可以選擇輸入適當的匯出選項，或者保留預設值 (在大多數設定中皆可正常運作)。如需匯出 NFS 共用所使用語法的更多相關資訊，請閱讀 匯出 線上文件。
- 6 按一下 完成。

之後，會自動啟動保存 SUSE Linux 安裝來源的 NFS 伺服器，並將啟動該伺服器的步驟整合到開機程序中。

如果您想要透過 NFS 手動匯出安裝來源，而不使用 YaST NFS 伺服器模組匯出安裝來源，請執行下列步驟：

- 1 登入為 root 使用者。
- 2 開啟檔案 `/etc/exports` 並輸入以下內容：

```
/productversion *(ro,root_squash,sync)
```

此操作可將目錄 `/productversion` 匯出至網路中的任何一台主機或任何可連線在此伺服器的主機。若要限制存取此伺服器，請使用網路遮罩或網域名稱，而不要使用一般萬用字元 `*`。如需詳細資訊，請參閱 匯出 線上文件。儲存並結束此組態檔案。

- 3 若要新增 NFS 服務至系統開機時要啟動的伺服器的清單中，請執行以下指令：

```
insserv /etc/init.d/nfsserver
```

```
insserv /etc/init.d/portmap
```

4 使用以下指令啟動 NFS 伺服器：

```
rcnfsserver start
```

如果您之後必須變更 NFS 伺服器的組態，請修改組態檔案並使用 `rcnfsserver restart` 重新啟動 NFS 精靈。

透過 **OpenSLP** 宣告 NFS 伺服器可以讓您網路中的所有用戶端都知道該伺服器的位址。

- 1 登入為 `root` 使用者。
- 2 輸入目錄 `/etc/slp.reg.d/`。
- 3 建立名為 `install.suse.nfs.reg` 的組態檔案，檔案中必須包含以下內容：

```
# Register the NFS Installation Server
service:install.suse:nfs: //$HOSTNAME/path_instsource/CD1,en,65535
description=NFS Installation Source
```

您伺服器上安裝來源的實際路徑將會取代 `path_instsource`。

- 4 儲存此組態檔案，並使用以下指令啟動 **OpenSLP** 精靈：

```
rcslpd start
```

如需 **OpenSLP** 的更多資訊，請參閱 `/usr/share/doc/packages/openslp/` 中的套件文件或 [章 39，網路中的 SLP 服務 \[543\]](#)。

1.2.3 手動設定 FTP 安裝來源

建立 FTP 安裝來源與建立 NFS 安裝來源非常相似。您也可以使用 **OpenSLP** 透過網路宣告 FTP 安裝來源。

1 依照 節 1.2.2, "手動設定 NFS 安裝來源" [31] 中的說明，建立一個保存安裝來源的目錄。

2 設定要發佈安裝目錄內容的 FTP 伺服器：

a 以 root 身分登入，接著使用 YaST 套件管理員安裝套件 `pure-ftpd` (小型的 FTP 伺服器)。

b 輸入 FTP 伺服器的根目錄：

```
cd/srv/ftp
```

c 建立保存在 FTP 根目錄中安裝來源的子目錄：

```
mkdir instsource
```

產品名稱將會取代 `instsource`。

d 將安裝 CD 上的所有內容複製到 FTP 伺服器的根目錄 (此步驟與 節 1.2.2, "手動設定 NFS 安裝來源" [31] 和 步驟 3 [31] 中描述的程序相似)。

另外，請將現有安裝儲存區中的內容裝載至 FTP 伺服器的變更根目錄環境中：

```
mount --bind  
path_to_instsource /srv/ftp/instsource
```

符合您設定的值將會取代 `path_to_instsource` 和 `instsource`。
如果您必須讓此組態永久生效，請將此組態新增至 `/etc/fstab`。

e 啟動 `pure-ftpd`：

```
pure-ftpd &
```

3 如果您的網路設定支援 OpenSLP，您即可透過 OpenSLP 宣告安裝來源：

a 在 `/etc/slp/reg.d/` 建立名為 `install.suse.ftp.reg` 的組態檔案，檔案中必須包含以下內容：

```
# Register the FTP Installation Server  
service:install.suse:ftp: // $HOSTNAME/srv/ftp/instsource/CD1,en,65535  
description=FTP Installation Source
```

您伺服器上安裝來源目錄的實際名稱將會取代 *instsource*。
service: 行輸入的內容則應為一連續的行。

- b** 儲存此組態檔案，並使用以下指令啟動 OpenSLP 精靈：

```
rcslpd start
```

1.2.4 手動設定 HTTP 安裝來源

建立 HTTP 安裝來源與建立 NFS 安裝來源非常相似。您也可以使用 OpenSLP 透過網路宣告 HTTP 安裝來源。

- 1 依照 [節 1.2.2, "手動設定 NFS 安裝來源" \[31\]](#) 中的說明，建立一個保存安裝來源的目錄。
- 2 設定要發佈安裝目錄內容的 HTTP 伺服器：

- a** 以 root 身分登入，接著使用 YaST 套件管理員安裝套件 `apache2`。

- b** 輸入 HTTP 伺服器的根目錄 (`/srv/www/htdocs`)，並建立要保存安裝來源的子目錄。

```
mkdir instsource
```

產品名稱將會取代 *instsource*。

- c** 建立一個從安裝來源位置連線到 Web 伺服器根目錄的符號連結 (`/srv/www/htdocs`):

```
ln -s /path_instsource /srv/www/htdocs/instsource
```

- d** 接著，請修改 HTTP 伺服器的組態檔案 (`/etc/apache2/default-server.conf`)，以使符號連結生效。行

```
Options None
```

將被

```
Options Indexes FollowSymLinks
```

- e** 使用 `rcapache2 restart` 重新啟動 HTTP 伺服器。

3 如果您的網路設定支援 OpenSLP，您即可透過 OpenSLP 宣告安裝來源：

- a** 在 `/etc/slp/reg.d/` 建立名為 `install.suse.http.reg` 的組態檔案，檔案中必須包含以下內容：

```
# Register the HTTP Installation Server
service:install.suse:http://$HOSTNAME/srv/www/htdocs/instsource/CD1/,en,65535
description=HTTP Installation Source
```

您伺服器上安裝來源的實際路徑將會取代 `path_to_instsource`。
`service:` 行輸入的內容則應為一連續的行。

- b** 儲存此組態檔案，並使用 `rcslpd restart` 啟動 OpenSLP 精靈。

1.2.5 管理 SMB 安裝來源

使用 SMB (Samba) 時，您可以從 Microsoft Windows 伺服器輸入安裝來源，甚至不需使用 Linux 機器就可啟動 Linux 部署。

若要設定保存 SUSE Linux 安裝來源的匯出 Windows 共用，請執行下列步驟：

- 1** 登入您的 Windows 機器。
- 2** 啟動「檔案總管」並建立一個保存完整安裝樹狀結構的資料夾，且可將資料夾命名為 `INSTALL` 或其他名稱。
- 3** 根據您 Windows 文件中概述的程序匯出此共用。
- 4** 進入此共用並建立名為 `product` 的子資料夾。請以實際的產品名稱取代 `product` (在此案例中，產品名稱為 SUSE Linux)。
- 5** 將每張 SUSE Linux CD 內容複製到個別的資料夾，並將資料夾命名為 `CD1`、`CD2`、`CD3` 等。
- 6** 進入匯出共用最上層的目錄 (此範例中為 `INSTALL`)，並將以下檔案和資料夾從 `product/CD1` 複製到此資料夾：`content`、`media.1`、`control.xml` 和 `boot`。
- 7** 在 `INSTALL` 下建立一個新的資料夾，並將它命名為 `yast`。

8 進入 `yast` 資料夾，並建立名為 `order` 和 `instorder` 的檔案。

9 開啟 `order` 檔案並輸入以下內容：

```
/NLD /CD1 smb: //user:password@hostname /productCD1
```

您在 Windows 機器上使用的使用者名稱將會取代 `user`，或者您也可以輸入 `Guest` 允許以 `Guest` 登入的使用者可使用此共用；您的登入密碼或任何其他供 `Guest` 登入的字串將會取代 `password`；您 Windows 機器的網路名稱則會取代 `hostname`。

10 開啟 `instorder` 檔案並新增以下內容：

```
/product /CD1
```

若要將裝載的 SMB 共用做為安裝來源，請執行下列步驟：

- 1 啟動安裝目標。
- 2 選取 安裝。
- 3 按 **[F4]** 以選取安裝來源。
- 4 選擇 SMB 並輸入 Windows 的機器名稱或 IP 位址、共用名稱 (在此範例中為 `INSTALL`)、使用者名稱和密碼。

當您按下 **[Enter]** 後，會啟動 YaST 且您可開始執行安裝。

1.3 準備啟動目標系統

此章節包含各種複雜開機方式的組態工作。其中包含可立即用於 DHCP、PXE 開機、TFTP 和網路喚醒功能的組態範例。

1.3.1 設定 DHCP 伺服器

您可以手動編輯適當的組態檔案，以完成 SUSE Linux 上的 DHCP 伺服器設定。此章節包含延伸 DHCP 伺服器組態，藉此提供 TFTP、PXE 和 WOL 環境中所需的資料。

手動設定 DHCP 伺服器

除了提供您網路用戶端自動位址配置，DHCP 伺服器需要執行的所有工作就是宣告 TFTP 伺服器的 IP 位址，以及宣告目標機器上安裝常式應使用的檔案。

- 1 以 root 身分登入代管 DHCP 伺服器的機器。
- 2 將以下內容附加到位於 `/etc/dhcpd.conf` 下的 DHCP 伺服器的組態檔案：

```
group {
    # PXE related stuff
    #
    # "next server" defines the tftp server that will be used
    next server ip_tftp_server;
    #
    # "filename" specifies the pxelinux image on the tftp server
    # the server runs in chroot under /srv/tftpboot
    filename "pxelinux.0";
}
```

TFTP 伺服器的實際 IP 位置將取代 *ip_of_the_tftp_server*。

如需更多 `dhcpd.conf` 中可用的選項資訊，請參閱 `dhcpd.conf` 線上文件。

- 3 執行 `rcdhcpd restart` 重新啟動 DHCP 伺服器

如果您計畫使用 SSH 來遠端控制 PXE 和網路喚醒功能安裝，請明確指定 DHCP 應提供給安裝目標的 IP 位址。若要完成這項工作，請根據以下範例修改上述 DHCP 組態：

```
group {
    # PXE related stuff
    #
    # "next server" defines the tftp server that will be used
    next server ip_tftp_server;
    #
    # "filename" specifies the pxelinux image on the tftp server
    # the server runs in chroot under /srv/tftpboot
    filename "pxelinux.0";
    host test { hardware ethernet mac_address;
                  fixed-address some_ip_address; }
}
```

主機陳述式將介紹安裝目標的主機名稱。若要結合特定主機的主機名稱和 IP 位址，您必須了解並指定系統的硬體 (MAC) 位址。用於此範例中符合您環境的實際值將取代所有變數。

重新啟動 DHCP 伺服器後，DHCP 伺服器會提供指定的主機一個靜態 IP，讓您可以透過 SSH 連線到系統。

1.3.2 設定 TFTP 伺服器

您可以使用 YaST 設定 TFTP 伺服器，或是在任何支援 xinetd 和 tftp 的 Linux 作業系統上手動進行設定。每當目標系統開機時，TFTP 伺服器即會傳送開機影像至目標系統，並傳送對目標系統的要求。

使用 YaST 設定 TFTP 伺服器

- 1 登入為 root 使用者。
- 2 啟動 YaST → 網路服務 → TFTP 伺服器 並安裝所需套件。
- 3 按一下 啟用以確定啟動伺服器，並將伺服器納入開機常式。為安全起見，xinetd 在開機時啟動 tftpd 時，您不需執行任何動作。
- 4 按一下 開啟防火牆中的連接埠，開啟在您機器上運作中防火牆的適當連接埠。如果您的伺服器上沒有執行任何防火牆，將無法使用此選項。
- 5 按一下 瀏覽，瀏覽開機影像目錄。

建立的預設目錄為 /tftpboot，且會自動選取該目錄。

- 6 按一下 完成，套用設定並啟動伺服器。

手動設定 TFTP 伺服器

- 1 以 root 身分登入，並安裝 tftp 和 xinetd 套件。
- 2 若無法安裝，請建立 /srv/tftpboot 和 /srv/tftpboot/pxelinux.cfg 目錄。

- 3 接著依照 節 1.3.3, "PXE 開機" [40] 中的說明，新增用於開機影像的所需適當檔案。
- 4 修改位於 `/etc/xinetd.d/` 下 `xinetd` 的組態，以確定開機時會啟動 TFTP 伺服器：
 - a 如果 `xinetd` 不存在，請使用 `touch tftp`，在此目錄下建立一個名為 `tftp` 的檔案。接著執行 `chmod 755 tftp`。
 - b 開啟檔案 `tftp` 並新增以下內容：

```
service tftp
{
    socket_type           = dgram
    protocol              = udp
    wait                  = yes
    user                  = root
    server                 = /usr/sbin/in.tftpd
    server_args            = -s /tftpboot
    disable                = no
}
```

- c 儲存檔案並使用 `rcxinetd restart` 重新啟動 `xinetd`。

1.3.3 PXE 開機

您可以參考「開機前執行環境 (PXE, Preboot Execution Environment) 規格」(<ftp://download.intel.com/labs/manage/wfm/download/pxespec.pdf>)，以了解一些技術背景資訊和 PXE 的完整規格。

- 1 輸入以下內容，以變更安裝儲存區的目錄，並將 `linux`、`initrd`、`message` 和 `memtest` 檔案複製到 `/srv/tftpboot` 目錄：

```
cp -a boot/loader/linux boot/loader/initrd
    boot/loader/message boot/loader/memtest /srv/tftpboot
```

- 2 請使用 YaST 直接從您的安裝 CD 或 DVD 安裝 `syslinux` 套件。
- 3 輸入以下內容，將 `/usr/share/syslinux/pxelinux.0` 檔案複製到 `/srv/tftpboot` 目錄：

```
cp -a /usr/share/syslinux/pxelinux.0 /srv/tftpboot
```

- 4 輸入以下內容，以變更安裝儲存區的目錄，並將 `isolinux.cfg` 檔案複製到 `/srv/tftpboot/pxelinux.cfg/default`：

```
cp -a boot/loader/isolinux.cfg /srv/tftpboot/pxelinux.cfg/default
```

- 5 編輯 `/srv/tftpboot/pxelinux.cfg/default` 檔案，並移除以 `gfxboot`、`readinfo` 和 `framebuffer` 開頭的行。

- 6 接著，在預設的 `failsafe` 和 `apic` 標籤附加行中，插入以下輸入內容：

`insmod=e100`

透過此輸入內容，用於 Intel 100MBit/s 網路卡的核心模組將會載入到 PXE 用戶端上。此輸入內容會因用戶端的硬體而異，所以必須視情況修改輸入內容。如果您是使用 Broadcom GigaBit 網路卡，輸入內容則應為 `insmod=bcm5700`。

`netdevice=eth0`

此輸入內容可定義必須用於網路安裝的用戶端網路介面。只有當用戶端配有數張網路卡時才必須輸入此內容，且必須視情況修改輸入內容。如果只有使用一張網路卡，則可省略此輸入項目。

`install=nfs://ip_instserver/path_instsource/CD1`

此輸入內容可定義用於用戶端安裝的 NFS 伺服器和安裝來源。安裝伺服器的實際 IP 位址將會取代 `ip_instserver`；而安裝來源的實際路徑則應取代 `path_instsource`。除了 `http`、`ftp` 或 `smb` 通訊協定字首外，HTTP、FTP 或 SMB 來源皆以類似方法指定位址。

重要

如果您需要傳送其他開機選項給安裝來源，例如 SSH 或 VNC 開機參數，那麼請將這些參數附加到 `install` 項目。您可以參考 [節 1.4, "啟動要進行安裝的目標系統" \[46\]](#)，以取得參數概觀和一些範例。

以下將介紹 `/srv/tftpboot/pxelinux.cfg/default` 範例檔案。調整安裝來源的通訊協定字首以符合您的網路設定，接著將 `vnc` 和 `vncpassword` 或 `ssh` 和 `sshpassword` 選項新增到 `install` 輸入內容，以指定連接到安裝程式的方式。輸入內容會以 \ 分隔，且應為一連續的行，不可有斷行，也不能使用 \。

```

default linux

# default
label linux
    kernel linux
        append initrd=initrd ramdisk_size=65536 insmod=e100 \
            install=nfs: //ip_instserver/path_instsource/product

# failsafe
label failsafe
    kernel linux
        append initrd=initrd ramdisk_size=65536 ide=nodma apm=off acpi=off \
            insmod=e100 install=nfs: //ip_instserver/path_instsource/product

# apic
label apic
    kernel linux
        append initrd=initrd ramdisk_size=65536 apic insmod=e100 \
            install=nfs: //ip_instserver/path_instsource/product

# manual
label manual
    kernel linux
        append initrd=initrd ramdisk_size=65536 manual=1

# rescue
label rescue
    kernel linux
        append initrd=initrd ramdisk_size=65536 rescue=1

# memory test
label memtest
    kernel memtest

# hard disk
label harddisk
    kernel
        linux append SLX=0x202

implicit      0
display       message
prompt        1
timeout       100

```

您設定中使用的值將會取代 *ip_instserver* 和 *path_instsource*。

以下章節可做為此安裝中所使用 **PXELINUX** 選項的簡短參考。如需更多可用選項的資訊，請參閱 `/usr/share/doc/packages/syslinux/` 中 `syslinux` 套件的文件。

1.3.4 PXELINUX 組態選項

此處列出的選項為所有可用於 PXELINUX 組態檔案選項的子集合。

DEFAULT kernel options...

設定預設的核心指令行。當 PXELINUX 自動啟動時，PXELINUX 除了做為執行自動開機時自動新增的自動選項，還會做為出現開機提示時，當您鍵入 DEFAULT 後執行的輸入內容。

如果目前沒有任何組態檔案或目前的組態檔案中沒有 DEFAULT 輸入內容，那麼預設值為不包含任何選項的核心名稱 "linux"。

APPEND options...

新增一個或一個以上的選項至核心指令行。這些新增的選項是用於自動和手動開機。新增的選項位於每個核心指令行的開頭。一般而言，您只要明確輸入核心選項即可覆蓋這些選項。

LABEL label KERNEL image APPEND options...

表示是否輸入應做為 PXELINUX 開機核心的 *label*，而非輸入開機 *image*；且應使用指定的 APPEND 選項，而非使用該檔案全域區段中指定的選項 (在第一個 LABEL 指令前)。*image* 的預設值與 *label* 相同，且如果沒有指定 APPEND，則會使用全域項目中的預設值 (如果有的話)。您最多可輸入 128 個 LABEL 項目。

請注意，GRUB 將使用以下語法：

```
title mytitle
  kernel my_kernel my_kernel_options
  initrd myinitrd
```

而 PXELINUX 是使用以下語法：

```
label mylabel
  kernel mykernel
  append myoptions
```

如果以檔案名稱做為標籤內容將會混淆標籤，所以標籤內容必須是唯一的。舉例來說，在 PXELINUX 中將無法辨識 "v2.1.30" 和 "v2.1.31" 這兩個標籤，因為兩個標籤皆使用相同的 DOS 檔案名稱。

且核心不一定要是 Linux 核心，它可以是開機磁區或 COMBOOT 檔案。

APPEND -

不附加。包含單一連字號的 APPEND 在 LABEL 區段中會被做為一個引數，且可用來附寫全域的 APPEND。

LOCALBOOT type

在 PXELINUX 上，指定 LOCALBOOT 0 而不指定 KERNEL 選項表示會呼叫此特定標籤，且會使用本地磁碟開機，而不使用核心開機。

引數	說明
0	執行正常開機
4	使用仍常駐在記憶體中的「通用網路驅動程式介面」(Universal Network Driver Interface, UNDI) 驅動程式執行本地開機。
5	使用仍常駐在記憶體中完整的 PXE 堆疊 (包括 UNDI 驅動程式) 執行本地開機。

其他值尚未定義。如果您對 UNDI 或 PXE 堆疊不甚了解，請指定 0。

TIMEOUT time-out

表示自動開機後開機提示等待的時間，單位為 1/10 秒。使用者只要在鍵盤上按下任何一個鍵，就會取消逾時，且會如預期開始執行使用者設定的指令。如果逾時時間設為 0，則會完全停用逾時功能 (此設定值亦為預設值)。

可能的最大逾時值為 (剛好少於一個小時)。

PROMPT flag_val

如果 flag_val 為 0，只有在按下 `[Shift]` 或 `[Alt]`，或者已設定 `[Caps Lock]` 或 `[Scroll lock]` 時才會顯示開機提示 (此設定值為預設值)。如果 flag_val 為 1，則永遠會顯示開機提示。

```
F2 filename
F1 filename
..etc...
F9 filename
F10filename
```


在開機提示按下功能鍵時，畫面上將會顯示指定的檔案。這可用來執行開機前線上說明 (應是用於核心指令行選項的說明)。若使用較舊的版本反向相容，可將 **F10** 當做 **F0** 輸入。請注意，目前還無法將檔案名稱結合至 **F11** 和 **F12**。

1.3.5 準備用於 PXE 開機的目標系統

請將 PXE 選項納入 BIOS 開機順序中的選項，以準備用於 PXE 開機的系統 BIOS。

警告

但請勿在 BIOS 中，將 PXE 選項置於硬碟開機選項之前。否則，此系統會在您每次重新開機時嘗試重新安裝系統。

1.3.6 準備用於網路喚醒功能的目標系統

您需要使用適當的 BIOS 選項才可在安裝前啟用網路喚醒功能 (WOL)。另外，請記下目標系統的 MAC 位址。啟動網路喚醒功能時將會用到此資料。

1.3.7 網路喚醒功能

「網路喚醒功能」可讓機器透過經由網路傳送且包含機器 MAC 位址的特定網路套件，來進行開機。因為世界上的每台機器都有一個獨一無二的 MAC 識別碼，所以您不需擔心會不小心啟動到錯誤的機器。

重要

如果控制機器與喚醒安裝目標的位置不在同一個網路區段，請將要傳送的 WOL 要求設定為多重廣播，或者遠端控制網路區段上傳送這些要求的機器。

1.3.8 網路喚醒功能

- 1 登入為 root 使用者。

2 啟動 *YaST* → 安裝和移除軟體 並安裝套件 `netdiag`。

3 開啟終端機並以 `root` 身分輸入以下指令以喚醒目標：

```
ether-wakemac_of_target
```

目標的實際 MAC 位址將取代 `mac_of_target`。

1.4 啟動要進行安裝的目標系統

基本上，除了 節 1.3.7, "網路喚醒功能" [45] 和 節 1.3.3, "PXE 開機" [40] 中提到的方法，還有兩種不同的方法可以自訂安裝的開機程序。您可以使用預設開機選項和 F 鍵，或者使用安裝開機畫面的開機選項提示，來傳送安裝核心可能需要用於此特定硬體的任何開機選項。

1.4.1 使用預設開機選項

之前已在 章 1, 使用 *YaST* 安裝 (↑啟動) 中詳細描述了開機選項的內容。

一般而言，只要選擇 安裝 就可以開始安裝開機程序。如果發生問題時，則需使用 安裝 — 關閉 *ACPI* 或 安裝 — 安全設定 選項。

若需更多安裝程序疑難排解的資訊，請參閱 節 9.2, "安裝問題" (章 9, 一般問題和解決方案, ↑啟動)。

1.4.2 使用 F 鍵

畫面下方的功能表列會一些提供部分設定所需的進階功能。如果您要輸入做為開機選項的參數，您可以在不清楚所需參數詳細語法的情況下，使用 F 鍵指定要傳送到安裝常式的其他選項 (請參閱 節 1.4.3, "使用自訂開機選項" [48])。

請參閱以下表格，以取得所有可用選項的資訊。

表格 1.1 安裝時使用 F 鍵

鍵	目的	可用選項	預設值
F1	提供說明	無	無
F2	選取安裝語言	所有支援的語言	英語
F3	變更安裝的畫面解析度	• 文字模式 • VESA • 解析度 #1 • 解析度 #2 • ...	• 預設值會因您的圖形硬體而異
F4	選取安裝來源	• CD-ROM/DVD • SLP • FTP • HTTP • NFS • SMB • 硬碟	CD-ROM/DVD
F5	套用驅動程式更新磁片	驅動程式	無

1.4.3 使用自訂開機選項

使用適當的開機選項設定可以協助您進行安裝程序。您之後也可以使用 `linuxrc` 常式來設定許多參數，但使用開機選項是較簡單的方法。在某些自動化設定中，`initrd` 和 `info` 檔案會提供一些開機選項。

以下表格列出本章節中提及的所有安裝方式，其中包括開機所需的參數和對應的開機選項。您只要依照順序將此表格中出現的內容附加到檔案中，即可開機選項字串送到安裝常式中。例如 (不要斷行)：

```
install=... netdevice=... hostip=...netmask=... vnc=... vncpassword=...
```

請使用適用於您設定的值來取代所有 值 (...)。

表格 1.2 用於此章節的安裝 (開機) 方式

安裝方式	用於開機的參數	開機選項
章 1, 使用 <i>YaST</i> 安裝 (↑ 啟動)	無：系統自動開機	不需要
節 1.1.1, "透過 VNC 執行簡易遠端安裝 — 靜態網路組態" [22]	• 安裝伺服器的位置 • 網路設備 • IP 位址 • 網路遮罩 • 閘道 • 啟用 VNC • VNC 密碼	• <code>install=(nfs, http, ftp, smb): //path_to_instmedia</code> • <code>netdevice=some_netdevice</code> (只有在有數個網路設備時才會需要) • <code>hostip=some_ip</code> • <code>netmask=some_netmask</code> • <code>gateway=ip_gateway</code> • <code>vnc=1</code> • <code>vncpassword=some_password</code>

安裝方式	用於開機的參數	開機選項
節 1.1.2, "透過 VNC 執行簡易遠端安裝 — 透過 DHCP 的動態網路組態" [23]	• 安裝伺服器的位置 • 啟用 VNC • VNC 密碼	• <code>install=(nfs, http, ftp, smb): //path_to_instmedia</code> • <code>vnc=1</code> • <code>vncpassword=some_password</code>
節 1.1.3, "透過 VNC 執行遠端安裝 — PXE 開機和網路喚醒功能" [24]	• 安裝伺服器的位置 • TFTP 伺服器的位置 • 啟用 VNC • VNC 密碼	不適用；透過 PXE 和 DHCP 管理的程序
節 1.1.4, "透過 SSH 執行簡易遠端安裝 — 靜態網路組態" [25]	• 安裝伺服器的位置 • 網路設備 • IP 位址 • 網路遮罩 • 閘道 • 啟用 SSH • SSH 密碼	• <code>install=(nfs, http, ftp, smb): //path_to_instmedia</code> • <code>netdevice=some_netdevice</code> (只有在有數個網路設備時才會需要) • <code>hostip=some_ip</code> • <code>netmask=some_netmask</code> • <code>gateway=ip_gateway</code> • <code>usessh=1</code> • <code>sshpassword=some_password</code>
節 1.1.5, "透過 SSH 執行簡易遠端安裝 — 透	• 安裝伺服器的位置	• <code>install=(nfs, http, ftp, smb): //path_to_instmedia</code>

安裝方式	用於開機的參數	開機選項
過 DHCP 的動態網路組態" [27]	• 啟用 SSH • SSH 密碼	• <code>usessh=1</code> • <code>sshpassword=some_password</code>
節 1.1.6, "透過 SSH 執行遠端安裝 — PXE 開機和網路喚醒功能" [28]	• 安裝伺服器的位置 • TFTP 伺服器的位置 • 啟用 SSH • SSH 密碼	不適用；透過 PXE 和 DHCP 管理的程序
提示 如需更多用來啟動 Linux 系統的 linuxrc 開機選項資訊，請參閱 <code>/usr/share/doc/packages/linuxrc/linuxrc.html</code>。		

1.5 監控安裝程序

有數種選項可以用來遠端監控安裝程式。如果在開機時已指定用於安裝的正確開機選項，將會使用 VNC 或 SSH 從遠端工作站來控制安裝和系統組態。

1.5.1 安裝 VNC

您可以使用 VNC 檢視器軟體，從任何虛擬的作業系統，遠端控制 SUSE Linux 安裝。此章節將介紹如何使用 VNC 檢視器應用程式或網頁瀏覽器來進行安裝。

準備安裝 VNC

準備安裝 VNC 時，您在安裝目標上必須執行的動作就是在第一次開始安裝時，提供適當的開機選項 (請參閱 節 1.4.3, "使用自訂開機選項" [48])。目標系統將會啟動到以文字為基礎的環境，接著，請等待 VNC 用戶端連線到安裝程式。

安裝程式將會宣告連線到安裝所需的 IP 位址和顯示編號。如果您有實體存取目標系統的權限，系統開機後會隨即提供用於安裝的資訊。當 VNC 用戶端軟體提示輸入資料時，請輸入資料並提供您的 VNC 密碼。

因為安裝目標本身會透過 OpenSLP 進行宣告，所以您可以透過 SLP 瀏覽器擷取安裝目標的位址資訊，且不需實體連線到您網路設定提供的安裝本身。另外，所有機器都必須支援 OpenSLP。

- 1 啟動 KDE 檔案和 Konqueror 網頁瀏覽器。
- 2 在位置列中輸入 `service: //yast.installation.suse`。

接著，Konqueror 畫面中會出現一個代表目標系統的圖示。按一下此圖示可啟動要執行安裝的 KDE VNC 檢視器。另外，請使用提供的 IP 位址執行您的 VNC 檢視器軟體，並在 IP 位置結尾加上 `:1`，以顯示安裝正在執行中。

連線到安裝程式

基本上，有兩種方法可以連線到 VNC 伺服器 (在此案例中是指安裝目標)。您可以在任何作業系統上啟動獨立的 VNC 檢視器應用程式，或是使用已啟用 Java 的網頁瀏覽器進行連線。

您可以使用 VNC，從任何其他作業系統 (包括其他 Linux 版本、Windows 或 Mac 作業系統) 控制 Linux 系統的安裝。

若是使用 Linux 機器，請確定已安裝套件 `tightvnc`。若是使用 Windows 機器，請安裝此應用程式 (可在 TightVNC 首頁 <http://www.tightvnc.com/download.html> 取得) 的 Windows 連接埠。

若要連線到在目標機器上執行的安裝程式，請執行下列步驟：

- 1 啟動 VNC 檢視器。
- 2 輸入由 SLP 瀏覽器或安裝程式本身提供的安裝目標 IP 位址和顯示編號：

```
ip_address:display_number
```

接著會在您的桌面上開啟一個視窗，並顯示做為一般本地安裝的 YaST 畫面。

使用網頁瀏覽器連線到安裝程式，可讓您完全不需理會 VNC 軟體或舊版的作業系統。只要瀏覽器應用程式已啟用 Java 支援，您就可以使用任何瀏覽器 (Firefox、Internet Explorer、Konqueror、Opera 等) 來執行 Linux 系統安裝。

若要執行 VNC 安裝，請執行下列步驟：

- 1 啟動您要使用的網頁瀏覽器。
- 2 在位址提示中輸入以下內容：

```
http://ip_address_of_target:5801
```

- 3 接著系統會提示您輸入 VNC 密碼。然後，瀏覽器視窗會顯示做為一般本地安裝的 YaST 畫面。

1.5.2 安裝 SSH

您可以使用 SSH，透過任何 SSH 用戶端軟體遠端控制 Linux 機器的安裝。

準備安裝 SSH

除了安裝適當的軟體套件 (用於 Linux 的 OpenSSH 和用於 Windows 的 PuTTY) 外，您只需要傳送適當的開機選項即可開始安裝 SSH。如需詳細資料，請參閱 [節 1.4.3, "使用自訂開機選項" \[48\]](#)。安裝程式會根據作業系統的不同，依預設值，在任何 SUSE Linux 上安裝 OpenSSH。

連線到安裝程式

- 1 擷取安裝目標的 IP 位址。

如果您有實體存取目標機器的權限，只要在初始開機後，在控制台提供安裝常式 IP 位址即可。或者，也可以提供在 DHCP 伺服器組態中指定給此特定主機的 IP 位址。

- 2 接著，在指令行輸入以下指令：

```
ssh -X root@ip_address_of_target
```

安裝目標的實際 IP 位址將取代 *ip_address_of_target*。

- 3 系統會提示輸入使用者名稱，接著請輸入 `root`。

- 4 當提示輸入密碼時，請輸入透過 SSH 開機選項設定的密碼。

完成驗證後，會出現一個指令行，提示您輸入安裝目標。

- 5 請輸入 `yast` 啟動安裝程式。

接著，會出現一個視窗，顯示 章 1, 使用 *YaST* 安裝 (↑ 啟動) 中說明的一般 YaST 畫面。

進階磁碟安裝

複雜的系統組態必須在特定磁碟上進行安裝。要取得 SCSI 設備的永久性設備名稱，請使用特定的啟動程序檔。LVM (Logical Volume Management，邏輯磁碟區管理) 是一個磁碟分割結構，這項設計比標準安裝中使用的實體分割方式更為靈活。它的快照功能可讓您更輕鬆地建立資料備份。另外，RAID (Redundant Array of Independent Disks，多餘低價磁碟陣列) 可確保大量資料的完整性、效能和容錯。

2.1 SCSI 設備的永久性設備名稱

啟動系統時，會以較動態的方式，指定設備檔案名稱給 SCSI 設備。只要設備的編號或組態不變更，就沒有問題。不過，如果新增 SCSI 硬碟，而核心先偵測到新硬碟，之後才偵測到舊硬碟，則會指定新名稱給舊硬碟，導致與裝載表格 `/etc/fstab` 中的項目將不再相符。

要避免此問題，可以使用系統啟動程序檔 `boot.scsidev`。使用 `/sbin/insserv` 並在 `/etc/sysconfig/scsidev` 設定它的參數，啟用此程序檔。程序檔 `/etc/rc.d/boot.scsidev` 會在啟動程序時處理 SCSI 設備的設定，並在 `/dev/scsi/` 底下輸入永久性設備名稱。然後，`/etc/fstab` 就可以使用這些名稱。。此外，`/etc/scsi.alias` 可以用來定義 SCSI 組態的永久性名稱。在 `/etc/scsi` 中設備的名稱網要在 `man scsidev` 中有所說明。

在 `runlevel` 的進階模式，啟動層次 B 的 `boot.scsidev`。啟動程序時用於產生名稱所需的連結，會建立在 `/etc/init.d/boot.d`。

提示: 設備名稱和 `udev`

至於 SUSE Linux，雖然仍支援 `boot.scsidev`，不過建立永久性設備名稱較好的方法，是使用 `udev` 在 `/dev/by-id/` 建立具永久性名稱的設備節點。

2.2 LVM 組態

本小節簡短地說明在 LVM 背後的原則，以及它在許多狀況下好用的基本功能。在 [節 2.2.2, "使用 YaST 進行 LVM 組態" \[58\]](#) 中，學習如何使用 YaST 設定 LVM。

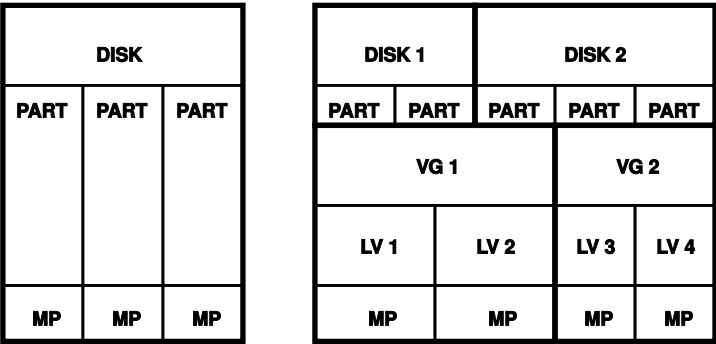
警告

使用 LVM 可能會增加風險，如遺失資料。這些危險也包括應用程式當機、電源中斷和錯誤指令。執行 LVM 或重新設定磁碟區前，請儲存您的資料。決不要在沒有備份的情形下工作。

2.2.1 邏輯磁碟區管理

邏輯磁碟區管理 (Logical Volume Manager, LVM) 可以在數個檔案系統上彈性地散佈硬碟空間。在安裝過程中的啟始分割已經完成後，有時需要變更硬碟空間的分割，因此開發此工具。因為要修改執行系統上的分割區很困難，LVM 提供記憶體空間的虛擬集區 (磁碟區群組，簡稱 VG)。如有必要，可從虛擬集區建立邏輯磁碟區 (LV)。作業系統可以存取這些 LV，而不是存取實體分割區。磁碟區群組可以延伸至一個以上的磁碟，因此數個磁碟或是磁碟的某些部份可能會構成單一的 VG。LVM 這種方法提供從實體磁碟空間擷取的方法，允許使用比實體重新分割更為簡單和安全的方式來變更分割。您可以在 "分割區類型" (章 1, [使用 YaST 安裝, ↑ 啟動](#)) 與 [節 3.7.5, "磁碟分割程式" \(章 3, \[使用 YaST 進行系統組態, ↑ 啟動\]\(#\)\)](#) 中找到有關實體分割的背景資訊。

圖形 2.1 實體分割與 LVM



圖形 2.1, "實體分割與 LVM" [57] 比較實體分割 (左邊) 與 LVM 分割 (右邊)。在左邊，單一個磁碟已分割為三個實體分割區 (PART)，每一個都會指定裝載點 (MP)，讓作業系統存取它們。在右邊，已經個別將兩個磁碟分割成兩個及三個實體分割區。已經定義兩個 LVM 磁碟區群組 (VG1 與 VG2)。VG1 包含 DISK1 的兩個磁碟區以及 DISK2 的一個磁碟區。VG2 包含 DISK2 其餘的兩個磁碟區。在 LVM 中，在磁碟區群組中合併的實體磁碟分割區稱為實體磁碟區 (PV)。在某些磁碟區群組中，已經定義四個邏輯磁碟區 (LV1 至 LV4)，作業系統可以透過指定的裝載點來使用。在不同的邏輯磁碟區之間的邊緣，不需要對齊任何分割區的邊緣。請參閱此範例中 LV1 與 LV2 之間的邊緣。

LVM 功能：

- 數個硬碟或分割區可以在大的邏輯磁碟區結合成一個。
- 如果組態適用，當可用空間耗盡時，可以擴大 LV (如 /usr)。
- 使用 LVM，甚至可以在執行的系統中新增硬碟或 LV。然而，這種作法需要能執行此動作的熱交換式硬體。
- 可以啟動「等量分割模式」，將邏輯磁碟區的資料流分散至數個實體磁碟區。如果這些實體磁碟區是在不同的磁碟上，這可改善讀寫效能，就像 RAID 0 一樣。
- 快照功能能夠讓執行系統中的備份 (特別是伺服器) 成為一致。

使用 LVM 的這些功能，對於使用頻繁的家用個人電腦或小型伺服器而言，在效能上可以看到改善。如果您的資料會一直累積，如資料庫、音樂歸檔或使用者目錄等，LVM 就是適合您的工具。這樣能夠允許比實體硬碟還大的檔案系統。

LVM 的另一個好處是最大可以增加到 256 個 LV。不過，請記住使用 LVM 與使用傳統分割區是不同的。有關設定 LVM 的說明及詳細資訊，請參閱官方網站的 LVM HOWTO (<http://tldp.org/HOWTO/LVM-HOWTO/>)。

從核心 2.6 版本開始，即可使用 LVM 2 版本，它可以向下相容之前的 LVM，而且可以繼續管理舊的磁碟區群組。建立新的磁碟區群組時，請決定要使用新的格式或能夠向下相容的版本。LVM2 不需要任何核心修補程式。它利用整合於核心 2.6 中的設備映射程式。此核心僅支援 LVM 2 版本，因此，談到 LVM 時，本節永遠是指 LVM 2 版本。

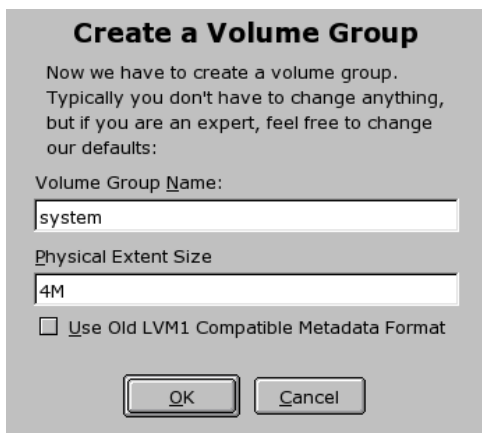
2.2.2 使用 YaST 進行 LVM 組態

YaST LVM 組態可以從 YaST 專家分割程式完成 (請參閱 節 3.7.5, "磁碟分割程式" (章 3, *使用 YaST 進行系統組態*, ↑啟動))。這個專業磁碟分割工具讓您編輯和刪除現有磁碟分割，以及建立應該與 LVM 一起使用的新磁碟分割。接著，先按一下 *建立* → *不要格式化*，以建立 LVM 分割區，再選取 *0x8E Linux LVM* 做為分割區的識別碼。在建立所有要與 LVM 一起使用的分割區後，按一下 *LVM* 以啟動 LVM 組態。

建立磁碟區群組

如果在系統上沒有磁碟區群組，將會提示您新增一個磁碟區群組 (請參閱 圖形 2.2, "[建立磁碟區群組](#)" [59])。您可以使用 *新增群組* 來建立其他的群組，但是通常單一磁碟區群組就已足夠。建議以 `system` 作為 SUSE Linux 系統檔案所在之磁碟區群組的名稱。實體擴充大小定義了磁碟區群組中實體區塊的大小。在磁碟區群組中的所有磁碟空間都會以此大小的區塊來處理。此值通常設為 4 MB，並允許將實體及邏輯磁碟區的最大容量設為 256 GB。只有在需要大於 256 GB 的邏輯磁碟區時，才需要增加實體擴充大小的容量 (例如，設為 8、16 或 32 MB)。

圖形 2.2 建立磁碟區群組



Create a Volume Group

Now we have to create a volume group.
Typically you don't have to change anything,
but if you are an expert, feel free to change
our defaults:

Volume Group Name:

Physical Extent Size

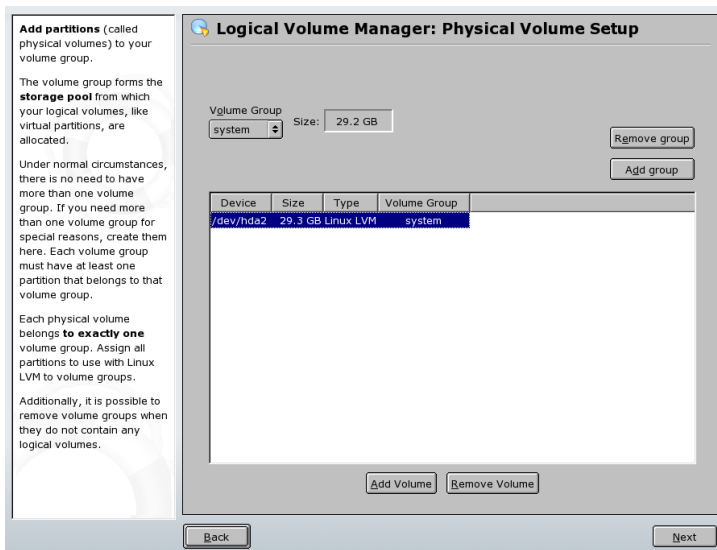
☐ Use Old LVM1 Compatible Metadata Format

設定實體磁碟區

一旦建立磁碟區群組，以下對話方塊就會列出具有 "Linux LVM" 或 "Linux native" 類型的所有分割區。不會顯示交換和 DOS 分割區。如果已經指定分割區給磁碟區群組，磁碟區群組的名稱就會顯示在清單中。未指定的分割區以 "--" 表示。

如果有數個磁碟區群組，請在左上角的選擇方塊中設定目前的磁碟區群組。右上角的按鈕可以建立其他的磁碟區群組以及刪除現有的磁碟區群組。僅能刪除沒有指定分割區的磁碟區群組。所有指定給磁碟區群組的分割區，又稱為實體磁碟區 (PV)。

圖形 2.3 實體磁碟區設定

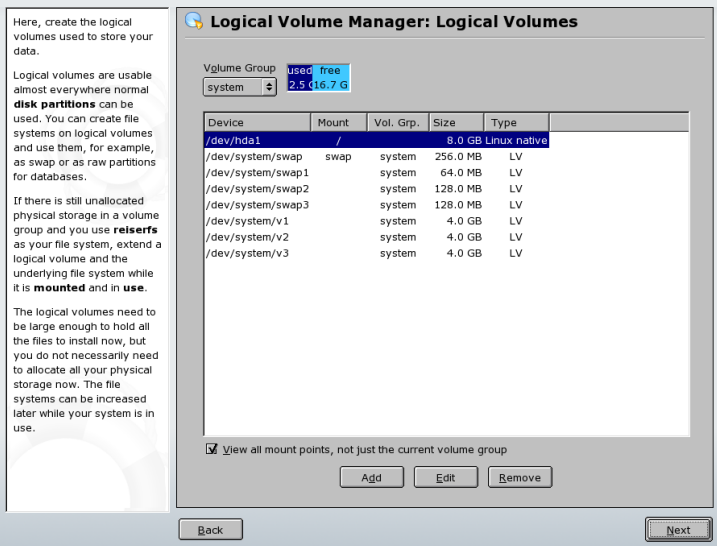


若要新增之前未指定的分割區給選取的磁碟區群組，請先按一下分割區，再按**新增磁碟區**。此時，磁碟區群組的名稱，會輸入到選取分割區的旁邊。指定為LVM保留的所有分割區給磁碟區群組。否則，仍然不會使用分割區上的空間。結束此對話方塊前，每個磁碟區群組必須指定至少一個實體磁碟區。在指定所有的實體磁碟區後，按一下**下一步**以繼續邏輯磁碟區的組態。

設定邏輯磁碟區

在磁碟區群組已經使用實體磁碟區填滿後，在下一個對話方塊中定義作業系統應該使用的邏輯磁碟區。在左上角的選項方塊中修改目前的磁碟區群組。接下來，會顯示目前磁碟區群組的可用空間。下方的清單包含該磁碟區群組中所有的邏輯磁碟區。指定裝載點的所有標準Linux分割區、所有交換分割區、以及所有已經存在的邏輯磁碟區都列示於此。視需要**新增**、**編輯**以及**移除**邏輯磁碟區，直到在磁碟區群組中的所有空間都使用完畢。至少指定一個邏輯磁碟區給每個磁碟區群組。

圖形 2.4 邏輯磁碟區管理

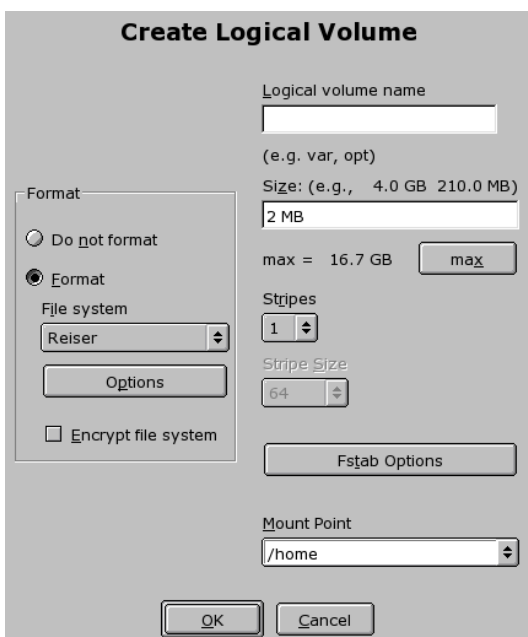


若要建立新的邏輯磁碟區，請按一下 **新增**，然後填寫開啟的快顯視窗。可以輸入磁碟分割、大小、檔案系統以及裝載點。一般而言，如 `reiserfs` 或 `ext2` 之類的檔案系統，是建立於邏輯磁碟區上，接著再指定裝載點。儲存於此邏輯磁碟區上的檔案，可以在已安裝系統的此裝載點上找到。此外也可以將邏輯磁碟區中的資料流分散至數個實體磁碟區 (等量分割)。如果這些實體磁碟區是在不同的硬碟上，通常可以改善讀寫效能 (像 `RAID 0` 一樣)。不過，具有 n 個等量磁區的等量 LV，只有在 LV 所需的硬碟空間可以平均分散給 n 個實體磁碟區時，才能正確建立。例如，如果只有兩個可用的實體磁碟區，那麼就不可能建立具有三個等量磁區的邏輯磁碟區。

警告：等量磁區

YaST 在此時沒有機會驗證您所輸入的等量磁區之正確性。在此所犯的錯誤只有稍後在磁碟上執行 LVM 時才會顯示出來。

圖形 2.5 建立邏輯磁碟區



The image shows a 'Create Logical Volume' dialog box. It has a title bar 'Create Logical Volume'. On the left, there is a 'Format' section with a radio button for 'Do not format' and a selected radio button for 'Format'. Below 'Format' is a 'File system' dropdown menu set to 'Reiser', an 'Options' button, and a checkbox for 'Encrypt file system'. On the right, there is a 'Logical volume name' text field, a hint '(e.g. var, opt)', a 'Size: (e.g., 4.0 GB 210.0 MB)' label, and a size input field set to '2 MB'. Below this is a 'max = 16.7 GB' label and a 'max' button. Further down are 'Stripes' and 'Stripe Size' dropdown menus, both set to '1' and '64' respectively. At the bottom right is an 'Fstab Options' button. At the very bottom is a 'Mount Point' dropdown menu set to '/home'. At the bottom left are 'OK' and 'Cancel' buttons.

如果已經在系統上設定 LVM，現在就可以輸入現有的邏輯磁碟區。在繼續前，請指定適當的裝載點到這些邏輯磁碟區。使用 下一步，返回「YaST 專家分割程式」，然後在那完成您的工作。

直接 LVM 管理

如果您已設定 LVM，而且只想變更某些項目，可以使用另一種方式來完成。在「YaST 控制中心」選取 *System* → *LVM*。基本上此對話方塊允許如上方所描述的相同動作，除了實體分割以外。它以兩個清單顯示現有的實體磁碟區及邏輯磁碟區，而且您可以使用已經描述的方法來管理 LVM 系統。

2.3 軟體 RAID 組態

RAID (多餘低價磁碟陣列，Redundant Array of Inexpensive Disks) 的用途是將數個硬碟分割區組合一個大型 虛擬硬碟，以達最佳化效能、資料安全性或是兩者兼具的功能。但是，使用此方法，是犧牲一個好處來換取另一個。大部分 RAID

控制器使用 SCSI 通訊協定，因為它可利用比 IDE 通訊協定更有效的方式處理較大量的硬碟，並且更適合指令的平行處理。有部分的 RAID 控制器支援 IDE 或 SATA 硬碟。請參照 <http://cdb.suse.de> 的 Hardware Database (硬體資料庫)。

2.3.1 軟體 RAID

由於 RAID 控制器，可能會很昂貴，軟體 RAID 就可以接下這些任務。SUSE Linux 提供結合數個硬碟至一個軟體 RAID 系統的選項，結合 YaST 的使用，對硬體 RAID 是非常好的替代方案。RAID 一詞是表示將數個硬碟結合成 RAID 系統的一些策略，每個都有不同的目標、優點及特色。這些變化通常稱為 *RAID 層級*。

常用 RAID 層級為：

RAID 0

此層級將每個檔案的區塊分散於多個磁碟，以提升您的資料存取效能。實際上，它不是真的 RAID，因為它不提供資料備份，但是此類型系統的名稱 *RAID 0* 已經成為標準。使用 RAID 0，就可將兩個以上的磁碟聚集在一起。效能非常好，但是如果其中一個硬碟錯誤，RAID 系統便會損毀而且資料會遺失。

RAID 1

此層級對資料提供足夠的安全性，因為資料是以 1: 1 複製到另一個硬碟。這就是所謂的 *硬碟鏡射*。如果其中一個硬碟損毀，另一個硬碟上有其內容的複本。只有其中一個會受到損害，但不會危害到資料。在複製程序中的寫入效能比使用單一磁碟存取時較差 (慢了百分之十到二十)，但是讀取存取卻較任何一般實體硬碟快得多，因為資料已複製，因此可平行掃描。一般而言，可以說「層級 1」比單一磁碟的讀取異動率快了將近兩倍，而且與單一磁碟的寫入異動率幾乎相同。

RAID 2 與 RAID 3

這些都不是一般的 RAID 實作。「層級 2」在是位元層級等量分割資料，而不是在區塊層級。「層級 3」提供具有專用同位磁碟的位元層級等量分割，但是無法同時服務多個要求。這兩個層級都很少使用。

RAID 4

「層級 4」提供與「層級 0」相同的區塊層級等量分割，並且結合專用的同位磁碟。在資料磁碟失敗時，會使用同位資料以建立替代的磁碟。不過，同位磁碟可能造成寫入存取的瓶頸。儘管如此，有時還是會使用「層級 4」。

RAID 5

RAID 5 是在「層級 0」與「層級 1」之間效能和備用方面最佳的折衷方法。硬碟空間等於使用的磁碟數減一。使用 RAID 0 可將資料分布至各個硬碟。在其中一個分割區上建立的同位區塊，即是基於安全性的因素。它們以 XOR 互相連結；在系統失敗時，透過 XOR 藉由對應同位區塊重新建構內容。使用 RAID 5，不會有一個以上的硬碟同時失敗。如果一個硬碟失敗，必須立即更換以避免資料遺失的風險。

其他的 RAID 層級

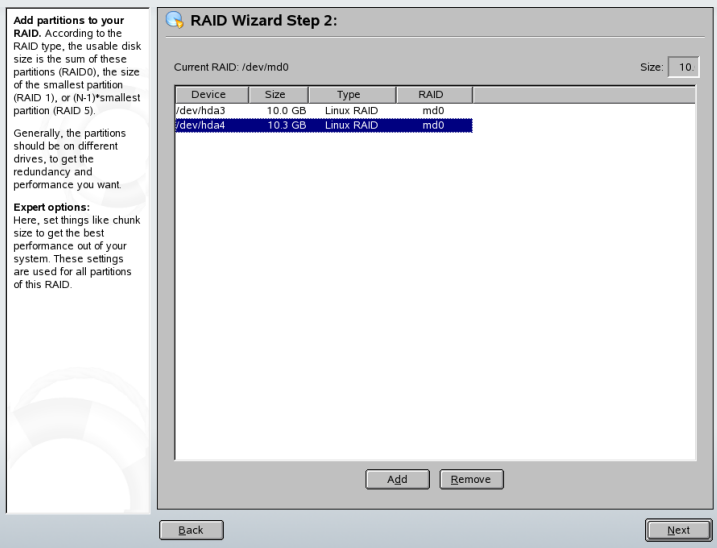
已經開發一些其他的 RAID 層級 (RAIDn、RAID10、RAID 10、RAID 0+1、RAID 30、RAID 50 等等)，有些是硬體廠商所建立的專用實作。這些層級並不是很普遍，因此在這裏並不作說明。

2.3.2 使用 YaST 進行軟體 RAID 組態

YaST 軟體 RAID 組態可以從 YaST 專家分割程式完成 (請參閱 節 3.7.5, "磁碟分割程式" (章 3, 使用 YaST 進行系統組態, ↑啟動))。這個專業磁碟分割工具讓您編輯和刪除現有磁碟分割，以及建立應該與軟體 RAID 一起使用的新磁碟分割。接著，先按一下 **建立** → **不要格式化**，以建立 RAID 分割區，再選取 *0xFD Linux RAID* 做為分割區的識別碼。對於 RAID 0 和 RAID 1，至少需要兩個分割區。對於 RAID 1，通常剛好兩個而不需更多。如果使用 RAID 5，至少需要三個分割區。建議您僅使用大小相同的分割區。RAID 分割區應該儲存在不同的硬碟中，以減少當其中一個損壞時 (RAID 1 和 5) 遺失資料的風險，並最佳化 RAID 0 的效能。在建立所有的磁碟區以便和 RAID 一起使用後，按一下 **RAID** → **建立 RAID** 以啟動 RAID 組態。

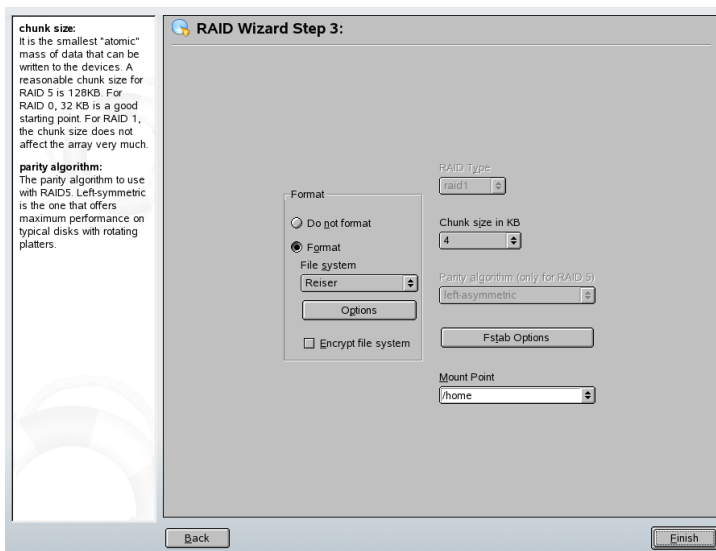
在下一個對話方塊中，在 RAID 層級 0、1 及 5 之間選擇 (請參閱 節 2.3.1, "軟體 RAID" [63] 以取得詳細資訊)。按一下 **下一步** 後，下列對話方塊會列出具有 "Linux RAID" 或 "Linux native" 類型的所有分割區 (請參閱 圖形 2.6, "RAID 分割區" [65])。不會顯示交換和 DOS 分割區。如果已經指定分割區給 RAID 磁碟區，RAID 設備 (例如 `/dev/md0`) 的名稱就會顯示在清單中。未指定的分割區以 "--" 表示。

圖形 2.6 RAID 分割區



若要新增之前未指定的分割區給選取的 RAID 磁碟區，請先按一下分割區，再按 **新增**。此時，RAID 設備的名稱，會輸入到選取分割區的旁邊。指定保留給 RAID 的所有分割區。否則，仍然不會使用分割區上的空間。在指定所有的分割區後，按一下 **下一步** 以進入設定對話方塊，在此您可以微調效能 (請參閱 [圖形 2.7, "檔案系統設定" \[66\]](#)。)

圖形 2.7 檔案系統設定



使用傳統分割時，會設定要使用的檔案系統、加密以及 RAID 磁碟區的裝載點。檢查 永久超級區塊 以確定在開機時可識別 RAID 分割區。在按一下 完成 以完成組態之後，請參閱 `/dev/md0` 設備及其他在專家分割程式中以 **RAID** 表示的設備。

2.3.3 疑難排解

檢查檔案 `/proc/mdstats` 以找出 RAID 分割區是否已損毀。當系統失敗時，請關閉 Linux 系統並使用以相同方式磁碟分割的新硬碟來更換損壞的硬碟。然後重新啟動系統，並輸入 `mdadm /dev/mdX --add /dev/sdX` 指令。使用特定的裝置識別碼取代 X。如此可將硬碟自動整合到 RAID 系統並完整地重新建構。

2.3.4 若需詳細資訊

可在下列網址的 HOWTO 中找到組態指南及軟體 RAID 的詳細資訊：

- `/usr/share/doc/packages/raidtools/Software-RAID.HOWTO.html`
- <http://en.tldp.org/HOWTO/Software-RAID-HOWTO.html>

也有 Linux RAID 郵寄清單可供參考，如 <http://www.mail-archive.com/linux-raid@vger.rutgers.edu>。

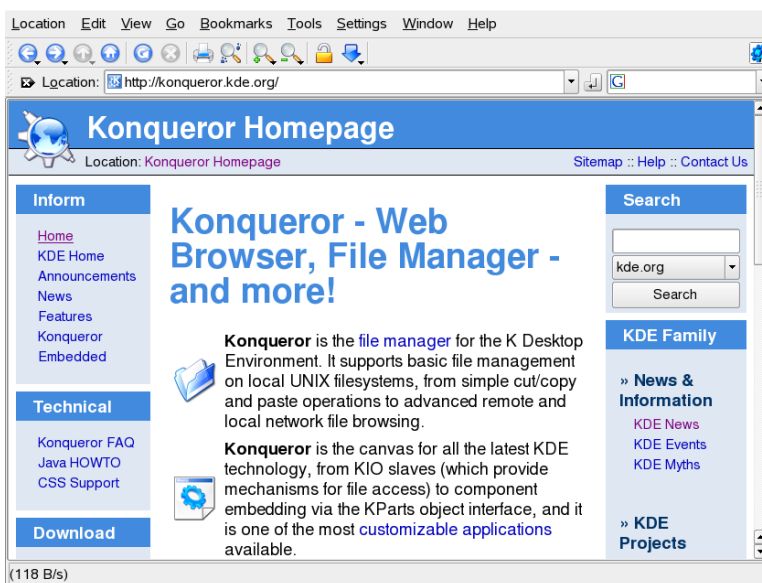
部 II. 網際網路

網頁瀏覽器 Konqueror

Konqueror 不僅是多用途檔案管理員。它也是現代化網頁瀏覽器。如果您以面板中的圖示啟動瀏覽器，Konqueror 就會使用網頁瀏覽器設定檔來開啟。就瀏覽器功能而言，Konqueror 提供標籤式瀏覽、儲存有圖形的網頁功能、網際網路關鍵字、書籤、以及 Java 和 JavaScript 的支援。

可以從主功能表啟動 Konqueror 或輸入指令 `konqueror`。若要載入網頁，請在位置列中輸入其網址，例如 <http://www.suse.com>。Konqueror 現在會試著連結到該網址並顯示其網頁。在網址的開頭處輸入通訊協定 (在此例中為 `http://`) 並非必要的。程式可以自動完成網址，不過這只會作用於網址。如果是 FTP 位址，則一定要在輸入欄位的開頭輸入 `ftp://`。

圖形 3.1 Konqueror 的瀏覽器視窗



3.1 標籤式瀏覽

如果您經常一次使用一個以上的網頁，標籤式瀏覽可以讓您更易於在網頁間切換。會將網站載入視窗內的個別標籤中。優點就是您可以保有更多的桌面控制權，因為您只有一個主視窗。登出之後，KDE 工作階段管理允許在 Konqueror 中儲存您的網頁工作階段。下次您登入時，Konqueror 會載入上次所造訪的確實 URL。

要開啟新的標籤，可以選取 視窗 → 新增標籤 或按下 **Ctrl** + **Shift** + **N**。若要變更標籤的行為，請移至 設定 → 設定 Konqueror。在開啟的對話方塊中，選擇 網頁瀏覽方式 → 標籤式瀏覽。若要開啟新的標籤而不是視窗，請啟動 在新標籤中開啟連結而不是在新視窗中。您也可以用 只有一個標籤開啟時，隱藏標籤列來隱藏標籤列。若要查看更多的選項，請按 進階選項。

您可以在設定檔中儲存有 URL 的標籤和視窗的位置。這與上述所提的階段作業管理有點不同。就設定檔來說，您手上有儲存的標籤，且沒有像使用階段作業管理那樣的密集啟動時間。

在 Konqueror 中，請移至 **設定 → 設定檢視設定檔** 並命名您的設定檔。您也可以設定檔中以相關的選項來儲存視窗的大小。請確定已選取 **在設定檔中儲存 URL**。請按一下 **儲存** 予以核准。下次若您需要您的「標籤集合」時，請移至 **設定 → 載入檢視設定檔** 並查看列在功能表中的名稱。選取該名稱之後，Konqueror 便會還原您的標籤。

3.2 儲存網頁與圖形

和其他瀏覽器一樣，您可以儲存網頁。若要儲存網頁，請選取 **位置 → 另存新檔**，並為 HTML 檔案指定名稱。不過，並不會儲存影像。若要將整個網頁 (包括影像) 一起歸檔，請選取 **工具 → 歸檔網頁**。Konqueror 會建議可接受的檔案名稱。檔案名稱以 `.war` 結尾，這是網頁歸檔的副檔名。若要在稍後檢視儲存的網頁歸檔，僅需按一下個別的檔案，則網頁會和其影像顯示在 Konqueror 中。

3.3 網際網路關鍵字

使用 Konqueror 來搜尋網路非常簡單。Konqueror 為您定義超過 70 種搜尋過濾器，所有的過濾器都有特定的捷徑。若要在網際網路上搜尋特定主題，請輸入捷徑和關鍵字，中間以冒號分隔。然後，便會顯示包含搜尋結果的相關頁面。

若要查看已定義的捷徑，請移至 **設定 → 設定 Konqueror**。在顯示的對話方塊中，選擇 **網路捷徑**。現在您可以看到搜尋提供者的名稱和捷徑。Konqueror 定義許多的搜尋過濾器：「典型的」搜尋引擎，例如 Google、Yahoo 及 Lycos，以及一些特殊用途的過濾器，例如縮寫資料庫、網際網路電影資料庫 或 KDE 應用程式搜尋。

如果您在這裡找不到喜歡的搜尋引擎，可以很簡單地定義一個新的搜尋引擎。例如，若要搜尋某些有興趣文章的支援資料庫，通常會移至 <http://portal.suse.com/>，接著尋找搜尋頁面，然後輸入您的查詢。這個步驟可以使用捷徑來簡化。在上述對話方塊中，選擇 **新增**，並在 **搜尋提供者名稱** 中命名捷徑。請在 **URI 捷徑** 中輸入您的縮寫。若縮寫多於一個，請用逗號分開。重要的文字欄位是 **搜尋 URI**。按下 **[Shift] + [F]** 然後按一下欄位會開啟一個小說明。搜尋查詢會指定為 `\{@}`。在正確的位置插入此符號是個挑戰。在此情況下，SUSE 支援資料庫的設定看起來如下：**搜尋提供者名稱** 是 SUSE 支援資料庫，**搜尋 URI** 是 (同一行) <https://portal.suse.com/PM/page/search.pm?q=\{@}&t=optionSdbKeywords&m=25&l=en&x=true>，而 **URI 捷徑** 是 `sdb_en`。

按一下 **確定** 加以確認兩次之後，請在 **Konqueror** 的位置列中輸入您的查詢，例如 `sdb_en: kernel`。結果會顯示在目前的視窗中。

3.4 書籤

除了記住並重新輸入經常造訪網站網址之外，您也可以使用 **書籤** 功能表將這些 URL 加入書籤。除了網頁位址之外，您也可以使用此方式將任何目錄加入書籤。

若要在 **Konqueror** 中建立新的書籤，請按一下 **書籤** → **新增書籤**。任何之前加入過的書籤，會在功能表中顯示為項目。以階層式的結構，按照主題排列書籤集合是一個不錯的想法，這樣您就不會遺失不同項目的標示。可使用 **新增書籤** 資料夾 來建立新的書籤子群組。選擇 **書籤** → **編輯書籤**，開啟書籤編輯器。使用此程式可組織、重新排列、新增和刪除書籤。

如果您使用 **Netscape**、**Mozilla** 或 **Firefox** 做為其他的瀏覽器，則不需要再重新建立書籤。書籤編輯器中的 **檔案** → **輸入 Netscape 書籤**，可讓您將 **Netscape** 和 **Mozilla** 書籤整合為您目前的書籤。也可以透過 **匯出為 Netscape 書籤** 反向使用此功能。

在項目上按一下滑鼠右鍵可變更書籤。會出現快顯功能表，可在其中選取想要的動作 (剪下、複製、刪除等等)。完成這些動作之後，請選取 **檔案** → **儲存** 來儲存書籤。若您只想變更名稱或連結，只要在書籤工具列的項目上按一下滑鼠右鍵，並選擇 **內容**。請變更名稱和位置，並按一下 **更新** 加以確認。

若要儲存書籤清單並要立即存取，請在 **Konqueror** 中顯示您的書籤。選取 **設定** → **工具列** → **書籤工具列 (Konqueror)**。書籤面板會自動顯示在目前的 **Konqueror** 視窗中。

3.5 Java 與 JavaScript

請勿將這兩種語言混淆。**Java** 是 Sun Microsystems 的物件導向、平台獨立式的程式設計語言。它通常使用於在網際網路上執行的小型程式 (**Applet**)，如用在線上銀行、聊天和購物等。**JavaScript** 是解譯的指令碼語言，主要使用於網頁的動態結構，例如功能表和其他效果。

Konqueror 可以讓您啟用或關閉這兩種語言。甚至可以用網域指定的方式來完成，代表您可以允許某些主機的存取權而封鎖其他主機的存取權。**Java** 和

JavaScript 通常因為安全性因素而關閉。不幸地，某些網頁必須要有 JavaScript 才能正確顯示。

3.6 若需詳細資訊

使用 Konqueror 若發生任何問題，請參閱應用程式的手冊，該手冊可於 *說明* 功能表取得。Konqueror 也有網頁，網址如下：<http://www.konqueror.org>。

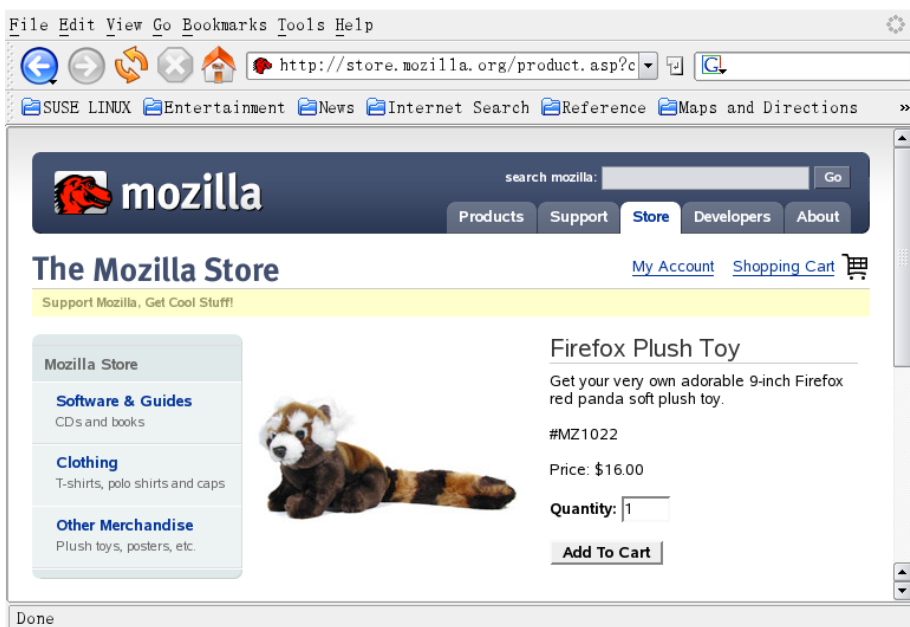
Firefox

SUSE Linux 附隨 Mozilla Firefox 網頁瀏覽器。透過標籤、封鎖快顯視窗、以及下載與影像管理等功能，Firefox 具備最新的網路技術。一個視窗中可以檢視多個網頁。可以抑制惱人的廣告以及關閉降低速度的影像。它易於存取不同的搜尋引擎，讓您輕鬆尋找需要的資訊。請從主要功能表或輸入 `firefox` 指令來啟動程式。以下小節將說明主要程式功能。

4.1 瀏覽網站

Firefox 的外觀和其他瀏覽器非常相似。如 [圖形 4.1, "Firefox 的瀏覽器視窗" \[78\]](#) 所示。瀏覽工具列含有 [上一頁](#) 和 [下一頁](#) 以及一個輸入網址的位置列。也有「書籤」可供您快速存取。如需各種 Firefox 功能的詳細資訊，請使用 [說明功能表](#)。

圖形 4.1 Firefox 的瀏覽器視窗



4.1.1 標籤式瀏覽

如果您經常一次使用一個以上的網頁，標籤式瀏覽可以讓您更易於在網頁間切換。會將網站載入視窗內的個別標籤中。

若要開啟新標籤，請選取 **檔案** → **新標籤**。Firefox 視窗中將開啟一個空白標籤。或者，也可以在連結按一下滑鼠右鍵，然後選取 **在新標籤開啟連結**。請在標籤上按一下滑鼠右鍵來存取其他標籤選項。您可以建立新標籤、重新載入某個標籤或所有現有的標籤、或關閉標籤。

4.1.2 使用提要欄位

請使用瀏覽器視窗的左側來檢視書籤或瀏覽歷程記錄。延伸程式可能也會加入新的使用方式來使用提要欄位。若要顯示提要欄位，請選取 **檢視** → **提要欄位**，然後選取想要的內容。

4.2 尋找資訊

Firefox 中尋找資訊的方法有兩種：搜尋列和尋找列。搜尋列會尋找頁面，而尋找列會在目前的頁面上尋找。

4.2.1 使用搜尋列

Firefox 含有一個搜尋列，可以存取各種引擎，例如，Google、Yahoo 或 Amazon。例如，如果您要使用目前的引擎尋找 SUSE 的相關資訊，請按一下搜尋列，接著鍵入 SUSE，然後按 **[Enter]**。結果將顯示在視窗中。若要選擇您的搜尋引擎，請按一下搜尋列中的圖示。會開啟一個功能表，裡面包含可用的搜尋引擎清單。

4.2.2 使用尋找列

若要在網頁內進行搜尋，請按一下 **編輯 → 搜尋這個頁面**，或按 **[Ctrl] + [F]** 來開啟尋找列。通常，它會顯示在視窗的下方。請在輸入欄位中鍵入您的查詢。Firefox 會反白顯示所有找到字句的例項。使用 **反白** 來啟用或關閉反白顯示功能。

4.3 管理書籤

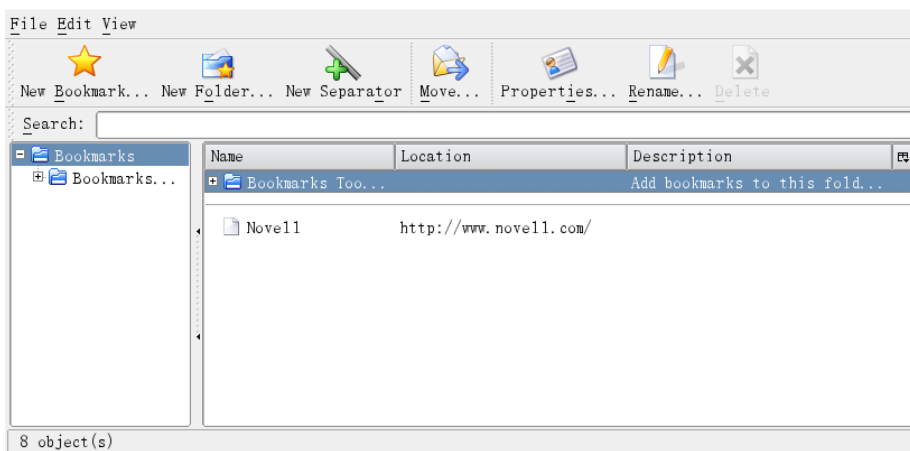
書籤是提供儲存我的最愛網站連結的便捷方法。若要新增目前的網站到書籤清單，請按一下 **書籤 → 將此頁設為書籤**。若您的瀏覽器目前在索引標籤上顯示多個網站，只有在目前選取索引標籤的 URL 會加入到您的書籤清單中。

當您新增書籤時，您可以對書籤指定其他名稱以及選取 Firefox 儲存書籤的資料夾。若要從書籤清單移除網站，請按一下 **書籤**，在清單中的書籤上按一下滑鼠右鍵，然後按一下 **刪除**。

4.3.1 使用書籤管理員

書籤管理員可以用來管理每一個書籤的內容 (名稱與位置) 以及依資料夾與段落組織書籤。它和 [圖形 4.2, "使用 Firefox 書籤管理員" \[80\]](#) 類似。

圖形 4.2 使用 Firefox 書籤管理員



若要開啟書籤管理員，請按一下 **書籤 → 管理書籤**。會開啟一個視窗並顯示您的書籤。使用 **新資料夾** 可以依名稱以及描述來建立新資料夾。如果您需要新的書籤，請按一下 **新增書籤**。您可以插入名稱、位置、關鍵字以及描述。關鍵字是書籤的捷徑。如果您想讓新建的書籤出現在提要欄位，請核取 **在提要欄位載入這個書籤**。

4.3.2 移轉書籤

如果您以前使用不同的瀏覽器，您可能也想在 Firefox 使用原有的我的最愛和書籤。目前，您可以從 Netscape 4.x、6、7、Mozilla 1.x 以及 Opera 輸入。

若要輸入您的設定值，請按一下 **檔案 → 輸入**。請選取您要輸入設定值的瀏覽器。按一下 **下一步** 後，會輸入您的設定值。請在新建立的資料夾中尋找輸入的書籤，並以 **從** 開始。

4.3.3 即時書籤

「即時書籤」可在書籤功能表中顯示標題，並讓您隨時得知最新消息。這樣可讓您掃視喜歡的站台，節省您的時間。

許多站台和部落格都支援此格式。支援此格式的網站會在右下角顯示一個內有 RSS 的橘色矩形。按一下它，然後選擇訂閱 *NAME OF THE FEED*。這樣會開啟對話方塊，讓您選取「即時書籤」的名稱和位置。請按 新增 加以確認。

有些站台雖然可支援新聞來源，但是不會告知 Firefox。若要手動新增「即時書籤」，您就需要新聞來源的 URL。請執行下列動作：

過程 4.1 手動新增即時書籤

- 1 以 書籤 → 管理書籤 開啟書籤管理員。會開啟新視窗。
- 2 選取 檔案 → 新的即時書籤。會開啟對話方塊。
- 3 插入「即時書籤」的名稱並新增 URL，例如 <http://www.novell.com/newsfeeds/rss/cool solutions.xml>。Firefox 會更新「即時書籤」。
- 4 關閉書籤管理員。

4.4 使用下載管理員

下載管理員可以協助追蹤您目前與過去的下載。若要開啟下載管理員，請按一下 工具 → 下載。Firefox 將開啟含有您下載的視窗。下載檔案時，會看到進度列和目前的檔案。可以暫停下載，必要時可以在稍後繼續。若要開啟下載的檔案，請按一下 開啟。使用 移除 可以從媒體移除該檔案。如果您需要檔案的相關資訊，請在檔案名稱按一下滑鼠右鍵，接著選擇 內容。

如果您想進一步設定「下載管理員」，請從 編輯 → 優先設定 開啟組態視窗，然後移至 下載標籤。請在此處決定下載資料夾、管理員的行為方式以及一些檔案類型的組態。

4.5 自訂 Firefox

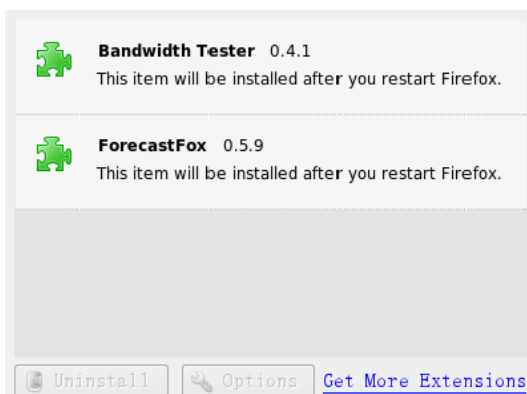
由於 Firefox 具有安裝延伸功能、變更佈景主題、以及新增線上搜尋的智慧型關鍵字的能力，因此可以廣泛的自訂。

4.5.1 延伸程式

Mozilla Firefox 是一個多功能應用程式，您可以下載並安裝附加程式 (稱為延伸程式)。例如，新增新的下載管理員與滑鼠動作。這可以讓 Firefox 保有精巧而不龐大的優點。

若要新增延伸程式，請按一下 **工具** → **延伸程式**。請在右下角按一下 **取得更多延伸程式** 來開啟 Mozilla 延伸程式更新網頁，您可以在其中選擇各種可用的延伸程式。按一下延伸程式來進行安裝，然後按一下安裝連結來下載並安裝。當您重新啟動 Firefox 時，新的延伸程式便會執行。您還可以到 <http://update.mozilla.org/> 查看各種延伸程式。

圖形 4.3 安裝 Firefox 延伸程式

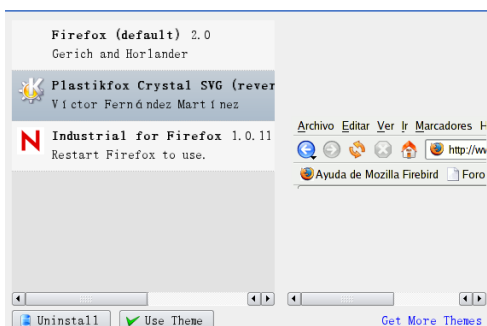


4.5.2 變更佈景主題

如果您不喜歡 Firefox 的標準外觀，請安裝新的佈景主題。佈景主題只會改變瀏覽器的外觀，不會改變功能。安裝佈景主題時，Firefox 會先要求確認。請選擇允許安裝或取消安裝。安裝成功後，您可以啟動新的佈景主題。

- 1 按一下 **工具** → **佈景主題**。
- 2 會開啟一個新對話方塊。按一下 **取得更多佈景主題**。如果您已經安裝佈景主題，請在清單中尋找，如 [圖形 4.4, "安裝 Firefox 佈景主題" \[83\]](#) 所示。

圖形 4.4 安裝 Firefox 佈景主題



- 3 會出現含有 <https://update.mozilla.org> 網站的新視窗。
- 4 請選擇一個佈景主題，然後按一下 **立即安裝**。
- 5 確認下載並安裝。
- 6 下載佈景主題後，會出現一個對話方塊並通知您佈景主題的清單。請使用 **使用佈景主題** 來啟動新佈景主題。
- 7 關閉視窗並重新啟動 Firefox。

如果已經安裝佈景主題，您只要按一下 **工具** → **佈景主題** 然後 **使用佈景主題**，便可以切換到不同佈景主題，不必重新啟動。如果不想再使用某個佈景主題，可以在同一個對話方塊中使用 **解除安裝** 來刪除。

4.5.3 新增智慧型關鍵字至線上搜尋

在網際網路進行搜尋是瀏覽器可以為您執行的一個主要工作之一。Firefox 可以讓您定義您想要的智慧型關鍵字：縮寫可用來搜尋網頁時做為 "指令"。例如，如果您經常使用 Wikipedia，請使用智慧型關鍵字來簡化這個工作：

- 1 請移至 <http://zh.wikipedia.org>。
- 2 Firefox 顯示網頁後，請查看搜尋文字欄位。在該欄位按一下滑鼠右鍵，然後從開啟的功能表中選擇 **新增此搜尋的關鍵字**。

- 3 會出現 **新增書籤** 對話方塊。在 **名稱** 欄位，指定這個網頁的名稱，例如，*Wikipedia (en)*。
- 4 在 **關鍵字** 欄位，輸入這個網頁的縮寫，例如，*wiki*。
- 5 使用 **建立於**，在書籤段落中選擇項目的位置。您可以將它放至 **快速搜尋** 中，但是其他層級也很適當。
- 6 使用 **新增** 來完成。

您已經成功產生新的關鍵字。每次想查看 **Wikipedia** 時，您不必使用整個 URL。只要鍵入 **wiki Linux** 便可以檢視 **Linux** 的相關項目。

4.6 從 Firefox 中列印

您可以使用 **頁面設定** 對話方塊來設定 **Firefox** 列印顯示內容的方式。請按一下 **檔案** → **頁面設定**，然後移至 **格式 & 選項** 標籤來選取列印工作的方向。您可以縮放或是讓其自動調整。若要列印背景，請選取 **列印背景 (色彩和影像)**。按一下 **邊界和頁首/頁尾** 標籤來調整邊界，並選取頁首和頁尾的內容。

完成設定後，請使用 **檔案** → **列印** 來列印網頁。請選取印表機或要儲存輸出的檔案。請使用 **內容** 來設定紙張大小、指定列印指令、選擇灰階或彩色以及決定邊界。如果您滿意您的設定，請使用 **列印** 確認。

4.7 若需詳細資訊

請至位於 <http://www.mozilla.org/products/firefox/> 的官方首頁來取得 **Firefox** 的詳細資訊。請參閱整合的說明來瞭解特定選項或功能的詳細資訊。

Linphone — Linux 桌面的 VoIP

Linphone 是 Linux 桌面的小型網路電話應用程式。它可讓您透過網際網路進行雙方通話。它不需要特殊的硬體項目：只要您有標準工作站，且已正確設定音效卡、麥克風和喇叭或耳機，就可以開始使用 Linphone。

5.1 設定 Linphone

開始使用 Linphone 之前，必須進行一些基本決策並完成一些組態工作。首先，請判斷並設定 Linphone 的執行模式，再判斷要使用的連線類型，然後啟動 Linphone 組態 (到 → 偏好設定) 以進行必要的調整。

5.1.1 判斷 Linphone 的執行模式

根據您執行的桌面類型和 Linphone 的組態而定，Linphone 可以在兩種不同的模式下執行。

一般應用方式

安裝 Linphone 軟體之後，就可以透過 GNOME 和 KDE 應用程式功能表或透過指令行來啟動它。如果 Linphone 未在執行中，就無法接收來電。

GNOME 面板 Applet

您可以將 Linphone 新增至 GNOME 面板。以滑鼠右鍵按一下面板中的空白區域，選取新增至面板，然後選取 Linphone。然後 Linphone 就會永久加入面板，並在登入時自動啟動。只要您未接聽任何來電，它就會在背景執行。

一有來電，主視窗就會開啟，您就可以接聽來電。若要開啟主視窗與某人通話，只需按一下 Applet 圖示即可。

5.1.2 判斷連線類型

Linphone 有數種方法可以進行通話。進行通話的方式以及與他人連絡的方式，都取決於您連接到網路或網際網路的方法。

Linphone 使用工作階段起始協定 (Session Initiation Protocol, SIP) 來建立與遠端主機的連線。在 SIP 中，是以 SIP URL 來辨識通話各方：

```
sip: username@hostname
```

username 是您在 Linux 機器的登入帳戶，而 *hostname* 則是您使用的電腦名稱。如果您使用 SIP 提供者，URL 看起來就像下列範例：

```
sip: username@sipserver
```

username 是您在 SIP 伺服器註冊時選擇的使用者名稱。*sipserver* 是 SIP 伺服器或 SIP 提供者的位址。如需有關註冊程序的詳細資料，請參閱 [節 5.1.5, "設定 SIP 選項" \[88\]](#)，然後查看提供者的註冊文件。如需適用於各項目的提供者清單，請查看 [節 5.8, "若需詳細資訊" \[94\]](#) 所述的網頁。

要使用的 URL 是由您選擇的連線類型來判定。如果您選擇與他人直接通話而不使用 SIP 提供者提供的任何進一步路由，則應輸入第一種類型的 URL。如果您選擇透過 SIP 伺服器與他人通話，則應輸入第二種類型的 URL。

在相同網路中通話

如果您想與屬於相同網路的朋友或同事通話，只需要正確的使用者名稱和主機名稱，就可以建立有效的 SIP URL。如果有人要與您通話也是如此。只要您與受話者之間沒有防火牆，就不需要進一步組態。

跨網路或網際網路進行通話 (靜態 IP 設定)

如果您使用靜態 IP 位址連接到網際網路，那麼想與您通話的任何人員只需要您的使用者名稱和您工作站的主機名稱或 IP 位址，就可以建立有效的 SIP URL，如 ["在相同網路中通話" \[86\]](#) 所述。如果您或發話者都位於可篩選內送和外寄傳

輸的防火牆後面，請開啟防火牆機器上的 SIP 連接埠 (5060) 和 RTP 連接埠 (7078)，才能跨防火牆進行 Linphone 傳輸。

跨網路或網際網路進行通話 (動態 IP 設定)

如果您的 IP 設定不是靜態的 — 也就是說，您每次連接到網際網路都會動態取得新的 IP 位址 — 那麼任何發話者都不可能根據您的使用者名稱和 IP 位址，來建立有效的 SIP URL。在此類情況下，可以使用 SIP 提供者提供的服務或是使用 DynDNS 設定，來確認外部發話者能夠連接到正確的主機。如需有關 DynDNS 的詳細資訊，請至 http://zh.wikipedia.org/wiki/Dynamic_DNS。

跨網路和防火牆進行通話

隱藏在防火牆後面的機器，不會向網際網路顯示其 IP 位址。因此，無論是誰都無法與使用這類機器工作的使用者直接通話。Linphone 使用 SIP 代辦或是將通話轉送到 SIP 提供者，藉以支援跨網路邊界和防火牆進行通話。如需使用外部 SIP 伺服器時的必要調整的詳細說明，請參閱 節 5.1.5, "設定 SIP 選項" [88]。

5.1.3 設定網路參數

網路標籤內含的設定大多不需要任何進一步調整。不變更這些設定，也應該能夠進行第一次通話。

NAT 遍遊選項

只有在您的私用網路位於防火牆後面，且您不想使用 SIP 提供者來路由通話時，才啟用此選項。選取核取方塊，並以點標記法 (Dot Notation) 輸入防火牆機器的 IP 位址，例如 192.168.34.166。

RTP 內容

Linphone 使用即時傳輸協定 (RTP) 來傳輸通話的音訊資料。RTP 的連接埠設為 7078，除非有另一個應用程式使用此連接埠，否則不得修改。Jitter Compensation 參數是用來控制 Linphone 實際播放音訊封包之前，緩衝處理的音訊封包數目。增加此參數就可以改善傳輸品質。緩衝處理的封包越多，播放較晚進行緩衝處理的封包的機會就越大。另一方面，增加緩衝處理的封包數目也會增加延遲 (Latency) -- 您聽到對方聲音時，會有某種程度的延遲。當您變更此參數時，請在這兩種因素之間小心取得平衡。

其他

如果您使用 VoIP 與傳統線路通話方式的組合，則可使用雙音多工頻率 (DTMF) 技術來觸發某些動作，例如按幾個特定鍵即可從遠端檢查語音信箱。Linphone 支援 DTMF 傳輸的兩種協定、SIP INFO 以及 RTP rfc2833。如果您需要 Linphone 的 DTMF 功能，請選擇可支援上述任一協定的 SIP 提供者。如需 VoIP 提供者的完整清單，請參閱 [節 5.8, "若需詳細資訊" \[94\]](#)。

5.1.4 設定音效設備

在 Linux 正確偵測到音效卡之後，Linphone 就會自動使用偵測到的設備作為預設音效設備。請保留 *使用音效設備* 的值不變。使用 *錄製來源* 來判定應該使用哪個錄製來源。這通常是麥克風 (micro)。若要選取自訂的鈴聲，請使用 *瀏覽* 來選擇一種鈴聲，再使用 *聆聽* 來測試您的選擇。按一下 *套用* 以接受變更。

5.1.5 設定 SIP 選項

SIP 對話方塊包含所有 SIP 組態設定。

SIP 連接埠

判定應該在哪個連接埠上執行 SIP 使用者代辦。SIP 的預設連接埠為 5060。除非您知道有任何其他應用程式或協定需要這個連接埠，否則請保留預設設定不變。

識別

想要直接與您通話但不使用 SIP Proxy 或 SIP 提供者的任何人員，都必須知道您的有效 SIP 位址。Linphone 會為您建立有效的 SIP 位址。

遠端服務

此清單保留的是您已經建立使用者帳戶的一或多個 SIP 服務提供者。您可以隨時新增、修改或刪除伺服器資訊。若要瞭解註冊程序，請參閱 [新增 SIP Proxy 和註冊遠端 SIP 伺服器 \[89\]](#)。

驗證資訊

若要註冊遠端 SIP 伺服器，請提供特定驗證資訊，例如密碼和使用者名稱。提供這些資料之後，Linphone 就會加以儲存。如果為了安全起見而要捨棄此資料，請按一下 *清除已儲存的所有驗證資料*。

遠端服務清單含有遠端 SIP Proxy 或服務提供者的數個位址。

過程 5.1 新增 SIP Proxy 和註冊遠端 SIP 伺服器

- 1 選擇合適的 SIP 提供者，並在該處建立使用者帳戶。
- 2 啟動 Linphone。
- 3 請至 *到* → *偏好設定* → *SIP*。
- 4 按一下 *新增 Proxy/註冊器* 以開啟註冊表單。
- 5 在 *註冊期間*、*SIP 識別*、*SIP Proxy* 以及 *路由* 填入適當的值。如果位於防火牆後面，請一律選取 *傳送註冊*，然後在 *註冊期間* 填入適當的值。這樣會在經過特定時間之後重新傳送原始註冊資料，使 Linphone 所需的連接埠的防火牆保持開啟。否則，如果防火牆未接收相同類型的其他封包，這些連接埠就會自動關閉。如果要讓 SIP 伺服器持續得知連線的目前狀態和發話者的位置，則也需要重新傳送註冊資料。在 *SIP 識別* 輸入應該用於本地通話的 SIP URL。如果這部伺服器也要用作 SIP Proxy，請在 *SIP Proxy* 輸入相同的資料。最後，視需要新增選擇性的路由，然後利用 *確定* 離開對話方塊。

5.1.6 設定音訊轉碼器

Linphone 支援數種用來傳輸語音資料的轉碼器。請在清單視窗中，設定連線類型並選擇偏好的轉碼器。不適合目前連線類型的轉碼器會以紅色標示且無法選取。

5.2 測試 Linphone

使用 `sipomatic` (一種可以接聽 Linphone 來電的小型測試程式) 來檢查 Linphone 組態。

過程 5.2 測試 Linphone 設定

- 1 開啟終端機。
- 2 在指令行提示輸入 `sipomatic`。

3 啟動 Linphone。

4 輸入 sip:robot@127.0.0.1:5064 作為 *SIP 位址*，然後按一下 *撥打或接聽*。

5 如果正確設定 Linphone，您就會聽到電話鈴聲，再過一會兒您就會聽到短訊。

如果您順利完成此程序，就可以確定音訊設定和網路設定能夠運作。如果此測試失敗，請檢查音效裝置是否正確設定，以及播放音量是否設為合理的值。如果還是聽不到聲音，請檢查網路設定，包括 SIP 和 RTP 的連接埠號碼。如果有任何其他應用程式或協定使用這些預設連接埠 (由 Linphone 建議)，請考慮變更連接埠然後重試。

5.3 進行通話

適當設定 Linphone 之後，就可以直接進行通話。通話程序會根據通話類型而有些許不同 (如需參考資料，請參閱 [節 5.1.2, "判斷連線類型" \[86\]](#))。

1 使用功能表或指令行啟動 Linphone。

2 在 *SIP 位址* 提示輸入對方的 SIP 位址。直接本地通話的位址看起來應該像 sip:username@domainname 或 username@hostname，而經過 Proxy 進行的通話或使用 SIP 提供者服務所進行的通話位址，看起來應該像 username@sipserver 或 userid@sipserver。

3 如果您使用 SIP 服務提供者或 Proxy，請在 *要使用的 Proxy* 中選取適當的 Proxy 或提供者，並提供此 Proxy 要求的驗證資料。

4 按一下 *撥打或接聽*，並等候對方拿起聽筒。

5 完成之後或是想要結束通話時，請按一下 *掛斷或拒絕* 並離開 Linphone。

如果您在通話期間需要調整音效參數，請按一下 *顯示更多* 以顯示四個含有更多選項的標籤。第一個標籤含有 *播放音量* 和 *錄製音量* 的音效選項。使用滑桿調整音量以符合您的需求。

我的狀態 標籤可讓您設定線上狀態。此資訊可以轉送給想要連絡您的任何人員。如果您永久離開，且想要將此狀況通知發話者，只需核取 *離開*。如果您只是忙

碌，但是希望發話者重試，請核取 *忙碌*，... 分鐘後回來，並指定您會有多久無法回應。只要您可以再次回應，就可以將狀態設回預設值 (*線上*)。他人是否能夠查看您的線上狀態，是取決於通訊錄中設定的 *訂閱原則*，如 [節 5.5, "使用通訊錄"](#) [91] 所述。如果通訊錄中列出的任何人員發佈了他們的線上狀態，您就可以使用 *我的線上朋友* 標籤來監視該狀態。

DTMF 標籤可用來輸入 *DTMF* 碼以檢查語音信箱。若要檢查語音信箱，請輸入適當的 *SIP* 位址，並使用 *DTMF* 標籤中的鍵台輸入語音信箱碼。最後按一下 *撥打或接聽*，就像進行一般通話一樣。

5.4 接聽通話

根據您在 Linphone 選取的執行模式而定，有數種方法可讓您注意到來電：

一般應用方式

只有在已經執行 Linphone 時，才能收到或接聽來電。然後您就會聽到耳機或喇叭發出鈴聲。如果 Linphone 未在執行中，就無法接收來電。

GNOME 面板 Applet

Linphone 面板 Applet 通常會自動執行，而不會通知您它已在執行。只要一有來電，這種狀態就會變更：Linphone 的主視窗會開啟，而您會聽到耳機或喇叭發出鈴聲。

一旦知道有來電，只需要按一下 *撥打或接聽* 即可拿起話筒開始通話。如果您不想接聽這次通話，請按一下 *掛斷或拒絕*。

5.5 使用通訊錄

Linphone 讓您能夠管理 *SIP* 連絡人。請使用 *到 → 通訊錄* 來啟動通訊錄。這樣會開啟空白的清單視窗。按一下 *新增* 以新增連絡人。

有效的連絡人必須具備下列項目：

名稱

輸入連絡人的姓名。這可以是全名，也可以使用綽號。請選擇便於記憶的名稱。如果您選擇查看此人的線上狀態，這個名稱就會顯示在主視窗的 *我的線上朋友* 標籤中。

SIP 位址

輸入連絡人的有效 SIP 位址。

要使用的 Proxy

視需要輸入用於這次特定連線的 Proxy。在大部分情況下，這就是您使用的 SIP 伺服器的 SIP 位址。

訂閱原則

訂閱原則可決定他人是否能夠追蹤您的連線狀態。

若要從通訊錄中與任何連絡人通話，請以滑鼠選取該連絡人，按一下 **選取讓位址** 出現在主視窗的位址欄位中，然後使用 **撥打** 或 **接聽** 開始通話。

5.6 疑難排解

我試著與某人通話，但是無法建立連線。

有數種原因會造成通話失敗：

您與網際網路的連線已中斷。

因為 Liphone 是使用網際網路來轉送通話，所以請確定電腦已正確連接到網際網路，且已針對網際網路進行正確設定。試著使用瀏覽器檢視網頁，就可以輕鬆測試是否符合上述條件。如果網際網路連線可以運作，則可能是對方無法回應。

通話對方無回應。

如果通話對方拒絕通話，您就無法與之連線。如果通話對方的電腦在您撥打時並未執行 Liphone，您也無法與之連線。如果通話對方與網際網路的連線已中斷，您就無法進行連線。

通話似乎已連線，但是我什麼也聽不到。

首先，請確定您的音效裝置已正確設定。啟動其他使用音效輸出的任何應用程式(例如媒體播放器)，即可進行此動作。確定 Liphone 有足夠的許可權能夠開啟此裝置。關閉其他使用音效裝置的所有程式，以避免資源衝突。

如果上述檢查順利進行，但您仍聽不到任何聲音，請在 **音效** 標籤之下，調高錄製音量和播放音量。

通話雙方的語音輸出聽起來斷斷續續的。

試著使用 *偏好設定* → *網路* 中的 *RTP* 內容來補償延遲的語音封包，以調整 *Jitter* 緩衝區。請注意，進行此動作會增加延遲 (*Latency*)。

DTMF 無法運作。

您嘗試使用 DTMF 鍵台檢查語音信箱，但是無法建立連線。雖然有三種不同的協定可用來傳輸 DTMF 資料，但是 Linphone 只支援其中兩種 (*SIP INFO* 和 *RTP/rfc2833*)。請洽詢提供者，以得知您的提供者是否支援其中一種協定。Linphone 使用的預設協定是 *rfc2833*，萬一它失敗，您就可以在 *偏好設定* → *網路* → *其他* 中，將協定設為 *SIP INFO*。如果使用上述任一種協定都無法運作，就無法使用 Linphone 完成 DTMF 傳輸。

5.7 詞彙

本文件述及的重要技術術語和協定，都可在此找到一些簡短說明：

VoIP

VoIP 表示 *網際網路電話 (Voice over Internet Protocol)*。此技術允許使用封包連結的路由，透過網際網路傳輸一般電話通話。語音資訊是以不同的封包傳送，就像是透過 IP 在網際網路上傳輸任何其他資料一樣。

SIP

SIP 代表 *工作階段起始設定 (Session Initiation Protocol)*。此協定是用來在網路上建立媒體工作階段。就 Linphone 的角度而言，SIP 是一種魔法，可觸發通話對方的電腦鈴聲，開始通話，也可以在任一方決定掛斷電話時結束通話。語音資料的實際傳輸動作是由 RTP 處理。

RTP

RTP 代表 *即時傳輸協定 (Real-Time Transport Protocol)*。它允許透過網路傳輸媒體資料流，且可透過 UDP 運作。資料是由已編號的不同封包 (帶有時間戳記) 來傳輸，這樣就可以正確排序並偵測遺失的封包。

DTMF

DTMF 編碼器就像一般電話一樣，使用一組一組的音調來代表不同的識別碼。每個識別碼都與一個高音調和一個低音調的唯一組合相關聯。而解碼器會將這些按鍵式音調組合轉譯回數字。Linphone 支援可觸發遠端動作 (例如檢查語音信箱) 的 DTMF 訊號。

轉碼器

轉碼器是特別設計來壓縮音訊和視訊資料的演算法。

Jitter

Jitter 是連線中的延遲變異。音訊裝置或連線導向系統 (例如 ISDN 或 PSTN) 都需要連續的資料流。為了加以補償，VoIP 終端機和閘道都實作 Jitter 緩衝區，而後者會先收集封包，再將封包轉送到音訊裝置或連線導向線路 (例如 ISDN)。增加 Jitter 緩衝區大小可減少遺失資料的可能性，但是會增加連線延遲。

5.8 若需詳細資訊

如需有關 VoIP 的一般資訊，請參閱 VoIP Wiki: <http://voip-info.org/tiki-index.php>。如需您所在位置的國家/地區中，有提供 VoIP 服務的完整提供者清單，請參閱 <http://voip-info.org/wiki-VoIP+Service+Providers+Residential>。

使用 KGpg 加密

KGpg是您系統上重要的加密基礎結構元件。有了此程式的協助，可產生與管理所有必要的金鑰、使用其編輯器功能以快速建立和加密檔案，或使用面板中的applet，藉由拖放來加密或解密檔案。其他程式如郵件程式(Kontact 或 Evolution)等都需要金鑰資料來處理已簽名或加密的資料。本章節涵蓋加密檔案日常工作所需的基本功能。

6.1 產生新的金鑰組合

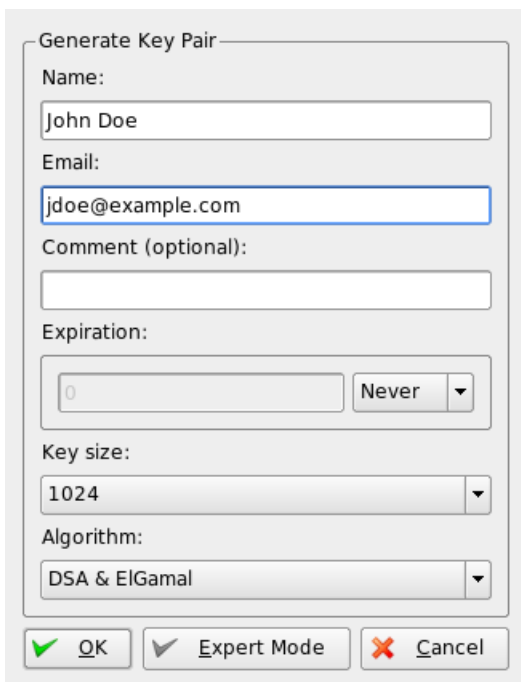
若要和其他使用者交換加密的訊息，首先要產生您自己的金鑰組合。其中一部分(公共金鑰)是配送給您的通訊夥伴，讓它可以用來加密所傳送的檔案或電子郵件訊息。金鑰組合的另一個部分(私密金鑰)是用來解密加密的內容。

重要：私密金鑰與公共金鑰

公共金鑰適用於公共場合，應該配送給您所有的通訊夥伴。但是，只有您才可以存取私密金鑰。請勿授權其他使用者存取此資料。

若要啟動 KGpg，您可以從主功能表選取 *公程式* → *KGpg* 或是從指令行輸入 kgpg。當您第一次啟動程式時，將出現助理帶領您完成組態。請依照指示在提示您的地方建立金鑰。請輸入名稱、電子郵件地址，以及選擇性的備註。如果您不喜歡提供的預設設定，請同時設定金鑰的到期時間、金鑰大小以及所使用的加[密演算法。請參閱 [圖形 6.1, "KGpg: 建立金鑰" \[96\]](#)。

圖形 6.1 KGpg: 建立金鑰

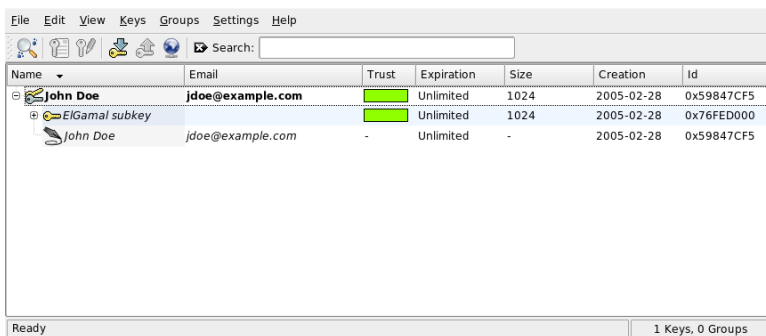


The dialog box titled "Generate Key Pair" contains the following fields and controls:

- Name:** Text input field containing "John Doe".
- Email:** Text input field containing "jdoe@example.com".
- Comment (optional):** Empty text input field.
- Expiration:** A numeric input field with "0" and a dropdown menu set to "Never".
- Key size:** A dropdown menu set to "1024".
- Algorithm:** A dropdown menu set to "DSA & ElGamal".
- Buttons:** "OK" (with a green checkmark icon), "Expert Mode" (with a checkmark icon), and "Cancel" (with a red X icon).

以 確定 確認您的設定值。下一個對話方塊將提示您輸入兩次密碼。程式接著產生一組金鑰並顯示摘要。最好立即儲存或列印撤銷證書。如果您忘記私密金鑰的密碼而需要撤銷時，便需要這個證書。您以 確定 進行確認後，KGpg 顯示自己的主視窗。請參閱 圖形 6.2, "金鑰管理員" [96]。

圖形 6.2 金鑰管理員



The Key Manager window shows a table of keys with the following data:

Name	Email	Trust	Expiration	Size	Creation	Id
John Doe	jdoe@example.com		Unlimited	1024	2005-02-28	0x59847CF5
ElGamal subkey			Unlimited	1024	2005-02-28	0x76FED000
John Doe	jdoe@example.com	-	Unlimited	-	2005-02-28	0x59847CF5

At the bottom, the status bar shows "Ready" and "1 Keys, 0 Groups".

6.2 匯出公共金鑰

在產生金鑰組合之後，要讓其他使用者使用公共金鑰。這樣可讓他們使用金鑰來加密或簽署傳送給您的訊息或檔案。若要讓其他人也可以使用公鑰，請選取 **金鑰** → **匯出公鑰**。開啟的對話方塊提供四個選項：

電子郵件

您的公共金鑰會傳送電子郵件給您所選擇的收件者。如果您啟動這個選項並以 **確定** 進行確認，將出現使用 **KMail** 建立新電子郵件訊息的對話方塊。請輸入收件者並按一下 **傳送**。收件者將收到您的金鑰並用來傳送加密內容給您。

剪貼簿

您可以在繼續處理之前先將公共金鑰放置在此處。

預設金鑰伺服器

若要讓大量的使用者使用您的公共金鑰，請將它匯出到網際網路上其中一個金鑰伺服器。如需詳細資訊，請參閱 [節 6.4, "金鑰伺服器對話方塊" \[98\]](#)。

檔案

如果您想要以檔案而非以電子郵件在資料媒體中配送您的金鑰，請按一下此選項來確認或變更檔案路徑與名稱，再按一下 **確定**。

6.3 匯入金鑰

如果您在檔案中收到金鑰 (例如，電子郵件附件)，可以使用 **匯入金鑰** 將它整合到您的金鑰環中，用來和傳送者進行加密通訊。這個程序和上述的輸出金鑰相似。

6.3.1 簽名金鑰

金鑰可以像任何其他檔案一樣簽名，以保證驗證性與完整性。如果您十分確定輸入的金鑰屬於擁有者指定的個人，您可以用簽名來表達您對金鑰驗證性的信任。

重要: 建立信任網路

加密的通訊僅可保障在特定使用者之間流通的公共金鑰之延伸。藉由交叉確認和簽署這些金鑰，便可促成信任網路的建立。

在金鑰清單中選取要簽署的金鑰。選取金鑰 → 簽署金鑰。在下列對話方塊中，指定要用於簽名的私密金鑰。簽名之前會出現警示，提醒您檢查這個金鑰的驗證性。如果您已經執行此檢查，請按一下 **繼續** 並在下一步輸入選取的私密金鑰密碼。其他使用者現在可以藉由您的公共金鑰來檢查簽名。

6.3.2 信任金鑰

通常，對應的程式將詢問您是否信任該金鑰 (您是否認為它確實被它的授權擁有者使用)。這在每次訊息需要解密或必須檢查簽名時都會發生。要避免這種情況，可以編輯新匯入金鑰的信任層級。

在新匯入的金鑰按一下滑鼠右鍵，以存取金鑰管理的小型內容功能表。從中選取 **在終端機編輯金鑰**。KGpg 將開啟文字主控台，以使用一些指令來設定信任層級。

在文字主控台的提示中 (Command>) 輸入 `trust`。可以利用 1 (不信任) 到 5 (完全信任) 的層級來表示您對輸入之金鑰的簽名者已經檢查金鑰擁有者之真實身份的信任程度。在提示 (Your decision?) 中輸入選取的值。如果您確實確認簽名者是可靠的，請輸入 5。輸入 `y` 來回答下列問題。最後，輸入 `quit` 以結束主控台並返回金鑰清單。現在金鑰具有信任層級 `Ultimate`。

金鑰環中的金鑰信任層級以金鑰名稱旁的彩色列指示。信任程度越低，表示您信任金鑰簽名者已經檢查簽名金鑰身份的程度也越低。您可能完全信任簽名者的身份，但簽名者可能疏於在簽名其他人的金鑰之前檢查他們的身份。因此，您可能仍然信任他們的金鑰，但是對於他們簽名的其他人的金鑰，則指定較低的信任層級。信任層級的目的是提供獨立的提醒。它不會由 KGpg 觸發任何動作。

6.4 金鑰伺服器對話方塊

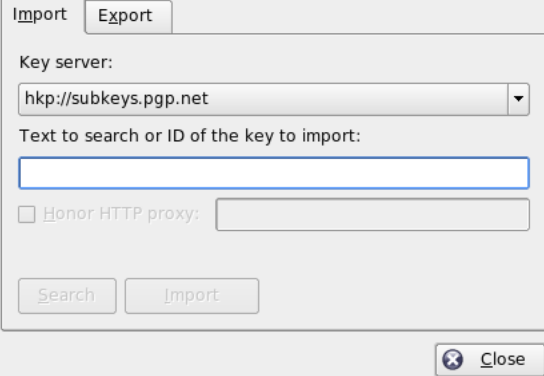
數個以網際網路為基礎的金鑰伺服器，提供許多使用者的公共金鑰。若想要與大量使用者進行加密通訊，請使用這些伺服器來配送您的公共金鑰。基於此原

因，請將您的公共金鑰匯出至其中一個伺服器。同樣地，KGpg 可供您在這些伺服器其中之一搜尋一些人的金鑰，並從伺服器匯入他們的公共金鑰。請從 [檔案](#) → [金鑰伺服器對話方塊](#) 來開啟金鑰伺服器對話方塊。

6.4.1 從金鑰伺服器匯入金鑰

利用金鑰伺服器對話方塊中的 [匯入](#) 標籤，可從其中一個網際網路為基礎的金鑰伺服器匯入公共金鑰。使用下拉式功能表選取一個預先設定的金鑰伺服器，並輸入搜尋字串 (通訊夥伴的電子郵件地址) 或要尋找的金鑰 ID。當您按一下 [搜尋](#) 時，系統會連線到網際網路，搜尋指定的金鑰伺服器以尋找符合您規格的金鑰。請參照 [圖形 6.3](#), "[匯入金鑰的搜尋畫面](#)" [99]。

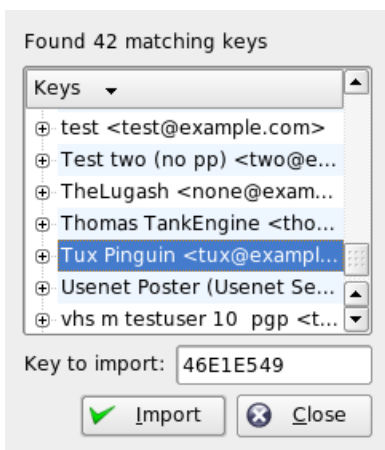
圖形 6.3 匯入金鑰的搜尋畫面



The image shows a dialog box titled "Import" with a tab labeled "Import". Inside the dialog, there is a section labeled "Key server:" with a dropdown menu showing "hkp://subkeys.pgp.net". Below this is a text input field labeled "Text to search or ID of the key to import:". There is also a checkbox labeled "Honor HTTP proxy:" which is currently unchecked. At the bottom of the dialog are two buttons: "Search" and "Import". A "Close" button with a window icon is located at the bottom right of the dialog box.

如果您在金鑰伺服器上的搜尋成功，取回的所有伺服器項目清單會顯示在新視窗中。選取想要包括在您金鑰環中的金鑰並按一下 [匯入](#)。請參閱 [圖形 6.4](#), "[點選與匯入](#)" [100]。按 [確定](#) 確認下列訊息，然後按 [關閉](#) 結束金鑰伺服器對話方塊。然後匯入的金鑰會出現在金鑰管理員的主要概觀中，並且可供使用。

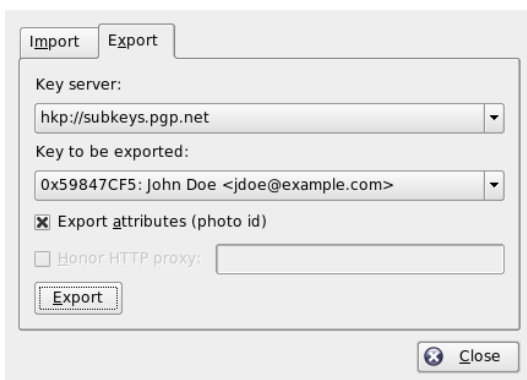
圖形 6.4 點選與匯入



6.4.2 將您的金鑰匯出至金鑰伺服器

若要將您的金鑰匯出到網際網路上可自由存取的金鑰伺服器，請選取金鑰伺服器對話方塊中的 **匯出** 索引標籤。利用兩個下拉式功能表指定目標伺服器 and 要匯出的金鑰。然後按 **匯出** 來開始匯出。

圖形 6.5 將金鑰匯出至金鑰伺服器



6.5 文字與檔案加密

KGpg 也提供加密文字或剪貼簿內容的功能。按一下掛鎖圖示並找到 *加密剪貼簿* 和 *解密剪貼簿* 的選項 (和開啟整合編輯器相同的選項)。

6.5.1 加密與解密剪貼簿

複製到剪貼簿的檔案僅需按幾下便可輕易加密。按一下 KGpg 圖示來開啟功能概觀。選取 *加密剪貼簿* 並指定要使用的金鑰。關於加密程序的狀態訊息會顯示在桌面上。現在，加密的內容可以視需求從剪貼簿加以處理。解密剪貼簿內容也是一樣容易。僅需開啟面板上的功能表，選取 *解密剪貼簿*，再輸入與私密金鑰相關的密碼。現在可以在剪貼簿和 KGpg 編輯器中處理加密版本。

6.5.2 藉由拖放來加密與解密

若要加密或解密檔案，請按一下桌面上或檔案管理員中的圖示，將它們拖曳到面板中的掛鎖，然後再將它們放開。如果檔案未加密，KGpg 將詢問要使用的金鑰。您選取金鑰後，檔案隨即進行加密而且不會有進一步訊息。在檔案管理員中，加密的檔案會以 .asc 字尾和掛鎖圖示來表示。這些檔案可以透過按一下檔案圖示或將檔案圖示拖曳到面板中的 KGpg 符號後放開，來進行解密。然後選取檔案是否應該解密並儲存，或是顯示在編輯器中。

6.5.3 KGpg 編輯器

如果不想先在外部分編輯器建立加密內容，然後再用上述方法之一進行檔案加密，您可以使用 KGpg 的整合編輯器來建立檔案。開啟編輯器 (從內容功能表選取 *開啟編輯器*)，輸入想要的文字，然後按一下 *加密*。接著選取要使用的金鑰，並完成加密程序。若要解密檔案，請使用 *解密*，並輸入與金鑰相關的密碼。

直接從編輯器產生和檢查簽名與加密一樣簡單。移至 *簽名* → *產生簽名*，接著從檔案對話方塊選取要簽名的檔案。然後指定要使用的私密金鑰，並輸入相關密碼。KGpg 會通知簽名產生成功。檔案也可從編輯器產生簽名，僅需按一下 *簽名/確認*。要檢查簽名的檔案，請移至 *簽名* → *確認簽名*，並在下列對話方塊選取要檢查的檔案。您確認選擇後，KGpg 將檢查簽名並報告作業結果。其他可能性是將簽名檔案載入編輯器並按一下 *簽名/確認*。

6.6 若需詳細資訊

若需關於加密方法的理論背景資訊，請參考 GnuPG 計畫頁面上簡略且清楚的說明，網址為 <http://www.gnupg.org/documentation/howtos.html.en>。本文件也提供進一步資訊來源的清單。

部 III. 多媒體

Linux 音效

Linux 包含了各式各樣的音效及多媒體應用程式。當中的一些應用程式屬於其中一種主要桌面環境的一部分。透過此處所述的應用程式，可以控制播放的音量與平衡、播放 CD 及音樂檔，還可以錄製及壓縮您自己的音效資料。

7.1 混音器

混音器可讓您方便地控制電腦音效輸出及輸入的音量與平衡。各種混音器的主要差異是使用者界面的外觀。不過，有幾種混音器則是針對特定硬體所設計。一個例子是 `envy24control` 混音器，專用於 Envy 24 音效晶片；另一個例子則是 `hdspmixer` 混音器，適用於 RME Hammerfall 音效卡。請從可用的混音器中，選取符合需求的混音器。

提示：測試您的混音器

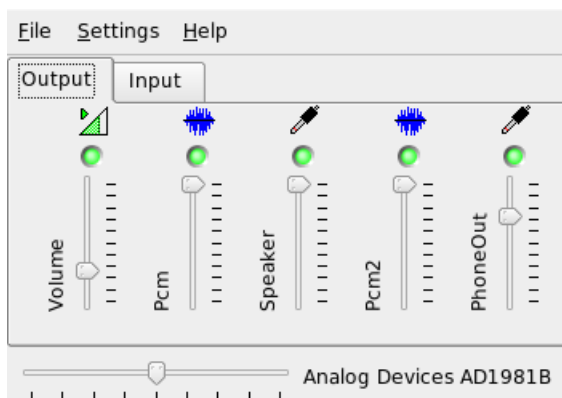
一般而言，建議先開啟混音器再開啟其他音效程式。然後使用混音器測試並調整音效卡輸入輸出的控制設定。

7.1.1 KDE 混音器 Applet

KMix 是 KDE 混音器應用程式。它已整合到 KDE 面板上，成為位於系統匣上的小型面板程式。按下面板圖示，以控制滑桿來控制喇叭的音量。如果用滑鼠右鍵按一下圖示，會出現 KMix 的內容功能表。選取 **靜音** 關閉音效輸出。接著面板圖示會改變它的外觀。再按一次 **靜音** 取消靜音。如果要微調您的聲音設定，請選取 **顯示混音器視窗**，然後設定 **輸出**、**輸入** 和 **切換**。這裡描述的每種

設備都有自己的內容功能表，以滑鼠右鍵按一下設備圖示即可開啟。您可以分別將每個設備靜音或隱藏。

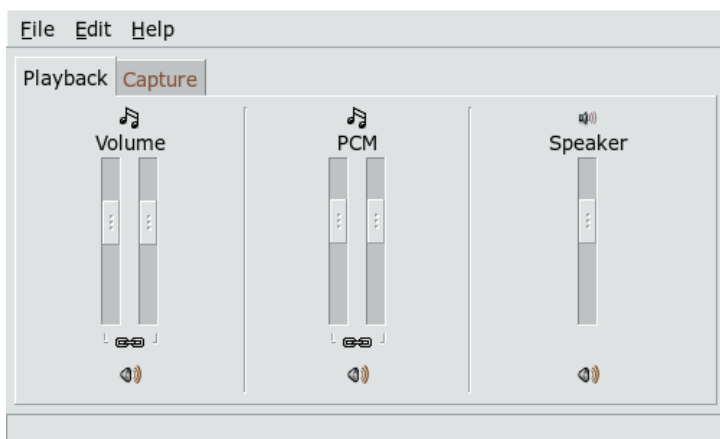
圖形 7.1 *KMix 混音器*



7.1.2 GNOME 混音器 Applet

GMix 是 GNOME 桌面的音量控制程式，已整合到 GNOME 面板。按一下面板圖示，以簡易的控制滑桿來控制喇叭的音量。如果要關閉音效輸出，用滑鼠右鍵按一下圖示，然後選取 **靜音**。接著音量控制圖示會改變它的外觀。如果要關閉音效輸出，用滑鼠右鍵按一下圖示，然後選取 **靜音**。選取 **開啟音量控制** 存取更進階的混音器功能，如 [圖形 7.2, "GNOME 混音器 Applet" \[107\]](#) 中所示。每種音效裝置都有自己的混音器標籤。

圖形 7.2 GNOME 混音器 Applet



7.1.3 alsamixer

`alsamixer` 可以在沒有 X 環境的情況下從指令行執行，因此能夠完全由鍵盤捷徑進行控制。`alsamixer` 視窗永遠包含以下元件：頂端列顯示音效卡和晶片類型、選取的檢視類型及混音器的基本資訊，然後在資訊區域下面是音量控制列。如果控制列無法在一個螢幕上完全顯示，請使用 `←` 和 `→` 鍵向左右捲動。控制列的名稱會顯示在各列下方，而目前選取的控制列會以紅色標示。任何混音器控制的靜音及取消靜音狀態可使用 `[M]` 進行切換。已靜音的控制列在其名稱下方會寫上 `MM`。任何具有擷取（錄音）功能的控制列都有紅色的擷取旗標。

`alsamixer` 具有三個不同的檢視模式：`播放`、`擷取` 和 `全部`。在預設情況下，`alsamixer` 會以 `播放` 模式啟動，僅顯示與播放（主聲道音量、PCM、CD 等）相關的混音器控制項。`擷取` 僅顯示用於錄音的控制項。而 `全部` 則會顯示所有可用的控制項。使用 `[F3]`、`[F4]` 和 `[F5]` 切換檢視模式。

使用 `→` 與 `←` 或是 `[N]` 與 `[P]` 按鍵選取聲道。利用 `↑` 與 `↓` 或是 `+` 與 `-` 按鍵調高及調低音量。各個立體聲聲道能夠單獨加以控制，使用 `[Q]`、`[W]` 和 `[E]` 可提高音量，而 `[Z]`、`[X]` 和 `[C]` 則可降低音量。`[0]` 到 `[9]` 的數字鍵可以快速改變絕對音量。這些數字對應的是整個音量從零到 90 的百分比。

7.1.4 混音器應用程式的外觀與風格

混音器應用程式的外觀與風格需視使用的音效卡類型而定。有些驅動程式，像是 SB Live!，具有多項可以控制(可微調的)混音器元件，而專業音效卡驅動程式上的元件名稱則可能完全不同。

主機板音效晶片

大部份的 PCI 主機板音效晶片以 AC97 轉碼器為基礎。主要控制了前端喇叭的主要音量。環繞、中央和 *LFE* 則是控制了後端、中間及重低音加強等喇叭的音量。每個都具有靜音開關。除此之外，一些主機板還有個別的耳機及主單音量控制。後者用於某些筆記型電腦上的內建喇叭。

PCM 控制數位 WAVE 播放的內部音量強度。PCM 是 Pulse Code Modulation (脈波符碼調變)的縮寫，這是數位訊號格式的其中一種。此控制項也有獨立的靜音開關。

其他如 *CD*、輸入、麥克風和輔助等音量，控制了從對應輸入到主要輸出的回送音量。這些選項不會影響錄音音量，而只會影響播放音量。

如果要錄音，開啟擷取開關。這是主要的錄音開關。擷取音量是作為錄音的音量輸入。在預設情況下，此開關是設定為零。選擇錄音來源如 *Line* 或 *Mic*。錄音來源僅是單一的，因此無法同時選擇其中兩個。*Mix* 是特殊的錄音來源。您可以從此來源錄製當下正在播放的訊號。

取決於 AC97 轉碼器晶片種類，AC97 轉碼器晶片也可能可以錄製如 3D 或低音/高音等特殊效果。

SoundBlaster Live! 與 Audigy 系列

SoundBlaster Live! 與 SB Audigy1 具有適用其 AC97 轉碼器晶片及 DSP 引擎的數種混音器控制項。除了上述的控制項外，它們還有 *Wave*、音樂和 AC97 音量，可以針對 PCM、WaveTable MIDI、及 AC97 混音的內部訊號的傳送及衰減。讓音量保持在 100% 的強度，以便聽清楚所有的效果。SB Audigy2 (取決於型號)相較於 SB Live 包含較少的控制項，不過仍有 *Wave* 與音樂兩個控制項。

SB Live 的錄音效果與主機板晶片的類似。您可以選擇 *Wave* 和音樂做為額外的錄音來源，來錄製播放的 PCM 及 WaveTable 訊號。

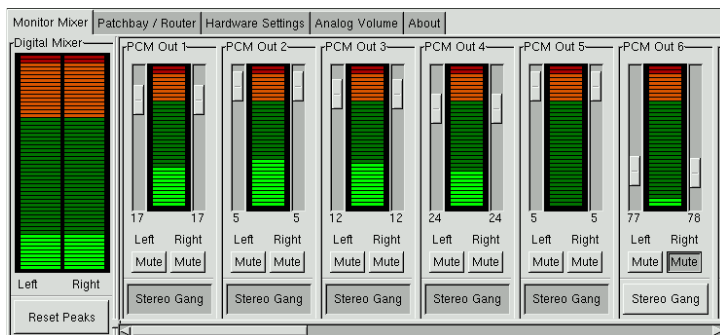
USB 音效裝置

USB 音效裝置通常具有較少的混音器控制項。有時候甚至沒有任何控制項。大部分的裝置會有 主要 或 *PCM* 控制開關可以用來控制播放音量。

7.1.5 Sound Chip Envoy24 混音器

envy24control 是使用 Envoy24 (ice1712) 晶片的音效卡混音器應用程式。由於 Envoy24 晶片的彈性，在不同的音效卡上可能有不同的功能。關於此音效卡的最新詳細資訊可由 `/usr/share/doc/packages/alsa-tools/envy24control` 中取得。

圖形 7.3 *envy24control* 的監聽器與數位混音器



envy24control 的 監聽器混音器 會顯示可在音效卡上數位混音的訊號強度。指定為 *PCM Out* 的訊號是由傳送 *PCM* 數據至音效卡的應用程式產生的。此類比輸入的訊號會顯示在 *H/W In* 下。*S/PDIF* 輸入會顯示在右方。在 類比音量 下設定類比聲道的輸入與輸出強度。

使用 監聽器混音器 滑桿來進行數位混音。在 數位混音器 中會顯示個別的強度。分配器 包含一系列圓鈕，可選擇每一個想要的輸出聲道來源。

在 類比音量 下調整類比至數位和數位至類比轉換器的擴音功能。對輸出聲道使用 *DAC* 滑桿，對輸入聲道則使用 *ADC* 滑桿。

在 硬體設定 下進行 *S/PDIF* 聲道設定。Envoy24 晶片會對於音量變更會有延遲現象，此延遲可經由 音量變更 來設定。

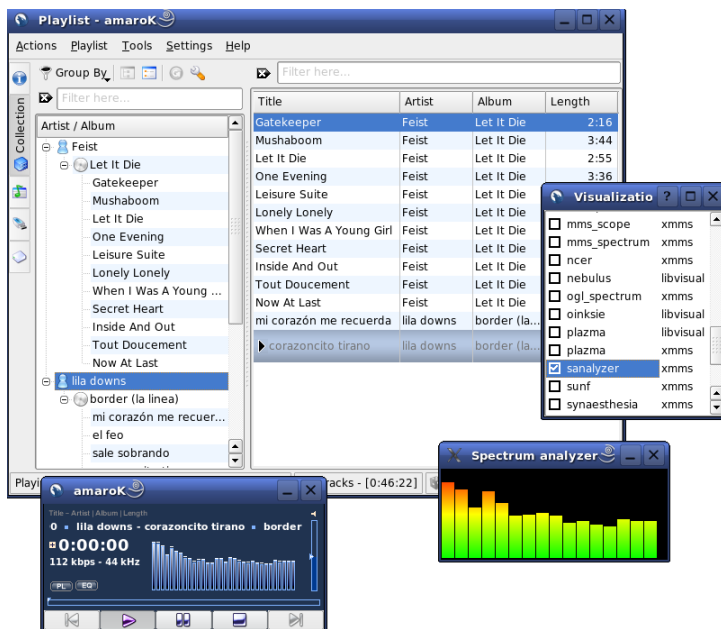
7.2 多媒體播放器

7.2.1 amaroK

amaroK 媒體播放器不但可處理各種音效格式，還能播放網際網路上廣播電台的資料流音訊廣播。此程式可播放各種後端音效伺服器 (目前是 aRts 或 Gstreamer) 所支援的檔案類型。

amaroK 首先啟用 第一次執行精靈，用來幫助設定 amaroK。在第一步驟中，先設定偏好的 amaroK 外觀與風格。選擇在不同的視窗中顯示播放器和播放清單 (請參閱 圖形 7.4, "amaroK 媒體播放器" [110]) 或將這些功能結合到單一視窗中。在第二步驟中，決定 amaroK 進行搜尋的音樂收藏位置。amaroK 會瀏覽這些資料夾尋找可以播放的媒體。在預設情況下，amaroK 是設定會遞迴瀏覽選取的資料夾 (在瀏覽時會包含其所有的子目錄)，監看選定目錄中內容的變更，並輸入位於其中的任何播放清單。透過精靈完成的所有設定值，之後只要使用 工具 → 第一次執行精靈 即可再次啟動精靈進行修改。

圖形 7.4 amaroK 媒體播放器



管理播放清單

一開始啟動 amaroK 時，它會根據在精靈中所做的設定，掃描檔案系統尋找多媒體檔案。播放清單視窗的右方會列出找到的任何播放清單。播放樂曲會依您的選擇順序而顯示。如果沒有找到播放清單，那就建立一個吧！最好的方式就是使用視窗左方的側方列來建立。在最左方有許多可用於開啟不同檢視的標籤。從每一種檢視下，拖曳個別樂曲或整個目錄當播放清單，來將它們包含於清單中。以下是每一個標籤的功能描述。

內容

使用此標籤，檢視與收藏及目前演唱者有關的資料。例如，檢視中可包含您的偏好樂曲、加入集合的最新樂曲，以及其他詳細資訊。*Home* 檢視提供您聆聽習慣的統計數字，列出您最喜愛、最新以及最不常播放的音軌。*目前曲目* 提供與當下正在播放之音軌的相關資料，例如唱片封面 (請參閱 "[封面管理員](#)" [112])、與此音軌相關的聆聽統計數字，以及其他訊息。如果您想要得知音軌的歌詞，使用 *歌詞* 標籤即可顯示歌詞。

收藏瀏覽器

使用此檢視來管理及顯示您的樂曲收藏。收藏檢視可能包含來自不同位置的檔案。工具列上的扳手圖示讓您決定要掃描哪些位置尋找音樂檔。一旦選定目錄後，就會自動開始掃描。結果會以樹狀結構顯示。使用 *主要* 及 *次要*，根據 *專輯*、*演唱者*、*音樂類型* 及 *年份* 準則，組織樹狀結構最上方的分支。一旦樹狀檢視完成，只要將樂曲名稱鍵入輸入欄位即可搜尋。當您輸入時，首先會自動跳出樹狀結構檢視中符合輸入內容的選項。如果要更新您的收藏資料，使用 *工具* → *重新掃描收藏* 重新掃描檔案系統。

播放清單瀏覽器

播放清單瀏覽器分為兩個部分。上部列出您的所有自訂播放清單，這些播放清單是將音軌拖曳到播放清單視窗並按一下 *另存播放清單* 而建立的。按一下播放清單名稱旁的 + 可以檢視其中的內容。使用拖放功能來修改這些播放清單。若要載入其中一個播放清單，在該清單上按兩下。

重要: 與其他播放器共用播放清單

將播放清單儲存成 *m3u* 或 *pls* 格式，這樣就可以與任何使用此格式的其他播放器共用。

amaroK 能夠立即編譯有用的播放清單(「智慧型播放清單」)。透過播放清單瀏覽器的下部選取其中一個智慧型播放清單，或是按一下 *建立智慧型播放清*

單定義自訂的智慧型播放清單。輸入名稱、搜尋準則、順序及選擇性的音軌限制。

檔案瀏覽器

此標籤可開啟檔案瀏覽器。它等同於標準 KDE 檔案選擇器對話方塊 (包含瀏覽檔案系統時的通用控制)。直接在文字輸入欄位中輸入 URL 或目錄。從顯示的內容拖曳元件到播放清單中來包含它們。您也可以執行遞迴搜尋在目的目錄搜尋檔案。若要執行此功能，請輸入樂曲標題的文字字串與要開始搜尋位置。然後選取 *搜尋*，結果將出現在視窗下方。

封面管理員

amaroK 提供封面管理員功能，讓您將音樂與影像資料與播放唱片上的保持一致。透過 *工具* → *封面管理員* 啟動 *封面管理員*。視窗左方的樹狀結構會列出收藏中的所有唱片。從 Amazon 取得的封面會顯示在視窗的右方。利用 *檢視* 選擇要在封面清單檢視中顯示的項目。無論您收藏中的唱片是否有封面影像，*所有專輯* 都會將其列出。*有封面的專輯* 僅列出有封面的唱片，而 *無封面的專輯* 則列出沒有封面的唱片。若是要取得封面資料，選擇您的 *Amazon 地區* 再按一下 *取得遺失的封面*，接著 amaroK 就會嘗試為您收藏中包含的所有唱片取得封面。

效果

在播放器視窗中選取 *FX* 按鈕，或是使用 amaroK 應用程式功能表，可開啟對話方塊，以啟用與設定數種音效，包括等化器、立體聲平衡、空間效果等等。請選取所要使用的效果，若有選項也一併為所選的效果逐一設定。

視覺效果

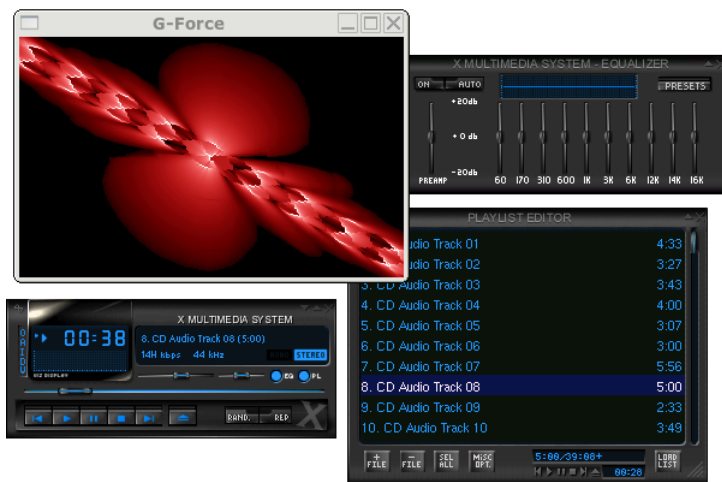
amaroK 內建數種視覺效果，可在播放音樂時顯示圖形的效果。amaroK 原始內建的視覺效果，顯示在播放器視窗中。按一下動畫，可循環選取各種可用的顯示模式。

除上述功能外，amaroK 亦支援 XMMS 媒體播放器的插件。若要使用這類插件，請先安裝 `xmms-plugins` 套件，然後從 amaroK 功能表中選取 *視覺效果*。接著便會開啟列示所有可用插件的視窗。XMMS 插件將顯示在另外的視窗中。在部分情況下，可選擇以全螢幕來顯示。使用部分插件時，除非電腦上配備 3D 加速卡，否則可能無法產生平順的視覺效果。

7.2.2 XMMS

XMMS 為另一個全功能的媒體播放器，對於音效的支援能力極佳，因此在播放時很少會發生不順或中斷情形。XMMS 非常容易使用。顯示功能表的按鈕，位於程式視窗的左上角。對於偏好 GNOME 樣式外觀及風格的使用者而言，另外提供了 XMMS 的 GTK2 版，Beep Media Player。只需安裝套件 `bmpl` 即可。不過，XMMS 的此連接埠不一定支援所有的 XMMS 插件。

圖形 7.5 XMMS 與 Equalizer、OpenGL Spectrum Analyzer 以及 Infinity 插件



以 選項 → 偏好設定 → 音效 I/O 插件 選取輸出插件模組。若已安裝 `xmms-kde` 套件，則可在此設定 aRts 音效伺服器。

重要: 使用 Disk Writer 插件

XMMS 若找不到已設定好的音效卡，會自動將其輸出重導至 *Disk Writer* 插件。在此狀況下，播放的檔案將寫入硬碟，成為 WAV 檔案。時間顯示將快於經由音效卡播放輸出的狀況。

使用 選項 → 優先設定 → 視覺效果插件 啟用各種視覺效果插件。若您的顯示卡具有 3D 加速功能，請選取 OpenGL spectrum analyzer 之類的應用程式。若已安裝 `xmms-plugins` 套件，則可使用 Infinity 插件。

在功能表按鈕下方左側共有 5 個按鈕，各有不同的字母顯示於上。這些按鈕可讓使用者快速存取額外的功能表、對話方塊與組態。使用 *PL* 開啟播放清單，使用 *EQ* 開啟等化器。

7.3 CD：播放與轉檔 (Ripping)

有許多種方式可以聆聽您最愛的音樂音軌。您可以播放 CD 或播放這些音樂的數位化版本。下節說明一些 CD 播放器應用程式，以及一些可以用來轉換及編碼音樂 CD 的應用程式。

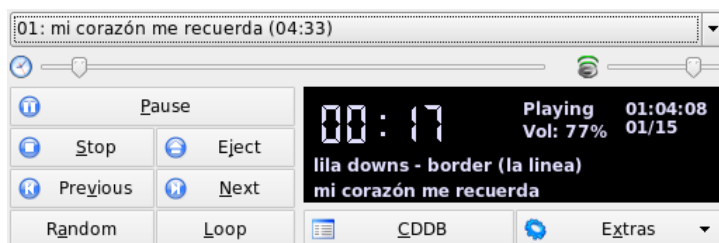
重要：CDDA 與類比 CD 播放

播放音樂 CD 有兩種不同的方式。能夠播放類比 CD 的 CD 與 DVD 光碟機，讀出音效資料並傳送到聲音輸出裝置。透過 PCMCIA、FireWire、或 USB 連接的一些外接磁碟機，需要使用 CDDA (Compact Disk Digital Audio) 先擷取音效資料，再將這些資料當作數位 PCM 進行播放。以下小節中描述的播放器不支援 CDDA。如果需要支援 CDDA，請使用 XMMS。

7.3.1 KsCD—音樂 CD 播放器

KsCD 是一個易於使用的音樂 CD 播放器。它已整合到 KDE 工作列，而且可以設定成插入 CD 後就能自動開始播放。若要存取組態功能表，選取 額外 → 設定 *KsCD*。從網際網路上的 CDDB 伺服器取得唱片及音軌資訊 (如果有相應設定 *KsCD*)。您也可以上載 CDDB 資訊與其他人分享。使用 *CDDB* 對話方塊來擷取及上載資訊。

圖形 7.6 *KsCD* 使用者介面



7.3.2 GNOME CD 播放器 Applet

這是整合到 GNOME 面板的簡易程式。使用工具圖示，設定其行為及選取主題。透過播放器視窗下方的按鈕控制播放，或是使用以滑鼠右鍵按下面板圖示或播放器視窗而開啟的內容功能表。

7.3.3 壓縮音效資料

音效的壓縮可以由各種工具進行處理。以下小節描述的指令行可以編碼及播放音效資料，以及一些能夠壓縮音效的圖形應用程式。

編碼及播放音效資料的指令行工具

Ogg Vorbis (套件 `vorbis-tools`) 為免費的音效壓縮格式，目前大部分音效播放器和可攜式 MP3 播放器皆支援此格式。此計劃的網頁為 <http://www.xiph.org/ogg/vorbis>。

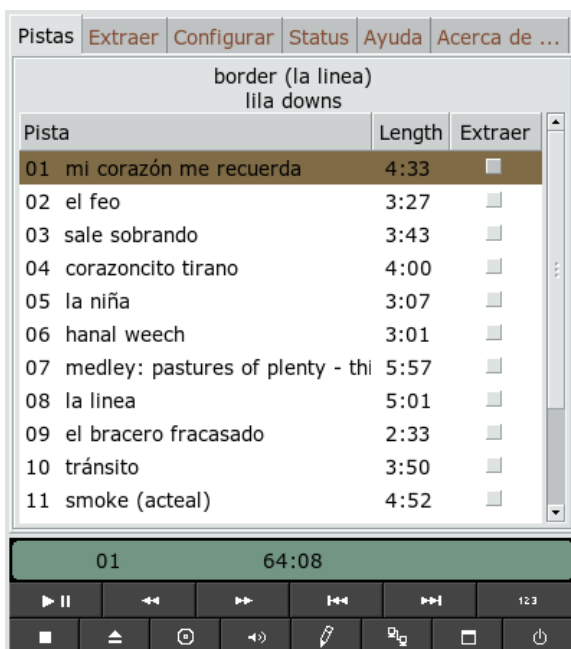
SUSE Linux 隨附數種支援 Ogg Vorbis 的工具。`oggenc` 是用來將 WAV 檔案編碼為 Ogg 的指令行工具。只需執行 `oggenc myfile.wav` 將指定的 .wav 檔案轉換為 Ogg Vorbis。`-h` 選項可概略說明其他參數。`Oggenc` 支援可變位元速率的編碼。利用此方式，可以達到更高的壓縮比例。除了位元速率，使用參數 `-q` 指定想要的品質；`-b` 決定平均位元速率；`-m` 和 `-M` 指定最小和最大位元速率。

`ogg123` 為指令行的 Ogg 播放器。請以 `ogg123 mysong.ogg` 之類的指令來啟動程式。

使用 Grip 壓縮音效資料

Grip 是 GNOME CD 播放器與轉換器 (Ripper) (請參閱 [圖形 7.7, "使用 Grip 轉換音樂 CD" \[116\]](#))。CD 播放器功能完全由視窗底部的按鈕控制。使用視窗頂端的標籤控制轉換及編碼功能。若要檢視及編輯音軌和唱片資訊或是選取要轉換的音軌，開啟 **音軌** 標籤。按一下音軌曲名旁邊的核取方塊來選取音軌。若要編輯音軌資訊，按一下 **切換音軌編輯器** 並送交您的修改。**Rip** 標籤可以選取偏好的轉換模式和控制轉換程序。在 **設定** 標籤下存取整個 Grip 組態。使用 **狀態** 檢查應用程式的狀態。

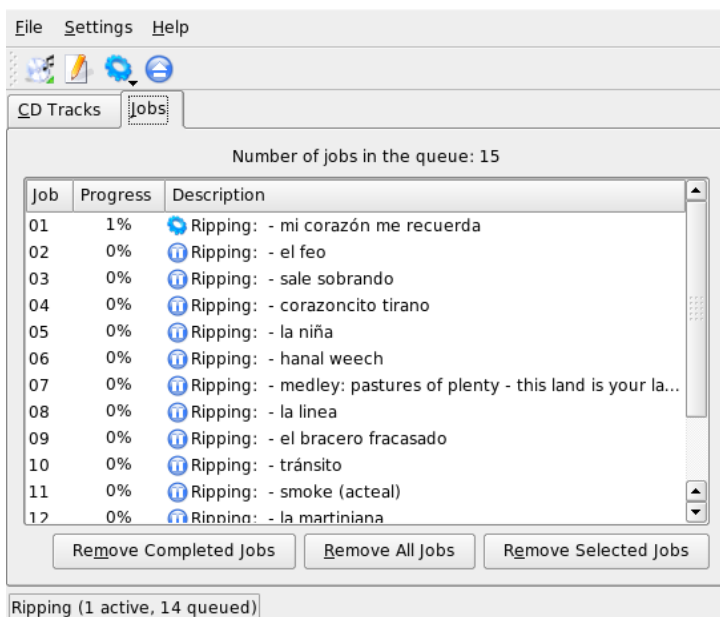
圖形 7.7 使用 Grip 轉換音樂 CD



使用 KAudioCreator 壓縮音效資料

KAudioCreator 是簡易型的 CD 轉換器應用程式 (請參閱 [圖形 7.8, "使用 KAudioCreator 轉換音樂 CD" \[117\]](#))。啟動後，它會在 *CD 音軌* 標籤中列出 CD 中的所有音軌。選取要轉換及編碼的音軌。若要編輯音軌資訊，使用 *檔案* → *編輯專輯* 下的 *專輯編輯器*。或者只要使用 *檔案* → *Rip* 選擇項目開始進行轉換及編碼。透過 *工作* 標籤觀察這些工作的進度。如果 KAudioCreator 已進行相應設定，也能夠為可以使用您的收藏的播放器 (如 amaroK 或 XMMS) 產生播放清單檔案。

圖形 7.8 使用 KAudioCreator 轉換音樂 CD

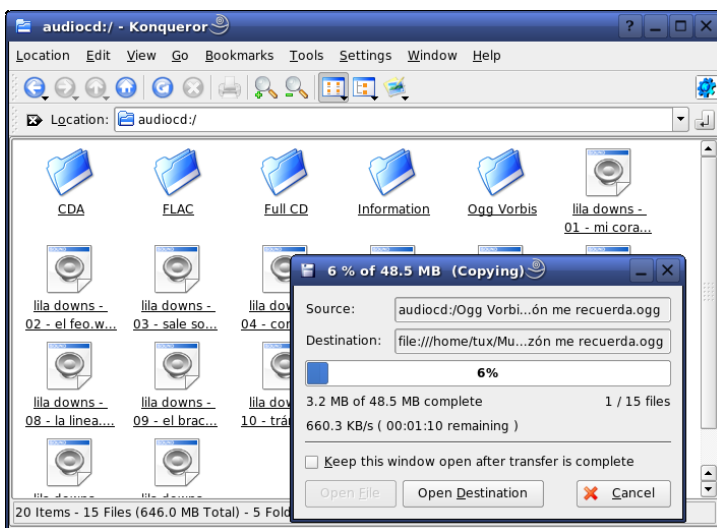


使用 Konqueror 壓縮音樂 CD

使用 Konqueror 開啟實際轉換程序前，請先在「KDE 控制中心」設定音樂 CD 及 Ogg Vorbis 編碼器的處理方式。選取 **聲音 & 多媒體** → **音樂 CD**。組態模組分為三個標籤：**一般**、**檔案名稱** 和 **Ogg Vorbis 編碼器**。一般來說，會自動偵測合適的 CD 裝置。除非自動偵測失敗，需要手動設定 CD 裝置，否則不要變更這些預設值。錯誤更正和解碼器順序也可在此設定。**Ogg Vorbis 編輯器** 標籤決定了編碼的品質。若要為已轉換的音樂資料設定唱片、音軌及演出者資訊的線上查詢，選取 **新增音軌資訊**。

將 CD 插入 CD-ROM 光碟機並在 **位置列** 輸入 `audiocd: /` 以便開始轉換程序。接著 Konqueror 會列出 CD 的音軌和一些資料夾 (請參閱 [圖形 7.9, "使用 Konqueror 轉換音效資料" \[118\]](#))。

圖形 7.9 使用 Konqueror 轉換音效資料

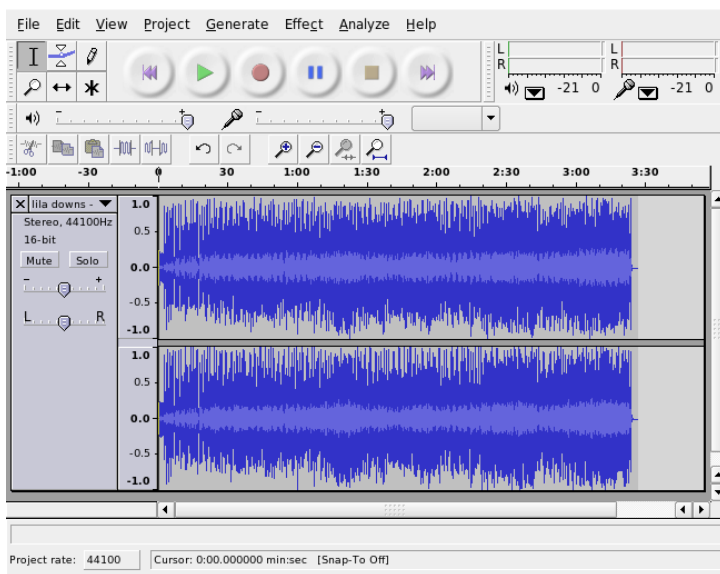


如果要將未經壓縮的音效資料保存在磁碟上，只需選取 .wav 檔案再拖曳到另一個 Konqueror 視窗，將它們複製到最後的目的地。若要開始 Ogg Vorbis 編碼，拖曳 Ogg Vorbis 資料夾到另一個 Konqueror 視窗。將 Ogg Vorbis 資料夾拖曳到其目的地後就會立刻開始編碼。

7.4 使用 Audacity 進行硬碟錄音

使用 audacity (套件為 audacity)，錄音與編輯音效檔案，這稱為硬碟錄音。當您首次啟動程式時，請選取語言。之後，若要變更語言設定，請至 檔案 → 優先設定 → 介面。變更的語言，在下次啟動程式時才會生效。

圖形 7.10 音效資料頻譜檢視



7.4.1 錄製 WAV 檔案與輸入檔案

按一下紅色錄音按鈕，可建立空白的立體音軌並開始錄音。若要變更標準參數，請在 **檔案** → **優先設定** 下進行想要的設定。**音效 I/O** 與 **品質** 為重要的錄音設定。即使已有音軌，按錄音按鈕仍然將建立新的音軌。使用者一開始可能會覺得困擾，因為在標準大小的程式視窗中看不到這些音軌。

若要輸入音效檔案，請選取 **專案** → **輸入音效**。程式支援 WAV 格式及壓縮的 Ogg Vorbis 格式。若需此格式的詳細資訊，請參閱 [節 7.3.3, "壓縮音效資料"](#) [115]。

7.4.2 編輯音效檔案

在音軌的左側可以開啟 **音軌功能表**。此功能表提供不同檢視及基本編輯操作的各種選項。若要重新命名音軌，選取 **名稱** 並輸入新名稱。Audacity 提供的不同檢視模式包括 *Waveform*、*Waveform (dB)*、*Spectrum* 和 *Pitch*。選擇適合您需要的模式。如果想要分別編輯音體音軌的每個聲道，選取 **分割音軌**。這樣就可以

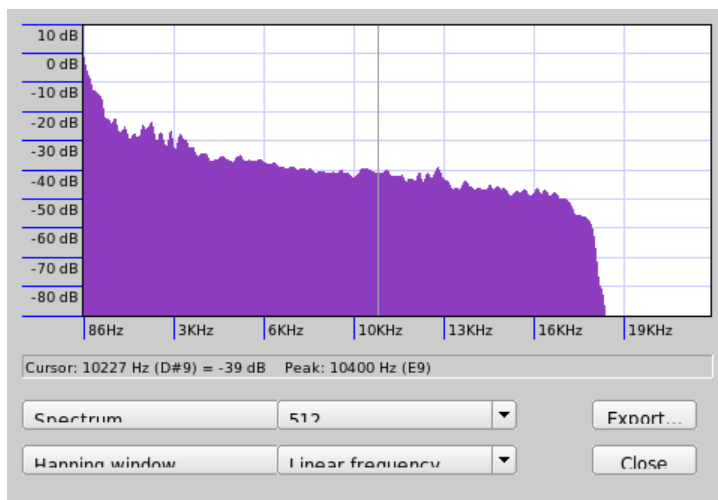
將每個聲道視為個別音軌。為每個音軌設定 **取樣格式**(單位為位元)和 **取樣速率**(單位為 Hz)。

在使用 **編輯功能表**提供的大部分工具前，請先選取要編輯的音軌聲道及區段。選取後，即可對其套用所有類型的修改和效果。

視所選的檔案類型而定，**檢視**→**設定選取格式**提供各種區段選項的檢視格式。利用 **設定銜接模式**，區段邊界將自動轉換為所選檢視格式。例如，若您選取 **PAL** 框架做為檢視格式並啟動 **銜接模式**，則選取區段邊界時，總是會選取多個框架。

所有的編輯工具都有工具提示，因此應該易於使用。使用 **檢視**→**歷程**，可存取 **復原歷程** 功能。這是個很好用的功能，可讓您檢視最近的編輯步驟，只要在清單中按一下，便可還原這些步驟。請小心使用 **捨棄**，因為會從清單中刪除編輯步驟。一旦放棄後，將無法還原這些步驟。

圖形 7.11 頻譜



內建的頻譜分析器，可協助您快速找出雜訊。請使用 **檢視**→**繪製頻譜**，檢視所選區段的頻譜。使用 **對數頻率**，以對數頻率刻度來顯示八度音。若在頻譜內移動滑鼠，將自動顯示波峰頻率與對應的音符。

以 **效果**→**FFT 濾波器** 移除不需要的頻率。在濾波的過程中，可能需要以 **增強** 來重新調整訊號的振幅。另外，使用 **增強** 來檢查振幅。**新波峰振幅** 預設為 0.0

dB。此一設定值代表所選音效格式中，最大的可能振幅。增強顯示將所選區段放大至此波峰振幅所需的值。負值代表過度增強。

7.4.3 儲存與輸出

若要儲存整個專案，請選取 檔案 → 儲存專案，或 另存專案。這樣可以產生描述專案的 XML 檔案，其副檔名為 .aup。以專案名稱再附加 _data 的目錄中，會儲存實際的音效資料。

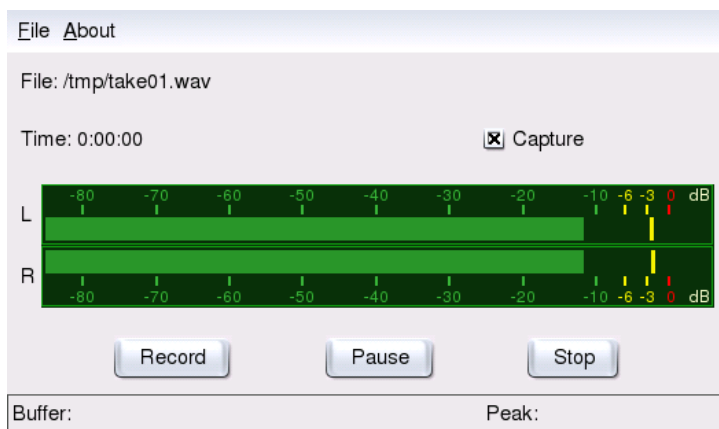
整個專案或目前所選的區段也可輸出成為立體聲 WAV 檔案。若要以 Ogg Vorbis 格式輸出專案，請參閱 節 7.3.3, "壓縮音效資料" [115] 中的資訊。

7.5 直接錄音與播放 WAV 檔案

alsa套件的 arecord與 aplay，提供PCM設備簡單且彈性的介面。arecord與 aplay 可用來錄製與播放 WAV 格式和其他格式的音效資料。arecord -d 10 -f cd -t wav mysong.wav 指令可以 CD 品質 (16 位元、44.1 kHz) 錄製 10 秒鐘的 WAV 檔案。若要列示所有的選項，請在執行 arecord 與 aplay 時，加上 --help 選項。

qaRecord(套件為 kalsatools)為簡單的錄音程式，具有圖形介面與強度顯示。由於此程式使用約 1 MB 的內部緩衝區(可利用 --buffersize 來設定)，即使硬體執行速度較慢，仍然可流暢地錄音。特別是以即時優先程度來執行此程式時，更是如此。在錄音期間，目前所用的緩衝區大小顯示在狀態列的 緩衝區 之下，而此錄音所用的最大緩衝區大小則顯示在 波峰 之下。

圖形 7.12 QARcord—簡易的硬碟錄製應用程式



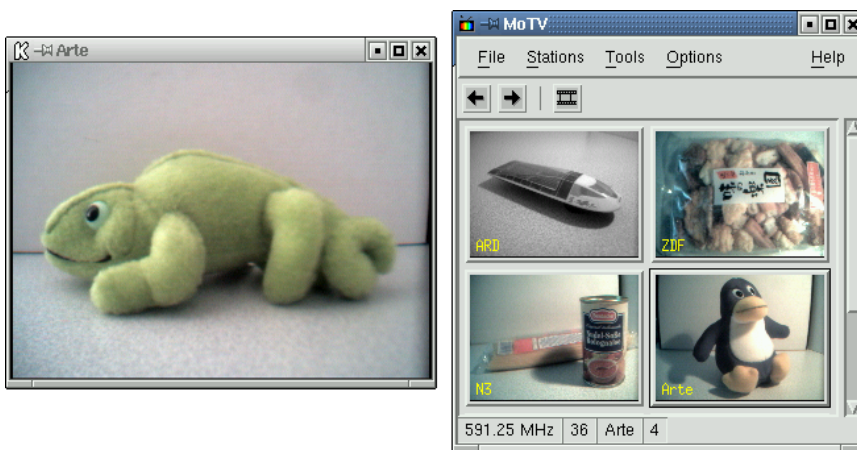
電視、視訊、廣播與網路攝影機

本章介紹部分基本的 Linux 視訊、廣播與網路攝影機應用程式。瞭解如何設定與使用 `motv` 來觀賞類比式電視、使用網路攝影機，以及瀏覽視訊文字。使用 `xawtv4` 於數位視訊廣播。使用 `gqcam` 可執行網路攝影機。使用 `nxtvepg` 或 `xawtv4` 可存取 EPG 資訊。

8.1 使用 `motv` 觀看電視

`motv` 為 `xawtv` 的後繼加強版。`motv` 在使用者介面整合了所有必要的功能。請使用 多媒體 → 視訊 → `motv` 來啟動應用程式。若要從指令行啟動，請輸入 `motv`。應用程式初次啟動時，只會顯示一個 TV 視窗。請在視窗上按一下滑鼠右鍵，開啟功能表視窗。

圖形 8.1 motv 電視應用程式



8.1.1 視訊來源與網路搜尋

請在 **設定** → **輸入** 中，選取視訊來源。若在此選取 **電視**，請在啟動應用程式之前，設定廣播網路。在 **設定** 功能表下，可自動執行網路搜尋。若您按一下 **儲存設定**，找到的網路將會輸入您的主目錄中的 `.xawtv` 檔案，在您下次啟動應用程式時便可使用。

提示: 選取頻道

若不要瀏覽所有可用的頻道，可利用 **Ctrl** + **[?]**，尋找下一個頻道。若有需要，後續可利用 **[?]** 或 **[?]** 來調整廣播頻率。

8.1.2 擷取音訊資料

電視卡的音效輸出連接至音效卡的 **Line** 輸入，再連至喇叭或擴大機。有些電視卡可變更音效輸出的音量。選取 **設定** → **滑動鍵**，便會出現滑桿，可用來設定音量。此視窗亦提供設定亮度、對比與色彩的滑桿。

若要使用音效卡來播放音樂，請使用 **gamix** 來檢查混音器的設定，詳細說明請參閱 [節 7.1, "混音器" \[105\]](#)。若音效卡符合 **AC97** 規格，請將 **Input-MUX** 設為 **Line**。音量可利用 **Master** 與 **Line** 滑桿來調整。

8.1.3 畫面比例與全螢幕模式

大部分電視影像的高度與寬度比例為 4:3。此比例可在 **工具** → **畫面尺寸** 中設定。若在此處選取 4:3(預設值)，即使顯示大小有所變動，仍然會自動維持畫面尺寸。

利用 **F** 或 **工具** → **全螢幕**，可切換至全螢幕模式。若在全螢幕模式中，電視影像未調整為整個螢幕的大小，則必須再做細部的設定。許多顯示卡可將全螢幕模式的電視影像，調整成為整個螢幕的大小，而無須變更圖形模式。若您的顯示卡未支援此功能，則在使用全螢幕模式時，必須將圖形模式切換為 640x480。相關組態請在 **設定** → **組態** 中建立。若您切換至全螢幕模式，重新啟動 **motv** 後，螢幕模式也會隨之變更。

提示： 將組態儲存於 **.xawtv**

按一下 **設定** → **儲存設定**，可自動建立與更新 **.xawtv** 檔案。在此，廣播電台將與組態一併儲存。有關組態檔案的詳細資訊，請參閱 **xawtvrc** 的 **man** 頁面。

8.1.4 啟動器功能表

使用啟動器功能表，以便啟動其他應用程式與 **motv** 一起執行。例如，使用鍵盤快速鍵來啟動 **gamix** 音訊混音器，以及 **alevt** 視訊文字應用程式。要從 **motv** 啟動的應用程式，必須在 **.xawtv** 檔案中輸入。設定項目如下：

```
[launch] Gamix = Ctrl+G, gamix AleVT = Ctrl+A, alevt
```

先是應用程式名稱，接著為快速鍵，然後才是用來啟動應用程式的命令。上述 **[launch]** 將出現在 **工具** 功能表中，便可用來啟動應用程式。

8.2 視訊文字支援

使用 **alevt** 來瀏覽視訊文字頁面。請以 **多媒體** → **視訊** → **alevt** 或在指令行使用 **alevt**，來啟動應用程式。

應用程式將儲存 motv 啟用所選站台的全部頁面。請直接輸入想要查看的頁號，或是按一下頁碼，便能瀏覽頁面。按一下視窗下緣的 << 或 >>，便可往前一頁或下一頁移動。

新版的 motv 與其後續的 xawtv4，皆包含本身的視訊文字檢視器應用程式：mtt (motv) 與 mtt4 (xawtv4)。mtt4 甚至支援 DVB 卡。

8.3 網路攝影機與 motv

若 Linux 支援您的網路攝影機，便可使用 motv 來存取網路攝影機。您可以在 <http://www.linux-usb.org> 找到支援的 USB 設備摘要。如果您在存取網路攝影機之前，已經使用 motv 來存取電視卡，表示已經載入 bttv 驅動程式。當您的網路攝影機連接至 USB 時，便會自動載入網路攝影機的驅動程式。從指令行以 `-c /dev/video1` 參數啟動 motv，便可存取網路攝影機。若要存取電視卡，請使用 `motv -c /dev/video0`。

當 bttv 驅動程式自動載入之前 (例如，啟動電視應用程式)，將網路攝影機連接至 USB 時，`/dev/video0` 將保留供網路攝影機使用。在此狀況下，若以 `-c /dev/video1` 參數啟動 motv 以存取電視卡，可能會出現錯誤訊息。這是因為 bttv 驅動程式未自動載入。請以 root 使用者的身份，使用 `modprobe bttv` 分別載入驅動程式，便可解決此問題。若要查看系統中可設定的視訊裝置，請使用 `motv -hwscan`。

8.4 nxtvepg 一個人電腦上的電視雜誌

部分廣播電台，可將視訊文字訊息連同 EPG 訊息 (Electronic Program Guide，電子節目指南) 一併傳送。使用 `nxtvepg` 程式，可輕鬆地檢視此一電子節目指南。若要使用此功能，您必須具有 bttv 驅動程式所支援的電視卡，且接收的頻道必須提供 EPG 的服務。

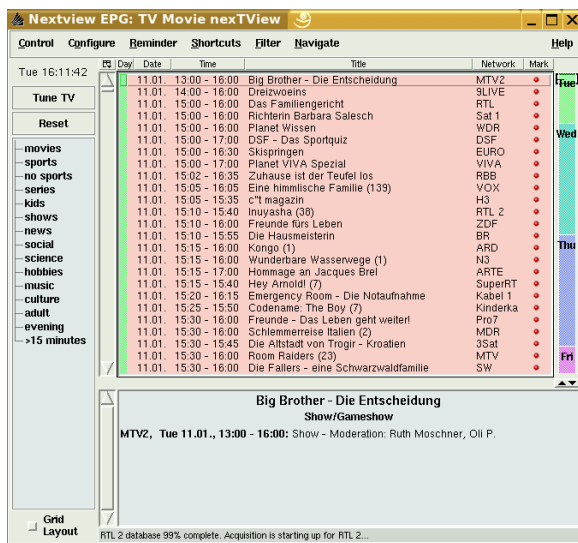
`nxtvepg` 可根據頻道與主題來排序廣播，例如，電影與體育，也可根據條件來過濾，例如，實況、立體聲或字幕。請以 `多媒體 → 視訊 → nxtvepg` 或在指令行使用 `nxtvepg`，來啟動應用程式。

8.4.1 匯入 EPG 資料庫

若要使用 EPG 訊息來設定與更新節目資料庫，請將電視卡的選台器設至提供 EPG 廣播的電台。利用 `motv` 或 `nxtvepg` 之類的電視應用程式，便可設定選台器。在同一時間，只能有一個應用程式存取選台器。

若您已在 `motv` 中設定 EPG 廣播電台，`nxtvepg` 便立即開始輸入目前的電視節目清單。您將可看到執行的進度。

圖形 8.2 `nxtvepg` 電子電視雜誌



若您尚未啟動電視應用程式，可讓 `nxtvepg` 搜尋 EPG 廣播電台。若要使用此功能，請執行 設定 → 掃描提供廠商。預設會啟用 使用 `.xatv`。這代表 `nxtvepg` 將存取儲存在此檔案中的廣播電台。

提示: 疑難排解

若出現問題，請檢查 電視卡輸入 下，是否選擇了適當的視訊來源。

請從 設定 → 選取提供廠商，選取找到的 EPG 提供廠商。設定 → 合併提供廠商 可在不同的提供廠商資料庫之間，建立更具彈性的關聯。

8.4.2 排序節目

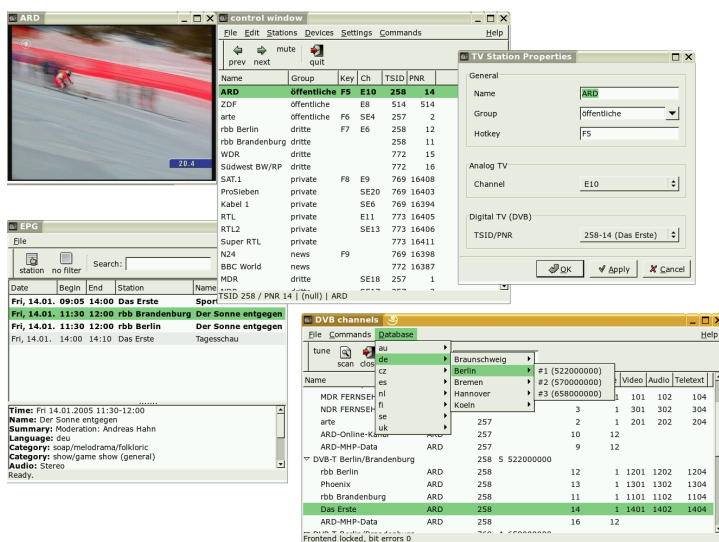
nxtvepg 提供便利的過濾功能，以便使用者輕鬆管理複雜的節目單。請使用 **設定** → **顯示網路** 來啟用網路選擇清單。**過濾功能表** 提供許多過濾功能。在節目清單上按一下滑鼠右鍵，可開啟特殊的過濾功能表，啟用內容過濾的功能。

值得注意的是 **導覽功能表**。這是直接從 EPG 資料所建立。此功能表的語言是由網路所提供。

8.5 以 xawtv4 觀賞數位視訊廣播

以 YaST 正確地設定硬體後，從主功能表 (**多媒體** → **視訊** → **xawtv4**) 啟動 xawtv4。在您實際開始觀賞喜愛的廣播節目前，請建置 DVB 電視台的資料庫。

圖形 8.3 執行 xawtv4



道編號_ (以所在位置的實際值取代_國家_與_頻道_編號_)，從 xawtv4 內建的資料庫中要求這些資訊。

一旦掃描器設定好頻道後，第一筆資料將顯示於瀏覽器視窗。使用 *指令* → *完整掃描*，開始完整掃描所有可用的電視台。掃描器仍在執行時，您可以選取喜愛的電視台，將之拖曳至控制視窗，便能加入電視台清單中。離開頻道掃描器，然後選取一個頻道，便可開始觀賞廣播節目。

提示: 編輯電視台清單

使用鍵盤快速鍵，經由鍵盤來控制頻道選取。若要為電視台清單中的電視台設定快速鍵，請選取電視台，按一下 *編輯* → *編輯電視台*。接著會開啟 *電視台內容* 對話方塊。輸入快速鍵，然後按一下 *確定* 以離開對話方塊。此對話方塊也可讓您定義子功能表，以包含電視台的群組 (例如，"新聞"或"私人")。

xawtv4 軟體套件包含數個實用的獨立多媒體應用程式：

pia4

由指令行控制的精簡電影播放程式，可用來播放由 xawtv4 所錄製的任何電影資料流。

mtt4

視訊文字瀏覽器 (請參閱 [圖形 8.4, "mtt4 視訊文字瀏覽器" \[130\]](#))。

圖形 8.4 mtt4 視訊文字瀏覽器



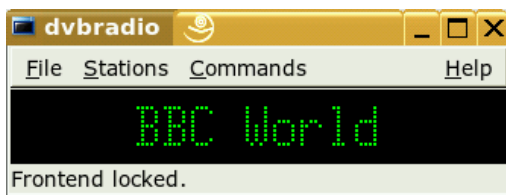
alexplorer

獨立的 DVB 頻道掃描器應用程式它的功能已整合至 xawtv4。

dvbradio

DVB 電台播放程式。當您完成起始的電台掃描後 (請參閱 圖形 8.5, "DVB 電台" [130]), 可利用它來收聽 DVB-S 電台資料流。

圖形 8.5 DVB 電台



dvbrowse

EPG 瀏覽器應用程式。當您完成起始的電台掃描後, 利用它來取得 EPG 資訊。

K3b -- 燒錄 CD 或 DVD

K3b 是一個多功能的程式，可燒錄資料與音樂 CD 和 DVD。請從主功能表啟動程式，或是輸入 `k3b` 命令來啟動程式。以下數節簡單介紹如何開始基本光碟燒錄程序，在 Linux 下完成第一張 CD 或 DVD。

9.1 建立資料光碟

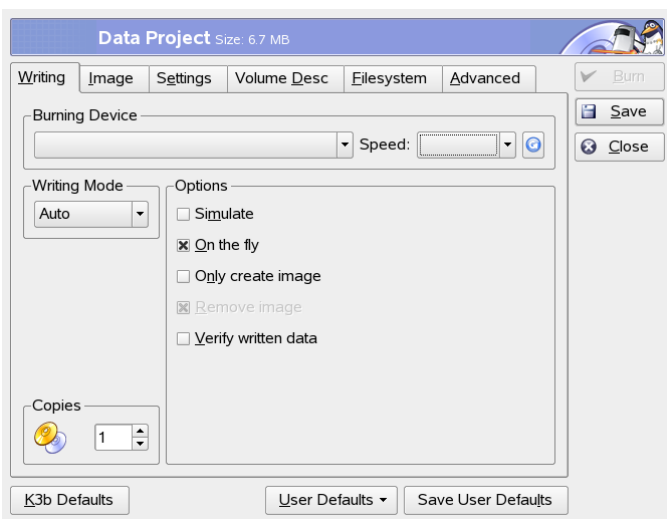
要建立資料光碟，請移至 檔案 → 新專案 → 新資料專案。視窗底端會出現專案檢視，如 [圖形 9.1, "建立新資料光碟" \[132\]](#) 所示。請將要燒錄的目錄或檔案，從您的主目錄拖放至專案資料夾。使用 檔案 → 另存新檔，以您指定的名稱來儲存專案。

圖形 9.1 建立新資料光碟



然後從工具列選取 燒錄，或按下 **Ctrl** + **B**。接著出現的對話方塊包含 6 個索引標籤，提供許多燒錄光碟的選項。請參閱 圖形 9.2, "自訂燒錄程序" [133]。

圖形 9.2 自訂燒錄程序



寫入索引標籤含有燒錄裝置、速度、燒錄選項的各種設定。以下為提供的選項：

燒錄設備

程式所偵測到的燒錄器將顯示在此快顯功能表下。您也可在此選擇速度。

警告：選取防燒壞寫入速度。

一般來說，您應該選擇 *自動*，選擇最高可能的寫入速度。但是，如果增加了速度，但是系統無法快速的傳送資料，可能會造成緩衝區不足。

寫入模組

此選項決定雷射如何寫入光碟。在 **DAO** (一次寫入光碟) 模式，寫入光碟期間不會停止雷射。若要燒錄音樂光碟，建議使用此模式。不過，並非所有燒錄器皆支援此模式。在 **TAO** 模式 (一次寫入音軌)，獨立的寫入程序是用於個別音軌。**RAW** 模式並不常用，因為在此模式下燒錄器不會執行資料校正。建議的設定是 *自動*，可讓 K3b 使用最合適的設定。

模擬

此功能可用來檢查您的系統是否支援所選的燒錄速度。整個過程並不會實際啟動雷射，目前只是為了測試系統。

即時燒錄

直接燒錄資料而不須建立映像檔案 (若系統效能不佳，請勿使用此功能)。映像檔案又稱為 ISO 映像，檔案中包含了整片光碟的內容，後續可用來燒錄光碟。

僅建立映像

此選項可建立映像檔案。請在 *暫存目錄* 中設定此檔案的路徑。映像檔案可在之後用來燒錄光碟。要這樣做，請使用 *工具* → *CD* → *燒錄 CD 映像*。若使用此選項，將會停用此區的所有其他選項。

移除映像

完成時請從硬碟移除暫存映像檔案。

確認寫入的資料

比較原始與燒錄資料的 MD5 總數，來檢查寫入資料的完整性。

只能在第一個索引標籤選擇選項 *僅建立映像* 時，才能存取 *映像* 索引標籤。如果是這樣，您可以決定要將 ISO 寫入哪個檔案。

設定索引標籤包含 2 個選項：*資料軌模式* 和 *多工作業模式*。*資料軌模式* 選項包含資料軌如何寫入的組態。一般認為 *自動* 選項是最好的方法。*多工作業模式* 是用來將資料加入到已經寫入但尚未完整燒錄的 CD。

在 *媒體說明* 索引標籤，請輸入一些用來識別這個資料專案的一般資訊，包括出版者和準備者，以及用來建立這個專案的應用程式和作業系統。

在 *檔案系統* 之下，請指定光碟的檔案系統 (RockRidge、Joliet 或 UDF)。同時也在此處指定如何處理符號連結、檔案權限與空白。在 *進階* 索引標籤，進階使用者可指定額外的設定。

依需求調整好所有設定值後，請使用 *燒錄* 開始實際燒錄程序。另外，請使用 *儲存* 來儲存這些設定值，供未來使用與調整。

9.2 建立音樂光碟

基本上，建立音樂光碟與建立資料光碟並無重大差異。請選取 *檔案* → *新音樂專案*。將個別音軌拖放至專案資料夾。音樂資料必須是 WAV 或 Ogg Vorbis 格式。請在專案資料夾中上下移動音軌來調整音軌的順序。

包含 CD 文字，您可以新增特定文字資訊到 CD，如 CD 標題、歌手名字和專輯名稱。支援這個功能的 CD 播放器可以讀取和顯示這些資訊。若要新增 CD 文字資訊到音軌，請先選擇音軌。再以滑鼠右鍵按一下，然後選擇 **內容**。就會開啟一個新的對話方塊，您可以在其中輸入文字。

燒錄音樂光碟的對話方塊，與燒錄資料光碟的對話方塊差異不大。不過，一次寫入光碟和一次寫入音軌模式比較重要。一次寫入音軌模式會在音軌之間，插入兩秒的間隔。

提示: 保留資料完整性

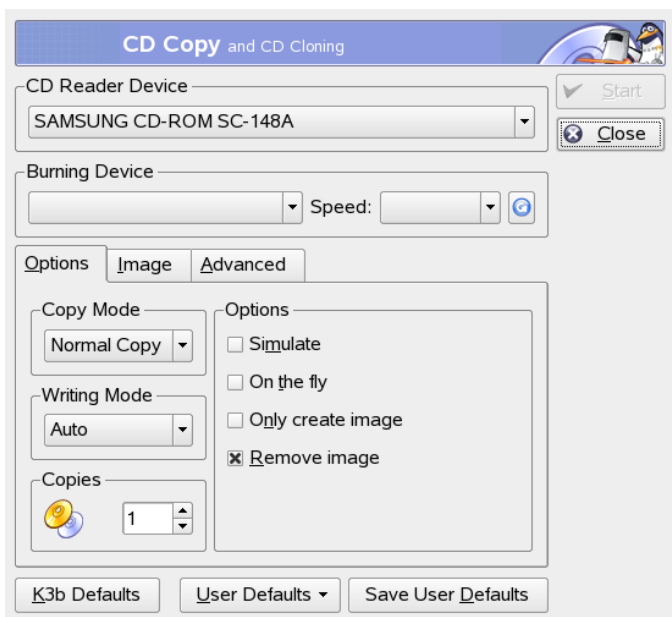
在燒錄音樂 CD 時，選擇較慢的燒錄速度可以降低燒錄錯誤的風險。

依需求調整好所有設定值後，請使用 **燒錄** 開始實際燒錄程序。另外，請使用 **儲存** 來儲存這些設定值，供未來使用與調整。

9.3 複製 CD 或 DVD

根據您的媒體，選取 **工具** → **複製 CD** 或 **工具** → **複製 DVD**。在開啟的對話方塊，請設定讀取與寫入裝置的設定值，如 [圖形 9.3, "複製光碟" \[136\]](#) 所示。也可以使用此處討論的寫入選項。另一個功能是讓使用者選擇複製多片 CD 或 DVD。

圖形 9.3 複製光碟



勾選 **即時燒錄** 在讀取時即可燒錄 CD，或選取 **僅建立映像** 可以在指定的路徑 (臨時目錄 → 寫入映像檔至) 建立映像並在稍後燒錄映像。

9.4 燒錄 ISO 映像

如果您已經有 ISO 映像，請移至 **工具** → **CD** → **燒錄 CD 映像**。將開啟一個視窗，在其中輸入 **要燒錄的映像** 的位置。K3b 將計算檢查總數，並顯示在 **MD5** 總數。若 ISO 檔案是從網路下載而來，MD5 Sum 可用來檢查下載是否成功。

使用 **選項** 與 **進階** 索引標籤可設定您的偏好設定。要燒錄光碟，請按一下 **開始**。

9.5 建立多工作業的 CD 或 DVD

可以使用多工作業的光碟，在包含一個以上燒錄作業，寫入資料。例如，當寫入備份小於媒體時就非常有用。在每個寫入作業，您可以新增其他的備份檔案。

有趣的是，這不受限於資料 CD 或 DVD。您也可以新增音樂燒錄作業到多工作業光碟。

若要開始新的多工作業光碟，請繼續以下步驟：

- 1 首先建立自己的資料光碟然後新增所有檔案。您無法使用音樂 CD 燒錄作業開始多工作業光碟。請確認不要填滿整個光碟，因為如此一來您將無法新增新的作業。
- 2 使用 **專案** → **燒錄** 燒錄您的資料。會出現對話方塊。
- 3 從 **設定** 索引標籤選擇 **開始多工作業**。
- 4 視需要設定其他選項。請參閱 [節 9.1, "建立資料光碟" \[131\]](#)。
- 5 要燒錄光碟，請按一下 **燒錄**。

成功完成燒錄後，您已經建立一張多工作業光碟。只要光碟上有足夠的空間，您可以加入其它的資料。只有在確定您不需要增加任何資料，或空間已不足，才結束多工作業光碟。

注：有關多工作業光碟的儲存空間

請注意，多工作業光碟需要額外的空間，記錄每個作業的項目。這會減少光碟可用的空間。減少的空間大小取決於燒錄作業的次數。

9.6 若需詳細資訊

除了上述兩項主要功能之外，K3b 亦提供其他功能，例如，複製 DVD、以 WAV 格式讀取音樂光碟、重覆燒錄光碟，以及以整合音樂播放器播放音樂。所有可用程式功能的詳細描述，可在 <http://k3b.sourceforge.net> 取得。

部 IV. 辦公室

OpenOffice.org Office 套裝軟體

OpenOffice.org 是一套威力強大的辦公室套裝軟體，為撰寫文字、使用試算表或是建立圖形和簡報等各類辦公室工作，提供所需的工具。使用 OpenOffice.org，可以在不同電腦平台之間使用相同的資料。若需要的話，您可以使用 Microsoft Office 格式開啟和儲存檔案，然後再將檔案儲存為原來格式。本章只涵蓋基本的技巧，讓您馬上可以開始使用 OpenOffice.org。請從 SUSE 功能表或使用 `ooffice` 指令來啟動應用程式。

OpenOffice.org 包含數個應用程式模組(子程式)，彼此之間均可互動。它們列示於 [表格 10.1, "OpenOffice.org 應用程式模組" \[141\]](#)。本章討論的重點在 Writer。線上說明中有每個模組的完整說明，請參閱 [節 10.6, "若需詳細資訊" \[147\]](#)。

表格 10.1 *OpenOffice.org 應用程式模組*

Writer	威力強大的文書處理程式
Calc	包含圖表公程式的工作表應用程式
Draw	建立向量圖形的繪圖應用程式
Math	建立數學公式的應用程式
Impress	建立簡報的應用程式
基本	資料庫應用程式

應用程式的外觀取決於所用的桌面或視窗管理員。除此之外，亦取決於桌面所用的開啟與儲存對話方塊格式。無論外觀如何，基本配置與功能皆相同。

10.1 與其他辦公室應用程式的相容性

OpenOffice.org 可以使用 Microsoft Office 文件、試算表、簡報與資料庫。它們可以像其他檔案一般順利地開啟，並儲存回原先的格式。由於 Microsoft 格式並未對外公開，其他應用程式無法取得規格，因此偶爾會出現格式化的問題。若您的文件出現問題，請從原來的應用程式開啟文件，然後重新儲存為開放性的格式，例如，文字文件的格式為 RTF，或試算表的格式為 CSV。

若要轉換多份文件 (例如，當初次切換至應用程式時)，請選取 **檔案** → **精靈** → **文件轉換器**。選擇轉換檔案格式。有幾個 StarOffice 以及 Microsoft Office 格式可以選擇。在選完格式之後，按一下 **下一步**，然後指定 OpenOffice.org 應該從哪裡尋找模板和要轉換的文件，以及轉換後檔案的放置目錄。在繼續之前，請確定所有其他的設定都正確。按一下 **下一步** 看一下要執行動作的摘要，您可以在這個時候看一下所有設定是否正確。最後，按一下 **轉換** 以開始轉換動作。

重要: 尋找 Windows 檔案

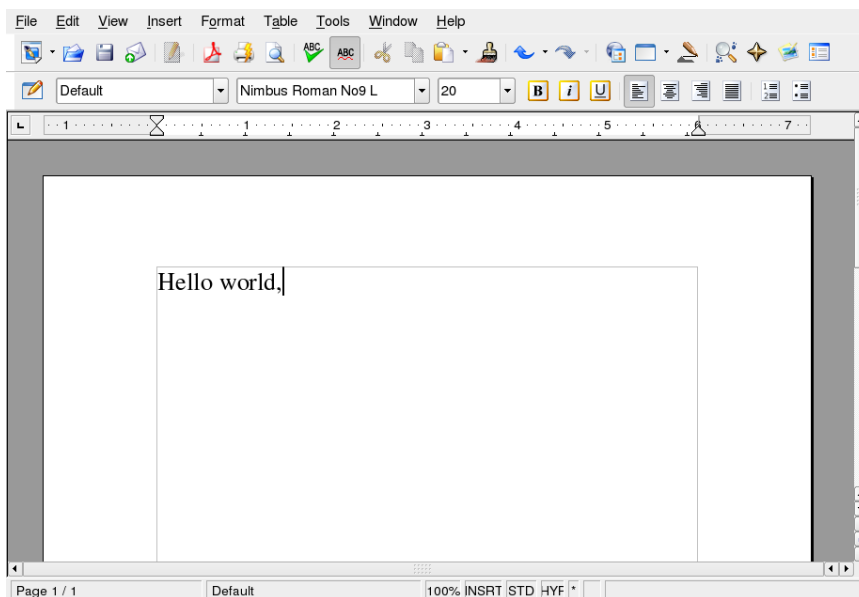
來自 Windows 分割區的文件通常位在 `/windows` 的子目錄之下。

與其他人共用文件時，您有數個選項。若接收者只需要讀取文件，請利用 **檔案** → **輸出為 PDF**，將它輸出為 PDF 檔案。利用 Adobe Acrobat Reader 之類的檢視器，可在各種平台上讀取 PDF 檔案。若要共用檔案以供編輯，請使用常用的文件格式。預設格式符合 OASIS 標準 XML 格式，使它們能夠與多種應用程式相容。TXT 與 RTF 格式雖然格式化的功能有限，但卻是文字文件不錯的選項。CSV 為試算表好用的選項。OpenOffice.org 也可能提供接收者偏好的格式，特別是 Microsoft 的格式。

OpenOffice.org 提供多種作業系統的版本。當一群使用者必須經常共用檔案，但是每人的電腦並非使用相同的系統時，OpenOffice.org 將會是最佳的工具。

10.2 使用 Writer 進行文字處理

圖形 10.1 *OpenOffice.org Writer*



有兩種建立新文件的方式。若要從新建立文件，請使用 **檔案** → **新增** → **文字文件**。若文件要使用標準的格式以及預先定義的元件，請使用精靈。精靈是一個小公用程式，可幫助您決定某些基本設定，進而從樣板直接產生立即可用的文件。例如，若要建立商業信函，請選取 **檔案** → **精靈** → **信函**。利用精靈的對話方塊，便可使用標準格式，輕易地建立基本文件。精靈對話方塊的範例會顯示在 [圖形 10.2, "OpenOffice.org 精靈" \[144\]](#)。

圖形 10.2 OpenOffice.org 精靈

The screenshot shows a wizard window titled "Specify the sender and recipient information". On the left, a "Steps" sidebar lists six steps: 1. Page design, 2. Letterhead layout, 3. Printed items, 4. Recipient and sender (highlighted in blue), 5. Footer, and 6. Name and location. The main area is divided into two sections. The "Sender's address" section has two radio buttons: "Use user data for return address" (selected) and "New sender address:". Below the second option are input fields for "Name:", "Street:", and "ZIP code/State/City:". The "Recipient's address" section also has two radio buttons: "Use placeholders for recipient's address" (selected) and "Use address database for mail merge". At the bottom, there are five buttons: "Help", "< Back", "Next >", "Finish", and "Cancel".

在文件視窗中，依照您的需要輸入文字。使用 **格式化工具列** 或 **格式功能表**，調整文件的外觀。請使用 **檔案功能表** 或 **工具列** 中對應的按鈕，列印或儲存您的文件。利用 **插入** 之下的選項，可在文件中加入額外的項目，例如，表格、圖片或圖表。

10.2.1 選擇文字

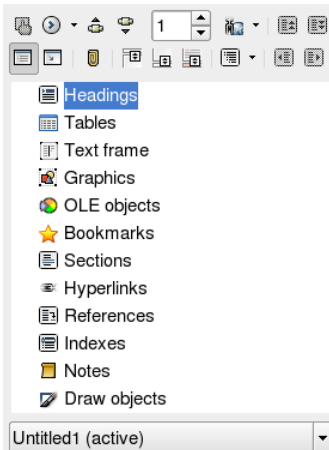
若要選擇文字，按一下選取範圍的開始點，持續按住滑鼠按鈕，移動游標至選取範圍的結束點 (選取範圍可以是文字、一整行或一整段)。選取所要的所有文字後，放開按鈕。當選取的時候，文字會以相反顏色顯示。在選取區域上按一下滑鼠右鍵以開啟內容功能表。使用內容功能表來變更字型、字型樣式以及其他文字屬性。

所選文字可剪下或複製到剪貼簿。剪下或複製的文字可貼回文件的其他位置。使用內容功能表、**編輯** 或對應的工具列圖示來存取這些功能。

10.2.2 在大型文件中瀏覽

Navigator 可以顯示關於文件內容的資訊。它同時也可以讓您快速地跳躍至包含的其他元件。例如，使用 Navigator 快速地綜覽文件內的全部章節或是影像清單。選取 **編輯** → *Navigator* 將它開啟。[圖形 10.3, "在 Writer 中的 Navigator" \[145\]](#) 顯示使用中的 Navigator。在 Navigator 中列示的元件可能因為 Writer 中載入文件的不同而有所差異。

圖形 10.3 在 Writer 中的 Navigator



10.2.3 以樣式進行格式化

使用 **格式** → *樣式與格式化* 所開啟的對話方塊，可協助您以數種方式格式化文字。若您將此對話方塊下方的下拉式清單設為 *自動*，OpenOffice.org 將會嘗試提供適用於目前工作的樣式選項。若您選取 *全部樣式*，「各類樣式」會從目前作用中的群組提供全部樣式。使用頂端的按鈕可以選擇群組。

此種格式化文字的方式稱為 *軟格式*，並未直接格式化文字。相反地，是將樣式套用於文字。您可以輕易地修改樣式，便會自動地套用格式變更至相同樣式的全部文字。

若要指定樣式至段落，請選擇要使用的樣式，然後按一下 *樣式與格式化* 中的繪圖水桶圖示。按一下要指定樣式的段落。按下 **[Esc]** 鍵或再按一下繪圖水桶圖示，便可停止指定樣式。

依照您的需要，使用 **格式功能表** 或 **工具列** 來格式化段落或字元，可讓您輕鬆地建立自己的樣式。選取格式化的項目，以便複製其樣式。然後在 **樣式與格式化** 中以滑鼠右鍵按住水桶圖示不放，從開啟的功能表中選取 **從選取區域中建立樣式**。輸入樣式的名稱，並按一下 **確定**。之後便可將此樣式套用至其他文字。

若要變更樣式的細節，請先從清單中選取樣式，在選取的樣式上按一下滑鼠右鍵，從功能表中選擇 **修改**。這會開啟對話方塊，列出所有可以修改的格式內容。

10.3 介紹 Calc

Calc 為 OpenOffice.org 的試算表應用程式。使用 **檔案** → **新增** → **試算表** 來建立新的試算表，或以 **檔案** → **開啟** 來開啟試算表。Calc 可讀取與儲存 Microsoft Excel 的格式。

在試算表的儲存格中，輸入固定資料或公式。公式可操作其他儲存格中的資料，在插入公式的儲存格中產生一個值。您也可以利用儲存格值來建立圖表。

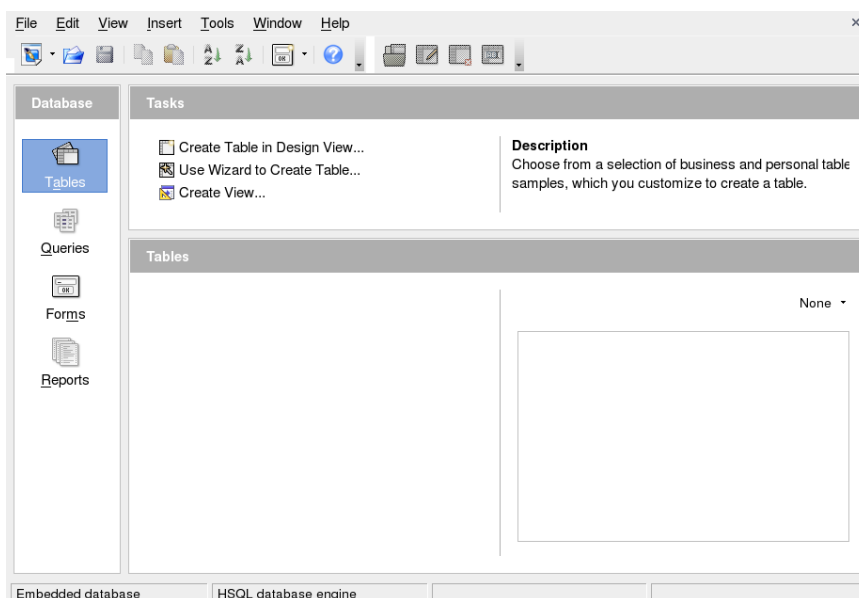
10.4 介紹 Impress

Impress 是設計用來建立簡報，以便用於螢幕顯示或列印 (例如，列印在投影片上)。利用 **檔案** → **新增** → **簡報**，可從新建立簡報。若要藉由精靈的協助來建立簡報，請使用 **檔案** → **精靈** → **簡報**。以 **檔案** → **開啟** 來開啟現有的簡報。Impress 可開啟與儲存 Microsoft PowerPoint 簡報。

10.5 介紹 Base

OpenOffice 2.0 引進一個新的資料庫模組。利用 **檔案** → **新增** → **資料庫**，可建立資料庫。接著會開啟精靈，協助您建立資料庫。Base 也能處理 Microsoft Access 資料庫。

圖形 10.4 Base--OpenOffice.org 的資料庫



藉由手動的方式，或使用方便的精靈，可建立表格、表單、查詢與報表。例如，表格精靈包含數個商務與個人使用的常見欄位。在 Base 中建立的資料庫，可作為資料來源，例如，在建立表單信函時使用。

10.6 若需詳細資訊

OpenOffice.org 包括數種資訊選項，可提供不同的資訊等級。若要全面瞭解主題，請選擇 **說明** → **OpenOffice.org 說明**。說明系統提供關於 OpenOffice.org 每一個模組 (Writer、Calc、Impress 等) 的深入資訊。

當應用程式首次啟動時，會提供 **秘訣** (滑鼠移至按鈕上方時，出現關於按鈕的簡短資訊) 與 **說明代辦程式** (根據執行的動作所提供的資訊)。除了 **秘訣** 所提供的資訊外，若要取得有關按鈕更為詳細的資訊，請使用 **說明** → **這是什麼**，然後將滑鼠移至按鈕的上方。若要結束 **這是什麼** 模式，再按一下即可。若您經常需要此功能，可考慮啟用 **工具** → **選項** → **OpenOffice.org** → **一般** 中的 **延伸秘訣**。在此也可啟用或停用 **說明代辦程式** 與 **秘訣**。

OpenOffice.org 網站為 <http://www.openoffice.org>。您可以在網站中找到郵件清單、文章與問題資訊。此網站提供各種不同作業系統的版本，讓使用者下載。

Evolution: 電子郵件與行事曆程式

11

Evolution 是一個套裝軟體，除了提供一般的電子郵件功能以外，還提供工作清單、行事曆等擴充的功能。Evolution 也提供完整的通訊錄功能，包括以 vCard 格式將連絡人資訊傳送給其他人。

請從主功能表或使用 `evolution` 來啟動 Evolution。第一次啟動時，Evolution 會提供組態助理。其使用方法描述於 [節 11.3.1, "設定帳戶" \[151\]](#) 中。

重要: Microsoft Exchange 帳戶

若要在 Microsoft Exchange 使用 Evolution，您必須安裝 `ximian-connector` 套件。使用 YaST 進行安裝。

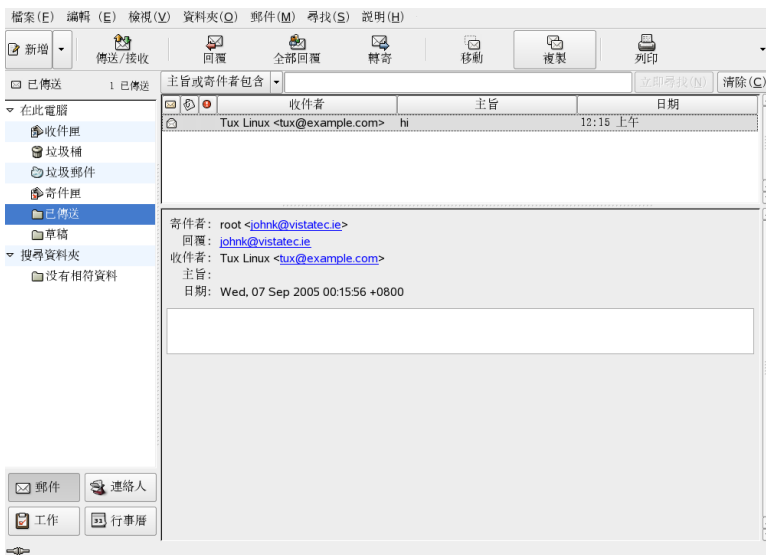
11.1 從其他郵件程式輸入電子郵件

若要從其他電子郵件程式 (例如 Netscape) 輸入電子郵件，請選取 **檔案** → **輸入**。若為 mbox 格式，請選取 **輸入單一檔案**。若為 Netscape，請選取 **從舊程式輸入資料和設定值**。如果要使用 maildir 格式的程式資料 (例如 KMail)，請設定一個存取該郵件目錄的帳戶。

11.2 Evolution 綜覽

預設視窗檢視顯示在 圖形 11.1, "含有郵件的 Evolution 視窗" [150] 中。可用的功能表、功能表項目以及工具列中的圖示，會隨開啟的元件而不同。請使用左框架來選取要顯示在右框架中的資訊。拖曳分隔列可調整框架的大小。

圖形 11.1 含有郵件的 Evolution 視窗



11.2.1 郵件

在此檢視中，上半部視窗會顯示目前資料夾的內容。下半部視窗為預覽窗格，用來顯示所選取的郵件訊息。若要顯示不同的資料夾，請在左框架的資料夾清單中選取一個資料夾。

使用搜尋列來搜尋資料夾中的訊息。若要按表格標題排序訊息，請按一下要做為排序依據的標題。向右箭頭會顯示要按照遞增或遞減順序來排序欄位。按一下欄位標題，使訊息按照您想要的方式來進行排序。

11.2.2 連絡人

此檢視顯示通訊錄中的全部地址。若要尋找特定的地址，請使用搜尋列，或按一下右方顯示連絡人姓氏第一個字母的按鈕。使用工具列新增連絡人或清單。

11.2.3 行事曆

啟始顯示包含目前日期的檢視，且右方窗格中會顯示月份與工作清單。可從工具列或檢視功能表，來使用週、工作週與月檢視。使用搜尋列可尋找已輸入行事曆的約會。使用工具列中的按鈕來新增約會與工作。您也可使用工具列逐頁瀏覽行事曆，或是跳至特定的日期。

11.2.4 工作

工作提供工作清單。選取的工作詳細資料，顯示在視窗的下半部。請使用 檔案 → 新增 → 工作 來新增工作。使用搜尋列來搜尋工作。在工作上按一下滑鼠右鍵然後選取 指定工作，可將工作指定給其他人。開啟此工作可用來新增更多詳細資料，例如截止日期及完成狀態。

11.3 郵件

Evolution 郵件元件可以各種格式處理多個帳戶。它提供有用的功能，例如，用來顯示搜尋結果和過濾垃圾郵件的虛擬資料夾。請在 編輯 → 優先設定 進行應用程式設定。

11.3.1 設定帳戶

Evolution 可以從多個郵件帳戶擷取電子郵件。撰寫訊息時，您可以選取要用來傳送電子郵件的帳戶。請在 編輯 → 優先設定 → 郵件帳戶 設定郵件帳戶。若要修改現有組態，請選取它再按一下 編輯。若要刪除帳戶，請選取帳戶，然後按一下 刪除。

若要新增帳戶，請按一下 新增。組態助理將會開啟。按一下 下一步 開始使用。在相關欄位中輸入您的名稱與電子郵件地址。如有需要可輸入選擇性的資訊。

請選取 *將此設為我的預設帳戶*，將此帳戶設成撰寫郵件時使用的預設帳戶。按一下 *下一步*。

在 *伺服器類型* 中，選取適合此地址的內送電子郵件格式。若要從遠端伺服器下載郵件，*POP* 是最常見的格式。*IMAP* 適用於特殊伺服器的郵件資料夾。請向您的 ISP 或伺服器管理員洽詢此項資訊。選取伺服器類型後，完成所顯示的相關欄位。完成後按一下 *下一步*。請選取想要的 *接收選項* (如果有的話)。按一下 *下一步*。

接著，請設定電子郵件傳送選項。若要傳送外寄電子郵件至本地系統，請選取 *Sendmail*。若要使用遠端伺服器，請選取 *SMTP*。請向您的 ISP 或伺服器管理員取得相關詳細資訊。若是選取「*SMTP*」，請完成所顯示的其他相關欄位。完成後按一下 *下一步*。

根據預設，電子郵件地址可做為識別帳戶的名稱。若有需要，可輸入其他名稱。按一下 *下一步*。按一下 *套用* 來儲存帳戶組態。

若要將帳戶設為傳送電子郵件的預設帳戶，請選取所要的帳戶，然後按 *預設*。若要停止從某一帳戶取得電子郵件，請選取該帳戶，然後按一下 *停用*。停用的帳戶仍可用來做為傳送電子郵件的地址，但是不會再檢查此帳戶的內送電子郵件。若有需要，可使用 *啟動* 來重新啟動帳戶。

11.3.2 建立訊息

若要撰寫新訊息，請按一下 *新增* → *郵件訊息*。回覆或轉寄訊息時，開啟相同的訊息編輯器。在 *寄件者* 旁，選取傳送訊息所用的帳戶。在收件者欄位中，輸入電子郵件地址，或是通訊錄中部分的名稱與地址。如果 Evolution 可以在通訊錄中找到與您輸入相符的內容，會顯示選擇清單。接著，按一下要寄送的連絡人；如果通訊錄中找不到任何相符的內容，則需完成輸入內容。若要直接從通訊錄中選取收件者，請按一下 *收件者* 或 *副本*。

Evolution 可以傳送純文字或 HTML 的電子郵件。若要設定 HTML 郵件格式，請在工具列中選取 *格式*。要傳送附件，請選取 *附加* 或 *插入* → *附件*。

若要傳送訊息，請按一下 *傳送*。若尚未準備立即傳送訊息，請在 *檔案* 下變更為其他選項。例如，將訊息儲存為草稿，或稍後再傳送訊息。

11.3.3 加密電子郵件與簽名

Evolution 支援 PGP 電子郵件加密。它可在電子郵件上簽名，並檢查經過簽名的電子郵件訊息。若要使用這些功能，請使用外部應用程式 (如 gpg 或 KGpg) 來產生與管理金鑰。

若想在傳送前，於電子郵件訊息上簽名，請選取 **安全性** → **PGP 簽名**。當您按一下 **傳送** 時，會出現對話方塊，提示您輸入私密金鑰的密碼。請輸入密碼，按一下 **確定** 以結束對話方塊，並傳送已簽名的電子郵件。若要在此工作階段的其他電子郵件上簽名，但不想重覆「解除鎖定」私密金鑰，請核取 **此工作階段後續皆記住此密碼**。

當您收到其他使用者簽名的電子郵件時，在訊息結尾處會顯示一個小掛鎖圖示。如果按一下這個符號，Evolution 將啟動外部程式 (gpg) 來檢查簽名。若簽名有效，則在掛鎖符號旁會顯示綠色核取記號。若簽名無效，則會顯示損毀的掛鎖。

加密與解密電子郵件非常容易。撰寫電子郵件訊息後，請至 **安全性** → **PGP 加密**，並傳送電子郵件訊息。當您收到經過加密的訊息時，接著出現的對話方塊會要求您輸入私密金鑰的密碼。請輸入通關密語以解密電子郵件訊息。

11.3.4 資料夾

最方便的作法是將電子郵件分類至數個不同的資料夾。您的資料夾目錄樹會顯示在左邊框架中。若經由 IMAP 存取郵件，則此資料夾列中也會顯示 IMAP 資料夾。至於 POP 與其他大部分的格式，資料夾會儲存在本地端，在 **本地端資料夾** 下分類排列。

預設會包含數個資料夾。**收件匣** 為從伺服器取得新訊息後最初存放的地點。**寄件備份** 用來儲存已送出電子郵件訊息的複本。**寄件匣** 會提供暫存空間，供您存放未寄出的電子郵件。若是離線工作，或是暫時無法連線外寄郵件伺服器，寄件匣便相當有用。**草稿** 可用來儲存尚未完成的電子郵件訊息。**垃圾筒** 資料夾用來暫存已被刪除的項目。**垃圾郵件** 提供 Evolution 垃圾郵件過濾功能。

新資料夾可建立在 **在此電腦上** 之下，或是做為現有資料夾的子資料夾。您可以依照本身的需求，建立複雜的資料夾階層。若要建立新資料夾，請選取 **檔案** → **新增** → **郵件資料夾**。請在「郵件資料夾」對話方塊中，輸入新資料夾的名稱。使用滑鼠來決定新資料夾的上層資料夾。使用 **確定** 結束對話方塊。

若要移動訊息至某個資料夾，請選取要移動的訊息。按一下滑鼠右鍵以開啟內容功能表。請選取 **移至資料夾**，接著會開啟目的資料夾的對話方塊。按一下 **確定** 以移動訊息。在原始資料夾中的訊息題將會顯示刪除線，代表訊息已標示為從該資料夾中刪除。訊息已儲存在新資料夾中。您也可以用類似的方法來複製訊息。

手動將多個訊息移至不同的目錄，可能需要花很多時間。過濾器可自動執行此程序。

11.3.5 過濾器

Evolution 提供一些過濾電子郵件的選項。過濾器可用來將訊息移至特定的資料夾，或是刪除訊息。利用過濾器，也可直接將訊息移至垃圾筒。建立新過濾器時，有兩種選項：從頭開始建立過濾器，或是根據要過濾的訊息來建立過濾器。若要過濾送至郵件清單的訊息，後者非常有用。

設定過濾器

選取 **工具** → **過濾器**。此對話方塊列示現有的過濾器，以便您編輯或刪除。按一下 **新增** 以建立新過濾器。另外，若要按訊息來建立過濾器，請選取訊息，然後選取 **工具** → **從訊息建立過濾器**。

在 **規則名稱** 中輸入新過濾器的名稱。選取過濾器所要使用的標準。可用的選項包括寄件者、收件者、來源帳戶、主旨、日期與狀態。在 **包含** 下拉式方塊中提供各類選項，例如，「**包含**」、「**是**」與「**不是**」。選擇適當的條件。輸入要搜尋的文字。按一下 **新增** 來新增更多的過濾器標準。使用 **執行動作**，決定必須符合所有標準或部分標準，才會套用過濾器。

在下半部視窗中，決定符合過濾器標準時，所要執行的動作。例如，可將訊息移至或複製到資料夾，或指定特殊的色彩。移動或複製訊息時，請按一下要選取的目的資料夾。在顯示的資料夾清單中，選取資料夾。若要建立新的資料夾，請按一下 **新增**。選取正確的資料夾後，按一下 **確定**。過濾器建立完成後，按一下 **確定**。

套用過濾器

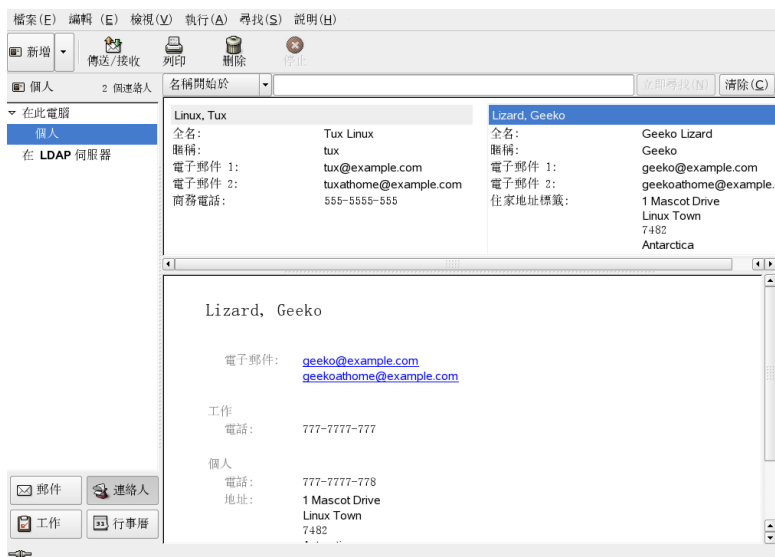
根據 **工具** → **過濾器** 開啟的對話方塊中所列示的順序來套用過濾器。若要變更順序，請反白選取過濾器，按一下 **向上** 或 **向下** 按鈕。完成後，按一下 **確定** 以關閉過濾器對話方塊。

過濾器將套用至所有新郵件訊息。過濾器不會套用至已存在您資料夾中的郵件。若要將過濾器套用至已接收的訊息，請選取要套用的訊息，然後選取 **動作** → **套用過濾器**。

11.4 連絡人

Evolution 可以使用多種不同的通訊錄。可用的通訊錄列在左邊框架中。請使用搜尋列來搜尋特定的連絡人。請使用 **檔案** → **輸入** 以多種格式新增連絡人至 Evolution 通訊錄。在連絡人上按一下滑鼠右鍵，將會開啟功能表，可在其中選取各種不同的選項，例如，轉寄連絡人或以 vCard 儲存連絡人。在連絡人上按兩下，可編輯該連絡人。

圖形 11.2 Evolution 通訊錄



11.4.1 新增連絡人

除了名稱和電子郵件地址外，Evolution 還可以儲存連絡人的其他地址與連絡資訊。若要快速新增寄件者的電子郵件地址，請在訊息預覽中以滑鼠右鍵按一下標示的地址即可。若要輸入完整的新連絡人，請在 *連絡人檢視* 中按一下 *新增連絡人*。兩種方法都會開啟對話方塊，可在其中輸入連絡人資訊。

在 *連絡人索引標籤* 中，請輸入連絡人名稱、電子郵件地址、電話號碼與即時訊息身份。*個人資訊* 是用來記錄網址和其他詳細資訊。請在 *郵件地址* 輸入連絡人的其他地址資訊。輸入所要的連絡人詳細資料後，請按一下 *確定*，將它加到通訊錄中。

11.4.2 建立清單

如果您經常傳送電子郵件訊息給一群人，您可以建立一個包含這些地址的清單，以簡化您的傳送程序。請按一下 *檔案* → *新增* → *連絡人清單*。接著會開啟連絡人清單編輯器。輸入清單的名稱。在方塊中輸入地址並按一下 *新增* 以便新增地址，或從 *連絡人檢視* 中將連絡人拖放至方塊中。切換 *隱藏地址* 以選取收件者是否能查看郵件的其他收件者。完成後，按一下 *確定*。清單現在會成為您的連絡人之一，當您在撰寫視窗中輸入清單的前幾個字母時，便會顯示清單。

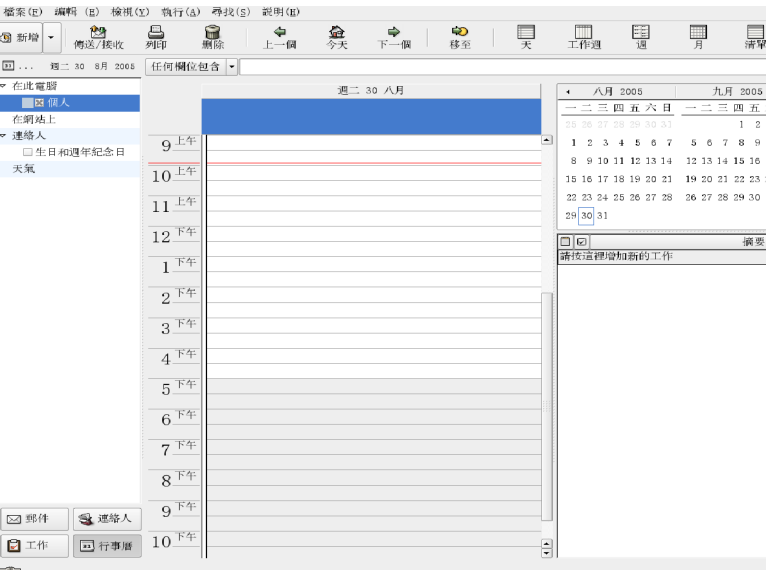
11.4.3 新增通訊錄

在該帳戶的帳戶組態中設定其他 GroupWise 和 Exchange 通訊錄。若要新增其他本地或 LDAP 通訊錄，請選取 *檔案* → *新增* → *通訊錄*。在開啟的對話方塊中，選取通訊錄類型並輸入必要的資訊。

11.5 行事曆

Evolution 可以和多個行事曆一起作業。請使用 *檔案* → *輸入* 來輸入 iCalendar 格式的行事曆。請使用行事曆來輸入約會以及排定會議。如有需要，您可以設定提醒功能，在您排定的約會即將開始前提醒您。

圖形 11.3 Evolution 行事曆



11.5.1 新增約會

若要在行事曆新增約會，請按一下 **檔案** → **新增** → **約會**。在 **約會** 索引標籤下，輸入約會的詳細資訊。若有需要，請選取類別以方便日後搜尋與排序。另外，您可以使用 **提醒** 選項來設定提醒功能，之後，**Evolution** 會在約會開始前提醒您。若要定期舉行的約會，您可以在 **週期** 索引標籤下設定定期舉行的日期。完成所有設定後，按一下 **確定**。新約會將顯示於行事曆中。

11.5.2 排程會議

若要安排會議，請選取 **檔案** → **新增** → **會議**。輸入您約會的相關資訊。請在 **邀請** 或 **排程** 中輸入出席者。若要從通訊錄輸入出席者，請使用 **連絡人** 來開啟通訊錄中的連絡人清單。**排程** 可以用來排定符合所有出席者的時間。設定好參與者後，請按 **自動選擇** 來自動尋找時間。

11.5.3 新增行事曆

您應在帳戶組態中設定 GroupWise 和 Exchange 行事曆。若要新增其他本地或網路行事曆，請選取 **檔案** → **新增** → **行事曆**。請選取想要的類型並輸入必要的資訊。

11.6 與掌上型裝置同步化資料

Evolution 的設計可讓資料和掌上型裝置同步，例如 Palm。同步使用 GNOME Pilot。請選取 **工具** → **Pilot 設定** 來開啟組態精靈。請參閱說明來取得詳細資訊。

11.7 用於 GroupWise 使用者的 Evolution

GroupWise 使用者在使用 Evolution 存取 GroupWise 帳戶時應該不會有太大的問題。因為 Evolution 和 GroupWise 使用的術語非常相似，所以熟悉其中一個系統的使用者，應該可以不費吹灰之力學會操作另一個系統。

11.7.1 設定 Evolution 以存取 GroupWise 系統

使用「Evolution 郵件組態助理」可設定 Evolution 存取 GroupWise 系統。若要啟動「Evolution 郵件組態助理」，請按一下 **偏好設定** → **郵件帳戶** → **新增**，接著按一下 **下一步**。

在「識別」頁面上輸入您在 GroupWise 系統中的電子郵件地址 (例如，joe@example.com)，接著按一下 **下一步**。

在「接收郵件」頁面的「伺服器類型」清單中選取 **IMAP**、在「主機」欄位中指定 GroupWise 伺服器的主機名稱、在「接收選項」頁面中設定其他適用您系統的設定值，接著按一下 **下一步**。

在「傳送郵件」頁面的「伺服器類型」清單中選取 *SMTP*、在「主機」來為中指定 **GroupWise** 伺服器的主機名稱、設定其他適用您系統的「傳送郵件」選項，接著按一下 下一步。

在「帳戶管理」頁面上，指定您要在「**Evolution** 設定」頁面上識別此帳戶的名稱，接著按一下 下一步。

按一下 套用即可建立 **GroupWise** 帳戶。接著，您的 **GroupWise** 信箱會出現在可用的電子郵件帳戶清單中。

11.8 若需詳細資訊

Evolution 提供詳盡的內部說明頁面。請使用 說明功能表來存取此資訊。如需有關 **Evolution** 的詳細資訊，請參閱專案網站 <http://www.gnome.org/projects/evolution/>。

Kontakt:電子郵件與行事曆程式

Kontakt 會將一些 KDE 應用程式的功能結合至個人資訊管理的便利單一介面中。這些應用程式包含電子郵件的 KMail、行事曆的 KOrganizer、通訊錄的 KAddressbook 以及備忘錄的 KNotes。您也可以與外部裝置同步化資料，例如掌上型裝置或其他裝置。Kontakt 能輕易地與其餘的 KDE 桌面整合，並與各種群組軟體伺服器連線。它包含額外的功能，例如垃圾郵件和病毒過濾，以及 RSS 閱讀程式。

從辦公室 → *Kontakt* (個人資訊總管) 啟動 Kontakt。或者，在指令行中輸入 `kontakt`。如果只需某些功能，您也可以開啟獨立的元件而不是組合的應用程式。

12.1 從其他郵件程式輸入電子郵件

若要從其他應用程式輸入電子郵件，請從 Kontakt 中的郵件檢視選擇 *工具* → *輸入訊息*。目前提供 Outlook Express、mbox 格式、電子郵件文字格式、Pegasus 郵件、Opera、Evolution 和其他的輸入過濾器。重要的公用程式也可以用指令 `kmailcvt` 來個別啟動。

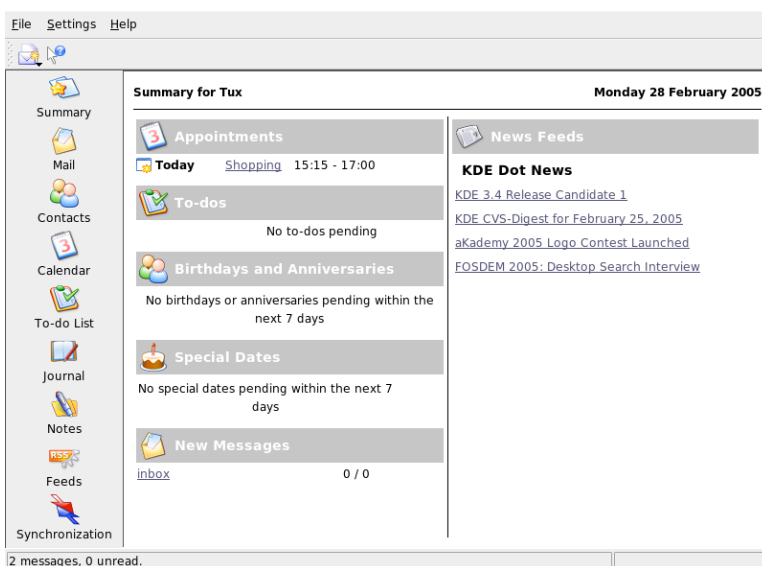
請選擇對應的應用程式，並按一下 *繼續* 加以確認。視選取的類型而定，必須提供檔案或資料夾。Kontakt 接著會完成處理程序。

12.2 Kontact 概觀

顯示摘要的預設視窗檢視，會顯示在 [圖形 12.1, "顯示摘要的 Kontact 視窗" \[162\]](#)。請使用左邊部分的按鈕以存取不同的元件。

摘要提供基本的資訊，如即將來臨的生日、待辦事項、天氣和 Kpilot 的狀態。可以存取 RSS 的新聞區段會提供您有興趣的新聞。請使用 [設定](#) → [設定摘要檢視](#) 來設定要顯示的資訊。

圖形 12.1 顯示摘要的 Kontact 視窗



12.2.1 郵件

左方的資料夾區域包含郵件資料夾清單 (信箱)，指出總訊息數和尚未讀取的訊息數。若要選取資料夾，僅需按一下即可。在此資料夾中的訊息會顯示在右上方框架中。此資料夾的訊息數也會顯示在應用程式視窗底部的狀態列中。

每一封訊息的主旨、寄件者和接收時間會列在右方的標題區域中。按一下訊息即可選取，並在訊息視窗中顯示。按一下其中一欄的標題 (主旨、寄件者、日期等) 便可排序訊息。目前選取訊息的內容會顯示在視窗的訊息框架中。附件會根據其 MIME 類型，在訊息結尾處以圖示顯示，或以插入的方式顯示。

可以使用不同的狀態旗標來標示訊息。要變更狀態可使用 **訊息** → **標示訊息**。您可以使用這個功能只訂訊息的狀態，例如重要或是忽略。例如，您可以反白選取想要記住的重要訊息。使用在搜尋列中的 **狀態** 僅會顯示有某種狀態的訊息。

12.2.2 連絡人

此元件的左上窗格會顯示目前啟動的通訊錄中的所有位址。左下窗格會列出您的通訊錄，並顯示通訊錄目前是否啟動中。右窗格會顯示目前選擇的連絡人。請使用最上面的搜尋列來尋找特定的連絡人。

12.2.3 待辦事項清單

待辦事項清單 顯示您的工作清單。按一下最上面的欄位以新增項目至清單中。在目前項目的欄位中按一下滑鼠右鍵，以變更該欄位的值。一個項目可以分成多個子項目。按一下滑鼠右鍵並選擇 **新增子待辦事項**，以建立子項目。也可以指定待辦事項給其他的人。

12.2.4 行事曆

行事曆檢視可分為一些窗格。根據預設，請檢視本月份的小型行事曆以及本週的一週檢視。也可以看到待辦事項清單、目前事件或待辦事項的詳細檢視以及所有狀態的行事曆清單。請從工具列或 **檢視功能表** 中選擇不同的檢視方式。

12.2.5 記事

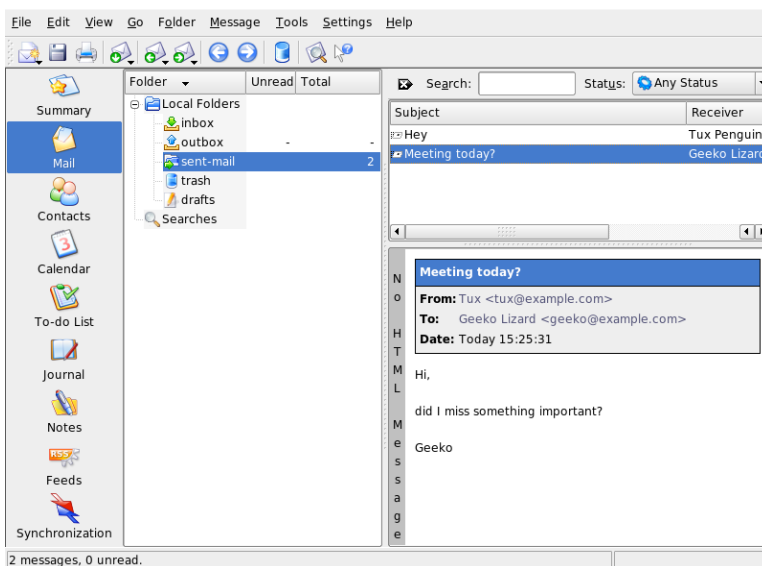
請使用「記事」元件來記錄容易忘記的事情。如果您正在使用 KDE，請使用系統夾中的 KNote 圖示來讓記事可以在桌面顯示。

12.3 郵件

Kontakt 會使用 KMail 做為其電子郵件元件。若要設定它，請開啟郵件元件，然後選擇 **設定** → **設定 KMail**。KMail 是功能完整的電子郵件用戶端，且支援一些通訊協定。工具包含許多實用的工具，可管理不要接收的電子郵件。請使用 **尋找** 來進行詳細的訊息搜尋。**反垃圾郵件精靈** 可以幫助管理工具過濾不想要的商

業電子郵件。掃毒精靈可以幫助管理電子郵件病毒掃描。這兩個精靈會使用外部的垃圾郵件和病毒軟體。如果這些選項停用，請安裝其他保護免受垃圾何病毒侵襲的套件。

圖形 12.2 Kcontact 郵件元件



12.3.1 設定帳戶

KMail 可以管理數個電子郵件帳戶，例如您的私人電子郵件地址和商務地址。在撰寫電子郵件時，請按一下 檢視 → 身份 選擇一個之前定義的辨識身份。若要建立新的身份設定檔，請選取 設定 → 設定 KMail，然後 身份 → 新增。在開啟的對話方塊中，請設定新身份的名稱，例如「私人」或「辦公室」。按一下 確定 以開啟對話方塊，可在其中輸入額外資訊。您也可指定身分到資料夾，回覆該資料夾中的訊息時，會選取指定的身分。

在 一般 索引標籤之下，輸入您的名稱、組織和電子郵件地址。請在 加密 下，選取您的金鑰，以便傳送經過數位簽名或加密的訊息。若要讓加密功能可正常運作，首先必須以 KGpg 建立金鑰，請參閱 章 6, 使用 KGpg 加密 [95]。

在 進階 下，您可輸入回覆與密件副本地址、選擇字典、選取草稿與寄件備份訊息的資料夾，以及定義應該如何傳送訊息。在 簽名 下，請決定每一封訊息是否

以結尾的一小段額外文字來簽名及如何簽名。例如，您可以在每封電子郵件簽名並附上聯絡資訊。若要啟用此功能，請選取 *啟用簽名*，並決定是否從檔案取得簽名、來自輸入欄位，或是指令的輸出。完成所有的身份設定後，請按一下 *確定* 以確認設定。

在 *網路* 下的設定會決定 *Kontact* 如何接收和傳送電子郵件。有兩個索引標籤，一個是傳送郵件，另一個則是接收郵件。這些設定會因為您的郵件伺服器所在的系統和網路的不同而異。如果您不確定要選取哪些設定或項目，請洽詢您的 ISP 或系統管理員。

若要建立建立外送信箱，按一下 *傳送* 索引標籤的 *新增*。選擇傳輸類型 *SMTP* 或是 *sendmail*。大部分情況下，應選擇 *SMTP*。選擇好之後，會出現一個視窗，可在其中輸入 *SMTP* 伺服器資料。輸入名稱與伺服器位址 (由 ISP 提供)。若伺服器要求驗證，請啟用 *伺服器要求驗證*。安全性設定是在 *安全性* 索引標籤之下。在此處指定您偏好的加密方法。

在 *接收* 索引標籤之下進行接收電子郵件的設定。使用 *新增* 以建立新帳戶。在不同取得郵件方法中選擇一項，例如本地 (儲存在 *Mbox* 或 *Maildir* 格式)、*POP3* 或 *IMAP*。讓此設定適合您的伺服器。

12.3.2 建立訊息

若要撰寫新訊息，請選擇 *訊息* → *新增訊息* 或按下工具列上對應的圖示。若要以不同電子郵件帳戶傳送訊息，請選取如 [節 12.3.1, "設定帳戶" \[164\]](#) 中所述的其中一個身份。在 *收件者* 中，輸入電子郵件地址，或是通訊錄中部分的名稱與地址。若 *Kontact* 可將您輸入的文字配對至通訊錄中的記錄，將會開啟選擇清單。按一下要寄送的連絡人；若在通訊錄中找不到任何相符的內容，則需完成您的輸入。若要直接從通訊錄中選取，請按一下「地址」欄位旁的 ... 按鈕。

若要在訊息附加檔案，請按一下迴紋針圖示並選取要附加的檔案。或者，從桌面或另一個資料夾將檔案拖曳到 *新增訊息* 視窗或在 *附加* 功能表中選取一個選項。通常，可正確地辨識檔案類型。如果無法辨認格式，請在該圖示上按一下滑鼠右鍵。從顯示的功能表中，選取 *內容*。請在下一個對話方塊設定格式和檔案名稱，並新增說明。此外，您可以決定是否要簽名或加密附加檔案。

在您撰寫完訊息之後，請按一下 *訊息* → *傳送* 立即傳送訊息，或按一下 *訊息* → *佇列* 將訊息移到寄件匣。如果傳送訊息，在成功傳送訊息後，會將訊息複製到 *sent-mail* 中。移動至 *寄件匣* 的訊息可以編輯或刪除。

12.3.3 加密電子郵件與簽名

若要加密電子郵件，首先要產生 [章 6, 使用 KGpg 加密 \[95\]](#) 中所述的金鑰組。若要設定加密程序的細節，請選取 **設定** → **設定 KMail** → **身份**，指定傳送加密和簽名訊息所用的身份。然後按 **修改**。按一下 **確定** 確認後，金鑰將顯示在對應的欄位中。按一下 **確定** 以關閉組態對話方塊。

12.3.4 資料夾

訊息資料夾可協助您組織訊息。按照預設狀況，他們會放在 `~/.kde/share/apps/korganizer` 目錄。第一次啟動 KMail 時，程式會建立數個資料匣。「收件匣」為從伺服器取得新訊息後最初存放的地點。「寄件匣」用來暫存送出佇列的訊息。「寄件備份」為已傳送訊息的備份。「垃圾筒」包含所有按 **Del** 鍵或選取 **編輯** → **刪除** 刪除掉的電子郵件副本。草稿 是儲存尚未完成訊息的地方。如果您正在使用 IMAP，IMAP 資料匣會列在本機資料匣下面。每個內送郵件伺服器，例如本機或 IMAP，在「資料夾」清單中都包含自己的資料夾。

若需要額外資料夾來組織您的訊息，可選取 **資料夾** → **新增資料夾**，以建立資料夾。如此可開啟視窗，讓您在指定新資料夾的名稱和格式。

在資料夾上按一下滑鼠右鍵，開啟提供數個資料夾操作的內容功能表。按一下 **過期** 指定讀取和尚未讀取的過期日，以及過期後如何處理郵件，郵件過期後是否刪除或移動至其他資料夾。若要使用資料夾儲存來自郵件清單的訊息，可在 **資料夾** → **郵件清單管理** 設定所需的選項。

若要在資料夾之間移動一或多個訊息，請反白要移動的訊息，然後按 **M** 或選取 **訊息** → **移動至**。在出現的清單中，選取要將訊息移至的資料夾。另一個移動訊息的方式是從上方視窗拖曳訊息，然後放在左方視窗中適當的資料夾內。

12.3.5 過濾器

過濾器是一種自動處理內送郵件的便利方式。它們使用郵件的項目，例如寄件者或大小，來將郵件移到某個資料匣、刪除不想要的郵件、將信退回給寄件者或執行一些其他的動作。

設定過濾器

若要重頭開始建立過濾器，請選取 **設定** → **設定過濾器**。若要在現有的訊息中建立過濾器，請在「標題」清單選擇想要的訊息，然後選取 **工具** → **建立過濾器** 以及想要的過濾條件。

請為過濾條件選擇符合的方式 (全部或任何)。然後選擇僅套用至想要訊息的條件。在 **過濾器動作** 中，請設定過濾器應該對符合該條件的訊息做什麼。**進階選項** 可以控制何時要套用過濾器，以及是否為這些訊息使用其他的過濾器。

套用過濾器

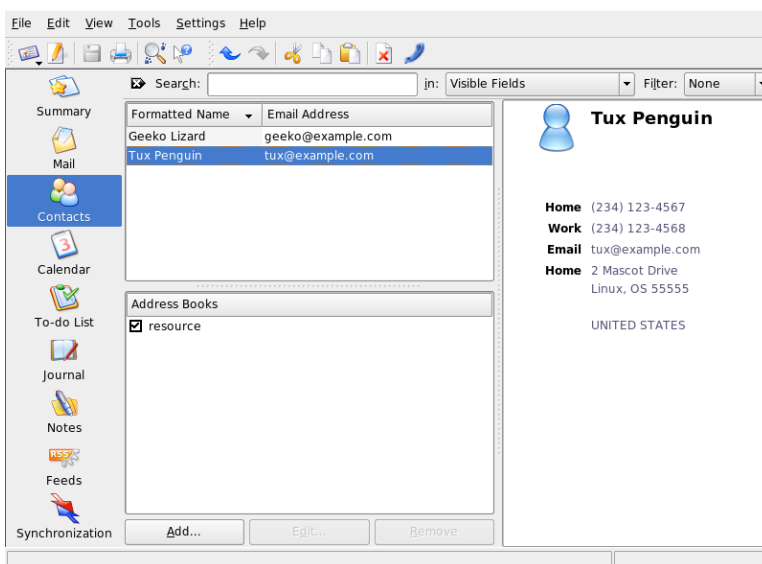
根據 **設定** → **設定過濾器** 開啟的對話方塊中所列示的順序來套用過濾器。選擇過濾器並按一下箭頭按鈕來變更順序。過濾器僅會套用到新內送的訊息，或依照過濾器的進階選項中所指定的方式來傳送訊息。若要將過濾器套用到現有的訊息，請選取要套用的訊息，然後選取 **訊息** → **套用過濾器**。

如果您的過濾器沒有如預期般運作，請選取 **工具** → **過濾器記錄檢視器** 來監控它們。如果啟用記錄功能時，此對話方塊中會顯示您的過濾器處理訊息的方式，並協助找出問題。

12.4 連絡人

連絡人元件會使用 KAddressBook。請選取 **設定** → **設定 KAddressBook** 來設定它。若要搜尋特定的連絡人，請使用搜尋列。按一下 **過濾器**，選擇僅在某個類別中顯示連絡人。在連絡人上按一下滑鼠右鍵開啟功能表，您可以從中選擇各種的選項，例如在電子郵件中傳送連絡人訊息。

圖形 12.3 *Kontakt* 通訊錄



12.4.1 新增連絡人

若要從電子郵件新增具有名稱及電子郵件地址的連絡人，請在郵件元件的地址處按一下滑鼠右鍵，並選取 *在通訊錄中開啟*。若不使用電子郵件新增連絡人，請在地址元件中選取 *檔案* → *新增連絡人*。兩種方法都會開啟對話方塊，讓您能輸入連絡人的資訊。

在一般索引標籤中，請輸入連絡人的基本資訊，例如姓名、電子郵件地址以及電話號碼。類別可以用來排序地址。*詳細資訊* 包含更多的特定資訊，例如生日和配偶姓名。

如果您的連絡人使用即時通訊程式，您可以將這些身份加到 *IM 地址* 中。如果您這樣做，且擁有 Kopete 或其他正在同時執行的 KDE 聊天程式，像 Kontakt，請檢視在 Kontakt 中關於這些身份的狀態資訊。在 *密碼設定* 中，請輸入連絡人的加密資料，例如公鑰。

其他有其他的資訊，例如照片以及該使用者的空間/忙碌資訊的位置。請使用 *自訂欄位* 將您自己的資訊加入連絡人或通訊錄中。

連絡人也可以以各種格式輸入。請使用 **檔案** → **輸入** 並選擇想要的格式。然後選擇要輸入的檔案。

12.4.2 建立配送清單

如果您經常寫電子郵件給相同的一群人，配送清單可讓您儲存多個電子郵件地址為一個單一的聯絡項目，如此一來，您不需在寄送電子郵件給群組時，輸入每個收件者的電子郵件地址。首先，按一下 **設定** → **顯示延伸工具列** → **配送清單編輯器**。在顯示的新區段中，按一下 **新增清單**。輸入清單的名稱，並按一下 **確定**。請從地址清單將連絡人拖曳到配送清單視窗中，來新增連絡人到清單中。建立電子郵件時，如同單一的聯絡人一般使用該清單。

12.4.3 新增通訊錄

重要：群組軟體通訊錄

新增群組軟體資源的最好方式便是使用 **Groupware Wizard** 的個別工具。若要使用這個工具，請關閉 **Kontakt** 並在指令列中輸入 `groupwarewizard` 或從 KDE 功能表的 **Office** 群組。請從提供的清單中選擇伺服器類型，例如 **SLOX**、**GroupWise** 或 **Exchange**，然後輸入位址和驗證資料。接著精靈會將可用的資源新增到 **Kontakt**。

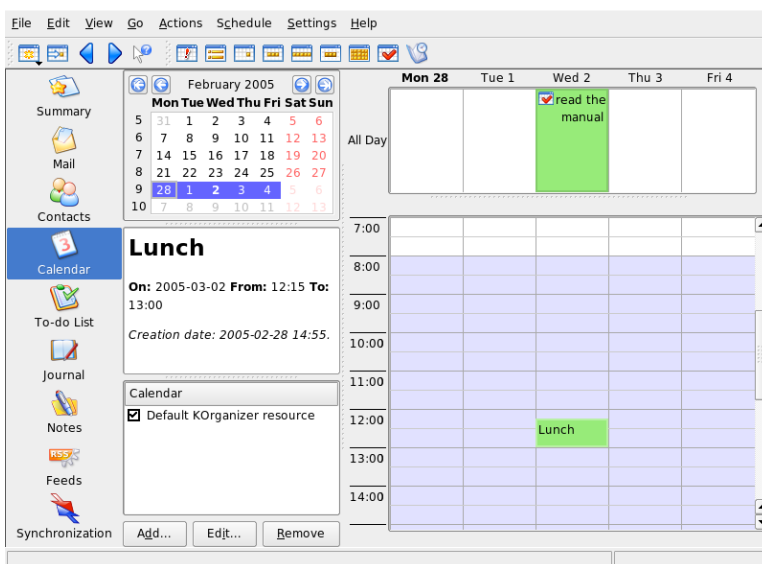
Kontakt 可以存取多個通訊錄，例如由 **Novell GroupWise** 或 **LDAP** 伺服器所提供的共用通訊錄。請選取 **設定** → **顯示延伸工具列** → **通訊錄** 以檢視現有的通訊錄。按下 **新增** 以新增一個通訊錄，然後選擇類型並輸入所需的資訊。

通訊錄前面的核取方塊會顯示每個通訊錄的啟動狀態。若要停止顯示通訊錄但不刪除它，請取消選取該核取方塊。移除 刪除清單中所選的通訊錄。

12.5 行事曆

Kontakt 使用 **KOrganizer** 做為其行事曆元件。若要設定它，請使用 **設定** → **設定 KOrganizer**。請使用行事曆輸入約會和與其他人見面的排程。如有需要，會提醒即將到來的約會。您也可以使用 **檔案** 中的選項來輸入、輸出及歸檔行事曆。

圖形 12.4 Kontact 行事曆



12.5.1 排程事件

可以選取 **動作** → **新增事件** 來新增新的事件或會議。輸入所要的詳細資訊。在 **提醒** 下，指定提醒出席者事件的確實時間 (在事件發生前的分、時或日)。若事件重覆發生，請指定適當間隔。要在行事曆特定時間點上建立事件的另一種方式，便是在程式的行事曆檢視之中，按兩下對應的欄位。如此所開啟的對話方塊視窗，與從功能表開啟的視窗相同。也可以在「行事曆」檢視按一下右鍵，選取時間範圍。

若要指定事件的出席者，請在對話方塊中手動輸入出席者的資料，或是從通訊錄中插入資料。若要手動輸入這些資料，請選取 **新增**。若要從通訊錄輸入資料，請按一下 **選擇位址**，然後從對話方塊中選取對應的項目。若要依據參與者的出席狀況來排程事件，請移至 **空間/忙碌** 並按一下 **選擇日期**。

請使用 **週期索引** 標籤來設定有規律性發生的事件。**附件** 可以輕鬆連結其他資訊與事件，例如會議的議程。

12.5.2 新增行事曆

重要：群組軟體行事曆

新增群組軟體資源的最好方式便是使用 Groupware Wizard 的個別工具。若要使用這個工具，請關閉 **Kontakt** 並在指令列中輸入 `groupwarewizard` 或從 KDE 功能表的 **Office** 群組。請從提供的清單中選擇伺服器類型，例如 **SLOX**、**GroupWise** 或 **Exchange**，然後輸入位址和驗證資料。接著精靈會將可用的資源新增到 **Kontakt**。

行事曆模組可以同時連線多個行事曆。這個功能非常的實用，例如連結個人行事曆與組織行事曆。若要新增行事曆，請按一下 **新增** 然後選擇行事曆類型。完成必須的欄位。

行事曆前面的核取方塊會顯示每個啟動狀態。若要停止顯示行事曆但不刪除它，請取消選取該核取方塊。移除 刪除清單中所選的行事曆。

12.6 與掌上型裝置同步化資料

設計 **Kontakt** 的目的就是要讓其資料與掌上型裝置同步化，例如 **Palm**。請在摘要中檢視關於 **KPilot** 狀態的資訊。請參閱 [章 13, 使用 KPilot 同步化掌上型電腦 \[175\]](#) 以取得設定和使用 **KPilot** 的相關資訊。

12.7 GroupWise 使用者的 Kontakt

如果習慣使用 **GroupWise**，您應該很快就可以對 **Kontakt** 上手。這兩個程式共享很多概念並提供許多相同的服務。這節將會討論明顯的術語變化，以及秘訣，供 **GroupWise** 使用者很快上手 **Kontakt**。

12.7.1 術語變化

下列表格列出 **Kontakt** 和 **GroupWise** 某些主要的術語變化。

表格 12.1 *Kontakt 和 GroupWise 術語變化*

GroupWise	Kontakt
約會	事件
忙碌搜尋	空間/忙碌
記事	日誌項目
已張貼，未張貼項目	沒有任何出席者張貼的事件。如果事件有出席者，則為「已傳送」項目。
工作	待辦事項

12.7.2 GroupWise 使用者的秘訣

這節將會討論幫助 GroupWise 使用者了解 GroupWise 和 Kontakt 的不同。

聯絡人資訊

您可以新增 GroupWise Messenger 和電子郵件聯絡人到 Kontakt 聯絡人資訊。然後可以在「聯絡人」檢視的聯絡人上按一下滑鼠右鍵，建立電子郵件或開啟即時訊息的作業。

色彩編碼

對 GroupWise 項目，以及其他來源的項目都非常有用。色彩編碼可輕易的掃描電子郵件、聯絡人和其他特定來源項目的其他資訊。

邀請出席者加入事件

不像 GroupWise，Kontakt 不會自動將您加入已排定事件의出席者。請記得邀請自己。

12.8 若需詳細資訊

Kontakt 包含本身和其不同元件的說明。請使用 *說明* → *Kontakt 手冊* 來存取。
此專案的網頁 <http://www.kontakt.org> 也具參考性。

使用 KPilot 同步化掌上型電腦

對於需要隨身管理排程、待辦清單與記事的使用者而言，掌上型電腦是廣為使用的工具。使用者經常需要將相同的資料存放在桌上型電腦與可攜式設備。KPilot 可同步化掌上型電腦的資料，以及屬於 Kontact 一部分的 KAddressBook、KOrganizer、KNotes 等 KDE 應用程式所用的資料。這正是 KPilot 流行的原因！

KPilot 的主要目的是讓掌上型電腦應用程式與相對的 KDE 應用程式之間，彼此共用資料。KPilot 具有內建的備忘錄檢視器、地址檢視器以及檔案安裝程式，但這些都無法在 KPilot 環境之外使用。除了檔案安裝程式之外，獨立的 KDE 應用程式也可提供上述所有功能。

為了能夠在掌上型電腦與不同的桌面程式之間通訊，KPilot 必須仰賴介質。KPilot 本身為管理兩部電腦設備之間資料交換的程式。要在桌上型電腦上使用掌上型電腦的特定功能，需要啟用與設定相關的介質。這些介質主要是針對與特定 KDE 程式互動而設計。因此，一般而言無法用於其他桌面應用程式。

時間同步介質相當特別，因為此介質並無使用者可見的程式。它會在每次同步作業時在背景啟動，但只有在使用網路時間伺服器來校正時間差的電腦上，才應啟用此介質。

開始進行同步作業時，介質將逐一啟用，以完成資料傳輸的工作。總共有兩種不同的同步方法：HotSync 作業僅同步化已啟用介質的資料；備份作業則是完整備份儲存在掌上型電腦的所有資料。

部分介質在同步作業期間會開啟檔案，這意味著在這段時間內不應執行對應的程式。特別是 KOrganizer，不應該在同步作業期間執行。

13.1 KPilot 所使用的介質

選取 **設定** → **設定 KPilot**，可啟用與設定 KPilot 所使用的介質。下列清單包含部分重要的介質：

通訊錄

此介質處理掌上型電腦通訊錄的資料交換。在 KDE 中管理這些連絡人的對應程式為 KAddressBook。請從主功能表或使用 `kaddressbook` 指令來啟動。

KNotes/備忘錄

此介質可讓您將 KNotes 所建立的記事，傳輸至掌上型電腦的備忘錄應用程式。請從主功能表或使用 `knotes` 指令來啟動 KDE 應用程式。

行事曆 (KOrganizer)

此介質負責同步化掌上型電腦的約會 (事件)。對應的桌面應用程式為 KOrganizer。

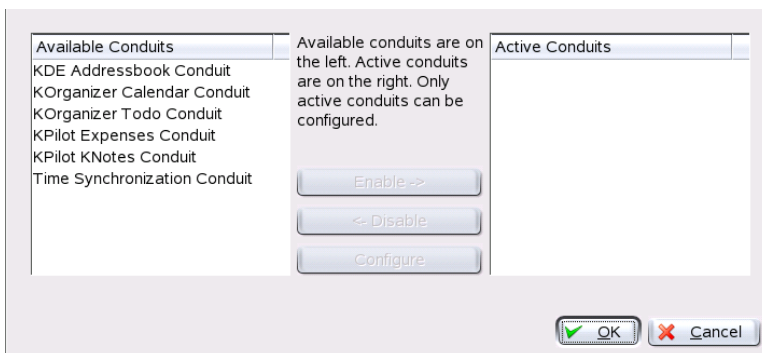
待辦 (KOrganizer)

此介質負責同步化待辦項目。對應的桌面應用程式為 KOrganizer。

時間同步化介質

啟用此介質可在每次同步化作業期間，將掌上型電腦的時間調整為與桌上型電腦相同。只有經常以時間伺服器來校正桌上型電腦的時間時，才適合使用此介質。

圖形 13.1 包含可用介質的組態對話方塊



13.2 設定掌上型電腦的連線

若要使用 KPilot，首先必須設定掌上型電腦的連線。組態取決於掌上型電腦所用的連接座類型 (銜接單位)。總共有兩種連接座類型：USB 連接座或纜線，與序列連接座或纜線。

13.2.1 從 KPilot 內設定連線

設定連線最簡單的方法，便是使用組態助理。選取 **設定** → **組態助理** 以啟動助理。在第一個步驟，輸入您的使用者名稱，以及掌上型電腦所連接設備的名稱。若您選取 **自動偵測掌上型電腦與使用者名稱**，助理會嘗試自行偵測。若自動偵測失敗，請參閱 [節 13.2.2, "建立 /dev/pilot 連結" \[177\]](#)。

按一下 **下一步** 確認後，助理將提示您指定用於同步化的應用程式。您可以選取 KDE 應用程式套件 (預設)、Evolution 或沒有可同步的應用程式。選取後，請利用 **完成** 關閉視窗。

13.2.2 建立 /dev/pilot 連結

序列掌上型電腦連接座的連線設定，有別於 USB 連接座的連線設定。依所用的連接座而異，您可能需要或不需要建立名為 `/dev/pilot` 的連結。

USB

一般而言，應可自動偵測出 USB 連接座，不需要建立上述的連結。

序列

若是使用序列連接座，您必須知道實際連接的序列埠。序列設備的命名方式為 `/dev/ttyS?`，從 `/dev/ttyS0` 開始為第一個連接埠。若要設定連接至第一個序列埠的連接座，請輸入下列指令：

```
ln -s /dev/ttyS0 /dev/pilot
```

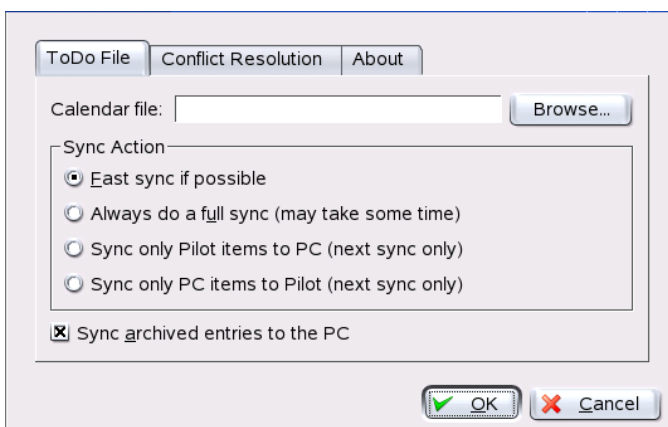
13.3 設定 KAddressBook 介質

起初，只需啟用 KAddressBook 介質，不需要變更任何預設值。在第一次同步化後，請設定下列細節：發生衝突的處理方式，備份資料庫的儲存方式，以及掌上型電腦上所儲存的特定欄位如何指派至 KAddressBook 所預期的欄位。

13.4 管理待辦項目與事件

在 KDE 桌面上，待辦項目 (工作) 與事件 (約會) 是由 KOrganizer 所管理。請從主功能表、使用 `korganizer` 指令，或是做為 Kontact 的一部分，來啟動應用程式。啟用 KPilot 的行事曆與待辦項目介質後，請在使用前設定部分組態選項。

圖形 13.2 KPilot 組態

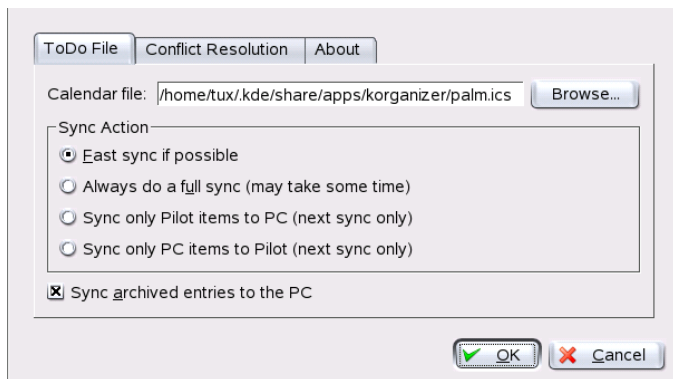


KOrganizer 將檔案儲存在 `~/.kde/share/apps/korganizer` 目錄。不過，由於 `.kde/` 目錄名稱是以點 (.) 為開頭，因此在檔案選取對話方塊中可能無法顯示。若發生此狀況，請在檔案選取對話方塊中，手動輸入完整的路徑，或明確切換隱藏檔案 (以點 (.) 開頭的檔案) 為顯示。此動作的預設捷徑為 `[F8]`。

開啟 `~/.kde/share/apps/korganizer` 目錄後，選取檔案以便 KOrganizer 用來做為行事曆檔案。在本例中，此檔案為 `palm.ics`。若使用者名稱為 `tux`，完整路徑與檔案名稱為 `/home/tux/.kde/share/apps/korganizer/palm`

.ics (如圖 圖形 13.3, "顯示 KOrganizer 行事曆檔案路徑的對話方塊" [179] 所示)。

圖形 13.3 顯示 KOrganizer 行事曆檔案路徑的對話方塊

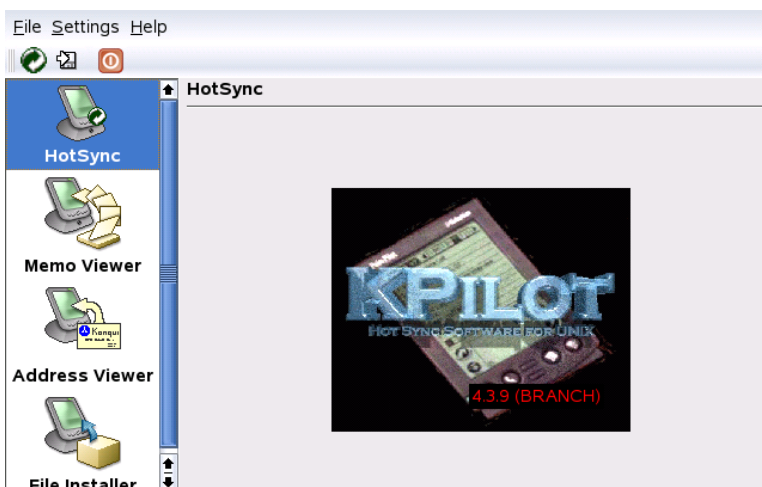


與掌上型電腦交換資料時，不應執行 KOrganizer。否則，KPilot 將無法執行同步化作業。

13.5 使用 KPilot

您可以輕易地同步化 KDE 應用程式與掌上型電腦的資料。只要啟動 KPilot，然後按下連接座或纜線的 HotSync 按鈕，便可開始執行同步化作業。

圖形 13.4 KPilot 的主視窗



13.5.1 備份掌上型電腦的資料

若要執行完整備份，請選取 檔案 → 備份。下一次同步化作業時將會執行備份。備份完畢後，請從功能表選取 檔案 → HotSync，以切換回原先的設定。否則，下一次同步化作業時仍然會執行耗時的完整備份。

執行完整備份後，掌上型電腦所有程式與資料庫的複本皆存放在 `~/.kde/share/apps/kpilot/DBBackup/使用者名稱`，其中的 使用者名稱 為註冊在掌上型電腦上的使用者名稱。

KPilot 內建的兩個檢視器可用來快速查看地址或備忘錄，但這兩個檢視器皆不適合用來實際管理這些資料。上述 KDE 應用程式比較適合用來從事這些工作。

13.5.2 在掌上型電腦上安裝程式

檔案安裝程式 模組是相當有趣也是很好用的工具，可用來安裝掌上型電腦的程式。這些程式的副檔名通常為 `.prc`，上傳至掌上型電腦後便立即可啟動。使用這類附加程式之前，請閱讀其授權與內附的說明。

13.5.3 同步化您的通訊錄與行事曆

若要同步化您的行事曆與地址，請使用 KDE 工具 MultiSynK。請以 `multisynk` 指令啟動工具。同步化資料前，請先建立 Konnector 組合。請至 檔案 → 新增，選取您的 Konnectors。利用 確定 即可離開。

名稱列示在主視窗。若要同步化您的掌上型電腦，請至 檔案 → 同步化。

使用 Beagle

Beagle 是一項會為個人資訊空間建立索引的搜尋工具，可協助您找出任何正在搜尋的資料。您可以使用 Beagle 來尋找文件、電子郵件、網站歷程記錄、即時訊息傳遞程式 Instant Messenge 和 ITC 對話、原始程式碼、影像、音樂檔案、應用程式，以及更多資料。

Beagle 支援下列資料來源：

- 檔案系統
- 應用程式啟動器
- Evolution 郵件和通訊錄
- Gaim 即時訊息日誌
- Firefox 網頁 (當您檢視時)
- Blam 與 Liferea RSS 彙總
- Tomboy 記事

它也支援下列檔案格式：

- OpenOffice.org
- Microsoft Office (doc、ppt、xls)
- HTML

- PDF
- 影像 (jpeg、png)
- 音效 (mp3、ogg、flac)
- AbiWord
- Rich Text Format (rtf)
- Texinfo
- man 頁面
- 原始程式碼 (C、C++、C#、Fortran、Java、JavaScript、Pascal、Perl、PHP、Python)
- 純文字

Beagle 會自動為主目錄的每項資料建立索引，但是您也可以選擇排除特定檔案或目錄，不為其建立索引。Beagle 也包括了各種可用來搜尋資料的工具。

14.1 索引資料

Beagle 精靈 (beagled) 會自動執行所有索引操作。根據預設，主目錄中的每項資料都會進行索引。Beagle 會偵測出主目錄的變更，並且為變更資料重新建立索引。

- 檔案一旦建立時就會立即建立索引，而且會在修改時重新建立索引，如果遭到刪除，就會從索引中移除。
- 電子郵件在送達時，就會馬上建立索引。
- 聊天時的每行 IM 對話也會逐行建立索引。

為資料建立索引會耗用相當數量的電腦資源，但是 Beagle 精靈會盡量不造成干擾。它所包含的排程器，可以依據工作站的使用情形來排列工作的優先順序，而且可以控制 CPU 用量。

14.1.1 避免檔案和目錄建立索引

如果您不希望目錄(及其所有子目錄)建立索引，請建立檔名為 `.noindex` 的空白檔案，並將此檔案放到該目錄中。您可以在此 `.noindex` 檔案中加入檔案和目錄的清單，避免這些檔案和目錄建立索引。`.noindex` 檔案可以使用萬用字元。

您也可以在主目錄中，放入 `.neverindex` 檔案和應該永遠不要建立索引的檔案清單。這個檔案也可以使用萬用字元。請使用在應用 `glob` 時的相同萬用字元使用方式(例如，`f*le???.txt`)。您也可以型式前後加入斜線，使用更有效的一般表示式(例如，`/file.*.txt/`)。如需更多詳細資訊，請參閱 `glob-UNIX` 網站(<http://docs.python.org/lib/module-glob.html>)內容。

14.1.2 手動建立索引

Beagle 所提供的有效系統，可以用來決定何時為檔案建立索引，而且會嘗試不干擾到正在執行的其他應用程式。它會特地根據負載量以及系統是否為閒置狀態，來估計索引建立時間，如此一來，就不會對桌面使用造成不良影響。但是，如果您希望立即為主目錄建立索引，請先在終端機視窗輸入下列指令，再執行 Beagle：

```
export BEAGLE_EXERCISE_THE_DOG=1
```

14.1.3 檢查索引狀態

Beagle 包括下列指令，供您用來檢查目前的索引狀態：

beagle-index-info

顯示有多少文件已經建立索引，以及哪些文件類型已經建立索引。

beagle-status

顯示目前 Beagle 精靈正在執行的工作(持續進行狀態)。

14.2 搜尋資料

Beagle 會提供下列工具，供您用來搜尋已經建立索引的資料。

14.2.1 Best

Best (Bleeding Edge Search Tool) 圖形工具，可以用來搜尋已經建立索引的所有資訊。Best 不會直接查詢索引；它會將搜尋詞彙傳到 Beagle 精靈，再由精靈將所有符合資料傳回給 Best。接著，Best 會轉譯結果，並允許您對符合物件執行動作。

如果要在 KDE 開啟 Best，請按一下「K 功能表」→「系統」→「檔案系統」→「Beagle 搜尋」。在 GNOME 中，按一下「應用程式」→「系統」→「檔案系統」→「Beagle 搜尋」。

如果要使用 Best，您只要在最上方的輸入方塊中輸入搜尋文字，再按 Enter 或是按一下「搜尋」。這樣 Best 就會查詢已經建立索引的檔案，並傳回結果。

圖形 14.1 Beagle 搜尋



您可以透過此時的結果清單，使用您的檔案管理員來開啟檔案、郵寄檔案、傳送即時訊息、重播檔案、轉遞檔案，或是顯示檔案。不同檔案類型會決定可以使用的選項。

您也可以使用「任何位置」，限制在例如通訊錄或網頁等特定位置中搜尋檔案，或是結果清單中只顯示特定類型檔案。

14.2.2 beagle-query

Beagle 會提供下列指令行工具，供您用來搜尋 Beagle 索引。如果要使用這項工具，請在終端機視窗中輸入下列指令：

```
beagle-query search
```

請將 *search* 替代成要搜尋的文字，接著 beagle-query 工具就會傳回結果。萬用字元可以配合這個指令使用。

使用 `beagle-query --verbose search`，顯示有關搜尋結果的詳細資訊。

部 V. 圖形

數位相機與 Linux

如果您有正確的工具，從相機來管理相片將變得輕鬆愉快。Linux 提供許多方便的公用程式供您排序或組織相片。這些公用程式包含 `gphoto2`、`Konqueror`、`Digikam` 和 `f-spot`。

完整的支援相機清單可在 <http://www.gphoto.org/proj/libgphoto2/support.php> 中取得。如果已經安裝 `gphoto2`，請使用指令 `gphoto2 --list-cameras` 來擷取清單。請使用 `gphoto2 --help` 來取得可用指令的相關資訊。

提示: 不支援的相機

如果您從 `gphoto` 的清單中找不到您的相機，請不要失望。您的相機非常有可能是由 USB 大量儲存設備所支援。您可以在 [節 15.2, "存取相機" \[192\]](#) 找到更多的資訊。

15.1 連接至相機

連接數位相機至電腦的最便捷的方法是透過 USB (如果核心、相機與電腦皆支援的話)。標準的 SUSE kernel 提供這項支援。適合的纜線也是必要的。

只要連接相機至 USB 埠並開啟相機即可。您可能需要切換相機至專用的資料傳輸模式。關於這個程序，請參閱您數位相機的手冊。

15.2 存取相機

有三種方法可以存取相機上的相片。它取決於相機以及相機支援的協定。通常它是 USB 大量儲存，可由 hotplug 系統或 PTP (又稱 PictBridge) 處理。某些相機型號不適用於任一通訊協定。為了支援這些相機型號，gphoto2 包含特定的驅動程式。

如果您的相機支援 USB 大量儲存，就最簡單不過了。如果您不確定，請閱讀相機的說明文件。有些則可支援兩種協定，例如，PTP 和 USB 大量儲存。不幸的是，有些使用的是專屬通訊協定，使得作業更為複雜。如果您的相機不支援 USB 大量儲存或 PTP，下列描述並不適用。請嘗試使用 `gphoto2 --list-cameras` 以及 <http://www.gphoto.org/> 中列出的資訊。

如果您的相機可以切換為 USB 大量儲存設備，請選取這個選項。當您連到電腦的 USB 埠並打開電源，hotplug 系統可以偵測到它。這會自動裝載裝置，因此非常方便。裝置成功後，KDE 桌面會顯示相機圖示。

相機裝置成功後，`/media` 下會看到一個以 `usb` 及一些數字開頭的新目錄。每一個廠商和產品都有一個號碼，因此您連接設備到電腦時，該設備會有相同名稱。請依據 USB 匯流排連接的內容來尋找其他項目。剩下的唯一問題是找出和相機對應的正確項目。請試著列示這些目錄的其中一種 (`DCIM/xxx`) 並查看結果。每一個相機都有不同的樹狀結構，沒有共通的規則。如果您在某個目錄中看到 JPEG 檔，那麼您可能找到了。

找到正確目錄後，您可以使用檔案管理員，如 Konqueror 或簡單的外圍程序指令，來複製、移動或刪除相機的檔案 (請參閱 [節 27.3, "重要的 Linux 指令" \[368\]](#) 和 [參考](#))。

15.3 使用 Konqueror

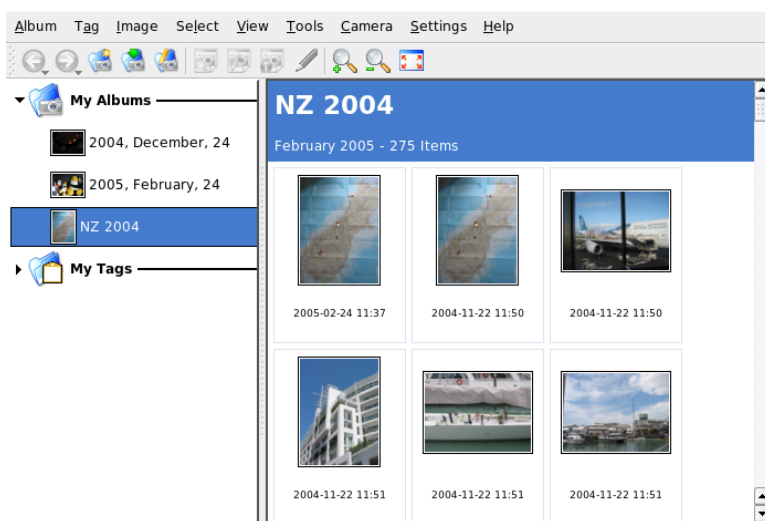
KDE 使用者可以利用常見的 Konqueror 介面輕易地存取數位相機。連接您的相機至 USB 埠。相機圖示應該會出現在桌面上。按一下這個圖示，在 Konqueror 中開啟相機。也可以在 Konqueror 中輸入 URL `camera:/` 以存取相機。瀏覽相機的目錄結構直到顯示檔案。請使用一般的 Konqueror 檔案管理功能來複製想要的檔案。有關如何使用 Konqueror 的詳細資訊，可參考 [章 3, 網頁瀏覽器 Konqueror \[71\]](#)。

15.4 使用 Digikam

Digikam 是用來從數位相機下載相片的 KDE 程式。第一次執行時，Digikam 會詢問相簿的存放位置。如果您輸入已經含有相片組的目錄，Digikam 會將每個子資料夾視為一本相簿。

啟動時，Digikam 顯示一個含有二個段落的視窗：您的相簿顯示在左邊，目前相簿相片顯示在右邊。請參閱 [圖形 15.1, "Digikam 的主視窗" \[193\]](#)。

圖形 15.1 Digikam 的主視窗



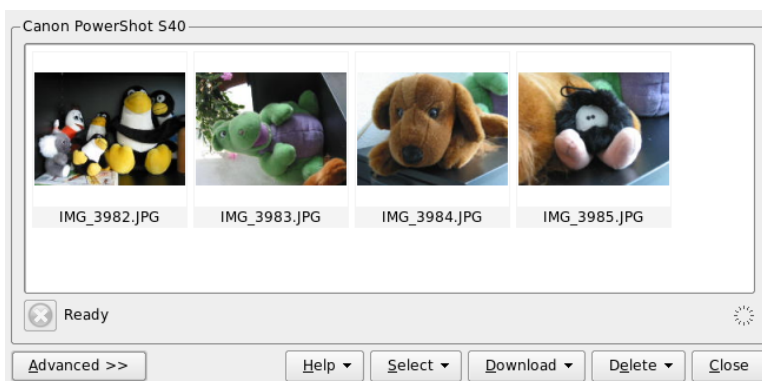
15.4.1 設定相機

要在 Digikam 中設定相機，請選取 **相機** → **新增相機**。首先，嘗試使用 **自動偵測**來自動偵測相機。如果失敗，請使用 **新增**來搜尋您的型號清單。如果清單中沒有您的相機型號，請嘗試用較舊的型號或使用 **USB/IEEE 大量儲存相機**。按 **確定** 來確認。

15.4.2 從相機下載相片

相機正確設定後，請使用 *相機* 功能表以及在 節 15.4.1, "設定相機" [193] 中的對話方塊所提供的名稱來連接相機。Digikam 會開啟一個視窗並開始下載並顯示縮圖，如 圖形 15.2, "從相機下載相片" [194] 所示。在影像上按一下滑鼠右鍵可以開啟快顯功能表，裡面的選項可用來 檢視、顯示一些 內容或 EXIF 資訊、下載或 刪除 影像。使用 進階，可選取重新命名選項以及處理相機提供資訊 (EXIF) 的方式。

圖形 15.2 從相機下載相片



如果您的相機不使用有意義的檔案名稱，重新命名選項會很方便。您可以讓 Digikam 自動重新命名相片。提供唯一化的字首，以及選擇性提供的日期、時間或序號。Digikam 可以完成其它工作。

請按滑鼠左鍵或按住 [Ctrl] 後按一下個別相片，來從相機選取所有相片。選取的相片會以相反的色彩顯示。按一下 下載。請從清單選取目的地，或使用 新增相簿 來建立新相簿。將以目前的日期自動建議檔案名稱。請按 確定 來開啟下載程序。

15.4.3 取得資訊

相片相關資訊的取得並不困難。如果您將滑鼠指標指向縮圖，會以工具提示方式顯示簡短摘要。如需較多的資訊，請在相片上按一下滑鼠右鍵，然後從功能表選擇 內容。會開啟內含三個標籤的對話方塊：一般、EXIF 和 長條圖。

一般列出名稱、類型、擁有者以及其他基本資訊。比較令人感興趣的部份是 *EXIF* 標籤。相機儲存每一張相片的某些中繼資料。Digikam 將讀取這些內容並顯示成清單。請尋找曝光時間、像素尺寸以及其他資訊。若要取得所選清單項目的詳細資訊，請按 **[Shift] + [F1]**。將顯示小工具提示。最後面的 *長條圖* 標籤，顯示某些統計資訊。

15.4.4 管理相簿

依預設，Digikam 會插入 *我的相簿* 資料夾，可存放您所有的相片。您稍後可以將這些照片儲存到子資料夾。相簿可以依目錄配置、相簿內容中設定的收藏名稱或相簿首次建立日期(這個日期可以在每一個相簿的內容中變更)來進行排序。

若要建立新相簿，有許多可用方法：

- 從相機上傳新相片
- 按一下工具列中的 *新相簿* 按鈕來建立新相簿。
- 從硬碟輸入現有的相片資料夾(選取 *相簿* → *輸入* → *輸入資料夾*)
- 在 *我的相簿* 按一下滑鼠右鍵，然後選取 *新相簿*

選取依喜好方式來建立相簿後，會出現一個對話方塊。請提供相簿的標題。可選擇性地選擇相片集、插入一些備註以及選取相簿日期。相片集是利用通用標籤來組織相簿的方法。當您選取 *檢視* → *排序相簿* → *依相片集* 時便會使用這個標籤。備註顯示在主視窗上方的標題中。當您選取 *檢視* → *相簿* → *依日期* 時會使用相簿日期。

Digikam 使用相簿中的第一張相片做為 *我的相簿* 清單中的預覽圖示。若要選取其他相片，請在個別相片上按一下滑鼠右鍵，然後從內容功能表選取 *設成相簿縮圖*。

15.4.5 管理標籤

管理各種相簿中的各種相片有時候會變得複雜。為了組織個別的相片，Digikam 提供 *我的標籤* 系統。

例如，假設您和朋友 John 在不同時間拍下許多相片，您想不依賴相簿收集所有相片。要找出所有相片非常簡單。首先，請按一下 *我的標籤* → *人員* 來建立新

標籤。從內容功能表，選擇 *新標籤*。在出現的對話方塊中，輸入 *John* 當作標題，同時設定一個選擇性的圖示。按 *確定* 來確認。

建立好標籤後，對它指定您要的相片。請移至每一個相簿選取個別相片。按一下滑鼠右鍵，然後從顯示的功能表中選擇 *指定標籤* → *人員* → *John*。另外，也可以將相片拖曳到 *我的標籤* 下，將相片放入標籤中。依需要在其他相簿重複執行。按一下 *我的標籤* → *人員* → *John* 來檢視所有影像。您可以對每一張相片指定多個標籤。

編輯標籤和備註可能非常煩瑣。若要簡化這個作業，請在相片上按一下滑鼠右鍵，然後選取 *編輯備註和標籤*。將會開啟對話方塊，內含預覽、備註欄位和標籤清單。現在，您可以插入所有必要的標籤並加入備註。使用 *下一頁* 和 *上一頁* 來瀏覽相簿。請使用 *套用* 來儲存變更並按 *確定* 離開。

15.4.6 匯出影像收藏

Digikam 提供一些匯出選項，協助您歸檔及發佈個人影像收藏。它可以歸檔到 CD 或 DVD (透過 k3b)、HTML 匯出，以及匯出到遠端藝廊。

若要將影像收藏歸檔到 CD 或 DVD，請執行下列步驟：

- 1 選取 *檔案* → *匯出* → *歸檔到 CD/DVD*。
- 2 在 *建立 CD/DVD 歸檔* 對話方塊中使用它的各種子功能表調整。調整好後，按一下 *確定*，開始燒錄。
 - a *選擇*：選取相簿和標籤，決定要歸檔的收藏部分。
 - b *HTML 介面*：決定能否透過 HTML 介面存取影像收藏，以及自動執行是否加到 CD/DVD 歸檔中。設定選擇標題和影像、字型和背景屬性。
 - c *媒體音量指標*：必要時，可以變更音量描述設定。
 - d *媒體燒錄*：必要時，可以調整燒錄選項。

若要建立影像收藏的 HTML 匯出，請執行下列步驟：

- 1 選取 *檔案* → *匯出* → *HTML 匯出*。

- 2 必要時，可以在 **建立影像圖庫** 中使用它的各種子功能表，調整設定。調整好後，按一下 **確定**，開始建立圖庫。
 - a **選擇**：選取相簿和標籤，決定要歸檔的收藏部分。
 - b **外觀**：設定 HTML 庫的標題和外觀。
 - c **相簿**：決定藝廊在磁碟上的位置，以及最後藝廊上顯示的影像大小、壓縮、格式和中繼資料數量。
 - d **縮圖**：包含藝廊瀏覽時所使用縮圖的目標影像、指定大小、壓縮和檔案類型。

若要將收藏匯出到網際網路上的外部影像藝廊，請執行下列步驟：

- 1 取得藝廊在外部網站的帳戶。
- 2 選取 **檔案** → **匯出** → **匯出到遠端藝廊**，並且按照提示，提供外部網站的 URL、使用者名稱和密碼。

Digikam 會連接您指定的網站，並且開啟新的視窗 **藝廊匯出**。

- 3 決定新相簿在藝廊中的位置。
- 4 按一下 **新相簿**，提供 Digikam 所需的資訊。
- 5 使用 **新增相片**，將影像上載到新相簿。

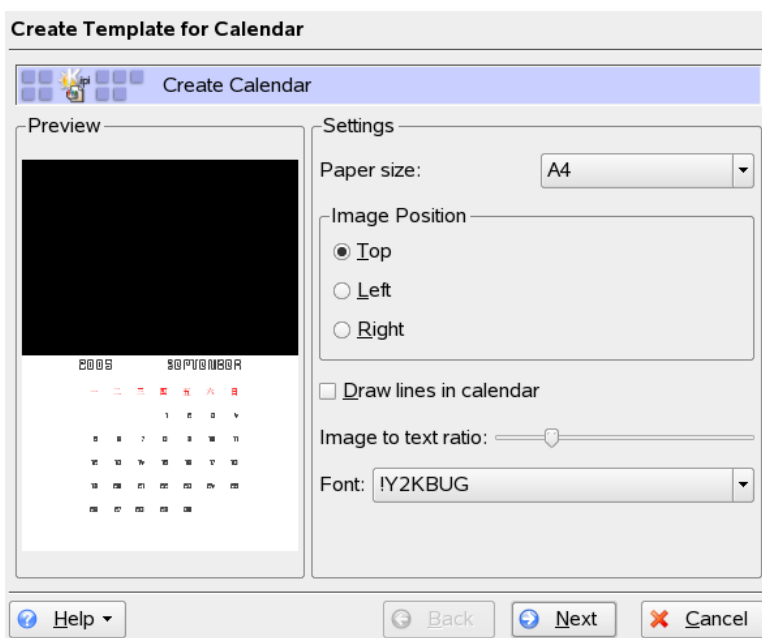
15.4.7 有用的工具

Digikam 提供許多可以簡化某些工作的工具。請到 **工具** 功能表尋找這些工具。以下是可用工具的簡略說明。

建立行事曆

如果您想取悅某人，自訂行事曆會是很好的禮物。請移至 **工具** → **建立行事曆** 來開啟精靈對話方塊，如 [圖形 15.3, "建立行事曆範本" \[198\]](#) 所示。

圖形 15.3 建立行事曆範本



自訂設定 (紙張大小、影像位置、字型等等)，然後按 下一步 以確認。您現在可以輸入年份以及選取要使用的影像。再按一下 下一步 後，便可看到摘要。最後的 下一步 會開啟 KDE 印表機對話方塊。您現在可以決定您要查看預覽、另存為 PDF 或直接列印。

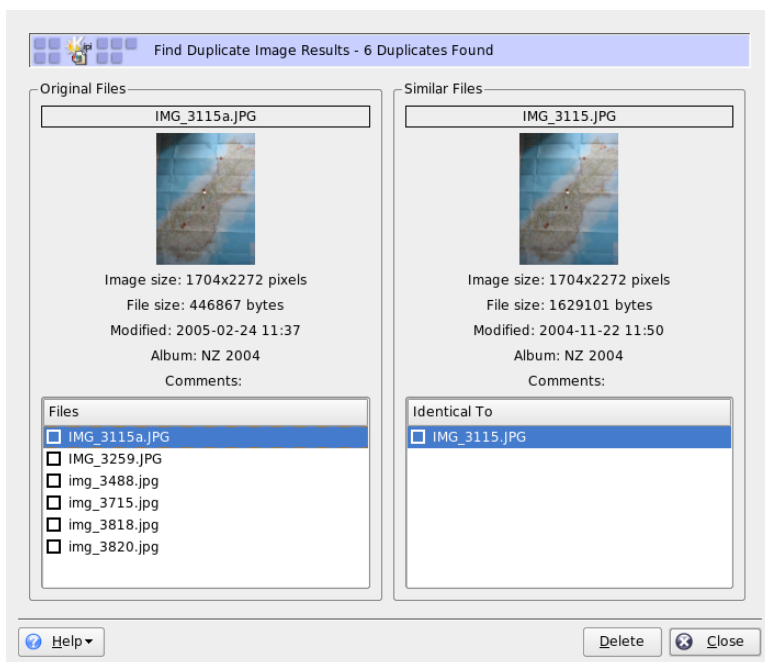
尋找重複的相片

有時候，相片內容很相似而您只想留下最好的一張。尋找重複 插件可以完成這個作業。

請移至 工具 → 尋找重複影像。請選取要處理的相簿或標籤。在 方法 & 快取下，請選擇搜尋方法：更精確或更快速的方法。按 確定 確認後，Digikam 會開始進行處理。

如果找到一些重複的相片，會將結果顯示於視窗中，如 圖形 15.4, "尋找結果" [199] 所示。請啟動核取方塊，決定哪些影像要刪除，然後按一下 刪除。按一下 關閉 離開視窗。

圖形 15.4 尋找結果



批次處理

Digikam 還提供一些批次處理，可以對多個檔案執行特定工作。可以更名、轉換、調整大小等等。在 **工具** → **批次處理** 下可以找到它們。

15.4.8 使用 Digikam 檢視及編輯基本影像

Digikam 有自己的小型影像檢視和編輯程式。只要按兩下影像縮圖，程式即可自動開啟。

您可以使用這個工具，對剛從相機下載的影像進行基本影像編輯。您可以裁切、旋轉或翻閱影像、執行基本的顏色調整、套用各種顏色過濾器(例如，彩色影像匯出為黑白影像)，以及有效減少直向相片的紅眼睛。

最重要的功能表為：

影像

使用 *編輯備註和標籤*，輸入特殊影像備，並且指定影像標籤 (類別)。內容可以叫出有三個標籤的視窗，提供一般資訊、EXIF 資訊和影像長條圖。

修正

這個功能表包含數位攝影最需要的某些編輯功能。*色彩* 會帶您進入子功能表，您可以在這裡修改所有基本色彩設定。您也可以讓整個相片或影像的一部分，變得模糊或更清晰。若要減少直向相片的紅眼睛，請按住左滑鼠指標，慢慢大致圈選臉上的眼睛部分，然後選取 *減少紅眼睛*，再根據您先前選擇整個區域或只有眼睛部分，選擇慢慢的或是迅速的減少。

轉換

轉換 功能表提供裁切、旋轉、翻閱和調整大小功能。您也可以使用 *外觀比例裁切* 選項，按照固定外觀比例裁切。

過濾器

如果您要將彩色相片轉換為黑白相片或是要讓相片看起來很老舊，請到 *過濾器* 功能表中選擇各種匯出選項。

如需這個工具的詳細資訊，請按一下 *Digikam* 功能表列中的 *說明* 按鈕，參閱 *digiKam 影像編輯器* 的 *Digikam* 線上說明。

提示: 進階影像處理

您可以使用 *GIMP* 執行專業影像編輯。關於 *GIMP* 的詳細資訊，請參閱 [章 17, 使用 GIMP 處理繪圖 \[215\]](#)。

15.5 使用 f-spot

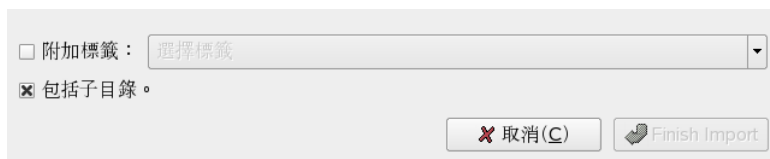
f-spot 是一種管理工具，可用來管理您為 *GNOME* 桌面量身訂製的數位影像收藏。它為影像指定不同的標籤進行分類，並且提供各種好用的影像編輯選項。

初次執行 *f-spot* 時，必須告訴它將哪裡的影像匯入到您的 *f-spot* 收藏。如果硬碟機上已經存有影像收藏，請輸入目錄路徑，您可以選擇包含子資料夾。*f-spot* 會將這些影像匯入到資料庫。

提示：匯入時貼上影像標籤

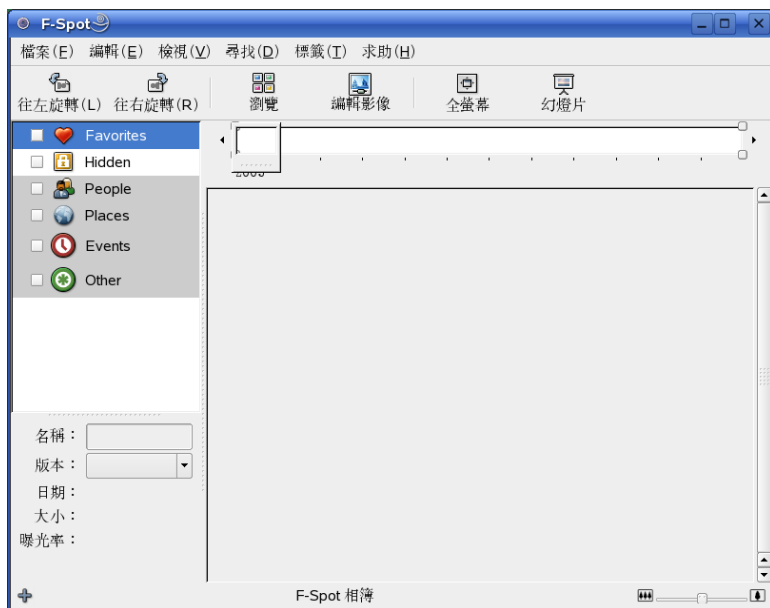
匯入的影像如果是相同類別，您可以在匯入時貼上適當的標籤。請選取 **貼上標籤**，再從下拉功能表中選擇適當的標籤。

圖形 15.5 影像匯入到 *f-spot*



f-spot 主視窗分為三個區域。左邊的側邊列顯示所選影像的類別、標籤和詳細資訊以及所選標籤或類別的所有影像縮圖，如果沒有選取標籤或類別，視窗右側顯示整個收藏。

圖形 15.6 *f-spot* 的主視窗



視窗頂端的功能表列可以讓您存取主功能表。下方的工具列提供以以圖示顯示的不同功能：

旋轉 (左或右)

這是變更影像方向的捷徑。

瀏覽

瀏覽模式可以讓您檢視及搜尋整個收藏或貼有標籤的子集。您也可以使用時間行，按照建立日期來搜尋影像。

編輯影像

這個模式可以讓您選取一個影像，進行某些基本處理處理。詳細資訊請參閱 [節 15.5.6, "使用 f-spot 處理基本影像" \[206\]](#)。

全螢幕

可以切換到全螢幕模式。

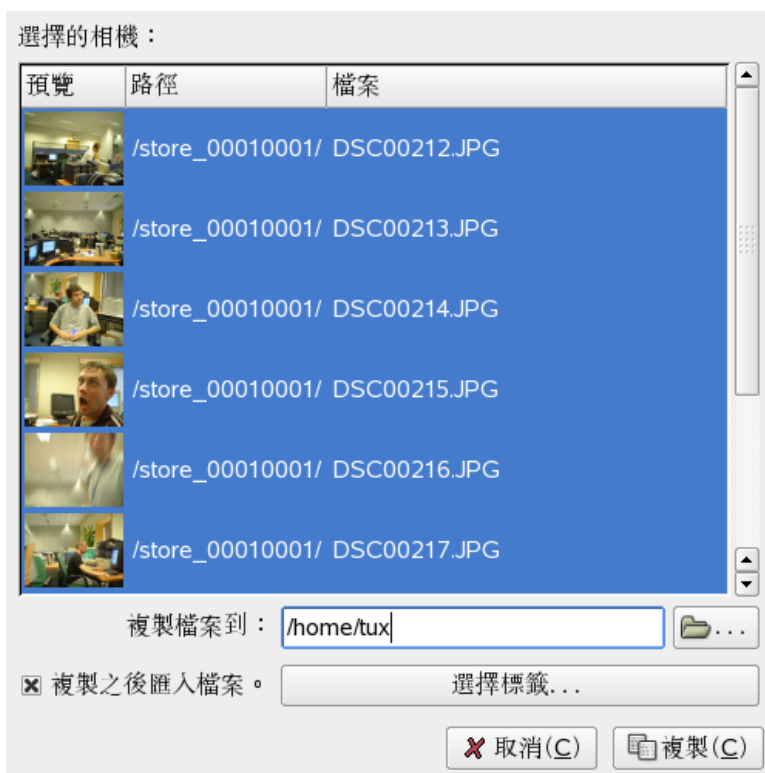
投影片

放映投影片。

15.5.1 從相機下載相片

如果數位相機連接電腦 USB 連接埠，您可以使用 檔案 → 從相機匯入，從數位相機匯入新影像。系統自動偵測相機類型。

圖形 15.7 從相機匯入



f-spot 會啟動預覽視窗，顯示可以從相機下載的所有影像。檔案會複製到您在 **檔案複製到** 中指定的目標目錄。如果您選擇 **複製後匯入檔案**，所有從相機複製的影像會自動匯入到 f-spot 資料庫。如果您在 **選取標籤** 中選取適當的標籤，匯入時會貼上標籤。如果您不想將相機中的所有影像匯入到資料庫，請在預覽視窗中取消選取不要匯入的影像。

15.5.2 取得資訊

選取影像後，視窗左下方會顯示這個影像的某些基本統計資訊。包含檔案名稱、版本 (複本或原始影像)、建立日期、大小，以及建立特殊影像時使用的曝光。若要檢視影像檔案的相關 EXIF 資料，請使用 **檢視** → **EXIF 資料**。

15.5.3 管理標籤

您可以使用標籤將影像分類，建立可管理的收藏子集。例如，如果您想將收藏中心愛的直向相片做某種排序，請執行下列步驟：

- 1 選取 f-spot 的 瀏覽 模式。
- 2 選取 f-spot 視窗左框架中的 人物 類別，在此類別上按一下滑鼠右鍵，選擇 建立新標籤。新標籤出現在 人物 類別下，成為它的子類別：
 - a 建立新標籤 朋友。
 - b 建立新標籤 家庭。
- 3 將這些標籤貼到所選影像或影像群組。在影像上按一下滑鼠右鍵，選擇 貼上標籤，再選取適合影像的標籤。如果要將標籤貼到影像群組，請按一下第一個影像，然後按 **[Shift]** 鍵，選取其他影像。請記住，要按住 **[Shift]** 鍵不放。在標籤功能表上按一下滑鼠右鍵，選取適當的類別。

影像分類後，即可按照標籤來瀏覽您的收藏。您只要勾選 人物 → 家庭，即可顯示貼有 家庭 標籤的收藏。另外，選取 尋找 → 依標籤尋找，也可以按照標籤來搜尋您的收藏。搜尋結果顯示在縮圖概觀視窗中。

移除影像或影像群組標籤的方法，類似貼上標籤。您也可以從頂端功能表列的 標籤 功能表，存取標籤編輯功能。

15.5.4 搜尋及尋找

如 節 15.5.3, "管理標籤" [204] 所述，標籤可用來尋找某些影像。f-spot 特有的尋找方法是使用工具列下方的 時間行。您只要沿著時間行拖曳小框架，即可讓縮圖概觀只顯示所選時間框架內的影像。f-spot 使用預設時間行，但您可以朝時間行的左右二端移動滑桿，編輯時間範圍。

15.5.5 匯出影像收藏

f-spot 在 檔案 → 匯出 下方，為您的收藏提供不同的匯出功能。最常用的功能可能是 匯出到網站藝廊 和 匯出到 CD。

若要將選擇的影像範圍匯出到網站藝廊，請執行下列步驟：

- 1 選取要匯出的影像。
- 2 按一下 檔案 → 匯出 → 匯出到網站藝廊，選取影像要匯往的藝廊，或新增藝廊。f-spot 會連接網站藝廊的網站位置。選取影像要匯往的相簿，並且決定是否自動調整影像大小以及是否匯出標題和註解。

圖形 15.8 影像匯出到網站藝廊



若要將選擇的影像範圍匯出到 CD，請執行下列步驟：

- 1 選取要匯出的影像。
- 2 按一下 檔案 → 匯出 → 匯出到 CD，再按 確定。

f-spot 會複製檔案並且開啟 CD 寫入對話方塊。請指定影像磁碟名稱，並且決定寫入速度。按一下 寫入，開始寫入 CD。

圖形 15.9 影像匯出到 CD



資訊

將光碟燒錄至(I)： 影像檔

光碟名稱(N)： 個人資料，西元2005年09

資料大小： 7 MiB

燒錄選項

燒錄速度(S)： 最大可能

燒錄完成後

☒ 退出光碟(J)

求助(H) 取消(C) 燒錄(W)

15.5.6 使用 f-spot 處理基本影像

f-spot 提供一些非常基本的影像編輯功能。按一下工具列中的 **編輯影像** 圖示或按兩下要編輯的影像，即可進入 f-spot 編輯模式。右下方的方向鍵可用來切換影像。您可以選擇下列編輯功能：

更清晰

使用 **編輯** → **更清晰**，即可存取這個功能。必要時，您可以調整 **數量**、**範圍** 和 **臨界點** 等值，然後按一下 **確定**。

裁切影像

若要將影像裁切成您選擇的範圍，您可以在左下方的下拉式功能表中選取固定比例裁切或 **不限制**，然後按一下比例功能表旁邊的剪刀圖示。

減少紅眼睛

在直向相片中選取臉上的眼睛部分，然後按一下紅眼睛圖示。

調整色彩

檢視用來建立相片的長條圖，必要時您可以更正曝光和色溫。

提示: 進階影像處理

您可以使用 GIMP 執行專業影像編輯。關於 GIMP 的詳細資訊，請參閱 [章 17, 使用 GIMP 處理繪圖 \[215\]](#)。

15.6 若需詳細資訊

如需使用數位相機與 Linux 的詳細資訊，請參考下列網站：

- <http://digikam.sourceforge.net/> 是 Digikam 的相關資訊
- <http://www.gphoto.org> 是 gPhoto2 的相關資訊
- <http://www.gphoto.org/proj/libgphoto2/support.php> 是支援的相機清單
- <http://www.thecompany.com/projects/gphoto/> 是 Kamera (gPhoto2 的 KDE 前端工具) 的相關資訊

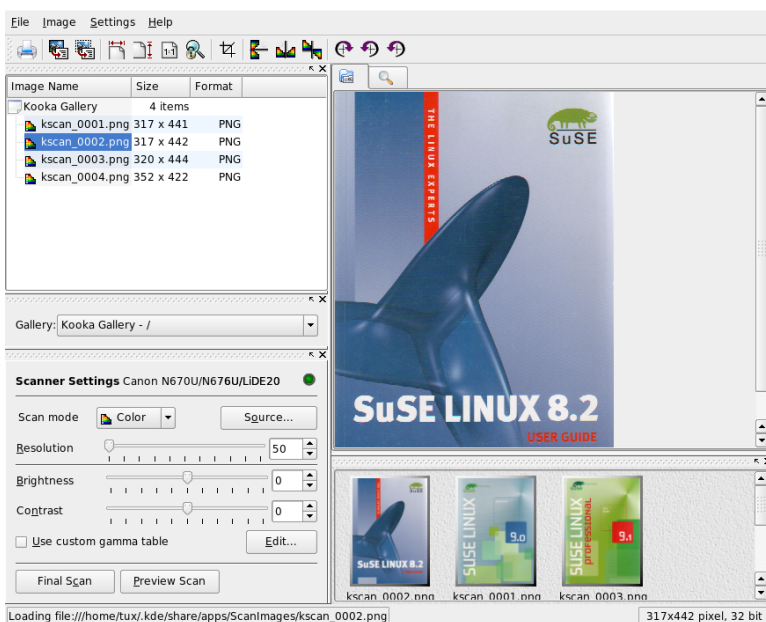
Kooka--掃描應用程式

Kooka 是用來掃描的 KDE 應用程式。本章節是說明此應用程式的使用者介面及功能。Kooka 除了能夠從照片或雜誌等平面媒體建立影像檔案之外，也具有字元辨識的功能。這意味著 Kooka 能夠將寫下的文字轉換為可編輯的文字檔案。

請從主功能表啟動 Kooka 或輸入指令 `kooka` 來啟動。啟動時 Kooka 會開啟三個框架的視窗，包括左上方的功能表列及其下方的工具列。所有視窗都可以使用滑鼠自由地重新調整或重新排列。也可以從 Kooka 視窗中完全卸離單一框架並配置於桌面。要移動框架，只要點選並拖曳框架上的細雙線即可。除了主視窗之外的任何框架，都可以隨意放置於其他框架中，並與框架的左、右、頂端、底端或中間對齊。在中間的視窗有相同的大小，可以堆疊，且可以利用索引標籤將視窗帶到前景。

影像檢視器與掃描預覽框架依預設共用一個視窗。可以使用索引標籤在兩個框架之間切換。左框架提供了藝廊。這是一個小型的檔案瀏覽器，以供存取掃描的影像。右下方的框架是由 OCR (optical character recognition, 光學字元辨識) 與縮圖所共用，只要按一下滑鼠就可以載入影像檢視器中。請參閱 [圖形 16.1, "Kooka 主視窗" \[210\]](#)。

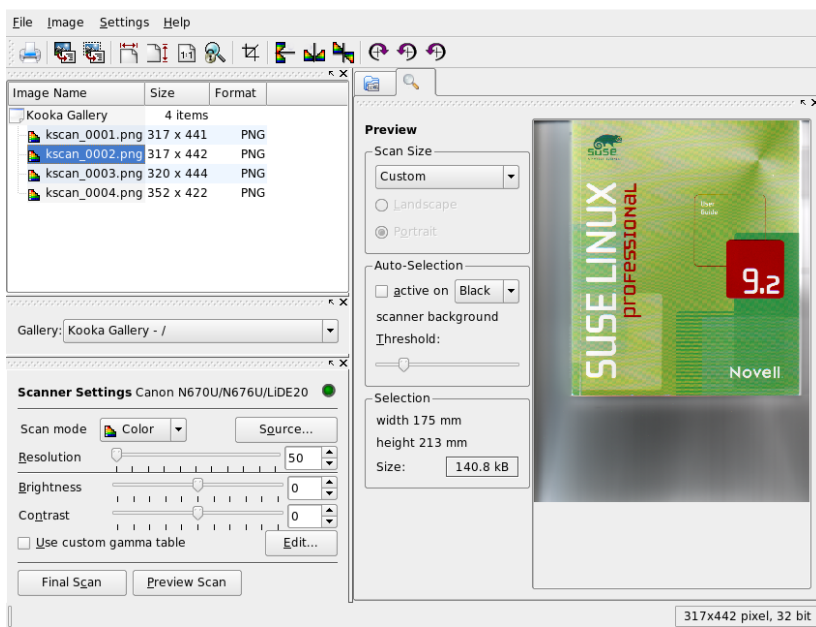
圖形 16.1 Kooka 主視窗



16.1 預覽

當要掃描的物件比整個掃描區域小時，都能建立預覽。在預覽框架的左邊可以設定一些參數。利用 *自訂* 或某一標準格式以選取掃描大小。請參閱 圖形 16.2, "Kooka 預覽視窗" [211]。自訂設定是最具彈性的，因為它允許使用滑鼠來選取想要的區域。一旦完成設定，就可以按一下 *掃描參數* 上的 *預覽掃描* 以預覽掃描的影像。

圖形 16.2 Kooka 預覽視窗

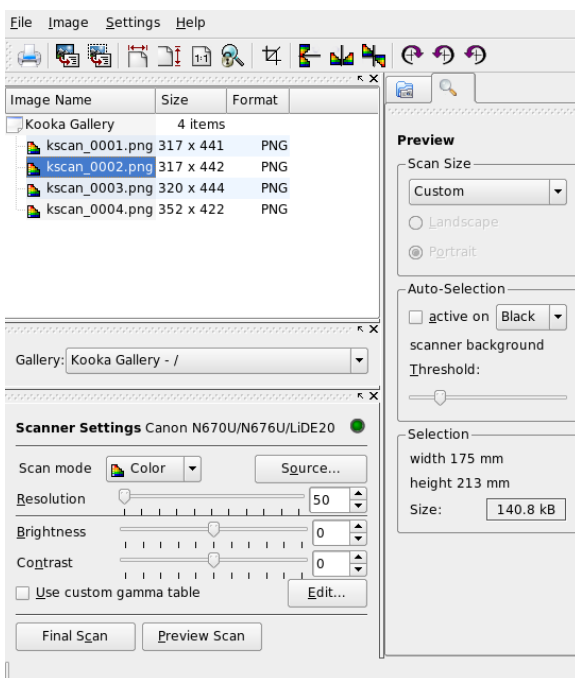


16.2 完成掃描

如果您選取 *自訂* 以調整掃描大小，可使用滑鼠選取要掃描的長方形區域。選取的區域為點狀框線所包圍。

選擇彩色及或黑白掃描，並以滑桿來設定解析度。請參閱 圖形 16.3, "Kooka 掃描參數" [212]。解析度越高，掃描的影像品質就越好。然而，在高解析度下也會產生相對較大的檔案，而且掃描的過程會花費很長的時間。啟用 *使用自訂 gamma* 表，然後按一下 *編輯* 以變更亮度、對比及 *gamma* 的設定。

圖形 16.3 Kooka 掃描參數



一旦所有設定已完成，就可以按一下 **完成掃描** 以掃描影像。掃描的影像會以縮圖顯示於影像檢視器中。當提示出現時，請選取要儲存影像的格式。要以該相同格式儲存未來的所有影像，請核取對應的方塊。請按 **確定** 加以確認。

16.3 功能表

工具列的某些功能也可以在 **檔案** 及 **影像** 功能表中使用。在 **設定** 中修改 Kooka 的優先設定。

檔案

使用此功能表以啟動 KPrinter 列印助理，為您的影像建立新資料夾，以及儲存、刪除與關閉檔案。掃描文字文件的 OCR 結果可以在此儲存。也可以使用此功能表來關閉 Kooka。

影像

影像功能表可讓您啟動用來進行後製處理的繪圖應用程式，或是影像的光學字元辨識。從 OCR 作業得到的辨識文字會顯示於自己的框架中。有許多種縮放、旋轉及翻轉影像的工具可以使用。這些功能也都可以從工具列中存取。從選取建立 允許儲存之前使用滑鼠來標記區域的影像。

設定

設定 調整 Kooka 的外觀與操作。工具列與狀態列都可以切換開關，而功能表項目的鍵盤捷徑也可以自行定義。設定工具列提供可以在工具列使用的所有功能的清單。設定 Kooka 會開啟組態對話方塊，可在此修改 Kooka 的外觀與操作。然而在正常情況下，預設值已經足夠使用了。在 工具檢視 中，可以啟用與關閉縮圖檢視器、預覽、藝廊、掃描參數，以及 OCR 結果視窗。

說明

說明 功能表可以存取 Kooka 線上說明手冊。也可使用它來存取有關問題與期望的意見回應通道。它也提供有關 Kooka 與 KDE 的版本、作者及授權資訊。

16.4 藝廊

藝廊視窗顯示 Kooka 儲存所有影像檔案的預設資料夾。請參閱圖 [圖形 16.4](#), "Kooka 藝廊" [213] 中顯示的範例。要儲存影像至您個人的主目錄中，請按一下縮圖以選取它，然後選取 檔案 → 儲存影像。然後進入您的個人主目錄中，給予該檔案描述性名稱。

圖形 16.4 Kooka 藝廊

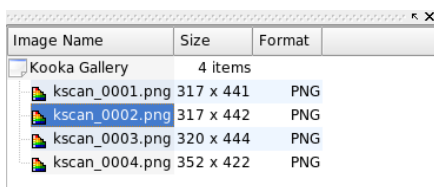


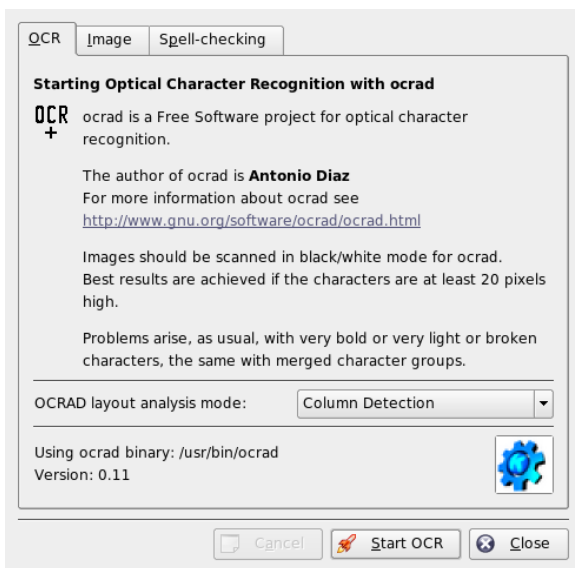
Image Name	Size	Format
Kooka Gallery 4 items		
kscan_0001.png	317 x 441	PNG
kscan_0002.png	317 x 442	PNG
kscan_0003.png	320 x 444	PNG
kscan_0004.png	352 x 422	PNG

若要新增影像至藝廊，只要將它們從 Konqueror 拖放過來即可。啟動 Konqueror，瀏覽包含要新增至藝廊的影像之目錄，並使用滑鼠拖曳影像至 Kooka 藝廊的資料夾中。

16.5 光學字元辨識

如果字元辨識模組已安裝，文件就可以使用 *lineart* 模式進行掃描，並以建議的格式儲存，然後從 影像功能表中對文字辨識進行處理。您可以決定要處理整份文件或只處理事先選取的區域。組態對話方塊會通知模組有關原始文字是要以列印類型、手寫或標準化類型進行處理。也可以設定語言，使模組可以正確地處理文件。請參閱 [圖形 16.5, "Kooka 的 OCR 功能" \[214\]](#)。

圖形 16.5 Kooka 的 OCR 功能



切換至 *OCR 結果文字* 視窗並檢查這些文字，您可能需要對 OCR 出來的文字進行校對。要這樣做，請使用 檔案 → 儲存 OCR 結果文字 來儲存文字。文字可以稍後使用 OpenOffice.org 或 KWrite 進行處理。

使用 GIMP 處理繪圖

GIMP (*GNU 影像處理程式*, *GNU Image Manipulation Program*) 是一個建立與編輯點陣圖形的程式。在許多方面，它的功能可媲美 Adobe Photoshop 以及其他商用程式。使用 GIMP 來調整大小並修飾相片、設計網頁的圖形、製作自訂 CD 的封面，或是任何其他繪圖專案。它可以滿足業餘及專業使用者的需求。

如同許多其他 Linux 程式一樣，GIMP 是由世界各地自願花時間並分享專案程式碼的開發人員所合作開發的。程式是持續發展的，所以包含在您的 SUSE Linux 的版本與在此討論的版本可能會有些微的不同。個別視窗與視窗區域的配置最有可能不同。

GIMP 是非常複雜的程式。本章節僅討論小範圍的功能、工具及功能表項目。請參閱 [節 17.6, "若需詳細資訊" \[221\]](#) 以得知可以找到更多有關此程式資訊的地方。

17.1 繪圖格式

有兩種主要的繪圖格式 -- 點陣圖形與向量圖形。GIMP 只處理點陣圖形，這是相片與掃描影像通用的格式。點陣圖形由彩色的小方塊所組成，集合起來建立整個影像。檔案很容易因此變得相當大。放大點陣影像的大小一定會降低品質。

和點陣圖形不同，向量圖形不儲存個別像素的資訊。向量圖形是儲存有關影像的點、線或面如何群組在一起的資訊。向量影像也可以很輕易地縮放。例如，OpenOffice.org 的繪圖應用程式便是使用這種格式。

17.2 啟動 GIMP

從主功能表中啟動 GIMP。或者，在指令行中輸入 `gimp &`。

17.2.1 啟始組態

第一次啟動 GIMP 時，就會開啟組態精靈以供設定預備的組態。預設設定可以滿足大部分的用途。在每個對話方塊中按 *繼續*，除非您很熟悉這些設定而且想要使用其他設定。

17.2.2 預設視窗

預設會出現三個視窗。可以在螢幕上排列它們，而且如果不再需要，可以關閉它們(工具箱除外)。關閉工具箱就會關閉應用程式。在預設組態中，GIMP 會在結束時儲存視窗配置。當您下次啟動程式時，對話方塊就會保持開啟的狀態重新出現。

工具箱

如 [圖形 17.1, "主視窗" \[217\]](#) 所示，GIMP 的主視窗含有應用程式的主要控制項。關閉它以結束應用程式。在最頂端，功能表列提供檔案功能、擴充、以及說明的存取。在其下面，會發現各種工具的圖示。將滑鼠移至某圖示的上方就會顯示關於該圖示的資訊。

圖形 17.1 主視窗



目前前景和背景的色彩會顯示於兩個重疊的方塊中。前景的預設色彩為黑色，而背景為白色。按一下方塊以開啟色彩選取對話方塊。使用方塊右上方的右彎箭號符號來交換前景與背景的色彩。使用左下方的黑白符號以重設色彩為預設值。

右邊顯示目前的筆刷、圖樣及漸層。按一下顯示的圖示以存取該選項對話方塊。視窗的下半部包含允許為目前工具設定各種選項的組態。

圖層、色版、路徑、復原

在第一小節中，使用下拉式方塊來選取索引標籤所參考的影像。按一下 *自動*，來控制是否要自動選擇作用中影像。依預設會啟用 *自動*。

*圖層*可顯示目前影像中不同的圖層，且可以用來操作圖層。*色版*可顯示並操作影像的色版。

路徑是選取部分影像的進階方法。它們也可以用來繪圖。*路徑*顯示可在影像中使用的路徑，並提供存取路徑的功能。*復原*顯示對目前影像所進行修改的有限歷程。

視窗底部包含三個索引標籤。使用它們來選取目前筆刷、漸層及圖樣。

17.3 GIMP 入門

雖然 GIMP 可能讓新的使用者感到怯步，可是一旦熟悉一些基本功能後，多半都會認為它容易使用。重要的基本功能包括建立、開啟與儲存影像。

17.3.1 建立新影像

要建立新影像，可以選取 **檔案** → **開新檔案** 或按下 **Ctrl** + **N**。這樣會開啟對話方塊來對新影像進行設定。如有需要，可以使用 **範本** 以選取新影像所依據的範本。GIMP 包含一些範本，範圍從 A4 紙張到 CD 封面，可以從此選取範本。若要建立自訂範本，請選取 **檔案** → **對話方塊** → **範本**，並使用開啟視窗提供的控制項。

在 **影像大小** 區域中，以像素或其他單位設定要建立的影像大小。按一下單位，並從可用單位的清單中選取其他單位。像素和單位之間的比例在 **解析度** 中設定，當 **進階選項** 區段開啟時即會出現。每英吋 72 個像素的解析度符合螢幕的顯示。這樣的解析度對於網頁圖形已經夠用。要列印的影像應該使用較高的解析度。對於大部分的印表機而言，每英吋 300 個像素的解析度才會有可以接受的品質。

在 **Colorspace** 中，選取影像是否要使用色彩 (**RGB**) 或 **灰階**。選取新影像的 **填滿類型**。**前景色彩** 與 **背景色彩** 使用在工具箱中選取的色彩。**白色** 使用白色背景於影像中。**透明** 建立透明的影像。**透明** 會以灰色的棋盤圖樣表示。在 **註解** 中輸入新影像的註解。

當設定符合您的需求時，請按 **確定**。若要還原預設設定，請按 **重設**。按下 **取消** 以中止新影像的建立。

17.3.2 開啟現有的影像

要開啟現有的影像，可以選取 **檔案** → **開啟舊檔** 或按 **Ctrl** + **O**。在開啟的對話方塊，選取想要的檔案。按一下 **確定** 以開啟選取的影像。按 **取消** 以略過開啟影像。

17.3.3 影像視窗

新增或開啟的影像會出現在各自的視窗中。在視窗頂端的功能表列提供存取所有的影像功能。或者，在影像上按一下滑鼠右鍵，或按一下尺規左邊角落的小型箭號按鈕以存取功能表。

檔案提供標準檔案選項，例如 *儲存與列印*。*關閉*可以關閉目前影像。*結束*會關閉整個應用程式。

利用 *檢視*功能表中的項目，控制影像及影像視窗的顯示。*新增檢視*會開啟目前影像的第二個顯示視窗。在某個檢視中進行的變更會反映在該影像的所有其他檢視中。對於放大要操作的局部影像，同時在另一個檢視中查看完整影像，替代檢視是非常有用的。使用 *縮放*來調整目前視窗的放大倍率。當選取 *調整視窗*時，影像視窗就會調整大小以符合目前影像的精確顯示。

17.4 儲存影像

沒有比 *檔案* → *儲存* 更重要的影像功能。常常儲存檔案會比偶而儲存檔案好。請使用 *檔案* → *另存新檔*，以新的檔案名稱儲存影像。以不同的名稱儲存影像階段，或製作一個備份於其他目錄，讓您可以輕易地還原到先前的狀態，都是很好的想法。

第一次儲存或使用 *另存新檔*時，將開啟一個對話方塊供您指定檔案名稱和類型。請在最上面的欄位輸入檔案名稱。在 *儲存在資料夾*，請從常用的目錄清單選取要儲存檔案的目錄。若要使用不同目錄或建立新目錄，請開啟 *瀏覽其他資料夾*。建議您將 *選取檔案類型*的設定保持為 *依副檔名*。透過該設定，GIMP 將依據附加在檔案名稱的副檔名來決定檔案類型。下列是經常會使用的檔案類型：

XCF

這是此應用程式的特有格式。它會將所有圖層和路徑資訊與影像本身一起儲存。即使您需要其他格式的影像，使用 XCF 來儲存複本以簡化未來的修改程序，通常是非常好的想法。

PAT

這是用於 GIMP 圖樣的格式。以此格式儲存影像就可以使用該影像做為 GIMP 的填滿圖樣。

JPG

JPG 或 JPEG 是一種用於相片及網頁的 (不含透明) 圖形之常見格式。它的壓縮方法可以縮減檔案大小，但是壓縮時會遺失一些資訊。在調整壓縮層級時使用預覽選項可能是一個好主意。使用 85% 到 75% 層級的適當壓縮通常能夠得到可接受的影像品質。也建議以品質降低較少的格式 (如 XCF) 來儲存備份。如果要編輯影像，只要將最終的影像儲存為 JPG 即可。一再重複載入 JPG 格式的影像然後儲存，容易得到較差的影像品質。

GIF

雖然圖形包含透明的特性在過去非常受歡迎，但是因為授權問題，GIF 現在比較不常用了。GIF 也可用於動畫影像。此格式只能儲存索引影像。如果只使用幾個色彩的話，檔案大小通常會相當小。

PNG

利用 PNG 所支援的透明、較不會降低品質的壓縮、自由的可利用性、以及越來越多的瀏覽器支援，PNG 已經取代 GIF 成為含有透明的網頁圖形中較受歡迎的格式。有一個額外的優勢就是 PNG 提供部分透明，這是 GIF 所無法提供的。這可以使彩色的區域更流暢的過渡到透明區域 (消除鋸齒)。

要以選取的格式儲存影像，請按 **儲存**。若要中止，請按 **取消**。如果影像有某些特性使其無法以選取的格式儲存，就會出現包含可解決此情況選項的對話方塊。如果有提供的話，請選取 **匯出**，通常會得到想要的結果。然後就會開啟一個包含格式選項的視窗。並提供適當的預設值。

17.5 列印影像

要列印影像，可以從影像功能表中選取 **檔案** → **列印**。如果您的印表機是在 SUSE 中設定，它應該會出現在清單中。在某些情況下，可能必須使用 **設定印表機** 來選取適當的驅動程式。在 **紙張大小** 中選取適當的紙張大小，而 **紙張類型** 中選取紙張類型。其他可用的設定在 **影像 / 輸出設定** 索引標籤中。

圖形 17.2 列印對話方塊



在視窗底部，調整影像大小。按下 **使用原始影像大小**，從影像本身取得這些設定。這會建議您是否設定適當的列印大小及影像解析度。利用 **位置** 中的欄位或在 **預覽** 中拖曳影像，以調整影像在頁面中的位置。

對這些設定滿意時，請按下 **列印**。要儲存這些設定以供未來使用，可以使用 **列印並儲存設定** 作為替代。取消會中止列印。

17.6 若需詳細資訊

以下是一些對 GIMP 使用者可能會有用的資源。不幸的是，不少資源適用於舊版。

- 說明可提供存取至內部說明系統。本說明文件也可以在<http://docs.gimp.org> 取得 HTML 和 PDF 格式的版本。
- GIMP 使用者群組在<http://gug.sunsite.dk> 提供包含許多情報及有趣的網站。
- <http://www.gimp.org> 是 GIMP 的官方首頁。
- Carey Bunks 的 *Grokking the GIMP* 是一本以舊的 GIMP 版本為基礎的好書。雖然程式的某些外觀已經改變，但是它可以提供有關影像操作的絕佳指導。線上版本可以在<http://gimp-savvy.com/BOOK/> 取得。
- <http://gimp-print.sourceforge.net> 是有關 GIMP 列印外掛程式的網頁。可從此網站取得的使用者手冊，使用者手冊提供有關設定及使用此程式的詳細資訊。

部 VI. 機動性

Linux 的行動計算功能

本章節提供使用 Linux 行動計算功能各方面的概觀。會簡短介紹使用的各種欄位，也會說明使用硬體的基本功能。特殊需求的軟體解決方案、最大效能的選項、及使用最少電源的可能性，皆合併涵蓋在內。本章節也包含有關主題最重要資訊來源的概述。

大部份的人將行動計算的概念與筆記型電腦、PDA、和行動電話以及其彼此間資料的交換聯想在一起。本章將這個焦點延伸到行動硬體元件，如外接硬碟、隨身碟或數位相機，這些物件可以連接到筆記型電腦或桌上電腦系統。

18.1 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標準的相關屬性。行動硬體的製造商已經發展出 PCMCIA (個人電腦記憶卡國際協會，Personal Computer Memory Card International Association) 標準。這個標準涵蓋了記憶卡、網路介面卡、ISDN 和數據卡、以及外接硬碟。Linux 如何支援這類硬體、組態時該考慮哪些事情、哪些軟體可以用來控制 PCMCIA、以及如何疑難排解可能出現的問題，都詳述於 [章 19, PCMCIA \[233\]](#)。

18.1.1 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。SUSE Linux 支援影響筆記型電腦電源使用的各種方法，而且對於電池電源的操作時間上會產生不同的效用。下列清單以遞減排序陳述對省電的貢獻：

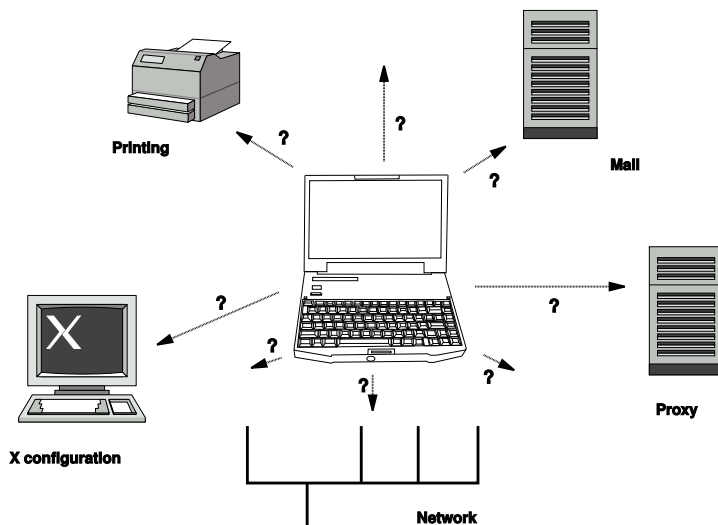
- 調節 CPU 速度
- 暫停時關閉顯示器亮度
- 手動調整顯示器亮度
- 中斷連接已啟用但未使用的熱插件配件 (USB CD-ROM、外接滑鼠、未使用的 PCMCIA 卡等)
- 閒置時降低硬碟轉速

有關 SUSE Linux 中電源管理及操作 YaST 電源管理模組的詳細背景資訊，請參閱 [章 21，電源管理 \[245\]](#)。

18.1.2 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 則會替您執行這項工作。

圖形 18.1 整合網路中的筆記型電腦



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：

網路組態

這包括指定 IP 位址、名稱解析、網際網路連線、以及與其他網路的連線。

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 Proxy

列印時，對應伺服器的清單必須是最新的。

設定 X

如果您的筆記型電腦暫時連接到視訊裝置或外接螢幕，需要使用不同的顯示器組態。

SUSE Linux 提供兩種方法，可以整合筆記型電腦到可結合的現有作業環境。

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。能夠一直設定檔之間切換。您可以在 [章 20, 系統組態設定檔管理\[235\]](#) 找到有關 SCPM 的資訊。KDE 中的 kicker applet Profile Chooser，能夠在設定檔之間切換。該應用程式切換前需要輸入 root 密碼。

SLP

服務位置通訊協定 (Service Location Protocol, 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。您可以在 [章 39, 網路中的 SLP 服務\[543\]](#) 找到有關 SCPM 的詳細資訊。

SCPM 的重要性在於能啟用和維護可複製的系統情況。SLP 可自動設定大部份的網路電腦組態，讓使用更為容易。

18.1.3 軟體選項

行動使用中有各種特殊工作，由專屬軟體執行：系統監控 (特別是電池充電)、資料同步化、以及與周邊設備和網際網路的無線通訊。以下小節涵蓋了 SUSE Linux 為每項工作提供的最重要應用程式。

系統監控

兩種 KDE 系統監控工具由 SUSE Linux 提供。筆記型電腦可充電電池的所有狀態顯示，由 kicker 中的 KPowerSave 程式處理。複雜的系統監控由 KSysguard 執行。使用 GNOME 時，所述的功能由 GNOME ACPI (作為面板程式) 及系統監看提供。

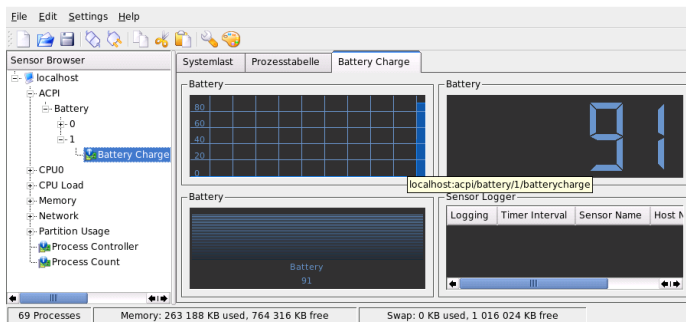
KPowerSave

KPowerSave 是顯示控制台中可充電電池狀態的 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。可以在其中設定不同電源供應類型下系統的行為。有關電源管理及相關 YaST 模組的資訊，請參閱 [章 21, 電源管理 \[245\]](#)。

KSysguard

KSysguard 是獨立的應用程式，將系統中所有可測量的參數收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。可自訂所收集資料的顯示及篩選。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。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。

圖形 18.2 使用 KSysguard 監控電池狀態



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：

電子郵件同步化

使用 IMAP 帳戶在公司網路中儲存您的電子郵件。接著會從任何中斷連線的 IMAP 工作站啟用電子郵件用戶端，來存取電子郵件，例如 Mozilla Thunderbird Mail、Evolution 或 KMail，如啟動中所述。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。如需詳細資訊，請參閱 [章 47](#)，[檔案同步化 \[665\]](#)。

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線而定。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。您可以在 [節 22.1](#)，["無線區域網路" \[267\]](#) 找到有關 WLAN 的詳細資訊。

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。如需有關藍芽、其應用程式和組態的詳細資訊，請參閱 [節 22.2](#)，["藍芽" \[276\]](#)。

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。您可以在 [節 22.3, "紅外線資料傳輸" \[285\]](#) 找到有關 IrDA 的詳細資訊。

18.1.4 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。

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。使用 SUSE Linux 建立加密磁碟分割的內容，請參閱 [節 23.3, "加密分割區和檔案" \[305\]](#) 的說明。

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 [節 23.4, "安全性與機密性" \[308\]](#)。與無線網路相關的安全性措施，請參閱 [章 22, 無線通訊 \[267\]](#)。

18.2 行動硬體

SUSE Linux 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，即可經由熱插自動偵測與

設定。subfs 和 submount 確保設備裝載到檔案系統中的對應位置。使用者完全不用手動裝載和卸載舊版 SUSE Linux 找到的設備。只要程式不需存取設備，即可直接拔除。

外接磁碟 (USB 和 Firewire)

只要系統正確辨認到外接磁碟，外接磁碟的圖示就會出現在 *我的電腦*(KDE) 或 *電腦*(GNOME) 中已裝載設備的清單內。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。若要重新命名磁碟的系統指定名稱，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/usb-xxx` 或 `/media/ieee1394-xxx` 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，在檔案管理員綜覽中也是顯示為外接磁碟機。KDE 允許讀取及存取 URL `camera: /` 上的圖片。而且影像可以使用 Digikam 或 The GIMP 進行處理。使用 GNOME 時，Nautilus 會將圖片顯示在自己的資料夾中。簡單的影像處理及管理公用程式是 f-spot。進階的相片處理則是使用 The GIMP。如需數位相機和影像管理的詳細資訊，請參閱 [章 15, 數位相機與 Linux \[191\]](#)。

18.3 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 ["無線通訊" \[229\]](#) 中提及。行動電話通訊協定的組態，於手冊中均有說明。有關 Linux 端的組態，於 [節 22.2, "藍芽" \[276\]](#) 和 [節 22.3, "紅外線資料傳輸" \[285\]](#) 中有詳細的說明。

Evolution 與 Kontact 內建的同步化已經支援 Palm, Inc. 生產的掌上型設備。開始與設備連線後，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)。這兩種群組軟體應用程式，詳述於 [啟動](#) 中。

整合於 Kontact 程式的 KPilot，也可當作獨立公用程式使用。此內容詳述於 [啟動](#) 中。KitchenSync 程式也可以用來同步化通訊錄資料。

18.4 若需詳細資訊

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.suse.com/archive/suse-laptop/>。在此清單上，使用者與開發人員討論了 SUSE Linux 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。

在筆記型電腦上使用 SUSE Linux 管理電源的問題，建議參閱 `/usr/share/doc/packages/powersave` 中的「讀我」檔案。這個目錄中通常包含測試者及開發人員的最新回應，因此對解決問題提供了寶貴的提示。

PCMCIA

本節包含一些筆記型電腦中與 PCMCIA 硬體和軟體特別相關的觀念。PCMCIA 是國際個人電腦記憶卡協會 (*Personal Computer Memory Card International Association*) 的英文縮稱，為相關軟硬體的統稱。

19.1 硬體

PCMCIA 卡是首要元件，有下列兩種類型：

PC 卡

是 PCMCIA 最初所使用的卡片。它們使用 16 位元匯流排來傳送資料，價格非常低廉。目前有些 PCMCIA 橋接器不易偵測到這類的卡片。不過，一旦偵測得到，通常就會運作順暢，不會發生任何問題。

CardBus 卡

這是較新的標準，使用了 32 位元匯流排，因而速度較快，但也較為昂貴。它們整合在像是 PCI 卡的系統中，運作十分順暢。

第二重要的元件，分別是 PCMCIA 控制器、PC 卡或 CardBus 橋接器，這些元件可以建立卡片與 PCI 匯流排的連線。所有常見型號皆能支援。如果是 PCI 設備，則使用 `lspci -vt` 指令可以得到更進一步的資訊。

19.2 軟體

使用目前的核心，PCMCIA 橋接器和 PCMCIA 卡會由熱插式子系統處理。`pcmcia_socket` 有橋接器事件和 `pcmcia` 事件。`Udevd` 載入所有必須模組，並呼叫相關工具設定這些裝置。這些動作是在 `/etc/udev/rules.d/` 定義。

`/etc/pcmcia/config.opts` 作為資源組態。需要的驅動程式由驅動程式中的裝置表格決定。插槽和卡的目前狀況資訊可透過 `pccardctl` 在 `/sys/class/pcmcia_socket/` 找到。

因為 PCMCIA 系統目前正在變更，本文件還不完整。若需完整綜攬，請參閱 `/usr/share/doc/packages/pcmciautils/README.SUSE`。

系統組態設定檔管理

透過 SCPM (系統組態設定檔管理) 的協助，可將您電腦的組態與不同作業環境或硬體組態搭配。SCPM 管理不同狀況的一組系統設定檔。SCPM 讓您可以輕易在系統設定檔之間切換，減少手動重新設定系統的需要。

有些情況要求修改的系統組態。大部份狀況是在不同位置操作的行動電腦。如果桌面系統要暫時使用與平常不同的硬體元件操作，SCPM 就非常好用。還原原始系統組態會變得很容易，而且可以重新產生系統組態的修改。使用 SCPM，系統組態的任何部份均可保存在自訂的設定檔中。

SCPM 應用程式的主要欄位是膝上型電腦的網路組態。不同的網路組態通常要求其他服務的不同設定，例如電子郵件和 Proxy。然後接著是其他元件，如不同的家用和公司用印表機、會議中的多媒體投影機之自訂 X 伺服器組態、路途上的特殊省電設定，或海外分公司的不同時區。

20.1 詞彙

下列為 SCPM 文件和 YaST 模組中常用的詞彙。

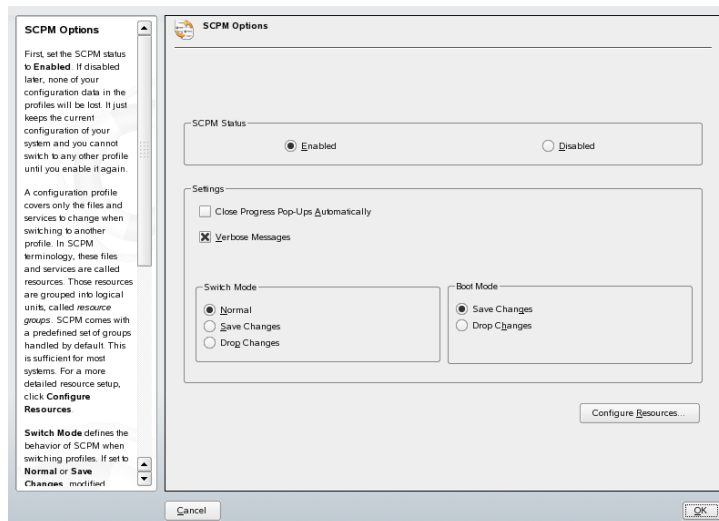
- 詞彙系統組態代表電腦的完整組態。它涵蓋所有基礎設定，如硬碟分割區的使用、網路設定、時區的選擇和鍵盤對應。
- 設定檔也稱為組態設定檔，是將狀態保留起來，可以隨時還原。
- 作用中的設定檔代表最後選取的設定檔。這並不表示目前的系統組態精確地對應到此設定檔，因為可隨時修改組態。

- 在 SCPM 內容中，資源是提供系統組態的元件。可以是包括中繼資料 (如使用者、權限或存取時間) 的檔案或軟式連結 (softlink)。也可以是在此設定檔中執行，但是在另一個設定檔中停用的系統服務。
- 每一個資源皆屬於某個特定的資源群組。這些群組包含邏輯上屬於一起的資源 -- 大部份群組同時包含服務和其組態檔。組合 SCPM 所管理的資源是很容易的，因為不需要想要服務之組態檔案的任何知識。SCPM 和預先設定的資源群組選項一起提供，對於大部份狀況應該已經夠用。

20.2 使用 YaST 設定檔管理員

從 YaST 控制中心使用 (系統 → 設定檔管理員) 啟動 YaST 設定檔管理員。在一開始啟動時，選取在 *SCPM* 選項對話方塊中的 *啟用*，如 [圖形 20.1, "YaST SCPM 選項" \[236\]](#) 中所示。在設定中，決定進度快顯視窗是否應該自動關閉，以及是否顯示 SCPM 組態進度的詳細資訊。*切換模式* 可以在切換設定檔時，決定是否應該儲存或放棄作用中設定檔的已修改資源。如果 *切換模式* 設為一般，在切換時會儲存作用中設定檔中的所有變更。若要在開機時定義 SCPM 的行為，請將 *開機模式* 設為 *儲存變更* (預設值) 或是 *放棄變更*。

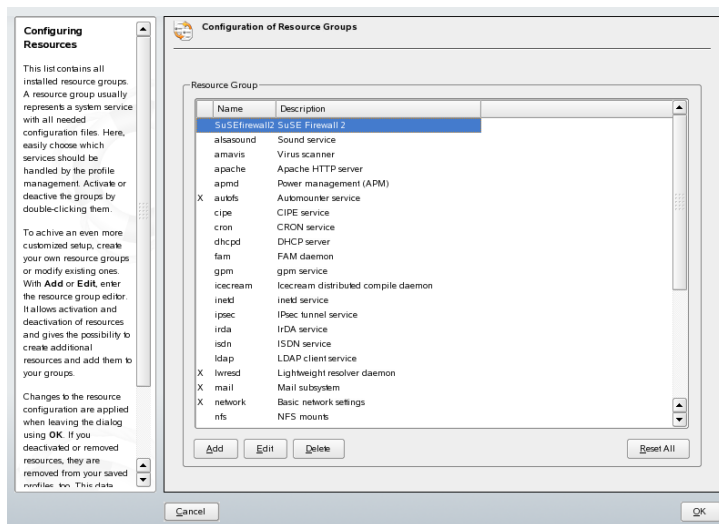
圖形 20.1 YaST SCPM 選項



20.2.1 設定資源群組

若要對目前的資源組態進行變更，請選擇在 *SCPM* 選項對話方塊中的 **設定資源**。下一個對方塊 (如 [圖形 20.2, "設定資源群組" \[237\]](#) 中所示) 會列出系統上所有可用的資源。若要新增或編輯資源群組，請指定或修改 **資源群組** 與 **描述**。以 LDAP 服務為例，請輸入 `ldap` 做為 **資源群組** 與 LDAP 用戶端服務 以做為 **描述**。然後輸入適當的資源 (服務、組態檔或兩者) 或修改現有的資源。刪除不使用的資源。若要重設選取資源的狀態 -- 放棄對它們所做的任何變更，並還原成起始的組態值 — 請選擇 **重設群組**。您的變更會儲存至作用中的設定檔。

圖形 20.2 設定資源群組



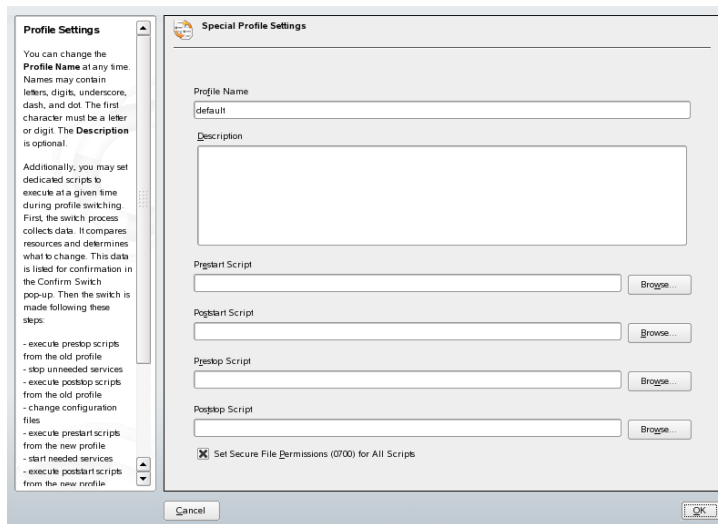
20.2.2 建立新設定檔

若要建立新的設定檔，請按一下啟動對話方塊中的 **新增**(系統組態設定檔管理)。在開啟的視窗中，選取新設定檔是否應該根據目前的系統組態 (*SCPM* 會自動擷取目前的組態並將它寫入設定檔) 或根據現有的設定檔。如果您使用目前的系統組態做為新設定檔的基礎，您可以將新設定檔標示為新作用中的設定檔。這並不會變更舊設定檔，而且也不會啟動或停止任何服務。

在下列對話方塊中提供新設定檔的名稱與簡短描述。若是 *SCPM* 要在切換設定檔時執行特殊的程序檔，請輸入每個執行檔的路徑 (請參閱 [圖形 20.3, "設定檔特](#)

殊設定"[238])。請參閱 節 20.3.4, "進階設定檔設定"[241]，以取得更多的資訊。SCPM 將針對新設定檔的資源進行檢查。在成功地完成此測試後，就可以開始使用新的設定檔。

圖形 20.3 設定檔特殊設定



20.2.3 修改現有設定檔

若要修改現有的設定檔，請選擇啟動對話方塊中的 **編輯(系統組態設定檔管理)**。然後依據需求修改名稱、描述、程序檔以及資源。

20.2.4 切換設定檔

若要切換設定檔，請開啟設定檔管理員。作用中的設定檔是以箭頭標示。選取要切換的設定檔，然後按一下 **切換到**。若要需要的話，SCPM 會檢查新的或修改過的資源，然後新增它們。

如果已修改資源，YaST 會開啟 **確認切換** 對話方塊。作用中設定檔的已修改資源群組 會列出已修改的作用中設定檔的所有資源群組，但不會儲存到作用中設定檔。目前選取的資源群組的 **儲存或忽略**，可以決定是否應該將此資源群組的變更儲存到作用中設定檔或者放棄變更。也可以選取每個資源，然後按一下 **詳**

細資訊 以詳細分析變更。這將顯示出屬於此已修改資源群組的所有組態檔或執行檔的清單。若要取得舊版與新版的逐行比較，請按一下 **顯示變更**。分析變更之後，請在 **動作** 中決定要如何處理它們。

儲存資源

將此資源儲存到作用中的設定檔，但是讓所有其他的設定檔維持不變。

忽略資源

使作用中資源維持不變。此變更已放棄。

儲存到所有設定檔

複製此資源的整個組態到所有其他的設定檔。

修補所有設定檔

僅套用最近的變更至所有的設定檔。

儲存或忽略全部 僅儲存或放棄此對話方塊中顯示的所有資源變更。

在確認作用中設定檔的變更後，請按一下 **確定** 以結束 **確認切換** 對話方塊。SCPM 接著會切換至新的設定檔。在切換時，它會執行舊設定檔的預先停止程序檔以及稍後停止程序檔，以及新設定檔的預先啟動程序檔以及稍後啟動程序檔。

20.3 使用指令行設定 SCPM

本節僅介紹 SCPM 的指令行組態。瞭解如何啟動它、設定它和使用設定檔。

20.3.1 啟動 SCPM 與定義資源群組

SCPM 必須在使用前啟用。以 `scpmenable` 啟用 SCPM。第一次執行時，SCPM 會花費數秒鐘起始化。以 `scpm disable` 隨時停用 SCPM 以避免意外切換設定檔。後續的重新啟動只會繼續起始化。

根據預設，SCPM 會處理網路和印表機設定以及 X.Org 組態。若要管理特殊服務或組態檔，請啟用對應的資源群組。若要列出預先定義的資源群組，請使用 `scpmlist_groups`。若僅要查看已經啟用的群組，請使用 `scpmlist_groups -a`。請在指令行以 `root` 的身份執行這些指令。

```
scpm list_groups -a
```

nis	網路資訊服務用戶端
mail	郵件子系統
ntpd	網路時間通訊協定精靈
xf86	X-伺服器設定
autofs	自動掛載器服務
network	基本網路設定
printer	印表機設定

以 `scpmactivate_group NAME` 或 `scpmdeactivate_group NAME` 啟用或停用群組。以相關群組名稱取代 `NAME`。

20.3.2 建立與管理設定檔

在啟用 SCPM 之後，名為 `default` 的設定檔就已經存在。以 `scpm list` 取得所有可用設定檔的清單。此一現有的設定檔也是作用中的設定檔，可使用 `scpm active` 來確認。設定檔 `default` 是基本組態，其他設定檔均從此衍生。因此，應該先進行在所有設定檔中均相同的設定。然後將這些修改以 `scpm reload` 儲存在作用中的設定檔中。`default` 設定檔可複製並重新命名作為新設定檔的基礎。

有兩種方法可新增設定檔。如果新設定檔 (在此處名為 `work`) 要以設定檔 `default` 為基礎，請以 `scpmcopy default work` 建立它。指令 `scpmswitch work` 可變為新設定檔，然後便可修改。您可能基於某些特殊原因想要修改系統組態，以及將變更儲存到新設定檔。指令 `scpmadd work` 可藉由將目前系統組態儲存在設定檔 `work` 中，並將它標示為作用中來建立新設定檔。執行 `scpm reload`，然後將變更儲存到 `work` 設定檔。

可使用指令 `scpmrename x y` 和 `scpmdelete z` 來重新命名或刪除設定檔。例如，若要將 `work` 重新命名為 `project`，請輸入 `scpmrename work project`。若要刪除 `project`，請輸入 `scpm delete project`。作用中的設定檔無法刪除。

20.3.3 切換組態設定檔

`scpm switch work` 指令可切換到另一個設定檔 (在此例中為 `work` 設定檔)。切換到作用中的設定檔可將系統組態修改的設定包括在設定檔中。這等同於指令 `scpmreload`。

在切換設定檔時，SCPM 首先檢查作用中設定檔的哪些資源被修改。然後查詢每一個資源的修改是否應新增到作用中的設定檔或捨棄。如果您偏好分別列出資源 (如先前版本的 SCPM)，請以 `-r` 參數使用切換指令：`scpm switch -r work`。

```
scpm switch -r work
```

檢查修改的資源

檢查要啟動或關閉的資源

檢查相依性

還原設定檔預設

然後 SCPM 會將目前系統組態與要切換的設定檔相比較。在此階段中，SCPM 會評估因為相互依存性或反映組態中變更而要停止或重新啟動的系統服務。這就像僅涉及小部份系統的部份系統重新開機，而沒有變更的其他部份仍可繼續運作。只有在此時系統服務會停止、所有修改的資源 (如組態檔案) 會寫入，而系統服務會重新啟動。

20.3.4 進階設定檔設定

您可以對每一個以 `scpmlist` 顯示的設定檔輸入描述。對於作用中的設定檔，以 `scpm set description "文字"` 設定。例如，以 `scpm set description "text" work` 提供非作用中設定檔的名稱。有時您可能想在切換設定檔時執行 SCPM 未提供的其他動作。最多可對每一個設定檔附加四個執行檔。可在切換過程的不同階段呼叫。這些階段有：

prestop

當離開設定檔時，在停止服務之前

poststop

當離開設定檔時，在停止服務之後

prestart

當啟用設定檔時，在啟動服務之前

poststart

當啟用設定檔時，在啟動服務之後

輸入 `scpm set prestop filename`、`scpm set poststop filename`、`scpm set prestart filename` 或 `scpm set poststart filename` 以 `set` 指令插入這些動作。程序檔必須為執行檔，並參考正確的解譯器。

警告: 整合自訂程序檔

其他可由 SCPM 執行的程序檔必須為超級使用者 (root) 可讀取且可執行。必須對其他所有使用者封鎖這些檔案的存取權。輸入 `chmod 700 filename` 與 `chown root:root filename` 以給予 root 對檔案的專用許可權。

以 `set` 與 `get` 查詢所輸入的其他所有設定。例如, `scpm get poststart` 指令可傳回 `poststart` 呼叫的名稱, 而若沒有附加則不會傳回任何內容。以 `"` 覆寫來重設此設定。`scpm set prestop "` 指令會移除附加的 `prestop` 程式。

所有 `set` 和 `get` 指令均可使用新增註解的方式套用於任意設定檔。例如, `scpm get prestop filename work` 或 `scpm get prestop work`。

20.4 使用 Profile Chooser Applet

GNOME 或 KDE 桌面面板中的 Profile Chooser Applet 可讓您輕鬆控制 SCPM 設定。透過 YaST 建立、修改或刪除設定檔 (如 [節 20.2, "使用 YaST 設定檔管理員" \[236\]](#) 所述), 以及切換設定檔。只要系統管理員允許一般使用者切換設定檔, 一般使用者就可以這樣做。使用系統 → 桌面 Applet → 設定檔選擇器, 從桌面啟動「設定檔選擇器」。

以滑鼠右鍵按一下桌面上的「設定檔選擇器」圖示, 並在開啟的功能表中選擇 *允許使用者切換*, 即可讓一般使用者切換設定檔。提供根密碼。從現在起, 系統中的任何已授權使用者都可以切換設定檔。

按一下「設定檔選擇器」圖示之後, 就會顯示在 YaST 中設定的所有設定檔, 無論這些設定檔是直接透過 YaST 呼叫所設定, 或是透過啟動 *YaST2 設定檔管理員模組* 所設定。使用游標按鍵來選取要切換的設定檔, SCPM 就會自動變更為新的設定檔。

20.5 疑難排解

本節涵蓋 SCPM 常遭遇的問題。請瞭解它們是如何產生的以及如何才能解決這些問題。

20.5.1 在切換過程中終止

有時 SCPM 在切換程序期間會停止工作。這可能是某些外部效果所造成，例如：使用者中止、電源中斷，或甚至 SCPM 本身的錯誤。如果發生，在下一次啟動 SCPM 時會顯示錯誤訊息表示 SCPM 已鎖定。這是基於系統安全，因為儲存在其資料庫中的資料可能和系統狀態不同。若要解決此問題，請執行 `scpm recover`。SCPM 會執行前一次執行的所有遺失作業。您也可以執行 `scpm recover -b`，它嘗試恢復前一次執行的所有已執行作業。如果您正在使用 YaST 設定檔管理員，將會在啟動時復原對話方塊，以執行上方所述的指令。

20.5.2 變更資源群組組態

若要在 SCPM 已經起始化時修改資源群組的組態，請在新增或移除群組之後輸入 `scpmrebuild`。利用此方法，新資源可新增到所有設定檔，而移除的資源會永久刪除。如果刪除的資源在不同設定檔中設定不同，此組態資料會遺失(除了系統中目前的版本之外，因為 SCPM 不會接觸到它)。如果您以 YaST 修改組態，不需要輸入重建指令，因為 YaST 會處理。

20.6 在系統開機時選取設定檔

若要在系統開機時選取設定檔，請在開機畫面時按下[F3]以存取可用設定檔的清單。使用方向鍵選取設定檔並以[Enter]確認選擇。然後選取的設定檔便可作為開機選項使用。

20.7 若需詳細資訊

SCPM info 頁面 (`info scpm`) 有最新的文件。給開發人員的資訊可在 `/usr/share/doc/packages/scpm` 中找到。

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。有兩個技術可：APM (進階電源管理，Advanced Power Management) 和 ACPI (進階組態與電源介面，Advanced Configuration and Power Interface)。除此之外，也可以控制 CPU 頻率比例，這有助於省電及降低噪音。您可以手動或使用特殊 YaST 模組，設定這些選項。

APM 之前只用於筆記型電腦的電源管理上，而硬體資訊與組態工具 ACPI 則可用於所有目前的電腦中 (筆記型電腦、桌上型電腦及伺服器)。電源管理技術都需要合適的硬體與 BIOS 常式。多數的筆記型電腦與目前許多桌上型電腦及伺服器都符合這項需求。

過去許多電腦都是使用 APM。由於 APM 大部份是由實作在 BIOS 中的一組功能所組成，使得 APM 支援的等級會因硬體而異。ACPI 更加有此特性，其組成也更加複雜。因此，實際上是無法說明哪種技術比較好。只有對硬體進行各種測試，然後再選出支援性最好的技術。

重要: AMD64 處理器的電源管理技術

AMD64 處理器具有 64 位元的核心，僅支援 ACPI。

21.1 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。這些主要功能在電源管理 APM 和 ACPI 中有下列用途：

待命

此操作模式關閉畫面顯示。在某些電腦上，會調節 (Throttling) 處理器的效能。並不是所有的 APM 實行都具有此功能。此功能等同於 ACPI 狀態 S1 或 S2。

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。使用 APM 的設備通常會在蓋子關閉時暫停，而在打開蓋子時啟動。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。

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。要從此狀況重新啟動需耗時 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 APM 與 ACPI 之外的核心常式來執行。

電池監視器

ACPI 與 APM 能檢查電池的變化狀態，並提供相關資訊。此外，兩個系統都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。

關閉系統元件

就整個系統而言，關閉硬碟是最省電的方式。依整個系統的穩定度而異，有時可讓硬碟進入休眠狀態。不過，睡眠時期的持續時間愈長，遺失資料的風險愈大。其他元件可透過 ACPI 停用 (至少在理論上)，或是在 BIOS 設定中永久停用。

處理器速度控制

與 CPU 有關的三種省電方式如下：頻率和電壓比例 (也稱為 PowerNow! 或 Speedstep)、調節 (Throttling) 以及讓處理器進入睡眠 (C 狀態)。依據電腦的操作模式，也可以合併這些操作方法。

21.2 APM

APM BIOS 本身能執行一些省電功能。許多筆記型電腦則可透過組合鍵或關閉蓋子等方式來啟動待命狀態或暫停狀態，而不需任何特別地操作系統功能。不過，如果要以指令啟動這些模式，必須先觸發一些動作才能使系統暫停。如果要檢視電池充電程度，需要有特別的程式套件與適用的核心。

SUSE Linux 核心內建即支援 APM。不過，APM 只在 ACPI 未在 BIOS 中實行以及偵測到 APM BIOS 時才會啟動。如果要啟動 APM 支援，必須在開機提示中輸入 `acpi=off` 以關閉 ACPI。請輸入 `cat /proc/apm` 以檢查 APM 是否啟動。輸出中含有各種數字的話，表示一切正常。現在若使用 `shutdown -h` 指令，應可關閉電腦。

BIOS 實行若未能完全符合標準則會導致 APM 發生問題。有些問題可以使用特定的開機參數來加以避免。所有參數是以 `apm=parameter` 的形式輸入於開機提示中：

on or off

開啟或關閉 APM 支援。

(no-)allow-ints

允許在 BIOS 功能執行期間出現中斷。

(no-)broken-psr

BIOS 的 "GetPowerStatus" 功能無法正常運作。

(no-)realmode-power-off

在關機前，重新設定處理器至真實模式。

(no-)debug

在系統記錄中記錄 APM 事件。

(no-)power-off

在關機後關閉系統電源。

bounce-interval=n

暫停事件發生後，要經過多少時間才能執行下個暫停事件，該時間以 0.01 秒為單位。

idle-threshold=*n*

執行 BIOS 的 `idle` 功能後，系統閒置的百分比 (0=永遠，100=永不)。

idle-period=*n*

系統啟動後所經過的時間，該時間以 0.01 秒為單位。

APM 精靈 (`apmd`) 已停用。其功能目前是由新的 `powersaved` 處理，且能支援 ACPI 和 CPU 頻率比例。

21.3 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。作業系統會使用這此資訊來執行工作，像是指定中斷或啟動和關閉元件。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 [節 21.3.4, "疑難排解" \[252\]](#)，以取得更多有關 ACPI 問題疑難排解的資訊。

21.3.1 ACPI 的動作

如果核心在系統啟動時偵測到 ACPI BIOS，會自動啟動 ACPI 並關閉 APM。有些較舊的機器可能需要用到開機參數 `acpi=force`。電腦需支援 ACPI 2.0 或以後的版本。請檢查 `/var/log/boot.msg` 中的核心開機訊息，以查看 ACPI 是否啟動。

接著，需載入一些模組。這會由省電精靈的啟動程序檔來完成。如果其中任何一個模組導致問題發生，該項模組就不會在 `/etc/sysconfig/powersave/common` 中載入或取消載入。系統記錄 (`/var/log/messages`) 內有模組的訊息，從中可以知道已偵測到哪些元件。

`/proc/acpi` 目前含有一些檔案，用來提供有關系統狀態的資訊，也可以用來對某些狀態進行變更。有些功能還不能使用，因為仍在開發中，而且有些功能的支援主要是依靠製造商是否在產品中實行。

所有檔案 (dsdt 與 fadt 除外) 都可使用 `cat` 來讀取。有些檔案可使用 `echo` 來修改設定，例如，`echo X > file` 可為 `X` 指定適合的值。此外，務必使用 `powersave` 指令來存取此資訊以及控制選項。最重要的檔案說明如下：

/proc/acpi/info

有關 ACPI 的一般資訊。

/proc/acpi/alarm

在此指定何時從睡眠狀態中喚醒系統。目前，尚未完整支援此功能。

/proc/acpi/sleep

提供可能的睡眠狀態的相關資訊。

/proc/acpi/event

所有事件都會會在此報告，並經由 Powersave 精靈來加以處理 (powersaved)。如果沒有精靈存取此檔案，快速按一下電源按鈕或是關閉蓋子等事件，可使用 `cat /proc/acpi/event` 來讀取 (按 `[Ctrl] + [C]` 來終止)。

/proc/acpi/dsdt 與 /proc/acpi/fadt

這些檔案包含 ACPI 的 (不同系統說明表格) DSDT 表格和 (固定 ACPI 說明表格) FADT 表格。可以使用 `acpidmp`、`acpidisasm` 與 `dmdecode` 來讀取它們。這些程式及其文件位於 `pmtools` 套件中。例如，`acpidmpDSDT | acpidisasm`。

/proc/acpi/ac_adapter/AC/state

顯示是否已連接 AC 轉換器。

/proc/acpi/battery/BAT*/{提醒、資訊、狀態}

有關電池狀態的詳細資訊。透過比較資訊中的上次完整電量以及狀態中的剩餘電量來讀取充電等級。另一種更方便的方法，便是使用在 [節 21.3.3, "ACPI 工具"\[252\]](#) 中所介紹的特別程式。可在提醒中指定觸發電池事件的充電等級。

/proc/acpi/button

此目錄包含各種切換的資訊。

/proc/acpi/fan/FAN/state

顯示風扇是否正在運作。以手動方式透過在此檔案中寫入 0 (開啟) 或 3 (關閉)，以開啟或關閉風扇。不過，在過熱時，核心與硬體(或 BIOS)中的 ACPI 程式碼都會覆寫此設定。

/proc/acpi/processor/*

系統中的每個 CPU 都有個別的子目錄。

/proc/acpi/processor/*/info

關於處理器的省電選項的資訊。

/proc/acpi/processor/*/power

關於目前處理器狀態的資訊。在 C2 旁邊有星號表示處理器閒置中。查看使用值時，這是最常出現的狀態。

/proc/acpi/processor/*/throttling

能用來設定調節處理器的時脈。通常，調節可以有八個層級。這和 CPU 的頻率控制是兩回事。

/proc/acpi/processor/*/limit

如果是由精靈來自動控制效能(過時)及調節功能，則可在此指定最大上限。部份限制是由系統所決定。部份則可由使用者來調整。

/proc/acpi/thermal_zone/

每個溫度區都有一個子目錄。溫度區是指一個具有類似的溫度屬性的區域，由硬體製造商指定區域的數值及名稱。不過大部份由 ACPI 所提供的可能選項，很少被實作。通常反而是使用 BIOS 來控制溫度。作業系統很少有干預的機會，因為這可能會減少硬體的使用壽命。因此，有一部份的檔案僅具理論價值。

/proc/acpi/thermal_zone/*/temperature

溫度區目前的溫度。

/proc/acpi/thermal_zone/*/state

指示是否一切都正常，或 ACPI 是否套用主動或被動冷卻規則。如果風扇控制獨立於 ACPI 之外，則狀態會一直是正常。

/proc/acpi/thermal_zone/*/cooling_mode

選取由 ACPI 控制的冷卻方法。從被動(效能較低，經濟型)選擇或是作用中的冷卻模型(完整的效能、風扇噪音)。

`/proc/acpi/thermal_zone/*/trip_points`

啟用判定溫度上限以觸發指定動作的功能，像是被動或主動冷卻、暫停 (過熱) 或關機 (嚴重)。在 DSDT (視設備而有所不同) 中定義可能動作。在 ACPI 指定中的啟動點分別是嚴重、過熱、被動、主動 1 及主動 2。即使不會每個都會實作，仍必須依此順序將它們輸入在此檔案中。例如，`echo 90:0:70:0:0 > trip_points` 項目即是將嚴重的溫度設為 90，將被動設為 70 (以攝氏測量的所有溫度)。

`/proc/acpi/thermal_zone/*/polling_frequency`

如果在 `temperature` 檔案中的值，未在溫度變更時自動更新，請在此切換輪詢模式。`echoX >`

`/proc/acpi/thermal_zone/*/polling_frequency` 指令能限定每 X 秒查詢一次溫度。設定 `X=0` 以關閉輪詢。

這些設定、資訊及事件都不需以手動方式編輯。可以使用 Powersave 精靈 (powersaved) 和各種應用程式來完成，像是 powersave、kpowersave 及 wmpowersave。請參閱 [節 21.3.3, "ACPI 工具" \[252\]](#)。因為 powersaved 包含較舊的 acpid 功能，所以已不再需要 acpid。

21.3.2 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

AMD 與 Intel 分別以 PowerNow! 與 Speedstep 來實作此技術。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 APM 或 ACPI 無關，需使用精靈來調整適用的頻率及目前所需的效能。您可以在 `/sys/devices/system/cpu/cpu*/cpufreq/` 目錄中進行設定。

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。共有三種狀態：C1、C2 及 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。通常會使用精靈來控制動態頻率比例，例如 `powersaved` 是最佳方法。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。

在 SUSE Linux 中，這些技術是由省電精靈所控制。組態的說明位於 [節 21.5, "powersave 套件" \[255\]](#)。

21.3.3 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 `wmacpimon` 等等。)、協助在 `/proc/acpi` 中存取架構或協助監控變更 (`akpi`、`acpiw`、`gtkacpiw`) 的工具，以及在 BIOS 中編輯 ACPI 表格的工具 (`pmtools` 套件)。

21.3.4 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=oldboot

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。

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 節 21.5.4, "疑難排解" [260] 中描述。

在核心組態中，有個啟動 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。

若需詳細資訊

ACPI 的其他文件和說明：

- <http://www.cpqlinux.com/acpi-howto.html> (詳細的 ACPI HOWTO，內含 DSDT 修補程式)
- <http://www.intel.com/technology/ia32/acpi/faq.htm> (ACPI FAQ @Intel)
- <http://acpi.sourceforge.net/> (Sourceforge 的 ACPI4Linux 計劃)
- <http://www.poupinou.org/acpi/> (Bruno Ducrot 的 DSDT 修補程式)

21.4 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。powersaved 和 YaST 電源管理模組能控制大部份這方面的功能，詳細資訊請參閱 節 21.6, "YaST 電源管理模組" [262]。

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。可將 x 換成以下內容：0 能關閉此機制，讓硬碟持續運作。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。核心更新精靈 (kupdated) 可以監控緩衝區。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。根據預設，kupdated 會設成較短的間隔，以最大化資料的完整性。它會每 5 秒檢查一次緩衝區，並會在資料時間大於 30 秒或緩衝區的容量已達 30% 時通知 bdflush。接著，bdflush 精靈會將資料寫入硬碟。它也可以在不經由 kupdated 之下寫入硬碟，例如，在緩衝區已滿時。

警告: 損害資料完整性

變更核心更新精靈的設定有害資料的完整性。

除了這些程序之外，像是 ReiserFS 與 Ext3 等日誌檔案系統，不經由 bdflush 會將中繼資料寫入硬碟，也會使得硬碟無法停止運作。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。不過，若增加 `kupdated` 的時間間隔，則此項設定將無關緊要。

21.5 powersave 套件

`powersave` 套件負責膝上型電腦在電池操作時的省電功能。它的部份功能對於一般工作站和伺服器也很有用，例如暫停、待命、ACPI 按鈕功能和使 IDE 硬碟休眠等。

此套件包含電腦的所有電源管理功能。它支援使用 ACPI、APM、IDE 硬碟和 PowerNow! 或 SpeedStep 技術的硬體。來自套件 `apmd`、`acpid`、`ospm` 和 `cpufreqd` (現在為 `cpuspeed`) 的功能已合併於 `powersave` 套件中。來自這些套件的精靈應該無法與 `powersave` 精靈同時執行。

即使您的系統不包含上述所有硬體元件，仍可以使用 `powersave` 精靈來控制省電功能。因為 ACPI 和 APM 互斥，僅可在電腦上使用其中一種系統。精靈會自動偵測硬體組態的變化。

21.5.1 設定 powersave 套件

一般而言，`powersave` 的組態散佈於數個檔案：

`/etc/sysconfig/powersave/common`

此檔案包含 `powersave` 精靈的一般設定。例如，可以增加變數 `DEBUG` 的值以增加 `/var/log/messages` 中的偵錯訊息數量。

`/etc/sysconfig/powersave/events`

`powersave` 精靈需要此檔案以處理系統事件。事件可以是指定的外部動作或精靈本身執行的動作。對於外部動作，精靈會嘗試執行 `/usr/lib/powersave/scripts/` 中的執行檔。預先定義內部動作：

- `ignore`

- throttle
- dethrottle
- suspend_to_disk
- suspend_to_ram
- standby
- do_suspend_to_disk
- do_suspend_to_ram
- do_standby

throttle 以 MAX_THROTTLING 中定義的值減緩處理器。此值視目前的配置而定。dethrottle 將處理器設為完整效能。suspend_to_disk、suspend_to_ram 和 standby 會觸發休眠模式的系統事件。這三個動作一般負責觸發休眠模式，但是它們應該永遠與特定系統事件關聯。

目錄 /usr/lib/powersave/scripts 包含處理事件的程序檔：

notify

藉由主控台、X 視窗或柔和式聲響訊號來通知事件。

screen_saver

啟用螢幕保護裝置。

switch_vt

在暫停或待命之後，如果螢幕發生錯置時會很有用。

wm_logout

儲存設定並從 GNOME、KDE 或其他視窗管理員登出。

wm_shutdown

儲存 GNOME 或 KDE 設定並關閉系統。

例如，如果設定

EVENT_GLOBAL_SUSPEND2DISK="prepare_suspend_to_disk
do_suspend_to_disk" 變數，兩個程序檔或動作會在使用者給 powersave

休眠模式指令 `suspend to disk` 時，立即以特定順序處理。精靈會執行外部程序檔 `/usr/lib/powersave/scripts/prepare_suspend_to_disk`。在此程序檔成功處理之後，精靈會執行內部動作 `do_suspend_to_disk`，並在程序檔卸載關鍵模組並停止服務之後，將電腦設為休眠模式。

休眠按鈕事件的動作可在 `EVENT_BUTTON_SLEEP="notify suspend_to_disk"` 中修改。在此例中，外部程序檔 `notify` 會通知使用者暫停。之後，會產生 `EVENT_GLOBAL_SUSPEND2DISK` 事件，導致執行上述動作及安全系統暫停模組。可使用 `/etc/sysconfig/powersave/common` 中的變數 `NOTIFY_METHOD` 自訂程序檔 `notify`。

`/etc/sysconfig/powersave/cpufreq`

包含最佳化動態 CPU 頻率設定的變數。

`/etc/sysconfig/powersave/battery`

包含電池限制和其他電池特定設定。

`/etc/sysconfig/powersave/sleep`

在此檔案中，啟用休眠模式並決定應卸載哪個關鍵模組，以及在暫停或待命事件之前應停止的服務。當系統繼續時，這些模組會重新載入並啟動服務。例如，您甚至可以延遲觸發的休眠模式以儲存檔案。預設值主要考量 USB 和 PCMCIA 模組。特定模組通常會造成暫停或待命的錯誤。請參閱 [節 21.5.4, "疑難排解" \[260\]](#) 以取得關於辨識錯誤的詳細資訊。

`/etc/sysconfig/powersave/thermal`

啟用冷卻和熱控制。關於此主題的詳細資訊也可在檔案 `/usr/share/doc/packages/powersave/README.thermal` 中找到。

`/etc/sysconfig/powersave/scheme_*`

這些為搭配特定部署狀況電源消耗的不同配置。數種配置已預先設定好並可供使用。可在此儲存自訂配置。

21.5.2 設定 APM 和 ACPI

暫停和待命

根據預設，休眠模式未啟用，因為它們在部份電腦上沒有作用。有三種基本 ACPI 休眠模式和兩種 APM 休眠模式：

Suspend to Disk (ACPI S4、APM 暫停)

將全部記憶體內容儲存到硬碟。電腦完全關閉，不消耗任何電力。

Suspend to RAM (ACPI S3、APM 暫停)

將所有設備的狀態儲存到主要記憶體。僅主要記憶體繼續消耗電力。

Standby (ACPI S1、APM 待命)

關閉部份設備 (視製造商而定)。

請確定檔案 `/etc/sysconfig/powersave/events` 中的下列預設選項設為暫停、待命和繼續的正確處理 (預設值為在安裝 SUSE Linux 之後)：

```
EVENT_GLOBAL_SUSPEND2DISK=
    "prepare_suspend_to_disk do_suspend_to_disk"
EVENT_GLOBAL_SUSPEND2RAM=
    "prepare_suspend_to_ram do_suspend_to_ram"
EVENT_GLOBAL_STANDBY=
    "prepare_standby do_standby"
EVENT_GLOBAL_RESUME_SUSPEND2DISK=
    "restore_after_suspend_to_disk"
EVENT_GLOBAL_RESUME_SUSPEND2RAM=
    "restore_after_suspend_to_ram"
EVENT_GLOBAL_RESUME_STANDBY=
    "restore_after_standby"
```

自訂電池狀態

在檔案 `/etc/sysconfig/powersave/battery` 中，定義三種電池充電強度 (以百分比)，當達到時會觸發警示或特定動作。

```
BATTERY_WARNING=20
BATTERY_LOW=10
BATTERY_CRITICAL=5
```

當充電強度掉到特定限制之下，要執行的動作或程序檔定義於組態檔 `/etc/sysconfig/powersave/events` 中。按鈕的標準動作可以修改為如 [節 21.5.1, "設定 powersave 套件" \[255\]](#) 中所述。

```
EVENT_BATTERY_NORMAL="ignore "  
EVENT_BATTERY_WARNING="notify "  
EVENT_BATTERY_LOW="notify "  
EVENT_BATTERY_CRITICAL="wm_shutdown "
```

將電源消耗與不同條件搭配

系統行為可與電源供應類型搭配。當系統從 AC 電源供應中斷連線並以電池操作時，系統的電源消耗應減少。相似地，系統一連線到 AC 電源供應時，效能應自動增加。CPU 頻率、IDE 的省電功能和許多其他參數均可修改。

當電腦連線到或從 AC 電源供應中斷連線時，要執行的動作定義於 `/etc/sysconfig/powersave/events` 中。請在 `/etc/sysconfig/powersave/common` 中選取要使用的配置：

```
AC_SCHEME="performance "  
BATTERY_SCHEME="powersave "
```

配置儲存在 `/etc/sysconfig/powersave` 中的檔案內。檔案名稱的格式為 `scheme_name-of-the-scheme`。範例參考兩個配置：`scheme_performance` 和 `scheme_powersave`。`performance`、`powersave`、`presentation` 和 `acoustic` 為預先設定。藉由 [節 21.6, "YaST 電源管理模組" \[262\]](#) 中描述的 YaST 電源管理模組的協助，可編輯、建立、刪除現有配置或和不同電源供應狀態相關聯。

21.5.3 其他 ACPI 功能

如果您使用 ACPI，可以控制系統對 *ACPI 按鈕* (電源、休眠、開蓋、關蓋) 的回應。請在 `/etc/sysconfig/powersave/events` 中設定動作的執行。請參考此組態檔以取得個別選項的說明。

EVENT_BUTTON_POWER="wm_shutdown"

當按下電源按鈕時，系統回應為關閉對應視窗管理員 (KDE、GNOME、fvwm 等)。

EVENT_BUTTON_SLEEP="suspend_to_disk"

當按下休眠按鈕時，系統設為休眠 (suspend-to-disk) 模式。

EVENT_BUTTON_LID_OPEN="ignore"

當開啟蓋子時，不發生任何事。

EVENT_BUTTON_LID_CLOSED="screen_saver"

當關上蓋子時，會啟動螢幕保護裝置。

如果 CPU 負載在特定時間未超過特定限制，則可以進一步節流 CPU 效能。在 `PROCESSOR_IDLE_LIMIT` 中指定負載限制，在 `CPU_IDLE_TIMEOUT` 中指定逾時。如果 CPU 負載未超過限制的時間大於逾時，則會啟動 `EVENT_PROCESSOR_IDLE` 中設定的事件。如果 CPU 再度忙碌，便執行 `EVENT_PROCESSOR_BUSY`。

21.5.4 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。下列小節涵蓋 `powersave` 最常見的問題。

以硬體支援啟動 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。請到您膝上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/tables> 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` (pmttools 套件) 編譯。請輸入 `iasl -sa file.asl` 指令。最新版本的 `iasl` (Intel

ACPI 編譯器) 可在 <http://developer.intel.com/technology/iapc/acpi/downloads.htm> 中找到。

- 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。

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。如果需要特殊核心模組或模組選項，可在檔案 /etc/sysconfig/powersave/cpufreq 中藉由變數 CPUFREQD_MODULE 和 CPUFREQD_MODULE_OPTS 加以設定。

暫停和待命沒有作用

有數種核心相關問題會使 ACPI 系統的暫停和待命無法使用：

- 目前超過 1 GB RAM 的系統不支援暫停。
- 目前多處理器系統和 P4 處理器 (具有超執行緒) 的系統不支援暫停。

錯誤也可能是導因於錯誤的 DSDT 執行方式 (BIOS)。如果是此狀況，請安裝新的 DSDT。

在 ACPI 和 APM 系統上：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。在此狀況下，在 /var/log/sleep mode 中的 powersave 精靈所產生的記錄檔非常有用。如果電腦無法進入休眠模式，原因在於最後卸載的模組。請操作 /etc/sysconfig/powersave/sleep 中的下列設定，在暫停或待命之前卸載有問題的模組。

```
UNLOAD_MODULES_BEFORE_SUSPEND2DISK=""
UNLOAD_MODULES_BEFORE_SUSPEND2RAM=""
UNLOAD_MODULES_BEFORE_STANDBY=""
SUSPEND2DISK_RESTART_SERVICES=""
```

```
SUSPEND2RAM_RESTART_SERVICES=""  
STANDBY_RESTART_SERVICES=""
```

如果您在變動網路環境中或以遠端裝載檔案系統 (如 Samba 和 NIS) 的連線使用暫停或待命，請在上述變數中使用自動裝載器以裝載或新增對應服務，例如 `smbfs` 或 `nfs`。如果應用程式在暫停或待命之前存取遠端裝載檔案系統，則服務無法正確停止，而檔案系統無法正確取消裝載。在系統繼續之後，檔案系統可能損毀並必須重新裝載。

使用 ACPI，而 Powersave 未注意到電池限制

使用 ACPI，作業系統可以要求 BIOS 在電池充電強度掉到特定限制之下時傳送訊息。此方法的好處在於不需要持續輪詢電池狀態，因為這可能損害電腦的效能。但是，當充電強度掉到特定限制之下時，即使假設 BIOS 支援此功能，仍可能不發生此通知。如果系統發生此狀況，請將檔案 `/etc/sysconfig/powersave/battery` 中的變數 `FORCE_BATTERY_POLLING` 設為 `yes` 以強迫電池輪詢。

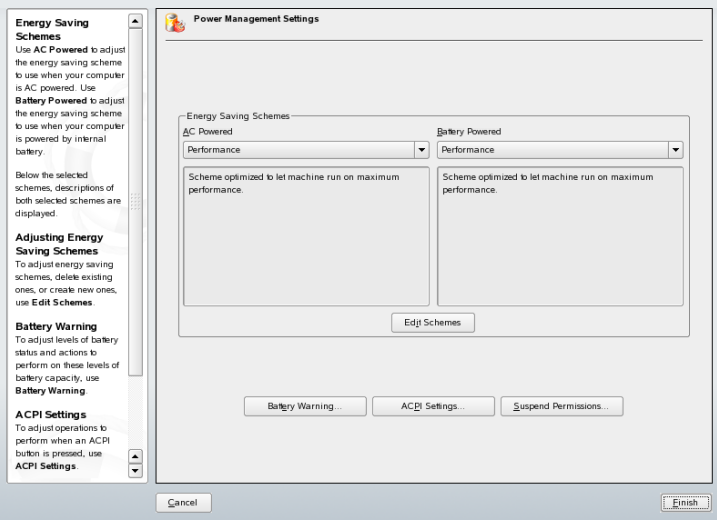
21.5.5 若需詳細資訊

關於 `powersave` 套件的資訊也可以在 `/usr/share/doc/packages/powersave` 中找到。

21.6 YaST 電源管理模組

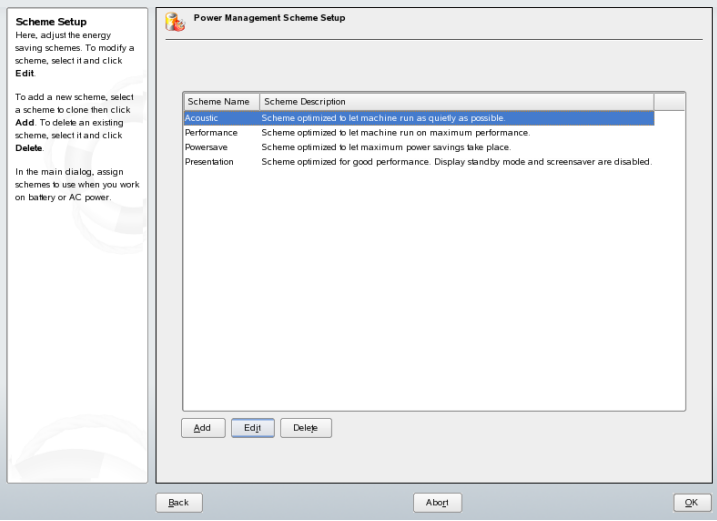
YaST 電源管理模組可以設定先前描述的所有電源管理設定。在您以 **系統 → 電源管理** 從「YaST 控制中心」啟動模組時，會開啟模組的第一個對話方塊。如圖形 21.1, "配置選取區域" [263] 所示。

圖形 21.1 配置選取區域



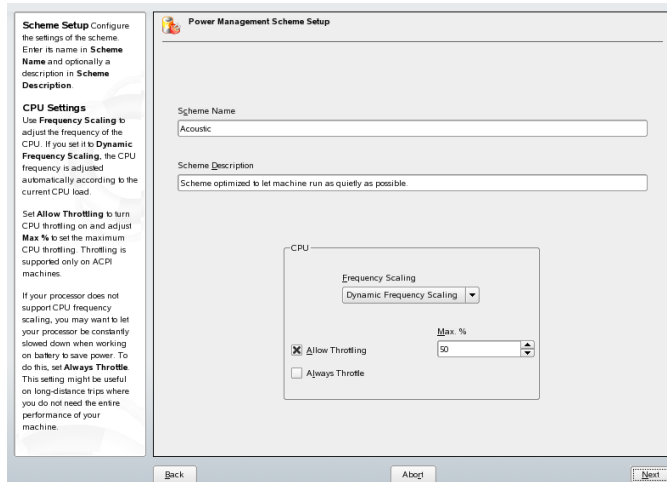
在此對話方塊中，選取電池操作和 AC 操作要使用的配置。若要新增或修改配置，請按一下 **編輯配置**，便可開啟如 [圖形 21.2, "現有配置的綜覽"](#) [263] 所示的現有配置綜覽。

圖形 21.2 現有配置的綜覽



在配置綜覽中，選取要修改的配置，再按一下 **編輯**。若要建立新的配置，請按一下 **新增**。兩種狀況開啟的對話方塊均相同，如 [圖形 21.3, "設定配置" \[264\]](#) 中所示。

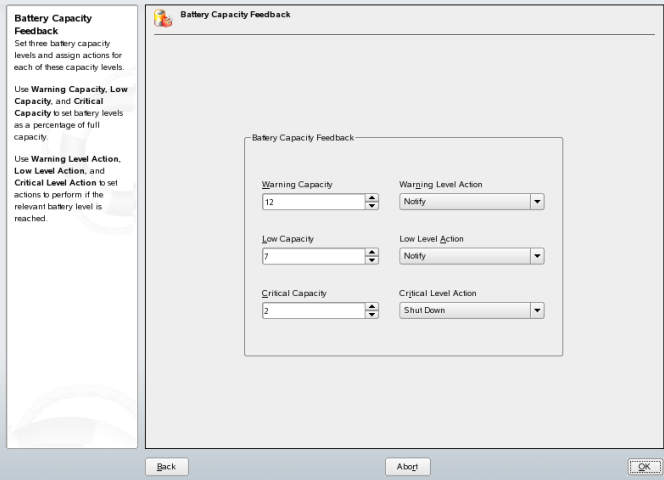
圖形 21.3 設定配置



首先，為新的或編輯的配置輸入適合的名稱和描述。判斷此配置是否應控制CPU效能及如何控制。決定是否應使用頻率比例和調節設定以及使用到什麼程度。在硬碟的下列對話方塊中，定義最大效能或省電的 **待命規則**。柔和式聲響規則控制硬碟的噪音強度 (少數硬碟支援)。冷卻規則 決定要使用的冷卻方法。不幸地，BIOS 很少支援此類型的熱控制。請讀取 `/usr/share/doc/packages/powersave/README.thermal` 以瞭解如何使用風扇和被動冷卻方法。

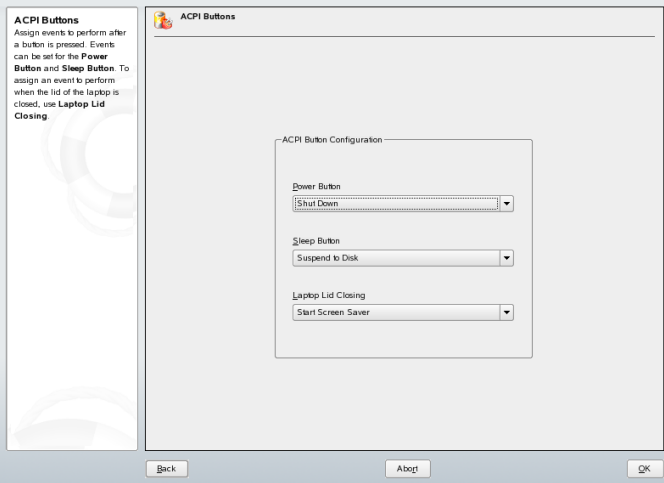
也可使用 **電池警告**、**ACPI 設定** 或 **啟用暫停** 從起始對話方塊進行全域電源管理設定。按一下 **電池警告** 以存取電池充電強度對話方塊，如 [圖形 21.4, "電池充電強度" \[265\]](#) 中所示。

圖形 21.4 電池充電強度



當充電強度掉到特定可設定的限制之下時，您系統的BIOS會通知作業系統。在此對話方塊中，定義三個限制：警告電量、低電量和極低電量。指定當充電強度掉到此限制之下時，要觸發的動作。通常，前兩種狀態僅觸發對使用者的通知。第三種重要強度會觸發關機，因為剩餘電力不足以繼續系統操作。選取適合的充電強度和想要的動作，然後按一下 **確定** 以回到開始對話方塊。

圖形 21.5 ACPI 設定



使用 *ACPI* 設定存取設定 *ACPI* 按鈕的對話方塊。如 [圖形 21.5, "ACPI 設定" \[265\]](#) 所示。*ACPI* 按鈕的設定決定系統對特定切換應如何回應。設定系統對按下電源按鈕、按下休眠按鈕和關閉膝上型電腦蓋子加以回應。按一下 *確定* 以完成組態並回到開始對話方塊。

按一下 *啟用暫停* 以進入對話方塊，決定此系統使用者是否可使用暫停或待命功能，以及使用方式。按一下 *確定* 以回到主要對話方塊。再按一下 *確定* 以結束模組並確認您的電源管理設定。

無線通訊

藉由使用 Linux 系統，您可以與其它電腦、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。本章節將介紹這三種技術及其組態設定。

22.1 無線區域網路

無線區域網路已成為行動電腦與網路不可或缺的一環。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。這些新標準對於類比調節、傳輸輸出及傳輸率等細節均有定義：

表格 22.1 WLAN 標準綜覽

名稱	頻段 (GHz)	最大傳輸率 (MBit/s)	附註
802.11	2.4	2	過時的；實際上無法取得終端設備
802.11b	2.4	11	普遍的
802.11a	5	54	較不普遍

名稱	頻段 (GHz)	最大傳輸率 (MBit/s)	附註
802.11g	2.4	54	與 11b 反向相容

此外還有一些專賣標準，像是德州儀器的 802.11b 變異標準，其最大傳輸率為 22 MBit/s (有時稱為 802.11b+)。不過使用此標準的網路卡數量有限。

22.1.1 硬體

SUSE Linux 不支援 802.11 卡。支援大部份使用 802.11a、802.11b 和 802.11g 的卡。新卡通常符合 802.11g 標準，但也有一些是 802.11b 的卡。一般來說，支援附有下列晶片的卡：

- Aironet 4500、4800
- Atheros 5210、5211 及 5212
- Atmel at76c502、at76c503、at76c504 及 at76c506
- Intel PRO/Wireless 2100、2200BG 及 2915ABG
- Intersil Prism2/2.5/3
- Intersil PrismGT
- Lucent/Agere Hermes
- Ralink RT2400、RT2500
- Texas Instruments ACX100 及 ACX111
- ZyDAS zd1201

也支援一些極少使用且無法取得的舊卡。在 http://www.linux-wlan.org/docs/wlan_adapters.html.gz 網站的 *AbsoluteValue Systems* 可以找到 WLAN 卡和 WLAN 卡所使用的晶片的延伸清單。<http://wiki.uni-konstanz.de/wiki/bin/view/Wireless/ListeChipsatz> 提供各種不同 WLAN 晶片的綜覽。

某些網路卡需要在啟動驅動程式時載入韌體影像。例如 Intersil PrismGT、Atmel 以及 TI ACX100 和 ACX111。使用 YaST 線上更新，即可輕易地安裝韌體。Intel PRO/Wireless 卡的韌體與 SUSE Linux 一起提供，並且會在偵測到此類型時，由 YaST 自動安裝。在裝好系統的 `/usr/share/doc/packages/wireless-tools/README.firmware` 可找到相關資訊。

您可以使用 `ndiswrapper` 應用程式來使用本來 Linux 不支援的卡。`Ndiswrapper` 會使用與多數 WLAN 卡一同提供的 Windows 驅動程式。若需有關 `ndiswrapper` 的說明，請參閱 `/usr/share/doc/packages/ndiswrapper/README.SUSE` (如果已安裝 `ndiswrapper` 套件)。若需 `ndiswrapper` 的進一步相關資訊，請參閱專案網站，網址為 <http://ndiswrapper.sourceforge.net/support.html>。

22.1.2 功能

在無線網路中，有各種不同的技術和組態，可用來確保快速、高品質的安全連線。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。

操作模式

基本上，無線網路可分為管理網路和臨機操作網路。受管理的網路具備的管理元件：存取點。在此模式(又稱為基礎結構模式)之下，網路上所有 WLAN 工作站的連線都會通過該存取點，此存取點亦可連線至乙太網路。臨機操作網路中沒有存取點。工作站之間直接進行通訊。臨機操作網路中的傳輸範圍及連線工作站的數目相當有限。因此，存取點的效率通常較高。WLAN 卡甚至可做為存取點。大部份的網路卡均支援此項功能。

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 "[安全性](#)" [275])，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 及其它驗證和加密方式。

驗證

管理網路使用各種不同的驗證機制，以確定只有獲得授權的工作站才可進行連線：

開放

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["加密" \[271\]](#))。

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。所有連線工作站及存取點都要有相同的金鑰。此金鑰長度為 256 位元，且通常以通關密語的方式輸入。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 提供三種不同的方法，用於連接至伺服器以及向伺服器進行驗證：TLS (Transport Layer Security)、TTLS (Tunneled Transport Layer Security) 以及 PEAP (Protected Extensible Authentication Protocol)。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，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。

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。

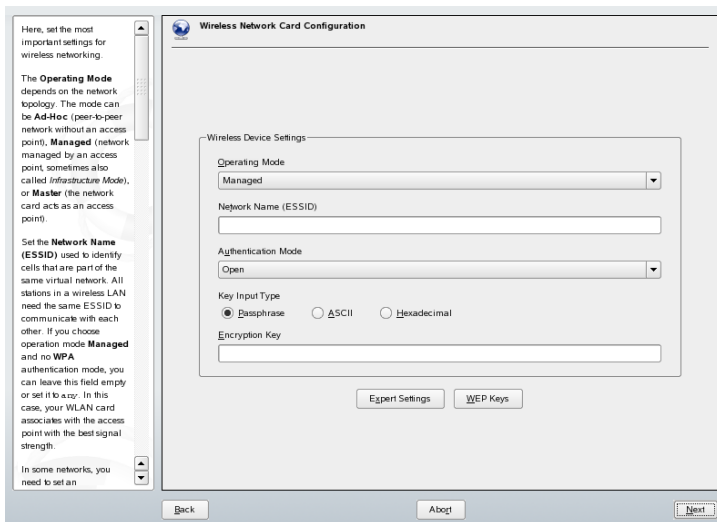
CCMP (定義於 IEEE 802.11i 中)

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。

22.1.3 利用 YaST 進行組態

若要設定您的無線網路卡，先啟動 YaST 網路卡模組。在 *網路位址設定* 中，選取設備類型 *無線* 然後按一下 *下一步*。在 *無線網路卡組態設定* 中 (顯示於 [圖形 22.1, "YaST: 設定無線網路卡" \[272\]](#) 中)，請指定 WLAN 的基本操作設定：

圖形 22.1 YaST: 設定無線網路卡



操作模式

工作站可用三種模式來連接 WLAN。依照要通訊的網路而定，適用不同的模式：**臨機操作模式**（沒有存取點的對等網路）、**管理模式**（由存取點管理網路）或**主要模式**（應該以您的網路卡為存取點）。若要使用任何 WPA-PSK 或 WPA-EAP 模式，就應該將作業模式設為 **管理模式**。

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。在未指定的情況下，網路卡會自動選擇一個存取點，該存取點可能不是您想要使用的。

驗證模式

為您的網路選取適用的驗證方式：**開放**、**共用金鑰**、**WPA-PSK** 或 **WPA-EAP**。如果您選取 WPA 驗證，則必須設定網路名稱。

進階設定

此按鈕會開啟一個對話方塊，說明 WLAN 連線的組態設定細節。稍後會提供此對話方塊的詳細說明。

完成基本設定之後，您的工作站即可部署在 WLAN 中。

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 連線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。請參閱 "[加密](#)" [271] 與 "[安全性](#)" [275] 以取得更多資訊。

視所選取的驗證方式而定，YaST 提示會要求您在其它對話方塊中微調您的設定。不需要設定 *開放* 的組態，因為它不需要驗證即可執行未加密的作業。

WEP 金鑰

設定金鑰輸入類型。選擇 *通關密語*、*ASCII* 或 *16 進位* 其中之一。您最多可使用四個金鑰來加密傳送的資料。按一下 *多個金鑰* 來開啟金鑰設定對話方塊。設定金鑰長度: *128 bit* 或 *64 bit*。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的金鑰作為預設金鑰。如果標準金鑰被刪除了，則必須手動標記其它金鑰為預設金鑰。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 (*通關密語*、*ASCII* 或 *16 進位*)。如果您選取了 *通關密語*，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 *通關密語* 輸入法或 *16 進位*。在 *通關密語* 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。

WPA-EAP

輸入網路管理員給您的證件。若為 TLS，請提供 *用戶端憑證* 和 *伺服器憑證*。TTLS 和 PEAP 需要 *識別* 和 *密碼*。*伺服器憑證* 是選用的。YaST 會在 `/etc/cert` 下搜尋任何證件，所以請將您的憑證儲存在這個位置，並將這些檔案的存取權限制為 0600 (擁有者讀取和寫入)。

按一下 *進階設定* 以離開 WLAN 連線的基本設定對話方塊，並進入進階組態。此對話方塊包含下列選項：

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 12 個通道中選取其中之一，讓您的工作站可與其它工作站通訊。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 [章 21, 電源管理 \[245\]](#)。

22.1.4 公用程式

hostap (hostap 套件) 用來使 WLAN 卡發揮存取點的功能。如需更多有關此套件的資訊，請參閱專案首頁 (<http://hostap.epitest.fi/>)。

kismet (kismet 套件) 是一種網路診斷工具，可用來傾聽 WLAN 的封包資料傳輸。您也可以藉此偵測您網路上任何侵入的企圖。如需更多相關資訊，請參閱 <http://www.kismetwireless.net/> 及手冊。

22.1.5 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，要看連線工作站是否能從其它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 公用程式或 KDE 中的 KInternet 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它地方，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，Linux 並不支援所有硬體元件上的 WPA。在開發此軟體時，WPA 只適用於使用 Atheros、Intel PRO/Wireless 或 Prism2/2.5/3 晶片的網路卡。就 Prism2/2.5/3 而言，只有在使用 hostap 驅動程式的情況下，才能使用 WPA (請參閱 "[Prism2 網路卡的問題](#)" [275])。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。

22.1.6 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。請參照 [節 22.1.1, "硬體" \[268\]](#)。以下段落為您說明一些可能出現的問題。

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 來設定這兩個設備 (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。<http://portal.suse.com> 的支援資料庫中有一篇與此相關的文章。如果要找這篇文章，請在搜尋對話方塊中輸入「DHCP」。

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`。

WPA

SUSE Linux 的新 WPA 支援仍處於開發階段。因此，YaST 並未支援所有 WPA 驗證方法的組態。並非所有無線 LAN 卡和驅動程式都支援 WPA。某些卡需要韌體更新才能啟用 WPA。如果您要使用 WPA，請參閱 `/usr/share/doc/packages/wireless-tools/README.wpa`。

22.1.7 若需詳細資訊

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。

22.2 藍芽

「藍芽」是連接各種設備(如：行動電話、PDA、週邊設備、膝上型電腦，或鍵盤或滑鼠等系統元件)的無線技術。此一名稱衍生自丹麥國王 Harold Bluetooth，他統一了斯堪的那維亞許多分裂的地區。「藍芽」標誌是以 "H" (像一顆星星) 和 "B" 的北歐文字為基礎。

可由許多重要方面辨識「藍芽」和 IrDA。首先，個別設備不需要直接「看到」彼此；其次，可將數個設備連線於一個網路中。但是，最大資料速率為 720 Kbps (目前的版本 1.2)。理論上，「藍芽」甚至可穿過牆面通訊。但是實際上，這要視牆面的材質和設備等級而定。有三種設備等級的傳輸範圍介於 10 至 100 公尺。

22.2.1 基本操作

下列小節簡述「藍芽」運作的基本原則。請瞭解需要符合哪些軟體需求，藍芽如何與系統互動，以及藍芽設定檔的運作方式。

軟體

若要使用「藍芽」，您需要「藍芽」介面卡(可以是內建介面卡或外部裝置)、驅動程式和「藍芽」通訊協定堆疊。Linux 核心已經包含使用「藍芽」的基本驅

動程式。Bluez 系統做為通訊協定堆疊使用。為了確保應用程式可與「藍芽」搭配使用，必須安裝基礎套件 `bluez-libs` 和 `bluez-utils`。這些套件提供許多所需的服務和公用程式。此外，部份介面卡，例如，Broadcom 或 AVM BlueFritz!，要求必須安裝 `bluez-firmware` 套件。`bluez-cups` 套件讓您可透過「藍芽」連線列印。

一般互動

「藍芽」系統由四個連鎖層所組成，提供需要的功能：

硬體

Linux 核心支援所需的介面卡和適合的驅動程式。

組態檔

用於控制「藍芽」系統。

精靈

組態檔案所控制並可提供功能的服務。

應用程式

應用程式允許使用者使用並控制精靈所提供的功能。

當您插入「藍芽」介面卡時，熱插拔系統會載入它的驅動程式。載入驅動程式之後，系統會檢查組態檔案以查看「藍芽」是否應該啟動。如果應該啟動，它會決定要啟動的服務。根據此資訊，會啟動對應的精靈。在安裝時會查探藍芽介面卡。如果找到一或多個介面卡，便會啟用藍芽。否則會停用藍芽系統。任何稍後新增的藍芽設備都必須以手動方式啟用。

設定檔

在「藍芽」中，服務是藉由設定檔(例如，檔案傳輸設定檔、基本列印設定檔和個人區域網路設定檔)所定義。若要讓設備使用其他設備的服務，兩者均必須了解相同設定檔，設備套件和手冊中通常缺少這項資訊。不幸地，部份製造商並未嚴格符合個別設定檔的定義。即使如此，設備之間的通訊通常可順暢運作。

在下列文字中，本地設備為實體上連線到電腦的設備。其他僅可透過無線連線存取的設備稱為遠端設備。

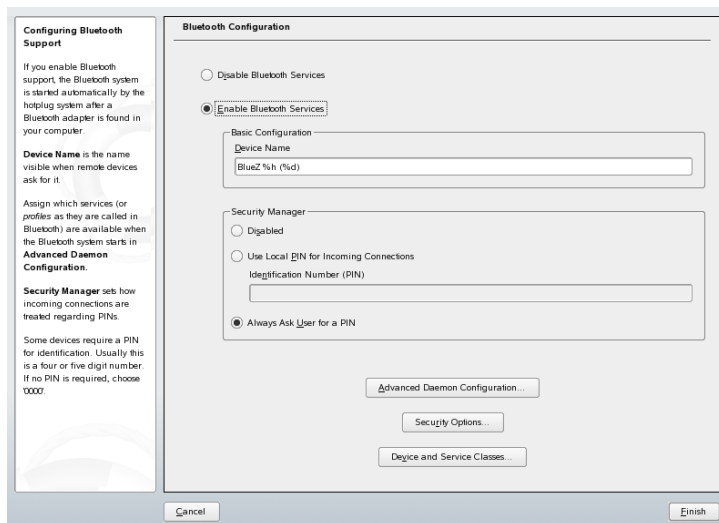
22.2.2 組態

此小節介紹藍芽的組態。請瞭解需要哪些組態檔、需要哪些工具以及如何以 YaST 或手動方式設定藍芽。

以 YaST 設定藍芽

使用「YaST 藍芽」模組 (請參閱圖 圖形 22.2, "YaST 藍芽組態" [278]) 來設定系統的「藍芽」支援。只要熱插拔一偵測到系統上的「藍芽」介面卡 (例如, 在開機期間或是在插入介面卡時), 便會自動以此模組中設定的設定值來啟動「藍芽」。

圖形 22.2 YaST 藍芽組態



在組態的第一步驟中, 決定是否要在系統上啟動「藍芽」服務。如果您已啟用藍芽服務, 可以設定兩個項目。首先, 裝置名稱。這是發現您的電腦時, 其他設備所顯示的名稱。一共有兩種可用的保留字元—%h 代表系統的主機名稱 (有用, 例如, 如果它是由 DHCP 動態指定), 而 %d 會插入介面編號 (如果您在電腦中有一個以上的「藍芽」介面卡才有用)。例如, 如果您在欄位中輸入 Laptop %h, 而 DHCP 指定 unit123 的名稱給您的電腦, 則其他遠端的設備將以 Laptop unit123 來識別您的電腦。

當遠端設備嘗試連線，安全管理員參數會與本機系統的行為相關。其差別在於處理 PIN 號碼。允許任何設備不需要 PIN 即可連線，或是判斷如何在需要 PIN 時選擇正確的 PIN。您可以在適當的輸入欄位輸入 PIN (儲存在組態檔案)。如果設備嘗試連線，它會先使用此 PIN。如果它失敗，它就會恢復不使用 PIN。就最佳的安全性而言，最好選擇永遠詢問使用者的 PIN。此選項允許您在不同 (遠端) 的設備上使用不同的 PIN。

然後按一下 *進階精靈組態* 以進入選取和設定可用服務的對話方塊 (在「藍芽」中也稱為「設定檔」)。所有可用的服務均顯示在清單中，可按一下 *啟用* 或 *停用* 加以啟用或停用。按一下 *編輯* 以開啟快顯對話方塊，您可以在其中指定選取服務 (精靈) 的其他引數。除非您對服務很熟悉，否則請勿做任何變更。完成精靈的組態之後，按一下 *確定* 以結束此對話方塊。

回到主要對話方塊，按一下 *安全性選項* 以進入安全性對話方塊，您可以在其中指定加密、驗證和掃描設定。然後結束安全性對話方塊以回到主要對話方塊。當您以完成關閉主要對話方塊之後，「藍芽」系統已經可以使用。

從主要對話方塊，也可以開啟 *設備和服務類別* 對話方塊。藍芽設備群組成不同的設備類別。在此對話方塊中，針對電腦選擇正確的類別，例如 *桌上型* 或 *膝上型*。設備類別與服務類別不同，並非很重要，同樣在此設定。有時像手機這一類的藍芽設備，如果它們可以偵測到您系統上所設定的正確服務類別，則只允許某些特定功能。這通常是發生在手機上，它們必須要有 *物件轉送* 的類別，才能從電腦轉送檔案或轉送檔案至電腦。您可以選擇多個類別。「為了以防萬一」而選取所有的類別是沒有太大幫助的。預設的選擇應該在大部份的情況都適用。

如果您想要使用「藍芽」來設定網路，請在 *進階精靈組態* 對話方塊中啟用 *PAND*，並以 *編輯* 設定精靈的模式。「藍芽」網路連線若要能運作，必須有一個 *pand* 在 *傾聽* 模式操作，而對等裝置則是在 *搜尋* 模式操作。根據預設，*傾聽* 模式為預先設定。請調整您本地 *pand* 的行為。此外，請在 *YaST 網路卡* 模組中設定 *bnepX* 介面 (X 代表系統中的設備號碼)。

手動設定藍芽

Bluez 系統個別元件的組態檔案位於 `/etc/bluetooth` 目錄中。唯一的例外是啟動元件的檔案 `/etc/sysconfig/bluetooth`，可由 *YaST* 模組修改。

以下所述的組態檔案僅可由使用者 *root* 修改。目前沒有變更所有設定值的圖形使用者介面。最重要的設定值可使用「*YaST 藍芽*」模組來設定，如 ["以 YaST 設定藍芽" \[278\]](#) 中所述一般。所有其他的設定，僅在特殊狀況的資深使用者才會用到。通常，預設值就足夠。

PIN 號碼提供基本保護，避免不需要的連線。行動電話在建立第一個連絡人時(或在設定電話的設備連絡人時)通常會查詢 PIN。兩個設備若要通訊，兩者皆必須以相同 PIN 辨識本身。在電腦上，PIN 位於檔案 `/etc/bluetooth/pin` 中。

重要: 藍芽連線的安全性

不管 PIN，兩個設備之間的傳輸可能不是完全安全。根據預設，「藍芽」連線的驗證和加密是停用的。啟動驗證和加密可能會導致與某些「藍芽」設備進行通訊時發生問題。

可以在組態檔案 `/etc/bluetooth/hcid.conf` 中變更各種設定，如設備名稱和安全性模式。通常，預設值就足夠。檔案包含描述各種設定選項的註解。

在所包括的檔案中的兩個區段是指定為 `options` 和 `device`。第一個區段包含 `hcid` 用以啟動的一般資訊。第二個區段則包含個別的本地「藍芽」設備的設定值。

`options` 區段最重要的設定之一為 `security auto;`。如果設為 `auto`，`hcid` 會嘗試使用本地 PIN 以進行內送連線。如果它失敗，它就會切換至 `none`，仍然嘗試建立連線。若要加強安全性，此預設值應設為 `user`，以確保使用者在每次建立連線時都會被要求輸入 PIN。

在 `device` 區段中設定電腦顯示在另一端的名稱。設備類別(例如，`Desktop`、`Laptop` 或 `Server`)是在此區段中定義。驗證與加密也是在這裏啟用或關閉。

22.2.3 系統元件和公用程式

「藍芽」的操作性視各種服務間的互動而定。至少需要兩個背景精靈：`hcid`(host controller interface daemon，主機控制器介面精靈)，做為「藍芽」設備和控制它的介面；以及 `sdpd`(service discovery protocol daemon，服務探索通訊協定精靈)，設備可藉此找出主機提供何種服務。如果在啟動系統時沒有自動啟用它們，`hcid` 和 `sdpd` 均可使用 `rcbluetooth start` 指令來啟用。此指令必須以 `root` 的身份執行。

以下段落簡略描述可與「藍芽」搭配使用的最重要外圍程序工具。即使現在提供了各種控制「藍芽」的圖形元件，看一下這些程式還是非常值得的。

部份指令必須以 `root` 的身份執行。包括測試遠端設備連線的指令 `l2pingdevice_address`。

hcitool

`hcitool` 可用來判斷是否偵測到本地或遠端設備。`hcitool dev` 指令可列出本地設備。輸出會對每一個偵測到的本地設備以 `interface_name device_address` 格式產生一行。

遠端設備可使用 `hcitool inq` 指令來搜尋。每一個偵測到的設備會傳回三個值：設備位址、時鐘偏移值和設備等級。設備位址相當重要，因為其他指令會使用它來辨識目標設備。時鐘偏移值主要為技術層次的用途。等級以十六進位值指定設備類型和服務類型。

指令 `hcitoolname device-address` 可用來判斷遠端設備的設備名稱。如果是遠端電腦，類別和設備名稱會與它的 `/etc/bluetooth/hcid.conf` 檔案中的資訊一致。本地設備位址會產生錯誤輸出。

hciconfig

指令 `/usr/sbin/hciconfig` 傳遞本地設備的進一步資訊。如果 `hciconfig` 以無引數執行，輸出會顯示設備資訊，如設備名稱 (`hciX`)、實體設備位址 (以 `00:12:34:56:78` 格式表示的 12 位數字) 和關於傳輸資料量的資訊。

`hciconfig hci0 name` 顯示當您的電腦接收到遠端設備要求時所傳回的名稱。除了查詢本地設備的設定之外，`hciconfig` 尚可用來修改這些設定。例如，`hciconfig hci0 name TEST` 可將名稱設為 `TEST`。

sdptool

程式 `sdptool` 可用來檢查特定設備提供的服務。指令 `sdptoolbrowse device_address` 會傳回設備的所有服務。`sdptool search service_code` 指令可用來搜尋特定服務。此指令會掃描所有可存取的設備以尋找要求的服務。如果有一個設備提供服務，程式會印出設備傳回的完整服務名稱以及簡短描述。輸入 `sdptool` 而無任何參數，則可檢視所有可能服務代碼清單。

22.2.4 圖形應用程式

在 Konqueror 中，輸入 URL `bluetooth:/` 以列出本地和遠端「藍芽」設備。連按兩下設備可取得設備提供的服務之綜覽。如果您以滑鼠在特定服務上移動，瀏覽器的狀態列會顯示服務所使用的設定檔。如果您按一下服務，會開啟對話方塊，詢問要做什麼：儲存、使用服務 (必須啟動應用程式才能執行) 或取消動作。如果您不想要再顯示對話方塊，可是想要永遠執行選取的動作，可以標示核取方塊。對於某些服務，尚未提供支援。對於其他服務，必須安裝額外的套件。

22.2.5 範例

本小節提供兩個可能的「藍芽」案例典型範例。第一個範例教導如何透過「藍芽」建立兩個主機之間的網路連線。第二個範例則是教導電腦與手機之間的連線。

兩台主機之間的網路連線

在第一個範例中，將在兩台主機 *H1* 和 *H2* 之間建立網路連線。這兩台主機的「藍芽」設備位址為 *baddr1* 和 *baddr2* (如上述在兩台主機上以指令 `hcitooldev` 所決定)。主機應該以 IP 位址 192.168.1.3 (*H1*) 和 192.168.1.4 (*H2*) 來辨識。

藉由 `pand` (Personal Area Networking Daemon, 個人區域網路精靈) 的協助建立「藍芽」連線。下列指令必須以使用者 `root` 的身份執行。描述重點在於「藍芽」特定動作，不提供網路指令 (`ip`) 的詳細說明。

輸入 `pand -s` 以啟動主機 *H1* 的 `pand`。然後，可使用 `pand -c baddr1` 建立主機 *H2* 的連線。如果您在其中一台主機上輸入 `ip link show` 以列出可用的網路介面，輸出應該包含如下的項目：

```
bnep0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop qlen 1000
       link/ether 00:12:34:56:89:90 brd ff:ff:ff:ff:ff:ff
```

除了 00:12:34:56:89:90，輸出應該包含本地設備位址 *baddr1* 或 *baddr2*。現在此介面必須指定 IP 位址並啟用。在 *H1* 上，可使用下列兩個指令來執行：

```
ip addr add 192.168.1.3/24 dev bnep0
ip link set bnep0 up
```

在 *H2* 上：

```
ip addr add 192.168.1.4/24 dev bnep0
ip link set bnep0 up
```

現在 *H1* 可從 *H2* 的 IP 192.168.1.3 之下存取。使用指令 `ssh 192.168.1.4` 從 *H1* 存取 *H2* (假設 *H2* 執行 `sshd`，在 SUSE Linux 中預設便會啟用)。指令 `ssh 192.168.1.4` 也可以一般使用者執行。

從行動電話到電腦的資料傳輸

第二個範例說明如何將含有內建數位相機的行動電話所建立的相片傳輸到電腦 (不產生多媒體訊息傳輸的額外成本)。雖然不同行動電話的功能表結構不同，但程序通常相似。如有需要，請參考您電話的手冊。此範例描述從 Sony Ericsson 行動電話傳輸相片到膝上型電腦。電腦上的 Obex-Push 服務必須為可用的，且電腦必須授與行動電話存取權。在第一步驟中，使膝上型電腦上的服務可用。可以藉由 `bluez-utils` 套件的 `opd` 精靈執行。以下列指令啟動精靈：

```
opd --mode OBEX --channel 10 --daemonize --path /tmp --sdp
```

使用了兩個重要的參數：`--sdp` 向 `sdpd` 註冊服務，而 `--path /tmp` 則是指定程式接收資料的儲存位置—在此例中是儲存於 `/tmp`。您也可以指定其他您具有寫入存取權的目錄。

現在行動電話必須認識電腦。要這樣做，開啟電話上的 *連線* 功能表並選取 *藍芽*。如有需要，在選取 *我的設備* 之前按一下 *開啟*。選取 *新設備* 並讓電話搜尋膝上型電腦。如果偵測到設備，它的名稱會出現在顯示幕上。選取和膝上型電腦相關的設備。如果您遭遇到 PIN 查詢，請輸入在 `/etc/bluetooth/pin` 中指定的 PIN。現在您的電話可辨識膝上型電腦，並且可和膝上型電腦交換資料。結束目前功能表並移至影像功能表。選取要傳輸的影像，並按下 *其他*。在下一個功能表中，按下 *傳送* 以選取傳輸模式。選取 *透過藍芽*。膝上型電腦應該列為目標設備。選取膝上型電腦以啟動傳輸。然後影像會儲存在 `opd` 指令所指定的目錄。音軌可以相同方式傳輸到膝上型電腦。

22.2.6 疑難排解

如果您無法建立連線，請根據下列清單繼續。請記住錯誤可能發生在連線的任一端或甚至在兩端。如果可能，以另一個「藍芽」設備重新建構問題，以確認設備沒有損壞。

本地設備是否列在 **hcitooldev** 的輸出中？

如果本地設備沒有列在此輸出中，表示 **hcid** 未啟動，或設備未被識別為「藍芽」設備。這可能有各種因素。設備可能損壞或正確的驅動程式可能遺失。內建「藍芽」的膝上型電腦通常有無線設備 (如 **WLAN** 和「藍芽」) 的開啟和關閉開關。請檢查您膝上型電腦的手冊以查看您的設備是否有此開關。以指令 **rcbluetoothrestart** 重新啟動「藍芽」系統，並檢查 **/var/log/messages** 中是否報告任何錯誤。

您的「藍芽」介面卡是否需要韌體檔案？

如果需要，請安裝 **bluez-bluefw** 並以指令 **rcbluetoothrestart** 重新啟動「藍芽」系統。

hcitoolinq 的輸出是否傳回其他設備？

請測試此指令一次以上。連線可能受到干擾，因為其他設備也使用「藍芽」的頻帶。

PIN 符合嗎？

檢查電腦的 **PIN** 號碼 (在 **/etc/bluetooth/pin** 中) 是否符合目標設備的。

遠端設備可以「看到」您的電腦嗎？

請嘗試從遠端設備建立連線。檢查此裝置是否看得到電腦。

可以建立網路連線嗎 (請參閱 ["兩台主機之間的網路連線" \[282\]](#))？

["兩台主機之間的網路連線" \[282\]](#) 中描述的設定可能不適用，這有幾個原因。例如，兩台電腦其中之一不支援 **ssh** 通訊協定。請嘗試 **ping192.168.1.3** 或 **ping192.168.1.4**。如果可以，請檢查 **sshd** 是否在作用中。其他問題可能是兩個設備其中之一已經有網路設定，與範例中的位址 **192.168.1.x** 衝突。如果是此狀況，請嘗試不同的位址，如 **10.123.1.2** 和 **10.123.1.3**。

膝上型電腦是否顯示為目標設備 (請參閱 ["從行動電話到電腦的資料傳輸" \[283\]](#))？

行動設備可識別膝上型電腦上的 **Obex-Push** 服務嗎？

在 *我的設備* 中，選取對應設備並檢視 *服務清單*。如果沒有顯示 **Obex-Push** (即使在更新清單之後)，則問題是因為膝上型電腦上的 **opd** 所造成。**opd** 是否在作用中？您對特定目錄是否有寫入存取權？

["從行動電話到電腦的資料傳輸" \[283\]](#) 中描述的狀況以相反方向是否能運作？

如果安裝了 **obexftp** 套件，在部份設備上可使用 **obexftp -b device_address -B 10 -p image** 指令來執行。已測試數種 **Siemens**

和 Sony Ericsson 型號並發現其可以作用。請參閱 `/usr/share/doc/packages/obexftp` 中的文件。

22.2.7 若需詳細資訊

「藍芽」的使用和組態的各種說明之廣泛綜覽可在 <http://www.holtmann.org/linux/bluetooth/> 中找到。其他有用的資訊和說明：

- 「藍芽」通訊協定堆疊的正式 howto 整合於核心中：<http://bluez.sourceforge.net/howto/index.html>
- 連線到 PalmOS PDA：<http://www.cs.ucl.ac.uk/staff/s.zachariadis/btpalmlinux.html>

22.3 紅外線資料傳輸

IrDA (紅外線資料關聯) 是以紅外線進行無線通訊的產業標準。目前販售的許多膝上型電腦均配備有 IrDA 相容接收器，可與其他設備 (如印表機、數據機、LAN 或其他膝上型電腦) 通訊。傳輸速率範圍從 2400 bps 到 4 Mbps。

IrDA 操作模式有兩種。標準模式 SIR 透過序列介面存取紅外線連接埠。此模式幾乎可在所有系統上使用，而且對於大部份需求已經夠用。較快的模式 FIR 必須要有 IrDA 晶片的特殊驅動程式。FIR 模式並不支援所有晶片類型，因為缺乏適當的驅動程式。請在電腦的 BIOS 中設定想要的 IrDA 模式。BIOS 也會顯示 SIR 模式所使用的序列介面。

您可以在 Werner Heuser 的 IrDA how-to 中找到的 IrDA 相關訊息，網址為 <http://tuxmobil.org/Infrared-HOWTO/Infrared-HOWTO.html>。除此之外，您可以參考 Linux IrDA Project 網站 <http://irda.sourceforge.net/>。

22.3.1 軟體

所需的核心模組包括在核心套件中。irda 套件提供所需協助程式應用程式以支援紅外線介面。安裝套件之後，可在 `/usr/share/doc/packages/irda/README` 找到文件。

22.3.2 組態

當系統啟動時，IrDA 系統服務不會自動啟動。請使用 YaST IrDA 模組以啟動。在此模組中僅可修改一個設定：紅外線設備的序列介面。測試視窗會顯示兩個輸出。一個是 `irdadump` 的輸出，記錄所有傳送和接收的 IrDA 封包。此輸出應包含電腦名稱和傳輸範圍中所有紅外線設備的名稱。這些訊息的範例顯示在節 22.3.4, "疑難排解" [287] 中。有 IrDA 連線存在的所有設備列示於視窗下方。

IrDA 消耗相當可觀的電池電量，它每隔幾秒便傳送探索封包以偵測其他週邊設備。因此，如果您依賴電池電力，則應該僅在需要時啟動 IrDA。輸入指令 `rcirdastart` 以啟用或 `rcirdastop` 以停用。在介面啟動時會自動載入需要的所有核心模組。

可在 `/etc/sysconfig/irda` 檔案中執行手動組態。此檔案僅包含一個變數 `IRDA_PORT`，它可以決定 SIR 模式所使用的介面。

22.3.3 使用方式

資料可傳送到設備檔案 `/dev/ir1pt0` 以列印。設備檔案 `/dev/ir1pt0` 功能與一般 `/dev/lp0` 有線介面類似，除了列印資料是以紅外線無線傳送之外。對於列印，請確認印表機在電腦紅外線介面的視覺範圍之內，以及紅外線支援已啟動。

透過紅外線介面操作的印表機可使用「YaST 印表機」模組加以設定。因為不會自動偵測它，可以按一下 *其他 (未偵測到者)* 以手動設定它。在下列對話方塊中，選取 *IrDA 印表機*。通常，`ir1pt0` 是正確連線。在章 31, *印表機操作* [431] 中，可取得在 Linux 中操作印表機的詳細資訊。

與其他主機和行動電話或其他類似設備的通訊，是透過設備檔案 `/dev/ircomm0` 所指導。例如，Siemens S25 和 Nokia 6210 行動電話可以使用紅外線介面，以

wvdial 應用程式撥號並連線到網際網路。假使對應應用程式的設備設定已設定為 /dev/ircomm0，也可以與 Palm Pilot 同步化資料。

如果您想要，可以僅處理支援印表機或 IrCOMM 通訊協定的設備。支援 IROBEX 通訊協定的設備 (如 3Com Palm Pilot) 可使用特殊應用程式 (如 irobexpalm 和 irobexreceive) 來存取。如需詳細資訊，請參考 *IR-HOWTO*(<http://tldp.org/HOWTO/Infrared-HOWTO/>)。設備所支援的通訊協定是列在 irdadump 輸出中設備名稱之後的括號內。IrLAN 通訊協定支援仍然是「進行中的工作」。

22.3.4 疑難排解

如果連線到紅外線連接埠的設備沒有回應，請使用指令 irdadump (以 root 的身份) 來檢查電腦是否辨識到其他設備。當 Canon BJC-80 印表機在電腦的可見範圍中時，會固定出現類似 範例 22.1, "irdadump 的輸出" [287] 的現象：

範例 22.1 irdadump 的輸出

```
21: 41:38.435239 xid:cmd 5b62bed5 > ffffffff S=6 s=0 (14)
21: 41:38.525167 xid:cmd 5b62bed5 > ffffffff S=6 s=1 (14)
21: 41:38.615159 xid:cmd 5b62bed5 > ffffffff S=6 s=2 (14)
21: 41:38.705178 xid:cmd 5b62bed5 > ffffffff S=6 s=3 (14)
21: 41:38.795198 xid:cmd 5b62bed5 > ffffffff S=6 s=4 (14)
21: 41:38.885163 xid:cmd 5b62bed5 > ffffffff S=6 s=5 (14)
21: 41:38.965133 xid:rsp 5b62bed5 < 6cac38dc S=6 s=5 BJC-80
                        hint=8804 [ Printer IrCOMM ] (23)
21: 41:38.975176 xid:cmd 5b62bed5 > ffffffff S=6 s=* earth
                        hint=0500 [ PnP Computer ] (21)
```

檢查介面的組態，是否沒有輸出或是其他設備沒有回覆。確認所使用的介面是正確的。紅外線介面有時候位在 /dev/ttyS2 或在 /dev/ttyS3，而有時候會使用 IRQ 3 以外的中斷。這些設定幾乎可以在所有膝上型電腦的 BIOS 設定功能表中檢查和修改。

簡單的視訊攝影機也可協助判斷紅外線 LED 燈號是否全部都亮。大部份視訊攝影機可以看到紅外線；但是人眼卻不能。

部 VII. 管理

Linux 的安全性

偽裝及防火牆可確保資料流及資料交換受到控制。SSH (保全外圍程序, Secure Shell) 可確保您透過加密的連線登入遠端主機。加密檔案或整個分割區能夠保護您的資料, 不讓第三方存取您的系統。加上技術指導, 找到有關 Linux 網路安全性的資訊。

23.1 偽裝與防火牆

只要 Linux 用於網路環境, 您可以使用允許操作網路封包的核心功能, 讓內部和外部網路區域之間保持區隔。Linux netfilter 架構提供建立有效防火牆的方法, 將不同的網路分開。利用 iptables (規則集定義的一般表格結構) 的幫助, 準確地控制允許傳遞網路介面的封包。這類封包過濾器只要透過 SUSEfirewall2 及對應 YaST 模組的協助即可輕鬆設定。

23.1.1 使用 iptables 過濾封包

netfilter 與 iptables 元件負責過濾及操作網路封包, 以及網路位址轉譯 (NAT, Network Address Translation)。過濾標準和任何與其關聯的動作以鏈結方式儲存, 在個別網路封包抵達時必須相互符合。符合的鏈結儲存於表格內。iptables 指令允許您變更這些表格和規則集。

Linux 核心維護三個表格, 每個針對封包過濾器的特定功能種類:

過濾器

此表格保存一組過濾器規則，因為它以更嚴格的方式執行封包過濾 (*packet filtering*) 機制，例如，決定封包是否允許通過 (ACCEPT) 或放棄 (DROP)。

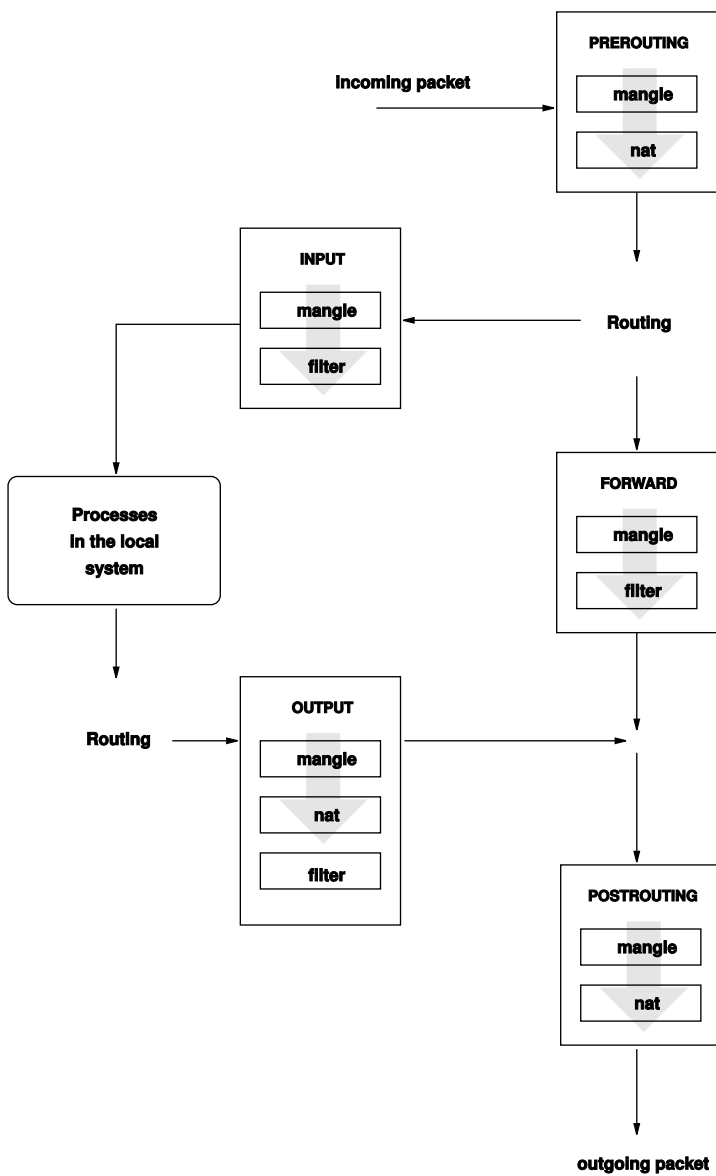
nat

此表格定義對封包之來源和目標位址的任何變更。使用這些功能也可以讓您執行「偽裝」(*masquerading*)，這是 NAT 用來連結私有網路與網際網路的一種特殊方式。

mangle

此表格中的規則可以操作儲存於 IP 標頭中的值 (如服務類型)。

圖形 23.1 iptables：封包的可能路徑



這些表格包含數個符合封包的預先定義鏈結：

PREROUTING

此鏈結套用到內送封包。

INPUT

此鏈結套用到預定送到系統內部程序的封包。

FORWARD

此鏈結套用到僅透過系統路由的封包。

OUTPUT

此鏈結套用到來自於從系統本身的封包。

POSTROUTING

此鏈結套用到所有外送封包。

[圖形 23.1, "iptables: 封包的可能路徑" \[293\]](#) 說明網路封包在指定系統上傳送的路徑。為了簡化的緣故，圖中將表格列為鏈結的各部份，但是實際上，這些鏈結是在表格本身內。

所有可能情況中最簡單的一種，系統本身預定的內送封包抵達 `eth0` 介面。封包先參照 `mangle` 表的 `PREROUTING` 鏈結，然後再參照 `nat` 表的 `PREROUTING` 鏈結。下一個步驟有關封包的路由，決定封包的實際目標是系統本身的程序。傳遞 `mangle` 與 `filter` 表的 `INPUT` 鏈結後，封包最後會到達其目標，也就是確定符合 `filter` 表的規則。

23.1.2 偽裝基本原則

偽裝是 NAT (網路位址轉譯) 的 Linux 專用形式。它可以用來連結小型 LAN (主機使用私用範圍的 IP 位址；請參閱 [節 38.1.2, "網路遮罩與路由" \[511\]](#)) 與網際網路 (使用正式 IP 位址)。若要讓 LAN 主機能夠連接網際網路，這些主機的私用位址會轉譯為正式位址。這是在路由器上完成的，路由器當作 LAN 與網際網路之間的閘道。基本原則很簡單：路由器擁有一個以上的網路介面，通常是網路卡以及另一個與網際網路連結的介面。後者連結路由器與外界，一或多個其他的則連結路由器與 LAN 主機。當區域網路中的這些主機連接到路由器的網路卡時 (如 `eth0`)，它們可以傳送任何不是預定給區域網路的封包到其預設的閘道或路由器。

重要：使用正確的網路遮罩

設定您的網路時，確定廣播位址及網路遮罩在所有本地主機上都是相同的。如果不相同，可能會造成封包無法正確的路由。

如所述，只要其中一個 LAN 主機傳送預定給網際網位位址的封包時，就會送到預設路由器。不過，路由器必須先設定才能轉遞這類封包。為了安全的緣故，SUSE Linux 在預設安裝時不會啟用此功能。若要啟用，將檔案 `/etc/sysconfig/sysctl` 中的變數 `IP_FORWARD` 設定為 `IP_FORWARD=yes`。

連線的目標主機可以看到您的路由器，但是並不知道您內部網路中產生封包的主機。這就是為什麼這個技術稱為偽裝。由於位址轉譯功能，所以路由器是任何回覆封包的第一個目的地。路由器必須識別這些內送封包並轉譯其目標位址，如此封包可轉遞到區域網路中的正確主機。

依據偽裝表格之內送交通的路由，是不可能從外面開啟對內部主機的連線。對於這類連線，表格中不會出現項目。此外，任何已經建立的連線在表格中都有指定給它的狀態項目，因此另一個連線無法使用此項目。

其結果是，您在一些應用程式通訊協定上可能會出現問題，如 ICQ、cucme、IRC (DCC、CTCP) 和 FTP (PORT 模式)。Netscape、標準的 FTP 程式以及許多其他通訊協定使用 PASV 模式。就封包過濾和偽裝而言，這種被動模式的問題較少。

23.1.3 防火牆基本原則

防火牆大概是使用最為廣泛的字詞，用來描述提供及管理網路之間的連結機制，同時還能控制其間的資料流。嚴格來說，本節中描述的機制稱為**封包過濾器**。封包過濾器根據特定標準(如通訊協定、通訊埠和 IP 位址)規範資料流。這樣允許您根據封包的位址，阻斷不應該送到您網路上的封包。例如，若要允許公用存取您的網頁伺服器，請明確開啟對應連接埠。不過，封包過濾器不會掃描具有正常位址的封包內容，如導向您網頁伺服器的那些封包。例如，即使內送封包意圖危害您網頁伺服器上的 CGI 程式，封包過濾器仍會讓這些封包通過。

更有效但更複雜的機制是數種系統類型的組合，如與應用程式閘道或 Proxy 互動的封包過濾器。這個時候，封包過濾器會拒絕預定給停用連接埠的任何封包。只有導向到應用程式閘道的封包才會被接受。此閘道或 Proxy 會假裝是伺服器的實際用戶端。以這種意義而言，這類 Proxy 會視為通訊協定層級上應用程式所使用的偽裝主機。這種 Proxy 的範例之一是 Squid，一種 HTTP Proxy 伺服器。

若要使用 Squid，瀏覽器必須設定成透過 Proxy 進行通訊。要求的任何 HTTP 頁面會從 Proxy 快取取得，而快取中找不到的頁面會透過 Proxy 從網際網路抓取。另一個範例則是，SUSE proxy-suite (proxy-suite) 為 FTP 通訊協定提供 Proxy。

下節將焦點放在 SUSE Linux 隨附的封包過濾器。如需有關封包過濾及防火牆的詳細資訊，請參閱 howto 套件中包含的「防火牆 HOWTO」內容。如果安裝了此套件，請使用 less

```
/usr/share/doc/howto/en/txt/Firewall-HOWTO.gz
```

 閱讀 HOWTO。

23.1.4 SUSEfirewall2

SUSEfirewall2 是讀取 /etc/sysconfig/SuSEfirewall2 中變數集的程序檔，可以產生一組 iptables 規則。它定義三個安全性區域，但是以下範例組態僅考慮第一個和第二個區域：

外部區域

假設沒有方法可以控制外部網路上發生的情況，因此需要保護主機。在大部分的情況中，外部網路就是網際網路，但是也可能是其他不安全的網路，例如 WLAN。

內部區域

這是指私用網路，通常是指 LAN。如果此網路上的主機使用私用範圍的 IP 位址 (請參閱 [節 38.1.2, "網路遮罩與路由" \[511\]](#))，請啟用網路位址轉譯 (NAT)，這樣內部網路上的主機即可存取外部網路。

廢除區域 (DMZ)

儘管外部及內部網路可以連接位於此區域中的主機，但是這些主機本身無法存取內部網路。這種設定可以在內部網路前加上額外的防護線，因為 DMZ 系統與內部網路是隔離的。

過濾規則集未明確允許的任何網路流量類型會由 iptables 封鎖。因此，具有內送流量的每個介面必須放置在三個區域中的其中一個。對於每個區域，定義允許的服務或通訊協定。規則集僅套用到遠端主機產生的封包。本地產生的封包不會被防火牆攔截。

使用 YaST 可以執行組態設定 (請參閱 ["使用 YaST 設定" \[297\]](#))。也可以在檔案 /etc/sysconfig/SuSEfirewall2 中手動進行，該檔案的註解完整。不僅如此，在 /usr/share/doc/packages/SuSEfirewall2/EXAMPLES 中還提供一些範例案例。

使用 YaST 設定

重要: 自動防火牆組態

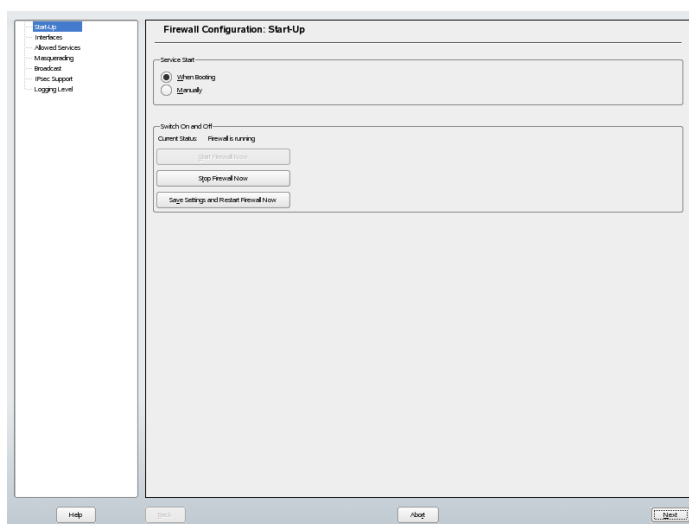
在安裝後，YaST 在所有已設定的介面上會自動啟動防火牆。如果已設定系統並在系統上啟動，YaST 會使用伺服器組態模組中的 *開啟防火牆* 中選取介面的 *連接埠* 或 *開啟防火牆的連接埠* 選項，以修改自動產生的防火牆組態。有些伺服器模組對話方塊具有 *防火牆詳細資料* 按鈕，可以啟動其他服務和連接埠。YaST 防火牆組態模組僅能啟動、關閉或重新設定防火牆。

YaST 的圖形組態對話方塊可從「YaST 控制中心」存取。選取 *安全性和使用者* → *防火牆*。該組態一共分成七個部份，可以在左手邊的三個架構直接存取。

啟動

在此對話方塊中設定啟動行為。預設的安裝中，會自動啟動 SUSEfirewall2。您也可以在此啟動和停止防火牆。若要在執行中的防火牆設定新的設定，請使用 *立即儲存設定並重新啟動防火牆*。

圖形 23.2 YaST 防火牆設定



介面

所有已知的網路介面都列在這裏。若要從區域移除介面，請選取介面，按 **變更後選擇 沒有指定區域**。若要新增介面至區域，請選取介面，按 **變更後選擇任何可用的區域**。您也可以使用 **自訂**，以自己的設定建立特殊的介面。

允許的服務

您需要此選項以從系統提供服務至受保護的區域。按照預設值，只會保護系統不受外部區域的侵犯。明確的規範允許外部區域可使用的服務。在 **選取區域所允許的服務** 中選取適當的區域後，啟動服務。

偽裝

偽裝可將內部網路隱身於外部網路，如網際網路，但允許內部網路的主機存取外部網路。從外部網路發出對內部網路的要求會遭到封鎖，但是從內部網路發出的要求，從外部看起來會像是從偽裝伺服器發出。如果內部機器的特殊服務需要開放給外部網路使用，可以針對服務增加特殊的重新指向規則。

廣播

在此對話方塊中，已設定允許廣播的 UDP 埠。新增必需的連接埠號碼或服務到適當的區域，由空格分隔。請參閱 `/etc/services` 檔案。

不被接受的廣播紀錄可以在此啟動。這有可能會有問題，因為 Windows 主機使用廣播瞭解彼此，也因而產生許多不被接受的封包。

IPsec 支援

在此對話方塊設定 IPsec 服務是否可供外部網路使用。在 **詳細資料** 下，設定可信任的封包。

記錄層次

記錄有兩種規則：可接受和不被接受的封包。不被接受的封包包括 **DROPPED** 或 **REJECTED**。從 **記錄全部**、**記錄關鍵** 或 **全部不要記錄** 來選取這兩個規則。

當完成防火牆組態時，請按 **下一步** 結束此對話方塊。開啟防火牆設定的區域導向摘要。在此，勾選所有設定。所有已允許的服務，埠和通訊協定都會列在此摘要中。若要修改組態，使用 **上一步**。按 **接受** 可儲存您的組態。

手動設定

以下段落提供成功組態的逐步指示。每個組態項目會依據是否與防火牆或偽裝功能相關加以標示。這裡並沒有涵蓋組態檔中所提與 **DMZ** (廢除區域) 相關的觀

點。這些觀點僅適用較大型組織的更複雜網路基礎結構 (公司網路)，因為需要全面的組態以及對該主題的深入瞭解。

首先，使用 YaST 模組系統服務 (Runlevel) 以啟動 runlevel 中的 SUSEfirewall2 (最有可能是 3 或 5)。它在 `/etc/init.d/rc?.d/` 目錄中會設定 SUSEfirewall2_* 程序檔的符號連結。

FW_DEV_EXT (防火牆、偽裝)

連結到網際網路的設備。如果是數據機連線，請輸入 `ppp0`。如果是 ISDN 連結，請使用 `ispnp0`。DSL 連線則使用 `dsl0`。指定 `auto` 使用與預設路由對應的介面。

FW_DEV_INT (防火牆、偽裝)

連結到內部私用網路的設備 (如 `eth0`)。如果沒有內部網路且防火牆僅保護執行的主機時，請留置空白。

FW_ROUTE (防火牆、偽裝)

如果需要偽裝功能，將此設定為 `yes`。外界將無法看到您的內部主機，因為網際網路路由器會忽略其私用網路位址 (例如，`192.168.x.x`)。

對於沒有偽裝的防火牆，如果想要允許存取內部網路，僅將此設定為 `yes`。在這種情況下，您的內部主機需要使用正式註冊的 IP。不過，通常您不應該允許外界存取您的內部網路。

FW_MASQUERADE (偽裝)

如果需要偽裝功能，將此設定為 `yes`。這可以提供網際網路主機，建立虛擬/直接的連線到網際網路。在內部網路和網際網路的主機之間使用 Proxy 伺服器較為安全。對於 Proxy 伺服器所提供的服務並不需要偽裝。

FW_MASQ_NETS (偽裝)

指定要偽裝的主機或網路，在個別項目之間加上空格。例如：

```
FW_MASQ_NETS="192.168.0.0/24 192.168.10.1"
```

FW_PROTECT_FROM_INT (防火牆)

將此設定為 `yes`，保護您的防火牆主機免於來自內部網路的攻擊。如果確實啟用，只有內部網路才可以使用服務。另請參閱 `FW_SERVICES_INT_TCP` 與 `FW_SERVICES_INT_UDP`。

FW_SERVICES_EXT_TCP (防火牆)

輸入應該可用的 TCP 連接埠。對於不應該提供任何服務的一般家用工作站，請留置空白。

FW_SERVICES_EXT_UDP (防火牆)

除非執行 UDP 服務且希望讓外界使用，否則請留置空白。使用 UDP 的服務包括 DNS 伺服器、IPSec、TFTP、DHCP 以及其他。如果要讓外界使用，輸入要使用的 UDP 連接埠。

FW_SERVICES_INT_TCP (防火牆)

利用此變數，定義可讓內部網路使用的服務。其表示法與 FW_SERVICES_EXT_TCP 相同，但是設定是套用到內部網路。只有在 FW_PROTECT_FROM_INT 設定為 yes 時，才需要設定變數。

FW_SERVICES_INT_UDP (防火牆)

請參閱 FW_SERVICES_INT_TCP。

設定防火牆後，測試您的設定。防火牆規則集的建立是以 root 身份輸入 `SUSEfirewall2 start`。接著使用 `telnet`，例如，從外部主機查看是否已實際拒絕連線。之後，檢視 `/var/log/messages`，在其中應該會看到如下內容：

```
Mar 15 13:21:38 linux kernel: SFW2-INext-DROP-DEFAULT IN=eth0
OUT= MAC=00:80:c8:94:c3:e7:00:a0:c9:4d:27:56:08:00 SRC=192.168.10.0
DST=192.168.10.1 LEN=60 TOS=0x10 PREC=0x00 TTL=64 ID=15330 DF PROTO=TCP
SPT=48091 DPT=23 WINDOW=5840 RES=0x00 SYN URGF=0
OPT (020405B40402080A061AFEB0000000001030300)
```

其他測試防火牆設定的套件為 `nmap` 或 `nessus`。`nmap` 的文件可在 `/usr/share/doc/packages/nmap` 中找到，而 `nessus` 的文件則是在安裝個別套件後的 `/usr/share/doc/packages/nessus-core` 目錄中。

23.1.5 若需詳細資訊

有關 `SUSEfirewall2` 套件的最新資訊及其他文件請參閱 `/usr/share/doc/packages/SUSEfirewall2`。`netfilter` 與 `iptables` 專案的首頁 (<http://www.netfilter.org>) 提供許多語言的大量文件。

23.2 SSH：安全網路作業

隨著有愈來愈多的電腦安裝在網路環境中，通常會需要從遠端位置存取主機。這通常是指使用者會傳送登入與密碼字串以供驗證。只要這些字串是以純文字傳送，它們就有可能被攔截和濫用以取得使用者帳戶的存取權，而授權的使用者卻完全未察覺。除了這將會把使用者的所有檔案暴露給攻擊者之外，不合法的帳戶有可能取得管理員或 `root` 存取權，或是滲透其他的系統。在過去，遠端連線是使用 `telnet` 建立，它並未提供對抗監聽加密形式或其他安全性機制的防護。有一些通訊通道是未防護的，像是傳統的 `FTP` 通訊協定以及某些遠端的複製程式。

SSH 套裝軟體可以提供必要的保護，即加密驗證字串 (通常是登入名稱與密碼) 以及所有在主機間的其他資料交換。有了 SSH，即使第三方記錄了資料流，但是內容已被加密，除非知道加密金鑰，否則無法將內容回復至純文字。因此 SSH 可以在非安全的網路上啟用安全通訊，例如網際網路。SUSE Linux 隨附的 SSH 為 `OpenSSH`。

23.2.1 OpenSSH 套件

SUSE Linux 預設會安裝 `OpenSSH` 套件。`ssh`、`scp` 以及 `sftp` 程式就變成了 `telnet`、`rlogin`、`rsh`、`rcp` 以及 `ftp` 的替代程式。在預設的組態中，只有使用 `OpenSSH` 公用程式且防火牆允許時，才能存取 SUSE Linux 系統。

23.2.2 ssh 程式

使用 `ssh` 程式，就有可能登入遠端系統並互動式地工作。它會取代 `telnet` 與 `rlogin`。`slogin` 程式只是指向 `ssh` 的符號連結。例如，使用 `ssh sun` 指令登入 Sun 主機。該主機接著會提示輸入 Sun 的密碼。

在成功地驗證後，即可使用遠端指令行，或使用互動式應用程式，例如 `YaST`。如果本地使用者名稱與遠端使用者名稱不同，您可以使用不同的登入名稱加上 `ssh -l augustine sun` 或 `ssh augustine@sun`。

此外，`ssh` 提供從遠端系統上執行指令 (亦即 `rsh`) 的可能性。在下列範例中，在 Sun 主機上執行 `uptime` 指令，並建立名為 `tmp` 的目錄。程式輸出會顯示在 Earth 主機的本地終端機上。

```
ssh otherplanet "uptime; mkdir tmp"
tux@otherplanet's password:
1:21pm up 2:17, 9 users, load average: 0.15, 0.04, 0.02
```

在這裏需要使用引號以使用一個指令傳送兩個指示。只有執行此動作，第二個指令才可以在 Sun 上執行。

23.2.3 scp--安全的複製

scp 會將檔案複製到遠端機器。它是 rcp 的安全與加密的替代程式。例如，scp MyLetter.tex sun: 從 Earth 主機將 MyLetter.tex 檔案複製到 Earth 主機。如果在 Earth 上的使用者名稱與 Sun 上的使用名稱不相同，請使用 username@host 格式指定後者的使用者名稱。此指令並沒有 -l 的選項。

在輸入正確的密碼後，scp 會啟動資料傳輸並顯示逐漸變長的星號列，以模擬進度列。除此之外，程式會永遠在進度列的右邊顯示到達的估計時間。指定 -q 選項以抑制所有的輸出。

scp 也會為整個目錄提供遞迴複製功能。scp -r src/ sun::backup/ 指令會複製 src 目錄的整個內容 (包括所有的子目錄) 至 Sun 主機上的 backup 目錄。如果這個子目錄不存在，就不會自動建立。

-p 選項會指示 scp 保留未變更檔案的時間戳記。-c 可以壓縮資料傳輸。這可能會減少要傳輸的資料量，但是會造成處理器更重的負擔。

23.2.4 sftp--安全的檔案傳輸

可以使用 sftp 程式取代 scp，以進行安全的檔案傳輸。在 sftp 工作階段期間，可以使用許多已知的 ftp 指令。sftp 程式可能是比 scp 更好的選擇，特別是在傳輸未知檔名的資料時。

23.2.5 SSH 精靈 (sshd) 伺服器端

若要使用 SSH 用戶端程式 ssh 與 scp，伺服器 (SSH 精靈) 必須在背景執行，監聽 TCP/IP port 22 的連線。精靈會在第一次啟動時產生三個金鑰組合。每個金鑰組合都是由私密與公用金鑰所組成。因此，此程序可稱為公用金鑰式。為了保證透過 SSH 通訊的安全性，必須限制只有系統管理員可以存取私密金鑰

檔。預設安裝會據此設定檔案權限。SSH 精靈只有在本地上需要私密金鑰，而且絕對不能將它給予任何人。公用金鑰元件 (以 .pub 的副檔名來識別) 會傳送至要求連線的用戶端。它們可讓所有的使用者讀取。

SSH 用戶端所啟始的連線。等待 SSH 精靈與要求 SSH 用戶端交換識別資料，以比較通訊協定與軟體版本，並防止透過錯誤埠的連線。因為原始 SSH 精靈的子處理序會回覆要求，所以可以同時建立數個 SSH 連線。

至於 SSH 伺服器與 SSH 用戶端之間的通訊，OpenSSH 可以支援 SSH 通訊協定的版本 1 與 2。新安裝的 SUSE Linux 系統預設值為版本 2。若要在更新後繼續使用版本 1，請遵照在 /usr/share/doc/packages/openssh/README.SuSE 中的指示。這個文件也描述如何只執行一些步驟，即將 SSH 1 環境轉換成工作 SSH 2 環境。

當使用 SSH 的版本 1 時，伺服器會傳送每小時由 SSH 精靈重新產生的公用主機金鑰與伺服器金鑰。這兩者都允許 SSH 用戶端加密自由選擇的工作階段金鑰，並且會將它傳送至 SSH 伺服器。SSH 用戶端也會指示伺服器要使用的加密方法 (密碼)。

SSH 通訊協定的版本 2 並不需要伺服器金鑰。兩邊都是使用根據 Diffie-Helman 的演算法來交換金鑰。

私密主機與伺服器金鑰絕對需要解密工作階段的金鑰，而且不能從公用部份產生。只有連絡的 SSH 精靈可以使用私密金鑰解密工作階段金鑰 (請參閱 man /usr/share/doc/packages/openssh/RFC.nroff)。若要仔細監看此啟始連線階段，可以開啟 SSH 用戶端的 -v 詳細偵錯選項。

預設會使用 SSH 通訊協定的版本 2。使用 -1 參數即可覆寫此預設以使用通訊協定的版本 1。用戶端會在第一次與遠端主機連繫後，將所有的公用主機金鑰儲存在 ~/.ssh/known_hosts 中。這將可防止外來的 SSH 伺服器進行「在中間攔截的攻擊」(外部 SSH 伺服器使用假冒的名稱與 IP 位址)。這類的攻擊通常都是由下列方式所偵測出來：在 ~/.ssh/known_hosts 中未包含主機金鑰，或伺服器在缺少適當的私密對照組時無法解密工作階段的金鑰。

建議將儲存在 /etc/ssh/ 中的私密與公用金鑰備份在安全的外部位置。以此方式，就可以偵測出金鑰的修改，並且可以在重新安裝後，再度使用舊的金鑰。這可讓使用者避免收到任何擾人的警告。儘管出現警告，但是如果已確認它確實是正確的 SSH 伺服器，就必須從 ~/.ssh/known_hosts 移除關於此系統的現有項目。

23.2.6 SSH 驗證機制

目前以最簡單的形式所執行的實際驗證，是由上方所提及的輸入密碼所組成。SSH的目標就是要引進安全且易於使用的軟體。因為它是用以取代rsh與rlogin，SSH也必須能夠提供適於平日使用的驗證方法。SSH是藉由使用者所產生的另一個金鑰組合來完成此動作。SSH套件為此提供了說明程式：ssh-keygen。在輸入ssh-keygen -t rsa或ssh-keygen -t dsa之後，就會產生金鑰組合，而且會提示您輸入儲存金鑰的基本檔案名稱。

確認預設值並回覆通關密語的要求。即使軟體建議空白的通關密語，一般建議輸入10至30個字元的文字以描述在此的程序。請勿使用簡短的文字或片語。重複通關密語以確認。接著，您將會看到儲存私密與公用金鑰的位置，在此例中，為id_rsa與id_rsa.pub檔案。

使用ssh-keygen -p -t rsa或ssh-keygen -p -t dsa以變更舊的通關密語。複製公用金鑰元件(在範例中為id_rsa.pub)至遠端機器，並將它儲存至~/.ssh/authorized_keys。下次建立連線時，將會要求您驗證自己的通關密語。如果沒有要求您驗證，請驗證這些檔案的位置與內容。

就長期而言，此程序比每次給密碼更為麻煩。因此，SSH套件提供另一個工具：ssh-agent。此工具可保留了X工作階段期間的私密金鑰。整個X工作階段是啟動成ssh-agent的子處理序。執行這個的最簡單的方法就是在.xsession檔案的開頭將useshh變數設為yes，並透過顯示管理員(例如KDM或XDM)登入。或者，輸入ssh-agent startx。

現在您可以照常使用ssh或scp。如果您已依照上方所述配送公用金鑰，就不會再提示您輸入密碼。請利用xlock之類的密碼保護應用程式，來終止或鎖定X工作階段。

在推出SSH通訊協定版本2後所造成的相關變更，也記載在/usr/share/doc/packages/openssh/README.SuSE檔案中。

23.2.7 X，驗證與轉寄機制

除了先前所述的安全性相關改善之外，SSH也會簡化遠端X應用程式的使用。如果您以-x選項執行ssh，就會自動在遠端機器上設定DISPLAY變數，而且所有的X輸出都會透過現有的SSH連線輸出至遠端機器。同時，未獲授權的個體將無法攔截以此方式檢視，並在遠端和本地上啟動的X應用程式。

透過新增 `-A` 選項，`ssh-agent` 驗證機制就會延續至下一部機器。以此方式，您不需輸入密碼就可以在不同的機器上工作，但僅限於您已配送公用金鑰至目的主機並將其適當地儲存在主機上。

兩種機制都是在預設值中停用，但是也可以隨時在整個系統的 `/etc/ssh/sshd_config` 組態檔中或使用者的 `~/.ssh/config` 中啟動。

`ssh` 也可以用於重新導向 TCP/IP 連線。在下列範例中，設定 SSH 分別重新導向 SMTP 與 POP3 埠：

```
ssh -L 25:sun:25 earth
```

使用此指令後，任何導向 Earth 埠 25 (SMTP) 的連線都會透過加密通道重新導向至 Earth 上的 SMTP 埠。這對於那些使用 SMTP 伺服器但卻沒有 SMTP-AUTH 或 POP-before-SMTP 功能的連線特別有用。從任何連線至網路的位置，都可以將電子郵件傳送至「主」郵件伺服器以進行傳遞。同樣地，在 Earth 上的所有 POP3 要求 (埠 110) 都可以使用此指令轉寄至 Sun 的 POP3 埠：

```
ssh -L 110:sun:110 earth
```

兩個指令都必須以 `root` 的身份執行，因為該連線必須以有權限的本地埠建立。電子郵件是由一般的使用者在現有的 SSH 連線中傳送和擷取。SMTP 與 POP3 主機都必須設成 `localhost` 才能正常運作。上方所述的每個程式的其他資訊都可以在手冊頁面中找到，也可以在 `/usr/share/doc/packages/openssh` 下的檔案中找到。

23.3 加密分割區和檔案

每一位使用者都有一些第三方無法存取的機密資料。當您連線越頻繁、越有機動性時，處理資料時就應更加小心。此時，若有其他人透過網路連線或直接實體存取您的電腦，則會建議您加密檔案或整個分割區。

警告：加密的媒體受到有限保護

請注意，您無法使用本章節中描述的方法來防止您正在運作的系統資料外洩。成功裝載加密的媒體後，使用者必須要有足夠的權限才可存取該媒體。如果您的電腦遺失或遭竊，加密的媒體將發揮作用，不讓未經授權的使用者讀取您的機密資料。

以下清單重點摘述一些可以想像的使用案例。

筆記型電腦

如果您旅行時攜帶筆記型電腦，為存有機密性資料的硬碟分割區加密將會是個很棒的點子。當您的筆記型電腦遺失或遭竊時，如果資料位於加密檔案系統或單一加密檔案中，其他人將無法讀取這些資料。

抽取式媒體

USB 隨身碟或外接式硬碟和筆記型電腦一樣容易遭到竊取。加密的檔案系統可防止第三方存取加密檔案。

工作站

在公司中，幾乎所有人都可以存取您的電腦，此時若加密分割區或單一檔案即可保護檔案。

23.3.1 使用 YaST 設定 Crypto 檔案系統

YaST 在安裝時會加密檔案或分割區，並加密已安裝系統中的檔案或分割區。因為加密檔案與現存分割區配置完全相合，所以任何時候都可建立加密檔案。若要加密整個分割區，請固定分割區以便在分割區配置進行加密。依照預設，YaST 建議的標準分割提案不包含已加密的分割區。在分割區的對話方塊手動新增。

在安裝時建立加密分割區

警告：密碼輸入

設定加密分割區的密碼時請注意密碼安全性警告，並將密碼牢記於心。若沒有輸入密碼，將無法存取或還原加密的檔案。

如節 3.7.5, "磁碟分割程式" (章 3, 使用 YaST 進行系統組態, ↑啟動) 中所述，YaST 分割區進階對話方塊會提供建立加密分割區所需的選項。按一下 **建立** 像是建立一般分割區時一樣。請在開啟的對話方塊中，輸入新分割區的分割參數，例如想要的格式化和裝置點。按一下 **加密檔案系統完成程序**。接著，在以下的對話方塊中輸入密碼兩次。按一下 **確定** 關閉分割對話方塊後，即可建立新的加密分割區。下次開機時，作業系統會在裝載分割區前要求輸入密碼。

如果您不想在啟動時裝載加密分割區，請在系統提示輸入密碼後，按一下 **[Enter]**。接著拒絕再次輸入密碼。在此情況下，將不會裝載加密的檔案系統，接著作業系統會繼續進行開機且不會存取您的資料。一旦分割區裝載完成，所有使用者都可以使用它。

如果只要在有需要時裝載加密的檔案系統，請啟用 *fstab* 選項對話方塊中的 *開機時不要裝載*。之後，系統開機時將不會裝載該分割區。之後若要裝載該分割區，請使用 `mount name_of_partition mount_point` 指令手動裝載。接著系統會提示您輸入密碼。分割區使用完畢後，請再使用 `umount name_of_partition` 指令解除裝載，以避免其他使用者進行存取。

在執行系統上建立加密分割區

警告：在執行系統啟動系統加密

在執行系統上建立加密分割區，也可以使用與安裝期間相同的方式。然而，加密現存分割區會損毀它所有的資料。

在執行中的系統上，選取 YaST 控制中心中的 *系統* → *分割區*。按一下 *是* 繼續。如前文所述，請按一下 *編輯*，而不要選取 *建立*。其他程序則相同。

安裝加密檔案

除了使用分割區之外，您還可以在保存機密資料的單一檔案中建立加密檔案系統。請從同一個 YaST 對話方塊，建立加密檔案系統。選取 *加密檔案*，然後輸入要建立的檔案路徑和檔案大小。接著，接受建議的格式化設定值和檔案系統類型。然後指定裝載點，同時決定系統開機時是否裝載加密檔案系統。

加密檔案有一個好處是，您不須重新分割硬碟即可新增加密檔案。他們會透過迴路設備進行裝載，且運作方式就跟一般分割區一樣。

使用 vi 加密檔案

使用加密分割區的缺點是，裝載分割區時，您至少要使用 `root` 使用者身分登入才可存取資料。若要改善此狀況，請在加密模式中使用 `vi`。

請使用 `vi -x filename` 編輯新檔案，接著 `vi` 會在加密檔案內容後提示您設定密碼。之後，每當您要存取該檔案時，`vi` 就會要求您輸入正確的密碼。

為確保更高的安全性，您可以在加密分割區中放置加密的文字檔。因為 `vi` 中使用的加密功能不是非常強大，所以會建議您建立一個加密文字檔。

23.3.2 加密抽取式媒體內容

YaST 會將抽取式媒體視為外接硬碟或 USB 快閃磁碟機，與任何其他硬碟一視同仁。您可以按照以上步驟來加密此類媒體上的檔案或分割區。但是，請勿選取在系統開機時裝載這些媒體的選項，因為通常只有在系統執行時這些媒體才會連線至系統。

23.4 安全性與機密性

Linux 或 UNIX 系統的其中一個主要特性是能夠同時 (多重使用者) 處理數個使用者，並允許這些使用者在同一部電腦上同時執行數個工作 (多工作業)。再者作業系統具有透明化的網路。使用者通常不知道他們正在使用的資料與應用程式是由其本地機器提供或是透過網路所提供。

使用多個使用者的功能，不同使用者的資料必須分開儲存。必須保證安全性與隱私。即使在電腦可以透過網路連結前，資料安全性就已經是一個重要的議題。就像今日，最重要的問題就是在遺失資料或損毀資料的媒體上 (大部份是指硬碟而言)，仍然能夠提供資料。

本小節主要是將焦點放在機密議題以及保護使用者隱私的方法上，但是它並未著墨太多關於全面安全性概念應該永遠包含的一些程序，即在適當的位置具有定期更新、可運作以及已測試的備份。如果沒有這些，您可能很難將資料還原不只是在某些硬體有缺陷的例子中，如果懷疑有人取得未授權的存取權並損毀檔案時也是如此。

23.4.1 本地安全性與網路安全性

存取資料有數種方式：

- 與具有所需資訊或具有電腦資料存取權的人員進行個人通訊
- 直接從電腦的主控台 (實體存取)
- 透過序列線
- 使用網路連結

在所有的這些例子中，在使用者存取資源或是有問題的資料前，應該先驗證其身份。網頁伺服器在這方面可能限制比較少，但是您仍然不應該對任何網友揭露所有的個人資料。

在上方的清單中，第一個例子是需要與人頻繁的互動，例如，當您正在與銀行人員聯繫時，需要證明您就是擁有該銀行帳戶的人員。接著會要求您提供簽名、PIN 或密碼以證明您就是您所聲稱的那個人。在某些狀況下，有可能從某個知道的人問出一些資訊，只要使用一些巧妙的措詞提及一些已知的片斷資訊，就有可能贏得該人員的信賴。這個受害者有可能被引導逐漸揭露更多的資訊，但卻不自知。在駭客之間，稱此為 **社會工程**。您只能以教育人們的方式以及以謹慎的方式來處理語言與資訊才能對抗這樣的受害。在攻擊者侵入電腦系統前，他們通常會嘗試鎖定接待人員、公司的客服人員或甚至是家庭成員。在許多狀況下，利用社會工程的攻擊通常都發現得很晚。

一個想要取得資料未授權存取權的人，也有可能使用傳統的方式，嘗試直接取得您的硬體。因此，應該善加保護機器，以防他人移動、替換或破壞其元件。這也適用於備份，甚至是任何的網路纜線或電線。另外也需要防護開機程序，因為有些眾所周知的按鍵組合，可能會引起不正常的行為。請設定 BIOS 與開機載入程式的密碼，以避免自己的電腦發生此狀況。

連接至序列埠的序列終端機仍然用於許多地方。不像網路介面，它們並不依賴網路通訊協定和主機通訊。它們使用簡單的纜線或紅外線埠，在設備之間來回傳送純文字字元。纜線本身是這類系統最弱的點：只要以舊型的印表機連接到系統，就可以輕易地記錄纜線上所執行的任何內容。印表機可以達成的動作也可以由其他方式來完成，端視攻擊所進行的動作而定。

讀取某個主機本地上的檔案所需的存取規則，就是在不同的主機上開啟與伺服器的網路連線。本地安全性與網路安全性之間是有區別的。這條界線在於資料必須放入封包以傳送到別的地方。

本地安全性

本地安全性是從電腦執行位置的實體環境開始。在符合您的期望與需求的適當位置安裝機器。本地安全性的主要目標就是要將使用者彼此區隔開來，如此就不會有其他的使用者可以擅用另一個使用者的權限或身份。這是一般所遵守的規則，但是對於擁有系統最高權限的 `root` 使用者更是特別需要遵守。因為 `root` 可以使用任何本地使用者的身份，而且系統不會提示它輸入密碼，就可以讀取任何本地儲存的檔案。

密碼

在 Linux 系統上，密碼不會以純文字儲存，而且輸入的文字字串也不會與儲存的型式直接符合。如果是這樣，只要有人取得該對應檔案的存取權，系統上的所有帳戶都將受到侵害。實際上，會加密儲存的密碼，而且每次輸入它時，還會再加密一次，然後再比較這兩個加密的字串。只要加密的密碼無法反轉計算成原始的文字字串，就不允許登入，這提供了更多的安全性。

實際上，這是由一種特殊的演算法所達成，也稱為 *暗門演算法*，因為它只能往一個方向運作。取得加密字串的攻擊者將無法以直接再次套用相同的演算法來取得密碼。因此，在找到一個組合與加密的密碼相似前，它將需要測試所有可能的字元組合。使用 8 個字元長的密碼，將需要計算相當多的可能組合。

在 70 年代，有人主張此方法比其他方法更為安全，因為使用了相當慢的演算法，它需要好幾秒才能加密一個密碼。然而，在另一方面，PC 已變得非常強大，可以在每秒內進行數十萬或甚至數百萬的加密。因此，加密的密碼不應該對一般的使用者顯示（一般使用者無法讀取 `/etc/shadow`）。更重要的一點是密碼必須不易猜測，以防止密碼因為某些錯誤而變成可見。因此，將密碼如 "tantalize"「翻譯」成 "t@nt@1lz3" 並沒有太大的用處。

以某些看起來與字母相似的數字取代一個字的某些字母並不夠安全。使用字典猜測的密碼破解程式也可以猜測到這樣的替換。最好的方法是使用一個沒有一般意義但只對您個人有意義的文字，像是某句子或某書名文字的第一個字母，例如 "The Name of the Rose" by Umberto Eco。這將可以給予下列安全的密碼："TNotRbUE9"。相反的，像是 "beerbuddy" 或 "jasmine76"，將很容易被對您只有一些瞭解的人士猜到。

開機程序

設定系統使其無法從軟碟或從 CD 開機，可以將設備整個移除或是設定 BIOS 的密碼，並將 BIOS 設成只允許從硬碟開機。一般而言，Linux 系統是由開機載入程式所啟動，允許您將額外的選項傳遞至開機核心。在 `/boot/grub/menu.lst` 中設定額外的密碼以防止其他人在開機時使用像是這樣的參數（請參閱 [章 29, 開機載入器\[399\]](#)）。這對於系統的安全性非常重要。不只是核心本身是以 root 權限執行，它也是第一個授權者可以在系統啟動時授與 root 權限。

檔案許可權

一般而言，儘可能以限制最多的權限來進行指定的工作。例如，閱讀或撰寫電子郵件絕對不需要使用 `root` 的身份。如果郵件程式有問題，而這個問題有可能造成在程式啟動時，利用它以相同的程式權限來進行攻擊。遵照上方原則，可以減少可能的損毀。

在 SUSE 版本中所包含的 200,000 個以上的檔案權限是經過小心選擇的。安裝其他軟體或其他檔案的系統管理員應該極為注意何時會這麼做，特別是在設定權限時。資深以及對安全性敏感的系統管理員，永遠會使用 `-l` 選項加上 `ls` 指令以取得龐大的檔案清單，這可允許它們立即偵測是否有任何不正確的檔案權限。不正確的檔案屬性並不只是代表檔案已經過變更或遭到刪除。這些修改過的檔案有可能是 `root` 曾經執行過它們，或是在組態檔中，程式可能以 `root` 的權限使用過這些檔案。這大幅地增加了攻擊者的機會。像這一類的攻擊稱為布穀鳥式借巢孵蛋，因為不同的使用者(鳥)執行(孵化)程式(蛋)，就像布穀鳥誘使其他的鳥孵化牠自己的蛋。

SUSE Linux 系統包含 `permissions`、`permissions.easy`、`permissions.secure` 以及 `permissions.paranoid` 檔案，全部都在 `/etc` 目錄中。這些檔案的目的就是用以定義特殊的權限，像是全球都可以寫入的目錄或是供檔案使用的 Setuser ID 位元(具有 Setuser ID 位元集的程式並不是以啟動它的使用者權限執行，而是以檔案擁有者的權限執行，大部份是指 `root` 而言)。管理員可以使用 `/etc/permissions.local` 檔案以新增他自己的設定值。

若要定義 SUSE 的組態程式使用上方哪些檔案以據此設定權限，請選取 YaST 中的 安全性。若要學習更多關於此主題的詳細資訊，請參閱 `/etc/permissions` 中的備註或是 `chmod` 的手冊頁(`man chmod`)。

緩衝區溢位與格式字串問題

每當程式應該處理使用者可以或可能變更的資料時，就應該特別謹慎處理，不過這個問題大部份是針對應用程式設計人員而言，而非一般使用者。程式設計人員務必確保其應用程式以正確的方式解譯資料，而不需將它們寫入空間不足以儲存它們的記憶體區域中。另外，程式應該使用為該目的所定義的介面，以一致性的方式傳遞資料。

如果在寫入緩衝區時，未考量記憶體緩衝區的實際大小，就有可能發生 緩衝區溢位。在某些例子中，此資料(由使用者所產生)使用了比緩衝區可用空間更多的空間。結果，所寫入的資料超過了緩衝區資料的結尾，因此在某些狀況下，

儘可能讓程式執行使用者 (而非程式設計人員) 所設定的程式順序，而不只是處理使用者資料而已。這一類的問題有可能造成嚴重的後果，特別是如果使用特別的權限執行程式的話 (請參閱 "檔案許可權" [311])。

格式字串問題的運作方式有些不同，但是它又是有可能導致程式出狀況的使用者輸入。大部份而言，是利用這些程式的錯誤以及特殊的權限 (Setuid 以及 Setgid 程式)，這也表示您可以移除對應的執行權限，以保護資料與系統免於這類的問題。此外，最好的方法是使用最低可能的權限來套用原則 (請參閱 "檔案許可權" [311])。

假使緩衝區溢出與格式字串問題是與使用者資料處理相關的問題，如果已指定存取權給本地帳戶，就不只可以利用它們。許多已報告的問題都有可能透過網路連結被利用。因此，應該將緩衝區溢位與格式字串問題，分類成與本地安全性以及與網路安全性相關。

病毒

與某些人的說法相反，有些病毒是在 Linux 上執行。然而，那些作者所發行的已知病毒是一種 概念證明，以證明該技術如預期運作。目前為止沒有病毒已經蔓延。

如果沒有寄主得以寄生，病毒將無法生存和蔓延。在此例子中，宿主將會是一個程式或是系統的重要儲存區域，例如主要開機記錄，也就是病毒的程式碼可以寫入的。由於多個使用者的功能，Linux 可以將存取權限制為某些檔案，特別是某些重要的系統檔案。因此，如果您以 root 權限進行一般工作，將會增加系統受到病毒感染的機會。相反的，如果您遵照上方所提及的原則，儘可能使用最低權限，則感染病毒的機率便微乎其微。

除此之外，您絕對不要從網際網路上某些您不了解的網站執行程式。SUSE 的 RPM 套件帶有加密的簽名以做為數位標籤，必須小心謹慎才能建立它們。病毒通常代表管理員或使用者缺乏必要的安全性認知，造成原先應該非常安全的系統有感染的風險。

病毒不應該與蠕蟲混淆，它完全屬於網路世界。蠕蟲並不需要宿主就可以散播。

網路安全性

網路安全性對於防護從外面開始的攻擊非常重要。需要使用者名稱與密碼以驗證使用者的一般登入程序，仍然是本地的安全性問題。在透過網路登入的特定

例子中，區別兩個安全性領域。一直到實際驗證所發生的事，都是網路安全性，而之後所發生的任何事則為本地安全性。

X 視窗系統以及 X 驗證

如一開始所提及，網路通透性是 UNIX 系統的其中一個主要特性。X 是 UNIX 作業系統的視窗系統，可以印象深刻的方式來利用此功能。使用 X 基本上就可以輕易地在遠端主機中登入並啟動圖形程式，接著就可以透過網路在您的電腦上顯示該程式。

應該使用 X 伺服器以遠端顯示 X 用戶端時，X 伺服器應該從未授權的存取來保護它(即該顯示)所管理的資源。更具體而言，某些權限必須指定給用戶端程式。使用 X Window 系統，就可以使用兩種方法來進行此動作，分別為以主機為基礎的存取控制以及以 Cookie 為基礎的存取控制。第一個是依賴用戶端應該執行的主機 IP 位址。負責控制的程式為 `xhost`。`xhost` 會將合法用戶端的 IP 位址輸入屬於 X 伺服器的小資料庫。然而，依賴 IP 位址的驗證並不安全。例如，如果有第二個使用者在傳送用戶端程式的主機上工作，則該使用者也可以存取 X 伺服器就像有些人偷竊 IP 位址一樣。由於這些缺點，在此將不對此驗證方法做詳細的說明，但是您可以使用 `man xhost` 來學習它。

在以 Cookie 為基礎的存取控制例子中，所產生的字元字串只能讓 X 伺服器以及合法的使用者知道，就像某種 ID 卡。此 Cookie (這個字並不是指一般的餅乾，而是指包含警語的中國幸運籤餅)是儲存在使用者主目錄的 `.Xauthority` 檔案中，並且提供給任何需要使用 X 伺服器的 X 用戶端來顯示視窗。使用者可以使用 `xauth` 工具來檢查 `.Xauthority` 檔案。如果您要重新命名 `.Xauthority` 或不小心從主目錄刪除檔案，您將無法開啟任何新的視窗或 X 用戶端。在 Xsecurity 的 `man` 頁面中，可以參閱更多關於 X Window 系統安全性機制的詳細資訊(`man Xsecurity`)。

SSH (安全外圍程序) 可用以完整地加密網路連線，並將它以背景作業方式轉遞至 X 伺服器，但是使用者不會察覺到該加密機制。這又可稱為 X 轉遞。X 轉遞是藉由模擬伺服器端的 X 伺服器並在遠端主機上設定外圍程序的 `DISPLAY` 變數來達成。您可以在 [節 23.2, "SSH: 安全網路作業" \[301\]](#) 找到有關 SSH 的詳細資訊。

警告

如果您未考慮您所登入的主機是否為安全的主機，請勿使用 X 轉遞。啟用 X 轉遞後，攻擊者可以透過 SSH 連線進行驗證，以入侵您的 X 伺服器並查看您的鍵盤輸入。

緩衝區溢位與格式字串問題

如["緩衝區溢位與格式字串問題"\[311\]](#)中所討論，應該將緩衝區溢位與格式字串問題，分類成與本地與網路安全性相關的問題。隨著這類問題的本地變化、網路程式的緩衝溢位，當成功地利用這些瑕疵時，大部份都會使用它們來取得 root 權限。即使不是這樣，攻擊者也可能使用該問題來取得未授權的本地帳戶，以利用其他可能存在於系統上的弱點。

透過網路連結以利用緩衝區溢位與格式字串問題，絕對是遠端攻擊最常見的形式。對於這些程式的不當利用以開拓新發現的安全性漏洞，通常會張貼在安全性郵件清單上。使用它們就可以鎖定弱點，而不需知道程式碼的詳細資訊。經過這些年來，經驗顯示提供這些不當利用的程式碼對於更加安全的作業系統貢獻良多，很明顯地這是因為作業系統設計者被迫必須修正其軟體中的問題。有了免費軟體，每個人都可以存取原始程式碼 (SUSE Linux 是隨附在所有可用的原始程式碼)，而且任何找到弱點及其不當利用程式碼的人，都可以提交修補程式以修正對應的問題。

拒絕服務

此種拒絕服務 (DoS) 攻擊的主要目的就是要封鎖伺服器程式或甚至是整個系統，有可能以數種方式達成：使伺服器超出負載、以垃圾封包使其忙碌，或利用遠端緩衝區溢位。通常 DoS 攻擊的唯一目的就是要使服務消失。然而，一旦特定的服務變成無法使用時，通訊就有可能變成 **攔截式攻擊** (封包攔截、TCP 連線攔截、偽裝式攻擊) 以及 DNS 定址攻擊的弱點。

攔截式：封包攔截、攔截、偽裝式攻擊

一般而言，由攻擊者所執行的遠端攻擊會將自己放在通訊主機之間，可稱為 **攔截式攻擊**。幾乎所有類型的攔截式攻擊共同點就是，受害者通常不會察覺正在進行的事情。這類的攻擊有許多可能的變化，例如，攻擊者有可能接收某個連線要求並將它轉遞給目標機器本身。現在受害者已無意識地建立與錯誤主機的連線，因為另一端假裝自己是合法的目標機器。

攔截式攻擊最簡單的形式稱為嗅探器攻擊者「只是」監聽經過的網路流量。至於更複雜的攻擊，「攔截式攻擊」有可能嘗試接收已建立的連線(攔截)。如果要這麼做，攻擊者將需要一些時間分析封包，才能夠預測屬於該連線的 TCP 序號。當攻擊者最後奪取目標主機的角色時，受害者將會注意到此，因為他們會取得錯誤訊息說明連線因失敗而終止。有些通訊協定並沒有透過加密以防止攔截，只有在建立連線時執行簡單的驗證程序，使其更容易成為攻擊者攻擊的弱點。

偽裝式攻擊是一種將封包加以修改以包含假的來源資料的攻擊，通常是 IP 位址。大部份主動攻擊的形式是依賴傳送這樣的假封包，這在 Linux 機器上只能由進階使用者執行 (root)。

許多所提及的攻擊都是結合 DoS 來執行。如果攻擊者見到有機會可以將某個主機出其不意地擊倒，即使只是很短的時間，這將使其更易於做主動的攻擊，因為該主機將有一段時間無法妨礙其攻擊。

DNS 定址攻擊

DNS 定址攻擊是指攻擊者破壞 DNS 伺服器的快取，即使用偽裝的 DNS 回覆封包以回覆它、嘗試使伺服器傳送某種資料給正在要求伺服器資訊的受害者。許多伺服器以 IP 位址或主機名稱來維持與其他主機的信任關係。攻擊者必須非常瞭解對主機之間信任關係的實際結構，以便將自己偽裝成其中一個信任的主機。通常，攻擊者會分析從伺服器所收到的一些封包以取得所需的資訊。攻擊者通常也需要在名稱伺服器鎖定適時的 DoS 攻擊。請使用加密連線，以便能夠驗證要連線的主機身份來保護自己。

蠕蟲

一般常會將蠕蟲與病毒混淆，但是在這兩者之間其實有一個明確的界限。不像病毒，蠕蟲並不需要感染某個宿主程式才能存活。它們相當擅長於在網路結構上儘可能迅速散播。過去所出現的蠕蟲(例如 Ramen、Lion 或 Adore)都是利用伺服器程式已知的安全性漏洞像是 bind8 或 lprNG。防止蠕蟲的入侵其實相當容易。假設在安全漏洞的發現以及蠕蟲攻擊伺服器之間有些間隔，這樣正是感染程式的更新版本準時提供的好機會。不過這只對管理員在有問題的系統上真正安裝了安全性更新才有用。

23.4.2 一些一般的安全性秘訣與技巧

為了能妥善處理安全性問題，則必須隨時使用新開發並隨時關心最新的安全性問題。保護系統免於這些問題的最好方法是盡快取得並安裝安全聲明中推薦的套件。SUSE安全聲明由郵件清單發布，可透過下列連結<http://www.novell.com/linux/security/securitysupport.html> 訂閱。suse-security-announce@suse.de 清單是第一手關於更新套件的資訊來源，並且包含 SUSE 主動貢獻者的安全性團隊成員。

郵件清單 suse-security@suse.de 是討論任何所關心的安全性問題的好地方。可在同一網頁訂閱

bugtraq@securityfocus.com 是全球其中一個最知名的安全性郵件清單。建議閱讀這個清單，它每天會收到 15 到 20 個張貼。在 <http://www.securityfocus.com> 可以找到更多的詳細資訊。

下列是關於處理基本安全性問題的有效規則清單：

- 根據每個工作儘可能都使用限制最多的權限集合原則，避免以 `root` 的身份執行一般工作。這將可減少得到布穀鳥式借巢孵蛋或病毒的風險，並防止自己犯錯。
- 如果有可能，請永遠嘗試使用加密的連線在遠端機器上工作。使用 `ssh` (安全外圍程式) 以取代 `telnet`、`ftp`、`rsh` 以及 `rlogin` 應該為標準的慣例。
- 避免使用僅依據 IP 位址的驗證方法。
- 請試著將最重要的網路相關套件保持在最新的狀態，並訂閱對應的郵件清單以接收這類程式新版的公告 (`bind`、`sendmail`、`ssh` 等等)。同樣的原則也適用於與本地安全性相關的軟體。
- 變更 `/etc/permissions` 檔案以最佳化對系統安全性很重要的檔案權限。如果您從程式移除 `Setuid` 位元，它有可能再也無法依照所需的方式執行工作。另一方面，在大部份的情況下，請想想程式也將不再有潛在性的安全性風險。您可以採取全球可以寫入的目錄與檔案的相似方式。
- 停用所有非必要的網路服務，讓伺服器正常運作。這可讓您的系統較為安全。開啟埠及 `LISTEN` 插槽狀態，都可以在 `netstat` 程式中找到。至於其選

項，建議使用 `netstat -ap` 或 `netstat -anp`。`-p` 選項允許您查看哪些程序正在佔據哪個名稱下的埠。

比較 `netstat` 的結果以及從主機外面對埠所做全面掃描的結果。進行此工作的最佳程式為 `nmap`，它不只檢查機器的埠，也會針對哪些服務在等待這些埠做出摘要。然而，埠掃描有可能被視為一種侵略行為，因此若無管理員的明確允許，請勿在主機上執行此動作。最後，請記得不只要掃描 TCP 埠，另外也必須掃描 UDP 埠 (`-sS` 與 `-sU` 選項)。

- 如果要以可靠的方式來監督系統檔案的完整性，請使用 `AIDE` (進階入侵偵測環境) 程式，可在 `SUSE Linux` 找到。加密 `AIDE` 所建立的資料庫以防有人篡改它。另外，在您的機器之外複製一份資料庫備份，將它儲存在未連接網路的外部資料媒體上。
- 在安裝協力廠商軟體時請謹慎小心。曾經有駭客將木馬程式建入安全軟體套件的 `tar` 歸檔中，所幸很快就被發現了。如果您要安裝二進位的套件，請確定您對所下載的網站沒有疑慮。

`SUSE` 的 `RPM` 套件是以 `GPG` 簽名。`SUSE` 用以簽名的金鑰為：

ID: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.rpm` 指令顯示解除安裝的套件，其檢查總數與簽名是否正確。尋找該版本第一張 CD 上的金鑰以及全球大部份金鑰伺服器上的金鑰。

- 定期檢查使用者與系統檔案的備份。請考慮如果您未測試備份是否可以使用，它有可能沒有用處。
- 檢查記錄檔。請儘可能隨時撰寫小型的程序檔以搜尋可疑的項目。不可否認的，這不完全是一個鎖碎的工作。最後只有您才知道哪些項目是不尋常的，哪些才是正常的。
- 使用 `tcp_wrapper` 以限制存取在機器上所執行的個別服務，因此您對於哪個 IP 位址可以連線至服務具有明確的控制權。如需關於 `tcp_wrapper` 的進一步資訊，請參閱 `tcpd` 與 `hosts_access` (`man 8 tcpd`、`man hosts_access`) 的手冊頁面。
- 使用 `SUSEfirewall` 以加強 `tcpd` 所提供的安全性 (`tcp_wrapper`)。

- 將您的安全性措施設計成重複型：出現兩次的訊息總比沒有訊息來的好。

23.4.3 使用集中式安全性報告位址

如果您發現安全性相關的問題 (請先檢查可用的更新套件)，請撰寫電子郵件給 security@suse.de。請輸入問題的詳細說明，以及有關的套件版本編號。SUSE 會盡快回復。您最好能以 PGP 加密電子郵件。SUSE 的 PGP 金鑰為：

```
ID: 3D25D3D9 1999-03-06 SUSE Security Team <security@suse.de>  
Key fingerprint = 73 5F 2E 99 DF DB 94 C4 8F 5A A3 AE AF 22 F2 D5
```

這個金鑰也可以從 <http://www.novell.com/linux/security/securitysupport.html> 下載。

Linux 存取控制清單

本章節簡略介紹 Linux 檔案系統的 POSIX ACL (存取控制清單) 相關背景和功能。ACL 可以看做是傳統檔案系統物件許可權概念的擴充。使用 ACL，定義許可權比傳統許可權概念更有彈性。

POSIX ACL 這個詞彙表示這是真正的 POSIX (可攜式作業系統介面) 標準。各自的草稿標準 POSIX 1003.1e 和 POSIX 1003.2c 已經因多種因素而撤銷了。然而，在屬於 UNIX 家族中很多系統發現的 ACL，是以這些草稿為基礎，而且本章節敘述的系統 ACL 實行，也是依循這兩個標準。要查閱它們，請到 <http://wt.xpilot.org/publications/posix.1e/>。

24.1 ACL 的優點

傳統上，會為 Linux 系統的每一個檔案物件，定義三組的許可權。這三組包括讀取 (r)、寫入 (w) 以及執行 (x) 許可權，用於三種使用者類型 (檔案擁有者、群組和其他使用者) 的每一個。此外，也可以設定設定使用者 *id*、設定群組 *id* 以及黏貼位元。這個小小的概念對於大部份實際情況已經足夠了。不過，對於更複雜的案例或進階應用程式，系統管理員以前必須使用一些訣竅才能克服傳統許可權概念的限制。

ACL 可以用於需要擴充傳統檔案許可權概念的狀況。它們允許指定許可權給個別使用者或群組，即使這些許可權並不對應至原始擁有者或所屬群組。存取控制清單是 Linux 核心的特性之一，目前 ReiserFS、Ext2、Ext3、JFS 和 XFS 都能支援。使用 ACL 可以理解複雜的情況，無需在應用程式層次上實行複雜的許可權模型。

以 Linux 伺服器取代 Windows 伺服器時，ACL 的優點特別明顯。部份連線的工作站還可以在移轉之後，繼續在 Windows 下執行。Linux 系統提供檔案和列印服務給含 Samba 的 Windows 用戶端使用。假設 Samba 支援存取控制清單，使用者許可權可以同時在 Linux 伺服器和使用圖形使用者介面的 Windows (限 Windows NT 和更新的版本) 上設定。透過 winbindd，還可以指定許可權給只存取 Windows 領域中而且沒有 Linux 伺服器任何帳戶的使用者。

24.2 定義

使用者類別

傳統 POSIX 許可權概念使用三種使用者類別，用於在檔案系統中指定許可權：擁有者、所屬群組以及其他使用者。三種許可權位元可以設定給每一個使用者類別，提供許可權來讀取 (r)、寫入 (w) 以及執行 (x)。

存取 ACL

所有檔案系統物件 (檔案和目錄) 的使用者和群組存取許可權，是透過存取 ACL 的方法來決定。

預設 ACL

預設的 ACL 只能套用至目錄。它們在檔案系統物件建立時，會決定從其上層目錄繼承的許可權。

ACL 項目

每一個 ACL 包含一組的 ACL 項目。ACL 項目包含類型 (請參閱 [表格 24.1, "ACL 項目類型" \[321\]](#))，使用者的識別符號或項目參照的群組，以及一組許可權。對於部份項目類型，群組或使用者的識別字尚未定義。

24.3 處理 ACL

[表格 24.1, "ACL 項目類型" \[321\]](#) 摘錄 6 種可能的 ACL 項目類型，每個類型定義一個使用者或一群組使用者的許可權。擁有者項目定義使用者所屬檔案或目錄的許可權。所屬群組項目定義檔案的所屬群組許可權。超級使用者可以使用 `chown` 或 `chgrp` 變更擁有者或所屬群組，在這樣情況下，擁有群組項目會參照新擁有者和所屬群組。每個命名的使用者項目定義使用者許可權，該項目指定於項目的辨識符號欄位，此欄位在 [表格 24.1, "ACL 項目類型" \[321\]](#) 中是以文字形式顯示在中央欄位。每個命名的群組項目定義群組許可權，該許可權指定於

項目的辨識符號欄位。只有命名的使用者和命名的群組擁有的辨識符號欄位不是空的。其他項目定義所有其他使用者許可權。

透過定義那些有效項目的許可項目以及遮罩的項目，遮罩項目會進一步限制由命名的使用者、命名的群組以及所屬群組項目授與的許可權。如果許可權存在於之前提及的任一項目和遮罩中，它們都是有效的。只有包含於遮罩或實際項目的許可權是無效的 -- 表示未授與此許可權。定義在使用者和所屬群組項目的所有許可權永遠有效。[表格 24.2, "遮罩存取許可權" \[321\]](#) 中的範例會示範此機制。

有兩種基本類型的 ACL：最小值 ACL 只包含擁有者、所屬群組和其他類型的項目，對應檔案和目錄的傳統許可權位元。延伸 ACL 則不僅如此。它必須包含遮罩項目而且可以包含命名使用者和命名群組類型的多個項目。

表格 24.1 ACL 項目類型

類型	文字形式
擁有者	user::rwx
命名的使用者	user:name:rwx
所有群組	group::rwx
命名的群組	group:name:rwx
遮罩	mask::rwx
其他	other::rwx

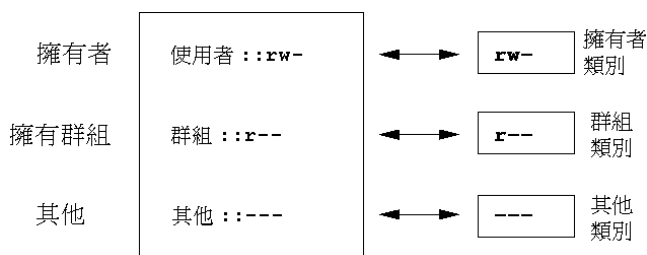
表格 24.2 遮罩存取許可權

項目類型	文字形式	許可權
命名的使用者	user:geeko:r-x	r-x
遮罩	mask::rw-	rw-
	有效許可權:	r--

24.3.1 ACL 項目和檔案模式許可權位元

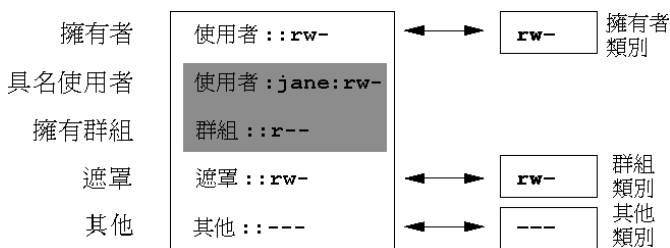
圖形 24.1, "最小 ACL: ACL 項目比較許可權位元" [322] 和 圖形 24.2, "延伸 ACL: ACL 項目比較許可權位元" [322] 列舉最小值 ACL 和延伸 ACL 的兩個案例。圖的結構分成三個區塊 — 左方區塊顯示 ACL 項目的類型格式，中間區塊顯示範例 ACL，而右方區塊則顯示傳統許可權概念的許可權位元，例如，以 `ls -l` 顯示。在二者案例中，擁有者類別許可權會對應至 ACL 項目擁有者。其他類別許可權對應個別 ACL 項目。不過，群組類別許可權的對應與這兩個案例不同。

圖形 24.1 最小 ACL: ACL 項目比較許可權位元



在最小 ACL 案例中(不含遮罩)，群組類別許可權會對應至 ACL 項目所屬群組。這會顯示於 圖形 24.1, "最小 ACL: ACL 項目比較許可權位元" [322]。在延伸 ACL 案例中(含遮罩)，群組類別許可權是對應至遮罩項目。這會顯示於 圖形 24.2, "延伸 ACL: ACL 項目比較許可權位元" [322]。

圖形 24.2 延伸 ACL: ACL 項目比較許可權位元



此對應方法可以確定應用程式互動順暢，無論它們是否擁有 ACL 支援。透過許可權位元方法所指定的存取許可權，代表藉由 ACL 所做的所有其他「微調」上限。對許可權位元所做的變更會由 ACL 反映，反之亦然。

24.3.2 含存取 ACL 的目錄

存取 ACL 的處理如以下範例所示範：

建立目錄之前，使用 `umask` 指令來定義每次建立檔案物件時，要遮罩的存取許可權。指令 `umask 027` 會指定完整範圍的許可權 (0) 給擁有者，拒絕群組寫入存取 (2) 以及不提供其他使用者任何許可權 (7)，以設定預設許可權。`umask` 實際上會遮罩相應的許可權位元或關閉它們。如需詳細資訊，請參閱相應的 `man` 頁面 (`man umask`)。

`mkdir mydir` 應該會以 `umask` 設定的預設許可權建立 `mydir` 目錄。使用 `ls -dl mydir` 檢查所有許可權是否指定正確。範例的輸出如下：

```
drwxr-x--- ... tux project3 ... mydir
```

使用 `getfacl mydir` 檢查 ACL 的啟始狀態。會提供類似以下的資訊：

```
# file: mydir
# owner: tux
# group: project3
user::rwx
group::r-x
other::---
```

`getfacl` 的輸出會精確反應許可權位元以及 ACL 項目的對應，說明於 [節 24.3.1, "ACL 項目和檔案模式許可權位元" \[322\]](#)。前三個輸出行顯示目錄的名稱、擁有者和所屬群組。接下來三行包含三個 ACL 項目擁有者、所屬群組和其他。實際上，在此最小 ACL 案例中，`getfacl` 指令不會產生任何您以 `ls` 指令無法取得的資訊。

以下列指令修改 ACL 以指定讀取、寫入和執行許可權給另一個使用者 `geeko` 以及另一個群組 `mascots`：

```
setfacl -m user:geeko:rwx,group:mascots:rwx mydir
```

選項 `-m` 提示 `setfacl` 修改現有的 ACL。以下引數指示要修改的 ACL 項目 (多個項目由逗點分開)。最後部份指定這些修改要套用至什麼目錄名稱。使用 `getfacl` 指令，查看產生的 ACL。

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
```

```
group::r-x
group:mascots:rwX
mask::rwX
other::---
```

除了為使用者 `geeko` 和群組 `mascots` 實施的項目之外，已經產生一個遮罩項目。會自動設定遮罩項目讓所有的許可權生效。除非使用 `-n` 停用此功能，`setfacl` 會將現有遮罩項目調整為修改過的設定值。遮罩會定義群組類別中所有項目的最大有效存取許可權。此包括命名使用者、命名群組以及所屬群組。`ls -dl mydir` 顯示的類別群組許可權位元，對應至遮罩項目。

```
drwxrwx---+ ... tux project3 ... mydir
```

輸出的第一欄現在包含額外的 `+`，指示此項目有一個**延伸 ACL**。

根據 `ls` 指令的輸出，遮罩項目的許可權包括寫入存取。傳統上，此類的許可權位元表示所屬群組(在此為 `project3`)也擁有目錄 `mydir` 的寫入存取。然而，所屬群組的有效存取許可權會對應至許可權的重疊部分，重疊部分定義在所屬群組和遮罩—在本範例中是 `r-x` (請參閱 [表格 24.2, "遮罩存取許可權" \[321\]](#))。就範例中所屬群組的有效許可權而言，即使新增 ACL 項目之後也不會變更有效所有權。

使用 `setfacl` 或 `chmod`，編輯遮罩項目。例如，使用 `chmod g-w mydir`。接著 `ls -dl mydir` 顯示如下：

```
drwxr-x---+ ... tux project3 ... mydir
```

`getfacl mydir` 提供下列輸出：

```
# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx          # effective: r-x
group::r-x
group:mascots:rwX       # effective: r-x
mask::r-x
other::---
```

執行 `chmod` 指令，從群組類別位元移除寫入許可權之後，`ls` 指令的輸出就可以知道遮罩位元必須同時變更：寫入許可權再次限制為 `mydir` 的擁有者。`getfacl` 的輸出可確認此項結果。此輸出針對有效許可權未對應至原始許可權的所有項目包含一個註解，因為它們是根據遮罩項目過濾的。可以隨時使用 `chmod g+w mydir` 復原原始許可權。

24.3.3 含預設 ACL 的目錄

目錄可以有一個預設的 ACL，它是特別的 ACL。建立物件時，它定義目錄中物件繼承的存取許可權。預設 ACL 會影響子目錄和檔案。

預設 ACL 的作用

子目錄的預設 ACL，其許可權傳送至檔案以及子目錄的方法有兩種：

- 子目錄繼承上層目錄的預設 ACL，當成預設 ACL，也當成存取 ACL。
- 檔案會將預設 ACL 繼承為存取 ACL。

建立檔案系統的所有系統呼叫，會使用 `mode` 參數，它會為新建立的檔案系統物件，定義存取許可權。如果上層目錄沒有預設 ACL，`umask` 定義的許可權位元會從 `mode` 參數傳送的許可權中去除，將結果指定給新物件。如果上層目錄的預設 ACL 存在，指定給新物件的許可權位元會對應至 `mode` 參數的許可權重疊部份，以及定義在預設 ACL 的許可權。`umask` 在此案例中不予處理。

預設 ACL 的應用

以下三個範例顯示目錄和預設 ACL 的主要作業：

1. 新增預設 ACL 至現有的目錄 `mydir`，透過：

```
setfacl -d -m group:mascots:r-x mydir
```

`setfacl` 指令的 `-d` 選項，提示 `setfacl` 在預設 ACL 執行以下修改 (`-m` 選項)。

仔細查看指令的結果：

```
getfacl mydir

# file: mydir
# owner: tux
# group: project3
user::rwx
user:geeko:rwx
group::r-x
group:mascots:rwx
mask::rwx
```

```

other: ---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other: ---

```

getfacl 傳回存取 ACL 和預設 ACL 二者。預設 ACL 是由開頭為 default 的所有行形成的。雖然您只是在預設 ACL 的 mascots 群組的一個項目執行 setfacl 指令，不過 setfacl 會自動從存取 ACL 複製所有其他項目來建立有效的預設 ACL。預設 ACL 對於存取許可權沒有立即的作用。它們只會在建立檔案系統物件時才有作用。這些新物件只會從其上層目錄的預設 ACL 繼承許可權。

2. 在下一個範例中，使用 mkdir，在 mydir 目錄中建立子目錄，它會繼承預設 ACL。

```

mkdir mydir/mysubdir

getfacl mydir/mysubdir

# file: mydir/mysubdir
# owner: tux
# group: project3
user::rwx
group::r-x
group:mascots:r-x
mask::r-x
other: ---
default:user::rwx
default:group::r-x
default:group:mascots:r-x
default:mask::r-x
default:other: ---

```

如預期般，新建立的子目錄 mysubdir 擁有上層目錄預設 ACL 的許可權。mysubdir 的存取 ACL 與 mydir 的預設 ACL 完全相同。目錄傳給其從屬物件的預設 ACL，也是如此。

3. 使用 touch 於 mydir 目錄建立檔案，例如，touch mydir/myfile。接著 ls -l mydir/myfile 會顯示如下：

```
-rw-r-----+ ... tux project3 ... mydir/myfile
```

getfacl mydir/myfile 輸出，如下：

```

# file: mydir/myfile
# owner: tux

```

```
# group: project3
user::rw-
group:r-x      # effective:r--
group:mascots:r-x  # effective:r--
mask::r--
other::---
```

建立新檔案時，touch 會使用值為 0666 的模式。這表示檔案建立後，所有使用者類別都有讀取和寫入的許可權，而且在 umask 或預設 ACL 中沒有其它限制 (請參閱 ["預設 ACL 的作用" \[325\]](#))。實際上，這表示所有未包含在 mode 值中的存取許可權，會從各自的 ACL 項目移除。雖然未從群組類別的 ACL 項目移除許可權，將遮罩項目修改為遮罩許可權，不在此模式中設定。

此方法可以確定應用程式 (例如編譯器) 與 ACL 的互動平順。您可以建立存取許可權受限的檔案，然後標示成可執行。mask 機制可以保證正確的使用者和群組可以執行它們。

24.3.4 ACL 檢查演算法

任何程序或應用程式授與存取 ACL 保護的檔案系統物件前，會先套用檢查演算法。有個基本規則是，ACL 項目會按照以下順序檢查：*擁有者、命名使用者、所屬群組或命名群組以及其他*。處理的存取合乎最符合程序的項目。許可權不會累積。

如果程序屬於一個以上的群組而且可能符合多個群組項目，就會變得更複雜。項目是從符合必要許可權的合適項目中隨機選取的。它與觸發最後的結果 "access granted" 的項目無關。同樣地，如果適合的群組項目不包含需要的許可權，隨機選取的項目會觸發最後的結果「拒絕存取」。

24.4 應用程式的 ACL 支援

ACL 可以用來實行很複雜的許可權情況，符合現在的應用程式需求。傳統許可權概念和 ACL 可以使用聰明的方法結合。基本檔案指令 (cp、mv、ls 等等) 都支援 ACL，Samba 也支援 ACL。

可惜的是，許多編輯器和檔案管理員都不支援 ACL。例如，使用 Konqueror 複製檔案時，這些檔案的 ACL 會遺失。使用編輯器修改檔案時，檔案的 ACL 有時候會保留，有時候則不保留，由編輯器的備份模式決定。如果編輯器將變更

寫入原始檔案，會保留存取 ACL。如果編輯器將更新的內容儲存至新檔案，後來重新命名成舊檔案名稱，則除非編輯器支援 ACL，否則 ACL 可能會遺失。除了 star 歸檔設備之外，目前沒有備份應用程式可以保存 ACL。

24.5 若需詳細資訊

如需關於 ACL 的詳細資訊，請參閱 <http://acl.bestbits.at/>。請參閱 `getfacl(1)`、`acl(5)` 和 `setfacl(1)` 的 man 頁面。

系統監視公用程式

有許多程式和機制可用來檢查您的系統狀態，其中一些會在這裡介紹。另外還會介紹一些例行工作使用的公用程式，及其重要參數。

所有介紹的指令都會有相關的輸出範例。在這些範例中，第一行為指令本身(在金錢符號提示之後)。會以方括號 [...] 表示備註，備註較長時會進行換行。因為備註過長而換行時，會以反斜線 (\) 斷行。

```
$ command -x -y
output line 1
output line 2
output line 3 is annoyingly long, so long that \
    we have to break it
output line 3
[...]
output line 98
output line 99
```

為了介紹更多公用程式，說明的部份保持簡短。如需所有指令的相關資訊，請參閱 **man** 頁面。大部份的指令都能解讀參數 `--help`，此參數可產生一份可能參數的簡短清單。

25.1 開啟檔案的清單：lsof

如果要檢視為了程序而開啟的所有檔案之清單及其程序 ID *PID*，可使用 `-p`。例如，要檢視目前外圍程序正在使用的所有檔案，可輸入：

```
$ lsof -p $$
COMMAND PID USER  FD   TYPE DEVICE SIZE      NODE NAME
zsh      4694  jj    cwd   DIR    0,18    144  25487368 /suse/jj/t
```

```
(totan: /real-home/jj)
zsh      4694   jj   rtd    DIR     3,2      608          2 /
zsh      4694   jj   txt    REG     3,2    441296      20414 /bin/zsh
zsh      4694   jj   mem    REG     3,2    104484      10882 /lib/ld-2.3.3.so
zsh      4694   jj   mem    REG     3,2     11648      20610
/usr/lib/zsh/4.2.0/zsh/rlimits.so
[...]
zsh      4694   jj   mem    REG     3,2    13647      10891 /lib/libdl.so.2
zsh      4694   jj   mem    REG     3,2     88036      10894 /lib/libnsl.so.1
zsh      4694   jj   mem    REG     3,2    316410     147725 /lib/libncurses.so.5.4
zsh      4694   jj   mem    REG     3,2    170563      10909 /lib/tls/libm.so.6
zsh      4694   jj   mem    REG     3,2   1349081      10908 /lib/tls/libc.so.6
zsh      4694   jj   mem    REG     3,2      56      12410
/usr/lib/locale/de_DE.utf8/LC_TELEPHONE
[...]
zsh      4694   jj   mem    REG     3,2      59      14393
/usr/lib/locale/en_US/LC_NUMERIC
zsh      4694   jj   mem    REG     3,2   178476      14565
/usr/lib/locale/en_US/LC_CTYPE
zsh      4694   jj   mem    REG     3,2   56444      20598
/usr/lib/zsh/4.2.0/zsh/computil.so
zsh      4694   jj      0u    CHR  136,48          50 /dev/pts/48
zsh      4694   jj      1u    CHR  136,48          50 /dev/pts/48
zsh      4694   jj      2u    CHR  136,48          50 /dev/pts/48
zsh      4694   jj     10u    CHR  136,48          50 /dev/pts/48
```

已使用特殊外圍程序變數 \$\$，其值為外圍程序的程序 ID。

指令 `lsuf` 列出所有目前開啟的檔案，使用時不需任何參數。因為通常其中有數千個檔案開啟，將這些檔案全部列出並不太有用。不過，可將這份包含所有檔案的清單結合搜尋功能，進而產生有用的清單。例如，將用到的字元設備全部列出：

```
$ lsuf | grep CHR
sshd      4685   root   mem    CHR     1,5          45833 /dev/zero
sshd      4685   root   mem    CHR     1,5          45833 /dev/zero
sshd      4693    jj     mem    CHR     1,5          45833 /dev/zero
sshd      4693    jj     mem    CHR     1,5          45833 /dev/zero
zsh       4694    jj      0u    CHR  136,48          50 /dev/pts/48
zsh       4694    jj      1u    CHR  136,48          50 /dev/pts/48
zsh       4694    jj      2u    CHR  136,48          50 /dev/pts/48
zsh       4694    jj     10u    CHR  136,48          50 /dev/pts/48
X         6476   root   mem    CHR     1,1        38042 /dev/mem
lsuf      13478    jj      0u    CHR  136,48          50 /dev/pts/48
lsuf      13478    jj      2u    CHR  136,48          50 /dev/pts/48
grep      13480    jj      1u    CHR  136,48          50 /dev/pts/48
grep      13480    jj      2u    CHR  136,48          50 /dev/pts/48
```

25.2 使用者存取檔案：fuser

這可用來判定哪些程序或使用者目前正存取著特定的檔案。例如，假設您要將裝載在 /mnt 上的檔案系統取消裝載，但 umount 指令傳回「設備忙碌」。接著可使用 fuser 指令來判定哪些程序正存取著此設備：

```
$ fuser -v /mnt/*

          USER          PID ACCESS COMMAND
/mnt/notes.txt
          jj            26597 f....  less
```

當另一個終端機上所執行的 less 程序結束時，即可成功地解除裝載檔案系統。

25.3 檔案內容：stat

指令 stat 會顯示檔案內容：

```
$ stat xml-doc.txt
  File: `xml-doc.txt'
  Size: 632          Blocks: 8          IO Block: 4096   regular file
Device: eh/14d Inode: 5938009      Links: 1
Access: (0644/-rw-r--r--)  Uid: (11994/   jj)   Gid: (   50/   suse)
Access: 2004-04-27 20:08:58.000000000 +0200
Modify: 2003-06-03 15:29:34.000000000 +0200
Change: 2003-07-23 17:48:27.000000000 +0200
```

參數 --filesystem 會產生特定檔案的檔案系統詳細內容：

```
$ stat . --filesystem
  File: "."
    ID: 0          Namelen: 255        Type: ext2/ext3
Blocks: Total: 19347388  Free: 17831731   Available: 16848938  Size: 4096
Inodes: Total: 9830400   Free: 9663967
```

使用 z 外圍程序 (zsh) 時，您必須輸入 /usr/bin/stat，因為 z 外圍程序有一個內建在 stat 中的外圍程序，並且包含不同的選項以及不同的輸出格式：

```
% type stat
stat is a shell builtin
% stat .
device 769
inode 4554808
mode 16877
nlink 12
```

```
uid      11994
gid      50
rdev     0
size     4096
atime    1091536882
mtime    1091535740
ctime    1091535740
blksize  4096
blocks   8
link
```

25.4 USB 設備：lsusb

指令 `lsusb` 會列出所有 USB 設備。加上選項 `-v`，會列出更多細節清單。詳細資訊讀取自目錄 `/proc/bus/usb/`。在 USB 隨身碟連接上之後，下列是 `lsusb` 的輸出。最後一行表示有新的設備存在。

```
Bus 004 Device 001: ID 0000:0000
Bus 003 Device 001: ID 0000:0000
Bus 002 Device 001: ID 0000:0000
Bus 001 Device 001: ID 0000:0000
Bus 001 Device 018: ID 0402:5634 ALi Corp.
```

25.5 SCSI 設備的相關資訊：scsiinfo

指令 `scsiinfo` 會列出 SCSI 設備的資訊。加上選項 `-l`，可列出所有系統已知的 SCSI 設備 (類似資訊經由指令 `lsscsi` 也可取得)。下列為 `scsiinfo -i /dev/sda` 的輸出，提供硬碟相關資訊。選項 `-a` 可提供更多資訊。

```
Inquiry command
-----
Relative Address          0
Wide bus 32               0
Wide bus 16               1
Synchronous neg.         1
Linked Commands           1
Command Queueing          1
SftRe                     0
Device Type               0
Peripheral Qualifier       0
Removable?                0
Device Type Modifier      0
ISO Version               0
ECMA Version              0
ANSI Version              3
```



```

AENC                                0
TrmIOP                              0
Response Data Format                 2
Vendor:                             FUJITSU
Product:                             MAS3367NP
Revision level:                      0104A0K7P43002BE

```

出現一個問題清單，內含兩個硬碟的錯誤區塊表格：第一個是由廠商所提供(廠商表格)，第二個是作業時出現的錯誤區塊清單(成長的表格)。若成長表格中的項目數量持續增加，此時最好更換硬碟。

25.6 處理程序：top

指令 `top` (即「程序表 (table of processes)」的縮寫) 會顯示一個程序清單，每隔兩秒自動更新一次。要終止程式，請按 `q`。在顯示一次程序清單之後，參數 `-n 1` 會終止程式。以下是 `top -n 1` 指令的輸出範例：

```

top - 14:19:53 up 62 days,  3:35, 14 users,  load average: 0.01, 0.02, 0.00
Tasks: 102 total,   7 running,  93 sleeping,   0 stopped,   2 zombie
Cpu(s):  0.3% user,   0.1% system,  0.0% nice,  99.6% idle
Mem:    514736k total,  497232k used,    17504k free,    56024k buffers
Swap:   1794736k total,  104544k used,  1690192k free,   235872k cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	Command
1426	root	15	0	116m	41m	18m	S	1.0	8.2	82:30.34	X
20836	jj	15	0	820	820	612	R	1.0	0.2	0:00.03	top
1	root	15	0	100	96	72	S	0.0	0.0	0:08.43	init
2	root	15	0	0	0	0	S	0.0	0.0	0:04.96	keventd
3	root	34	19	0	0	0	S	0.0	0.0	0:00.99	ksoftirqd_CPU0
4	root	15	0	0	0	0	S	0.0	0.0	0:33.63	kswapd
5	root	15	0	0	0	0	S	0.0	0.0	0:00.71	bdflush
[...]											
1362	root	15	0	488	452	404	S	0.0	0.1	0:00.02	nscd
1363	root	15	0	488	452	404	S	0.0	0.1	0:00.04	nscd
1377	root	17	0	56	4	4	S	0.0	0.0	0:00.00	mingetty
1379	root	18	0	56	4	4	S	0.0	0.0	0:00.01	mingetty
1380	root	18	0	56	4	4	S	0.0	0.0	0:00.01	mingetty

如果您在 `top` 正在執行時按 `F`，會出現一個功能表，您可以利用它來變更輸出格式。

參數 `-U UID` 只監視與特定使用者有關的程序。以使用者的使用者 ID 來取代 `UID`。 `top -U $(id -u username)` 指令會傳回使用者 `UID`，即使用者名稱，並顯示使用者的處理程序。

25.7 程序清單：ps

指令 `ps` 會產生一份程序清單。如果加入參數 `r`，則只會顯示正在使用運算時間的程序。

```
$ ps r
  PID TTY          STAT       TIME COMMAND
22163 pts /7        R          0:01  -zsh
 3396 pts /3        R          0:03  emacs new-makedoc.txt
20027 pts /7        R          0:25  emacs xml /common/utilities.xml
20974 pts /7        R          0:01  emacs jj.xml
27454 pts /7        R          0:00  ps r
```

寫入此參數時不可包含減號。寫入的參數有時會包含減號，有時則不包含。`man` 頁面很可能會嚇壞新的使用者，不過還好有 `ps --help` 指令，它會產生一個簡短的說明頁面來提供協助。

如果要核對執行中的 `emacs` 程序數目，請使用：

```
$ ps x | grep emacs
 1288 ?          S          0:07  emacs
 3396 pts /3      S          0:04  emacs new-makedoc.txt
 3475 ?          S          0:03  emacs .Xresources
20027 pts /7      S          0:40  emacs xml /common/utilities.xml
20974 pts /7      S          0:02  emacs jj.xml

$ pidof emacs
20974 20027 3475 3396 1288
```

參數 `-p` 會利用程序 ID 來選擇處理程序：

```
$ ps www -p $(pidof xterm)
  PID TTY          STAT       TIME COMMAND
 9025 ?          S          0:01  xterm -g 100x45+0+200
 9176 ?          S          0:00  xterm -g 100x45+0+200
29854 ?          S          0:21  xterm -g 100x75+20+0 -fn \
-B&H-LucidaTypewriter-Medium-R-Normal-Sans-12-120-75-75-M-70-iso10646-1
 4378 ?          S          0:01  xterm -bg MistyRose1 -T root -n root -e su -l
25543 ?          S          0:02  xterm -g 100x45+0+200
22161 ?          R          0:14  xterm -g 100x45+0+200
16832 ?          S          0:01  xterm -bg MistyRose1 -T root -n root -e su -l
16912 ?          S          0:00  xterm -g 100x45+0+200
17861 ?          S          0:00  xterm -bg DarkSeaGreen1 -g 120x45+40+300
19930 ?          S          0:13  xterm -bg LightCyan
21686 ?          S          0:04  xterm -g 100x45+0+200 -fn \
lucidasanstypewriter-12
23104 ?          S          0:00  xterm -g 100x45+0+200
26547 ?          S          0:00  xterm -g 100x45+0+200
```

程序清單可依照您的需要進行格式化。選項 `-L` 會傳回所有關鍵字의清單。輸入下列指令來發出一份按記憶體的使用排序的程序清單：

```
$ ps ax --format pid,rss,cmd --sort rss
  PID  RSS  CMD
    2     0 [ ksoftirqd/0]
    3     0 [ events/0]
   17     0 [ kblockd/0]
[... ]
10164 5260 xterm
31110 5300 xterm
17010 5356 xterm
 3896 29292 /usr/X11R6/bin/X -nolisten tcp -br vt7 -auth
/var/lib/xdm/authdir/au
```

25.8 程序樹：pstree

指令 `pstree` 會產生一份樹狀程序清單：

```
$ pstree
init--+-atd
      |-3*[ automount]
      |-bdflush
      |-cron
[... ]
      |-usb-storage-1
      |-usb-storage-2
      |-10*[ xterm---zsh]
      |-xterm---zsh---mutt
      |-2*[ xterm---su---zsh]
      |-xterm---zsh---ssh
      |-xterm---zsh---pstree
      |-ypbind---ypbind---2*[ ypbind]
      `--zsh---startx---xinit4--X
                                `--ctwm--+-xclock
                                        | -xload
                                        `--xosview.bin
```

參數 `-p` 會將程序 ID 新增至指定的名稱。如果要同時顯示指令行，請使用 `-a` 參數：

```
$ pstree -pa
init,1
  |-atd,1255
[... ]
  `--zsh,1404
      `--startx,1407 /usr/X11R6/bin/startx
          `--xinit4,1419 /suse/jj/.xinitrc [... ]
              |-X,1426 :0 -auth /suse/jj/.Xauthority
```

```
`-ctwm,1440
    |-xclock,1449 -d -geometry -0+0 -bg grey
    |-xload,1450 -scale 2
    `--xosview.bin,1451 +net -bat +net
```

25.9 功能介紹：w

使用指令 `w`，可清楚掌握登入系統的使用者和他們的動作。例如：

```
$ w
 15:17:26 up 62 days,  4:33, 14 users,  load average: 0.00, 0.04, 0.01
USER      TTY      LOGIN@   IDLE   JCPU   PCPU WHAT
jj         pts/0    30Mar04  4days  0.50s  0.54s xterm -e su -l
jj         pts/1    23Mar04  5days  0.20s  0.20s -zsh
jj         pts/2    23Mar04  5days  1.28s  1.28s -zsh
jj         pts/3    23Mar04  3:28m   3.21s  0.50s -zsh
[...]
jj         pts/7    07Apr04  0.00s   9.02s  0.01s w
jj         pts/9    25Mar04  3:24m   7.70s  7.38s mutt
[...]
jj         pts/14   12:49    37:34   0.20s  0.13s ssh totan
```

最末一行顯示使用者 `jj` 已建立一個安全的外圍程序 (`ssh`) 連線至電腦 `totan`。如果其它系統的使用者從遠端登入，參數 `-f` 會顯示該使用者用來連線的電腦。

25.10 記憶體的使用：free

公用程式 `free` 會檢查 RAM 的使用狀況。顯示可使用和已使用的記憶體 (及交換區域) 的詳細資訊：

```
$ free
              total        used        free      shared    buffers     cached
Mem:        514736        273964        240772           0        35920        42328
-/+ buffers/cache:        195716        319020
Swap:        1794736        104096        1690640
```

使用 `-m`，會以 MB 為單位來顯示大小：

```
$ free -m
              total        used        free      shared    buffers     cached
Mem:           502          267          235           0           35           41
-/+ buffers/cache:          191          311
Swap:         1752          101         1651
```

下面這一行包含相當有趣的資訊：

```
-/+ buffers/cache:      191      311
```

它可以計算緩衝區和快取所佔用的記憶體空間。參數 `-d` 延遲 會確定每隔 延遲 秒自動更新顯示內容。例如，`free -d 1.5` 每 1.5 秒會更新一次。

25.11 核心環狀緩衝區：dmesg

Linux 核心會將某些訊息保存在環狀緩衝區內。如果要檢視這些訊息，可輸入指令 `dmesg`：

```
$ dmesg
[...]
sdc : READ CAPACITY failed.
sdc : status = 1, message = 00, host = 0, driver = 08
Info fld=0xa00 (nonstd), Current sd00:00: sense key Not Ready
sdc : block size assumed to be 512 bytes, disk size 1GB.
sdc: test WP failed, assume Write Enabled
sdc: I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 2097144
I/O error: dev 08:20, sector 0
I/O error: dev 08:20, sector 0
unable to read partition table
I/O error: dev 08:20, sector 0
nfs: server totan not responding, still trying
nfs: server totan OK
```

最後一行表示 NFS 伺服器 `totan` 中有一個暫時性的問題。在此之前的幾行是由插入 USB 快閃磁碟機所觸發的。較早的事件記錄在檔案 `/var/log/messages` 和 `/var/log/warn` 中。

25.12 檔案系統及其使用：mount、df 和 du

指令 `mount` 會顯示哪個檔案系統 (設備和類型) 已裝載於哪個裝載點：

```
$ mount
/dev/hdb2 on / type ext2 (rw)
proc on /proc type proc (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/hda1 on /data type ext2 (rw)
```

```
shmfs on /dev/shm type shm (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
automount(pid1012) on /suse type autofs \
    (rw,fd=5,pgrp=1012,minproto=2,maxproto=3)
totan:/real-home/jj on /suse/jj type nfs \
    (rw,nosuid,rsize=8192,wsiz=8192,hard,intr,nolock,addr=10.10.0.1)
```

使用指令 `df` 來取得檔案系統總使用量的資訊。參數 `-h` (或 `--human-readable`) 可將輸出轉換為一般使用者容易了解的形式。

```
$ df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hdb2              7.4G  5.1G   2.0G   73% /
/dev/hda1              74G   5.8G   65G    9% /data
shmfs                 252M    0   252M    0% /dev/shm
totan:/real-home/jj   350G  324G   27G   93% /suse/jj
```

NFS 檔案伺服器 `totan` 的使用者應立刻清除他們的主目錄。可使用指令 `du` 來顯示指定目錄及其子目錄下所有檔案的大小。參數 `-s` 可以不顯示輸出的詳細資訊。`-h` 同樣可以將資料轉換為一般人容易了解的形式。使用這個指令：

```
$ du -sh ~
361M    /suse/jj
```

檢視您的主目錄佔用的空間。

25.13 /proc 檔案系統

`/proc` 檔案系統是一個虛擬檔案系統，核心會在此系統內以虛擬檔案的形式來保存重要資訊。例如，使用這個指令可顯示 CPU 類型。

```
$ cat /proc/cpuinfo
processor       : 0
vendor_id     : AuthenticAMD
cpu family    : 6
model         : 8
model name    : AMD Athlon(tm) XP 2400+
stepping      : 1
cpu MHz       : 2009.343
cache size    : 256 KB
fdiv_bug      : no
[...]
```

可用下列指令來查詢配置及中斷的使用：

```
$ cat /proc/interrupts
CPU0
```

```

0: 537544462      XT-PIC timer
1: 820082        XT-PIC keyboard
2: 0             XT-PIC cascade
8: 2             XT-PIC rtc
9: 0             XT-PIC acpi
10: 13970         XT-PIC usb-uhci, usb-uhci
11: 146467509     XT-PIC ehci_hcd, usb-uhci, eth0
12: 8061393       XT-PIC PS/2 Mouse
14: 2465743       XT-PIC ide0
15: 1355          XT-PIC ide1
NMI: 0
LOC: 0
ERR: 0
MIS: 0

```

以下為一些重要檔案及其內容：

/proc/devices

可用設備

/proc/modules

載入的核心模組

/proc/cmdline

核心指令行

/proc/meminfo

有關記憶體使用狀況的詳細資訊

/proc/config.gz

gzip-壓縮核心目前正在執行的組態檔案

如需更多資訊，請參閱 `/usr/src/linux/Documentation/filesystems/proc.txt`。有關執行中處理程序的資訊，請參閱 `/proc/NNN` 目錄，其中 `NNN` 為相關程序的程序 ID (PID)。在 `/proc/self/` 中可以找到每個程序的特性：

```

$ ls -l /proc/self
lrwxrwxrwx 1 root root 64 Apr 29 13:52 /proc/self -> 27585

$ ls -l /proc/self/
total 0
dr-xr-xr-x 2 jj suse 0 Apr 29 13:52 attr
-r----- 1 jj suse 0 Apr 29 13:52 auxv
-r--r--r-- 1 jj suse 0 Apr 29 13:52 cmdline
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 cwd -> /suse/jj/t
-r--r--r-- 1 jj suse 0 Apr 29 13:52 delay
-r----- 1 jj suse 0 Apr 29 13:52 environ

```

```

lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 exe -> /bin/ls
dr-x----- 2 jj suse 0 Apr 29 13:52 fd
-rw----- 1 jj suse 0 Apr 29 13:52 mapped_base
-r--r--r-- 1 jj suse 0 Apr 29 13:52 maps
-rw----- 1 jj suse 0 Apr 29 13:52 mem
-r--r--r-- 1 jj suse 0 Apr 29 13:52 mounts
lrwxrwxrwx 1 jj suse 0 Apr 29 13:52 root -> /
-r--r--r-- 1 jj suse 0 Apr 29 13:52 stat
-r--r--r-- 1 jj suse 0 Apr 29 13:52 statm
-r--r--r-- 1 jj suse 0 Apr 29 13:52 status
dr-xr-xr-x 3 jj suse 0 Apr 29 13:52 task
-r--r--r-- 1 jj suse 0 Apr 29 13:52 wchan

```

maps 檔中含有可執行檔和程式庫的位址指定：

```

$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 22890          /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 22890          /bin/cat
0804d000-0806e000 rwxp 0804d000 00:00 0
40000000-40016000 r-xp 00000000 03:02 10882         /lib/ld-2.3.3.so
40016000-40017000 rw-p 00015000 03:02 10882         /lib/ld-2.3.3.so
40017000-40018000 rw-p 40017000 00:00 0
4002b000-40135000 r-xp 00000000 03:02 10908         /lib/tls/libc.so.6
40135000-4013d000 rw-p 0010a000 03:02 10908         /lib/tls/libc.so.6
4013d000-40141000 rw-p 4013d000 00:00 0
bffffe00-c0000000 rw-p bffffe00 00:00 0
fffffe00-fffff000 ---p 00000000 00:00 0

```

25.14 vmstat、iostat 及 mpstat

公用程式 `vmstat` 會報告虛擬記憶體統計數據。它會讀取檔案 `/proc/meminfo`、`/proc/stat` 以及 `/proc/*/stat`。適合用來找出系統效能瓶頸。指令 `iostat` 可以報告 CPU 的統計數據以及設備與分割區的輸出與輸入。顯示的資訊來自於 `/proc/stat` 檔案與 `/proc/partitions` 檔案。輸出可用來改善硬碟間輸出與輸入的負載平衡。指令 `mpstat` 會報告 CPU 的相關統計數據。

25.15 procinfo

使用指令 `procinfo` 來摘述 `/proc` 檔案系統的重要資訊：

```

$ procinfo
Linux 2.6.4-54.5-default (geeko@buildhost) (gcc 3.3.3 ) #1 1CPU [roth.suse.de]

```



```

Memory:      Total      Used      Free      Shared      Buffers
Mem:         516696     513200      3496         0         43284
Swap:        530136      1352     528784

Bootup: Wed Jul  7 14:29:08 2004      Load average: 0.07 0.04 0.01 1/126 5302

user  :      2:42:28.08    1.3%  page in :      0
nice  :      0:31:57.13    0.2%  page out:      0
system:    0:38:32.23    0.3%  swap in :      0
idle   :    3d 19:26:05.93 97.7%  swap out:      0
uptime:    4d  0:22:25.84      context :207939498

irq 0: 776561217 timer                irq 8:      2 rtc

irq 1:  276048 i8042                  irq 9:    24300 VIA8233

irq 2:      0 cascade [4]            irq 11: 38610118 acpi, eth0, uhci_hcd

irq 3:      3                      irq 12: 3435071 i8042

irq 4:      3                      irq 14: 2236471 ide0

irq 6:      2                      irq 15:  251 idel

```

如果要檢視所有資訊，請使用參數 `-a`。參數 `-nN` 每隔 N 秒即更新資訊。在此例中，請按 `Q` 來終止程式。

依照預設，會顯示累計的值。參數 `-d` 會產生差值。`procinfo -dn5` 會顯示過去 5 秒內變更的值。

```

Memory:      Total      Used      Free      Shared      Buffers      Cached
Mem:         0          2        -2         0          0          0
Swap:         0          0          0

Bootup: Wed Feb 25 09:44:17 2004      Load average: 0.00 0.00 0.00 1/106 31902

user  :      0:00:00.02    0.4%  page in :      0  disk 1:      0r      0w
nice  :      0:00:00.00    0.0%  page out:      0  disk 2:      0r      0w
system:    0:00:00.00    0.0%  swap in :      0  disk 3:      0r      0w
idle   :      0:00:04.99 99.6%  swap out:      0  disk 4:      0r      0w
uptime:    64d  3:59:12.62      context : 1087

irq 0:      501 timer                irq 10:      0 usb-uhci, usb-uhci
irq 1:      1 keyboard              irq 11:    32 ehci_hcd, usb-uhci,
irq 2:      0 cascade [4]            irq 12:    132 PS/2 Mouse
irq 6:      0                      irq 14:      0 ide0
irq 8:      0 rtc                  irq 15:      0 idel
irq 9:      0 acpi

```

25.16 PCI 資源：lspci

指令 `lspci` 會列出 PCI 資源：

```
$ lspci
00:00.0 Host bridge: VIA Technologies, Inc. \
    VT8366/A/7 [Apollo KT266/A/333]
00:01.0 PCI bridge: VIA Technologies, Inc. \
    VT8366/A/7 [Apollo KT266/A/333 AGP]
00:0b.0 Ethernet controller: Digital Equipment Corporation \
    DECchip 21140 [FasterNet] (rev 22)
00:10.0 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.1 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.2 USB Controller: VIA Technologies, Inc. USB (rev 80)
00:10.3 USB Controller: VIA Technologies, Inc. USB 2.0 (rev 82)
00:11.0 ISA bridge: VIA Technologies, Inc. VT8235 ISA Bridge
00:11.1 IDE interface: VIA Technologies, Inc. VT82C586/B/686A/B \
    PIPC Bus Master IDE (rev 06)
00:11.5 Multimedia audio controller: VIA Technologies, Inc. \
    VT8233 AC97 Audio Controller (rev 50)
01:00.0 VGA compatible controller: Matrox Graphics, Inc. \
    MGA G550 AGP (rev 01)
```

使用 `-v` 可產生更詳細的清單：

```
$ lspci -v
[...]
01:00.0 \
    VGA compatible controller: Matrox Graphics, Inc. MGA G550 AGP (rev 01) \
    (prog-if 00 [VGA])
    Subsystem: Matrox Graphics, Inc. Millennium G550 Dual Head DDR 32Mb
    Flags: bus master, medium devsel, latency 32, IRQ 10
    Memory at d8000000 (32-bit, prefetchable) [size=32M]
    Memory at da000000 (32-bit, non-prefetchable) [size=16K]
    Memory at db000000 (32-bit, non-prefetchable) [size=8M]
    Expansion ROM at <unassigned> [disabled] [size=128K]
    Capabilities: <只有 root 可以使用>
```

有關設備名稱解析的資訊，請參閱檔案 `/usr/share/pci.ids`。本檔中未列出的 PCI ID 標記為「不明設備」。

參數 `-vv` 可產生所有可供程式查詢的資訊。如果要檢視純數值，應使用參數 `-n`。

25.17 執行程式的系統呼叫：strace

公用程式 `strace` 讓您可以追蹤執行中處理程序的所有系統呼叫。將 `strace` 新增至指令行的開頭，以正常方式來輸入指令：

```
$ strace -e open ls

execve("/bin/ls", [ "ls"], [ /* 88 vars */] ) = 0
uname({sys="Linux", node="edison", ...}) = 0
brk(0) = 0x805b000
old_mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0x40017000
open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=76333, ...}) = 0
old_mmap(NULL, 76333, PROT_READ, MAP_PRIVATE, 3, 0) = 0x40018000
[...]
ioctl(1, SNDCTL_TMR_TIMEBASE or TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=53, ws_col=110, ws_xpixel=897, ws_ypixel=693}) = 0
open(".", O_RDONLY|O_NONBLOCK|O_LARGEFILE|O_DIRECTORY) = 3
fstat64(3, {st_mode=S_IFDIR|0755, st_size=144, ...}) = 0
fcntl64(3, F_SETFD, FD_CLOEXEC) = 0
getdents64(3, /* 5 entries */, 4096) = 160
getdents64(3, /* 0 entries */, 4096) = 0
close(3) = 0
fstat64(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 48), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) \
    = 0x40018000
write(1, "ltrace-ls.txt myfile.txt strac...", 41) = 41
munmap(0x40018000, 4096) = 0
exit_group(0) = ?
```

例如，要追蹤所有試圖開啟某一特定檔案的動作，可使用下列指令：

```
$ strace -e open ls myfile.txt

open("/etc/ld.so.preload", O_RDONLY) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
open("/lib/tls/librt.so.1", O_RDONLY) = 3
open("/lib/libacl.so.1", O_RDONLY) = 3
open("/lib/libselinux.so.1", O_RDONLY) = 3
open("/lib/tls/libc.so.6", O_RDONLY) = 3
open("/lib/tls/libpthread.so.0", O_RDONLY) = 3
open("/lib/libattr.so.1", O_RDONLY) = 3
open("/proc/mounts", O_RDONLY) = 3
[...]
open("/proc/filesystems", O_RDONLY) = 3
open("/proc/self/attr/current", O_RDONLY) = 4
```

如果要追蹤所有子程序，可使用參數 `-f`。可充分掌控 `strace` 的行為和輸出格式。如需更多資訊，請參閱 `man strace`。

25.18 一個程式執行階段的程式庫呼叫：ltrace

指令 `ltrace` 讓您可以追蹤處理程序的程式庫呼叫。這個指令的用法與 `strace` 相似。參數 `-c` 會輸出程式庫呼叫的數目和持續時間。

```
$ ltrace -c find /usr/share/doc
% time      seconds    usecs/call      calls      errors syscall
-----
 86.27      1.071814          30      35327              write
10.15      0.126092          38       3297      getdents64
 2.33      0.028931           3      10208      lstat64
 0.55      0.006861           2       3122      1 chdir
 0.39      0.004890           3       1567      2 open
[...]
 0.00      0.000003           3           1      uname
 0.00      0.000001           1           1      time
-----
100.00      1.242403                    58269      3 total
```

25.19 指定需要的程式庫：ldd

指令 `ldd` 可用來釐清哪些程式庫將載入引數指定的動態可執行檔：

```
$ ldd /bin/ls
linux-gate.so.1 => (0xffffe000)
librt.so.1 => /lib/tls/librt.so.1 (0x4002b000)
libacl.so.1 => /lib/libacl.so.1 (0x40033000)
libseline.so.1 => /lib/libselinux.so.1 (0x40039000)
libc.so.6 => /lib/tls/libc.so.6 (0x40048000)
libpthread.so.0 => /lib/tls/libpthread.so.0 (0x4015d000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
libattr.so.1 => /lib/libattr.so.1 (0x4016d000)
```

靜態二進位不需要任何動態程式庫：

```
$ ldd /bin/sash
not a dynamic executable
$ file /bin/sash
/bin/sash: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), \
for GNU/Linux 2.2.5, statically linked, stripped
```

25.20 其它有關 ELF 二進位的資訊

可使用 `readelf` 公用程式來讀取二進位的內容。甚至可將它用在為其它硬體架構而建立的 ELF 檔案：

```
$ readelf --file-header /bin/ls
ELF Header:
  Magic:      7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00
  Class:      ELF32
  Data:      2's complement, little endian
  Version:    1 (current)
  OS/ABI:     UNIX - System V
  ABI Version: 0
  Type:      EXEC (Executable file)
  Machine:    Intel 80386
  Version:    0x1
  Entry point address: 0x8049b40
  Start of program headers: 52 (bytes into file)
  Start of section headers: 76192 (bytes into file)
  Flags:      0x0
  Size of this header: 52 (bytes)
  Size of program headers: 32 (bytes)
  Number of program headers: 9
  Size of section headers: 40 (bytes)
  Number of section headers: 29
  Section header string table index: 26
```

25.21 程序間的通訊：ipcs

指令 `ipcs` 可產生一份使用中的 IPC 資源清單：

```
$ ipcs
----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x000027d9   5734403   toms       660        64528      2
0x00000000   5767172   toms       666        37044      2
0x00000000   5799941   toms       666        37044      2

----- Semaphore Arrays -----
key          semid      owner      perms      nsems
0x000027d9   0          toms       660        1

----- Message Queues -----
key          msqid      owner      perms      used-bytes  messages
```

25.22 使用 **time** 進行時間管理

可利用 `time` 公用程式來決定指令所花費的時間。這個公用程式有兩種版本：做為外圍程序內建使用和做為程式 (`/usr/bin/time`) 使用。

```
$ time find . > /dev/null
```

```
real    0m4.051s
user    0m0.042s
sys     0m0.205s
```

部 VIII. 系統

64 位元系統環境的 32 位元和 64 位元應用程式 26

SUSE Linux 可用於多種 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 支援在 64 位元系統環境中使用 32 位元應用程式。本章簡略說明這項支援在 64 位元 SUSE Linux 平台的執行方式。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 以及 32 位元應用程式如何在 64 位元核心下執行的相關資訊。

SUSE Linux 針對 64 位元平台設計的 AMD64 和 EM64T，讓現有的 32 位元應用程式可以在此 64 位元環境中執行，且「表現出色」。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。

26.1 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。

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。必須有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`、`/usr/lib` 和 `/usr/X11R6/lib` 之下找到的 64 位元物件檔案，現在放在 `/lib64`、`/usr/lib64` 以及 `/usr/X11R6/lib64` 底下。這表示在 `/lib`、`/usr/lib` 和 `/usr/X11R6/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。

物件目錄中的子目錄，如果其資料內容不取決於文字大小，也不會移動。例如，X11 字型仍然可以在 `/usr/X11R6/lib/X11/fonts` 底下的一般位置找到。此配置與 LSB (Linux Standards Base) 以及 FHS (File System Hierarchy Standard) 相符。

26.2 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題檔案相符的 API (應用程式設計介面)。標準 SUSE 環境是根據此原則所設計。對於手動更新的程式庫，請自行解決這些問題。

26.3 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` 套件的個別標題和程式庫，以及 `rpmname-devel-32bit` 第二個架構的開發程式庫。

大部份的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，方法是執行帶有其他環境變數的 `configure` 程序檔。

以下範例，將 x86 的 AMD64 或 EM64T 系統指示為第二個架構：

1. 將 `autoconf` 設定成使用 32 位元編譯器：

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

2. 指示連結器處理 32 位元物件：

```
LD="ld -m elf64_i386"
```

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：

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

5. 判斷程式庫是否儲存在 lib 子目錄中：

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

6. 判斷是否使用 32 位元 X 程式庫：

```
--x-libraries=/usr/X11R6/lib/
```

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 -m64" \
LD_FLAGS="-L/usr/lib64;" \
    .configure \
        --prefix=/usr \
        --libdir=/usr/lib64
make
make install
```

26.4 核心規格

AMD64 和 EM64T 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 位元應用程式可以和 64 位元核心溝通，方法與和 32 位元核心溝通相同。

64 位元核心中，系統呼叫的 32 位元模擬，不支援系統程式使用的某些 API。這要視平台而定。因此，有少數應用程式，像 `lspci` 或 `LVM` 管理程式，必須如同 64 位元程式編譯，才能正常運作。

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 **SUSE**，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。

使用外圍程序

圖形化使用者介面對 Linux 來說變得日益重要，但是使用滑鼠並不一定是進行日常工作的最好方法。指令行提供了高度的彈性及效率。當您控制慢速網路上的電腦，或是要在 `xterm` 的指令行以根的身分執行工作時，以文字為基礎的應用程式更為重要。`Bash` 外圍程序是 SUSE Linux 的預設指令行解譯器。

Linux 是一種多重使用者系統，可存取由使用者權限所控制的檔案。無論是使用指令行或 GUI，它都有助於瞭解權限概念。使用指令行時，有數種重要的指令。從指令行設定系統時，通常會使用 `vi` 文字編輯器。許多系統管理員和開發人員都常使用它。

27.1 使用指令行上的 `Bash`

在 KDE 工作列中，有一個繪製為貝殼的顯示器之圖示。當您按下這個圖示，就會開啟可在其中輸入指令的終端機視窗。主控台就是一般執行 `Bash` 的終端機程式 (Bourne again shell)，這是 GNU 專案所開發程式的一部分。請在 GNOME 桌面上，按一下上方面板顯示電腦螢幕的圖示，啟動一般執行 `Bash` 的終端機。

一旦您開啟該外圍程序，會在第一列看到提示符號。提示名稱通常由使用者名稱、主機名稱和現行路徑組成，也可以自行設定。當游標出現在這個提示名稱之後，您可以將指令直接傳送給電腦系統。

27.1.1 輸入指令

指令是由幾個元件所組成。第一個元件一定是實際的指令，後面跟著參數或選項。指令會在您按 **Enter** 時執行。在這麼做之前，您可以輕易地編輯指令行、新增選項或修正打錯的字。有一個最常用的指令是 `ls`，可以單獨或和引數一起使用。只輸入 `ls` 指令可以顯示目前目錄的內容。

選項會以連字號開始。例如，指令 `ls -l` 可以顯示同一個目錄內容的全部細節 (長式清單)。在每一個檔案名稱旁可以看到檔案建立日期、檔案大小 (以位元組為單位)，以及進一步的細節，這些將在稍後說明。許多指令都有一個重要的選項 `--help`。輸入 `ls --help`，可顯示有關 `ls` 指令的所有選項。

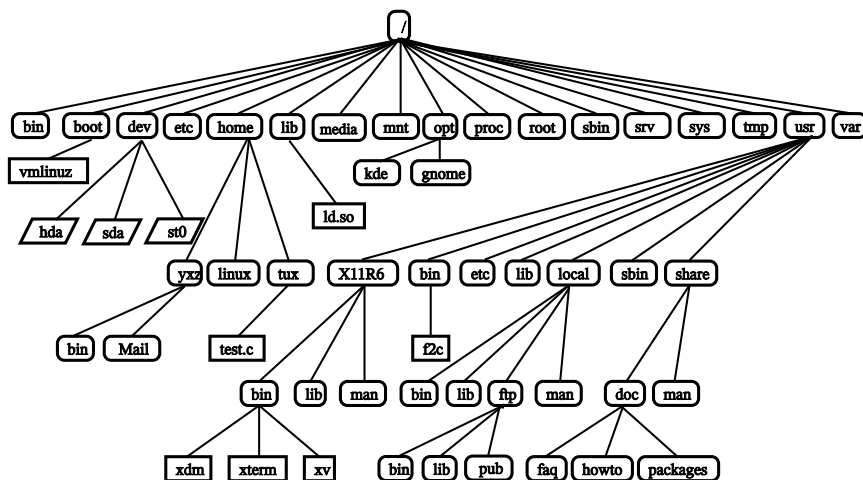
您必須取得"引號"使用權限。如果檔名中包含空格，請使用反斜線 (\) 來避免發生空格，或是將檔名包括在單引號或是雙引號中。否則，**Bash** 會將類似 `My Documents` 的檔案名稱，解譯成為兩個檔案或是兩個目錄。單引號和雙引號的差別，在於使用雙引號時會產生不同的展開方式。使用單引號，可以確保外圍程序只會解譯單引號所括住的字串。

27.1.2 檔案與目錄

若要有效使用外圍程序，瞭解一些有關 Linux 系統的檔案與目錄結構知識非常有用。您可以將目錄視為儲存檔案、程式及子目錄的電子資料夾。階層中最上層的目錄就是根目錄，以 `/` 表示。從此處可以存取其他所有目錄的位置。

`/home` 目錄包含每個使用者可以儲存個人檔案的目錄。[圖形 27.1, "摘錄自標準目錄網路樹" \[355\]](#) 顯示 Linux 中的標準目錄樹，以及範例使用者 `xyz`、`linux` 和 `tux` 的主目錄。Linux 系統的目錄網路樹有一個置於 *Filesystem Hierarchy Standard* (FHS) 後的功能性結構。下列清單提供 Linux 中標準目錄的簡短描述。

圖形 27.1 摘錄自標準目錄網路樹



/
根目錄，目錄網路樹的起點

/home
私人使用者目錄

/dev
代表硬體元件的裝置檔案

/etc
系統組態的重要檔案

/etc/init.d
啟動程序檔

/usr/bin
一般可存取的程式

/bin
啟動過程早期需要的程式

/usr/sbin
保留給系統管理員的程式

/sbin

保留給系統管理員而且啟動時需要的程式

/usr/include

C 編譯器標題檔

/usr/include/g++

C++ 編譯器標題檔

/usr/share/doc

各種文件檔

/usr/share/man

系統手冊頁面 (man 頁面)

/usr/src

系統軟體的原始程式碼

/usr/src/linux

核心原始程式碼

/tmp、/var/tmp

暫存檔

/usr

所有應用程式

/var

組態檔 (例如，連結自 /usr 的檔案)

/var/log

系統日誌檔

/var/adm

系統管理資料

/lib

共用的程式庫 (用於動態連結的程式)

/proc

程序檔案系統

/sys

system 檔案系統，為核心之所有設備資訊的集中存放位置

/usr/local

本機，分散獨立延伸

/opt

選擇性軟體，較大的附加套裝軟體 (例如 KDE、GNOME、Netscape)

27.1.3 Bash 功能

外圍程序提供兩個可以讓工作更輕鬆的重要功能：

歷程記錄

若要重複之前輸入的指令，請按 **↑**，直到該指令顯示在指令提示字元。按 **↓** 往前翻閱以前輸入的指令清單。要編輯指令行，只要使用方向鍵將游標移至想要的位置，然後開始輸入。使用 **Ctrl + R** 可在歷程中搜尋。

補齊功能

在輸入開頭的字母時，若足以識別某個特定檔案的名稱，便補齊為該檔案的完整檔案名稱。若要這樣做，請輸入第一個字母，然後按 **Tab** 鍵。如果有多個檔案名稱以相同的字母開頭，可按兩次 **Tab** 來取得清單。

第一個範例：管理檔案

現在您已經知道指令的樣子、SUSE Linux 有哪些目錄以及如何提高使用 Bash 的速度，請在小小練習中應用這些知識。

1. 按一下外圍程序圖示，從 KDE 或 GNOME 桌面開啟主控台。
2. 輸入 `ls` 指令，查看主目錄的內容。
3. 使用指令 `mkdir` (表示建立目錄 (*make directory*))，輸入 `mkdir test` 建立名為 `test` 的新子目錄。

4. 現在，按 `[Alt] + [F2]`，為 KDE 的 Kate 輸入 `kate` 或為 GNOME 的 Gedit 輸入 `gedit`，來啟動編輯器。在編輯器輸入幾個字母，然後將檔案儲存成 `Testfile` 放到您的主目錄。Linux 會區分大小寫。以此範例為例，是使用大寫的 `T`。
5. 再次檢視主目錄的內容。不必再次輸入 `ls`，只要按 `[↑]` 兩次，`ls` 指令就會重新顯示在指令提示。要執行指令，請按 `[Enter]`。新建的目錄 `test` 應該會顯示成藍色字母，而 `Testfile` 應該是黑色的。這是目錄和檔案在主控台區分的方式。
6. 使用 `mv` 指令，將 `Testfile` 移至子目錄 `test`。要加速這個動作，可使用展開功能：只要輸入 `mv T`，然後按 `[Tab]`。只要目錄中沒有以這個字母開頭的檔案，外圍程序就會展開檔案名稱並加上字串 *estfile*。否則，新增一到兩個字母，然後測試 `[Tab]`，看看外圍程序現在是否展開名稱。最後，請在展開的檔案名稱後方鍵入一個空格和 `test`，接著按 `[Enter]` 來執行指令。
7. 此時，`Testfile` 應該會從目錄中消失。再次輸入 `ls`，再檢查一次。
8. 要知道檔案是否移動成功，請使用 `cd test` 指令，移入至 `test` 目錄。現在再輸入 `ls`。您應該會在清單中看到 `Testfile`。只輸入 `cd`，可以從任何地方回到主目錄。
9. 要建立檔案複本，請使用 `cp`。例如，輸入 `cp Testfile Testbackup`，會將 `Testfile` 複製至 `Testbackup`。再次使用指令 `ls` 可用來查看二者檔案是否在目錄中。

27.1.4 指定路徑

使用檔案或目錄時，指定正確路徑十分重要。不過，您不需要從相關檔案的根目錄，輸入整個(絕對)路徑。您可以從目前的目錄開始。請使用 `~` 來直接設定主目錄位置。這表示有兩種方式可列出 `test` 目錄中的 `Testfile` 檔案：使用 `ls test` 輸入相對路徑或使用 `ls ~/test` 指定絕對路徑。

要列示其他使用者的主目錄內容，請輸入 `ls ~username`。在此網路目錄樹中，其中一個範例使用者是 `tux`。在本案例中，`ls ~tux` 會列示 `tux` 的主目錄內容。

使用一個點 (.) 可表示目前的目錄。網路樹中的上一層目錄是由兩個點 (..) 表示。輸入 `ls ..`，查看目前目錄的父目錄。指令 `ls .. /..` 會顯示階層中上兩層目錄中的內容。

第二個範例:使用路徑

這裡有另一個範例，示範如何在 SUSE Linux 系統的目錄間移動。

1. 使用指令 `cd`，變更至您的主目錄。然後輸入 `mkdir test2`，在其中建立一個名為 `test2` 的目錄。
2. 使用 `cd test2` 移至新目錄，然後在裏面建立名為 `subdirectory` 的子目錄。使用擴充功能移至該目錄：輸入 `cd su`，然後按 `[Tab]`。外圍程序將展開目錄名稱的其餘部份。
3. 現在試著將前面建立的檔案 `Testbackup` 移至目前的目錄 (`subdirectory`) 無需再次變更目錄。要達成此目的，可指定檔案的相對路徑：`mv ../ ../ /test/Testbackup .` (請注意出現在最後的點)。此指令結尾處的點是必要的，可以告訴外圍程序目前的目錄是檔案移動的目的地。此範例中的 `../ ../ /` 會參照您的主目錄。

27.1.5 萬用字元

外圍程序提供的另一個便利之處，就是可以在路徑展開時使用萬用字元。Bash 共用三種不同的萬用字元類型：

?

完全相符的任何字元

符合任何數目的字元

[set]

符合方括號中指定群組中的某個字元，在此是由字串 *set* 表示。在使用 *set* 時，您也可以使用語法 `[class:]` 來指定字元類別，其中可指定 `alnum`、`alpha` 與 `ascii` 等類別。

在群組 (*[!set]*) 開頭使用 `!` 或 `^`，可以比對不同於 *set* 所指定的字元。

假設您的 `test` 目錄包含 `Testfile`、`Testfile1`、`Testfile2` 與 `datafile` 等檔案，指令 `ls Testfile?` 會列示 `Testfile1` 和 `Testfile2` 這兩個檔案。使用 `ls Test*` 會在清單中加入 `Testfile`。 `ls *fil*` 顯示所有範例檔案。最後，您可以使用 `set` 萬用字元，找出最後一個字元是數字的所有範例檔案：`ls Testfile[1-9]`、使用類別，以及 `ls Testfile[[:digit:]]`。

在三種萬用字元中，應用最廣泛的是星號。它可以用來將某一目錄中的所有檔案，複製到另一個目錄，或者使用一個指令來刪除所有檔案。例如指令 `rm *fil*`，會刪除目前目錄中，名稱中包含字串 `fil` 的所有檔案。

27.1.6 Less 和 More

Linux 包括兩個小程式，可以直接在外圍程序檢視文字檔案。不必啟動編輯器就可以讀取檔案，例如 `Readme.txt`，只要輸入 `less Readme.txt` 便可在主控台視窗中顯示文字。使用 `[空格]` 鍵往下捲動一頁。使用 `[Page Up]` 和 `[Page Down]` 鍵，往前或往回捲動文件。若要結束 `less`，請按 `[Q]`。

除了 `less` 外，您也可以使用較舊的 `more` 程式。不過，因為它不能讓您往回捲動，所以不是很方便。

程式 `less` 的名稱來源是得自 *less is more* (少即是多) 的俗語，也可以方便用來檢視指令輸出。要知道如何使用，請參閱 [節 27.1.7, "管道和重新導向" \[360\]](#)。

27.1.7 管道和重新導向

正常情況下，外圍程序的標準輸出是您的螢幕或主控台視窗，而標準輸入是鍵盤。若要將指令的輸出轉送到應用程式，例如 `less`，請使用管道。

要檢視 `test` 目錄的檔案，可輸入指令 `ls test | less`。`test` 目錄的內容即改成使用 `less` 顯示。如果使用 `ls` 的標準輸出太長的時候，才有意義。例如，如果您檢視 `dev` 目錄內容時使用 `ls /dev`，視窗中只會看到一小部份。此時，請使用 `ls /dev | less` 檢視完整清單。

也可以將指令的輸出儲存至檔案。例如 `echo "test one" > Content` 就會產生名為 `Content` 的新檔案，而且其中會包含 `test one` 等字。可使用 `less Content` 檢視該檔案

您也可以使用檔案來輸入指令。例如，`tr` 會從檔案 `Content` 重新導向的標準輸入替代字元，並且將結果寫入標準輸出：呼叫 `tr t x < Content`，將 `t` 替代成為 `x`。`tr` 的輸出會傳送至螢幕。

如果需要內含該輸出的新檔案，請將 `tr` 的輸出重新導向至檔案。如果要測試這個動作，請變更到 `test`，並輸入指令 `tr t x < .. /Content > new`。最後，使用 `less new` 來檢視 `new`。

和標準輸出一樣，標準錯誤輸出也會傳送至主控台。不過，如果要將標準錯誤輸出重新導向至名為 `errors` 的檔案，請在相對應的指令附加 `> errors`。如果您附加 `>& alloutput`，標準輸出和標準錯誤會儲存至名為 `alloutput` 的檔案。最後，將指令的輸出附加至現有的檔案，指令後面應該是 `>> instead of >`。

27.1.8 歸檔和資料壓縮

現在您已經建立一些檔案和目錄，請考慮歸檔和資料壓縮的用途。假設您將整個 `test` 目錄包裝成一個檔案，您可以儲存至 USB 隨身碟當成備份或透過電子郵件傳送。要這樣做，請使用指令 `tar` (用於磁帶歸檔設備)。使用 `tar --help`，檢視 `tar` 指令的所有選項。在此說明最重要的選項：

-c

(用於建立) 建立新歸檔。

-t

(用於表格) 顯示歸檔的內容。

-x

(用於擷取) 解開歸檔。

-v

(用於詳細) 建立歸檔時，在螢幕顯示所有檔案。

-f

(用於檔案) 選擇要歸檔的檔案名稱。建立歸檔時，此選項必須永遠在最後面。

要將 `test` 目錄和所有檔案與子目錄包裝至一個名為 `testarchive.tar` 的歸檔，請使用 `-c` 和 `-f` 選項。為了測試用途，也請新增 `-v` 來追蹤歸檔的進度，

但這不是必要的選項。使用 `cd` 變更至 `test` 目錄所在的主目錄後，輸入 `tar -cvf testarchive.tar test`。接下來，使用 `tar -tf testarchive.tar` 檢視封存檔案的內容。`test` 目錄及其所有檔案與目錄會在磁碟中保留不變。要解開歸檔，請輸入 `tar -xvf testarchive.tar`，但此時先不要嘗試此選項。

如果要進行檔案壓縮，較常使用的選擇是使用 `gzip`，或者使用 `bzip2` 來取得更好的壓縮比例。只要輸入 `gzip testarchive.tar` (或 `bzip2 testarchive.tar`，但是這個範例是使用 `gzip`)。使用 `ls`，現在查看檔案 `testarchive.tar.gz` 是否已經不在該處，而且已經建立 `testarchive.tar.gz` 檔案。此檔案比較小，因此更適合透過電子郵件傳送或儲存在 USB 隨身碟。

現在，在先前建立的 `test2` 目錄解開此檔案。要這樣做，請輸入 `cp testarchive.tar.gz test2`，將檔案複製至該目錄。使用 `cd test2` 變更至目錄。副檔名為 `.tar.gz` 的壓縮歸檔，可以使用 `gunzip` 指令解壓縮。輸入 `gunzip testarchive.tar.gz`、就會產生檔案 `testarchive.tar`，然後需要使用 `tar -xvf testarchive.tar` 來解開或展開。您也可以透過 `tar -xvf testarchive.tar.gz` (不用再加入 `-z` 選項)，用一個步驟就解壓縮和擷取壓縮歸檔。使用 `ls`，您可以看到新的 `test` 目錄已經建立，而且與主目錄的 `test` 目錄，內容相同。

27.1.9 mtools

`mtools` 是一組指令，就像在使用 MS-DOS 檔案系統一樣。包括在 `mtools` 的指令可以讓您將第一個軟碟機設定成 `a:`，和在 MS-DOS 一樣，而且指令和 MS-DOS 指令類似，除了前面加上一個 `m` 不同：

`mdir a:`

顯示磁碟機 `a:` 的磁片內容

`mcopy Testfile a:`

將檔案 `Testfile` 複製到磁片

`mdel a:Testfile`

刪除 `a:` 的 `Testfile`

mformat a:

以 MS-DOS 格式，製作磁片格式 (與使用 `fdformat` 指令的結果相同)

mcd a:

讓 a: 做為您的目前目錄

mmd a:test

在磁片建立子目錄 test

mrdd a:test

刪除磁片的子目錄 test

27.1.10 清除

在此課程之後，您應該熟悉 Linux 外圍程序或指令行的基礎。您可以使用 `rm` 和 `rmdir` 指令，刪除不同的測試檔案與目錄，清除您的主目錄。請在 [節 27.3, "重要的 Linux 指令" \[368\]](#) 中，找出最重要指令的清單及其功能的簡要說明。

27.2 使用者和存取許可權

因為它早在 1990 年代便開始，Linux 已經開發成多重使用者系統。任何數目的使用者可以同時在上面工作。使用者需要登入系統，才能在他們的工作站開始工作階段。每一個使用者都有一個使用者名稱以及相應的密碼。這種使用者區隔，可以確保未授權的使用者，無法看到他們沒有存取許可權的檔案。對系統的較大變更，例如安裝新程式，一般使用者通常也無法執行或被限制。只有 `root` 使用者或超級使用者，可以不受限制對系統進行變更以及自由存取所有檔案。在需要時才以完整 `root` 存取權登入，只要是遵循此概念的人就可以免除意外損失資料的風險。因為在正常情況下，只有 `root` 才可以刪除檔案或格式化硬碟，因此特洛伊木馬效應或意外輸入破壞性指令的情形，都可以大大降低。

27.2.1 檔案系統許可權

基本上，Linux 檔案系統的每一個檔案都屬於一個使用者和一個群組。可以授權這些私有群組和所有其他人寫入、讀取或執行這些檔案。

在本案例中，群組可以定義成一組連線的使用者，具備特定的結合權限。例如，定義可以在特定專案 `project3` 工作的群組。Linux 系統的每一位使用者，至少屬於一個私有群組的成員，一般是 `users`。需要時可以在系統新增很多群組，但只有 `root` 才可以新增群組。每一個使用者都可以使用指令 `groups`，知道成員的所屬群組。

檔案存取

檔案系統中的許可權組織，會因檔案和目錄而不同。檔案許可權資訊可以使用指令 `ls -l` 顯示。輸出如 [範例 27.1, "顯示檔案許可權的範例輸出" \[364\]](#) 所示。

範例 27.1 顯示檔案許可權的範例輸出

```
-rw-r----- 1 tux project3 14197 Jun 21 15:03 Roadmap
```

如第三欄顯示，此檔案屬於使用者 `tux`。它被指定至群組 `project3`。要探索 `Roadmap` 檔案的使用者許可權，必須更詳細檢查第一欄。

-	rw-	r--	---
類型	使用者權限	群組權限	其他使用者的權限

此欄是由一個前導字元，後面加上 9 個字元 (3 個為一組) 所組成的。前 10 個字母標示檔案系統元件的類型。連字號 (-) 表示它是一個檔案。也可以表示目錄 (d)、連結 (l)、區塊裝置 (b) 或字元裝置。

後面的三個區塊依循標準形態。前三個字元指出檔案是否可以讀取 (r) 或無法讀取 (-)。中間部份的 w 代表可以編輯相應的物件，而連字號 (-) 表示無法寫入檔案。第三個位置的 x，代表物件可以執行。因為此範例中的檔案是文字檔，而且不是可以執行的檔案，所以不需要此特殊檔案的執行權限。

在此範例中，`tux` 是檔案 `Roadmap` 的擁有者，擁有讀取 (r) 和寫入存取權 (w)，但無法執行它 (x)。群組 `project3` 的成員可以讀取檔案，不過無法修改或執行。其他使用者沒有此檔案的任何存取權。其他許可權可以透過 ACL (存取控制清單) 來指定。請參閱 [節 27.2.6, "存取控制清單" \[367\]](#) 以取得背景資訊。

目錄許可權

目錄的存取許可權類型為 `d`。針對目錄，個別許可權在意義上會稍稍不同。

範例 27.2 顯示目錄許可權的範例輸出

```
drwxrwxr-x 1 tux project3 35 Jun 21 15:15 ProjectData
```

在 [範例 27.2, "顯示目錄許可權的範例輸出" \[365\]](#) 中，可以輕易辨別目錄 `ProjectData` 的擁有者 (`tux`) 和所屬群組 (`project3`)。對照 [檔案存取 \[364\]](#) 的存取許可權，設定讀取許可權 (`r`) 表示可以顯示目錄的內容。寫入許可權 (`w`) 表示可以建立新檔案。執行許可權 (`x`) 表示使用者可以變更至此目錄。在上面的範例中，這表示使用者 `tux` 以及群組 `project3` 的成員可以變更至 `ProjectData` 目錄 (`x`)、檢視內容 (`r`) 以及新增或刪除檔案 (`w`)。另一方面，其他的使用者被授與較少的存取權。他們可以進入目錄 (`x`) 以及瀏覽該目錄 (`r`)，但無法插入任何新檔案 (`w`)。

27.2.2 修改檔案許可權

變更存取許可權

擁有者和 `root` 可以使用指令 `chmod` 加上變更許可權的參數和一或多個檔案名稱，變更檔案或目錄的存取許可權。形成不同類別的參數：

1. 相關使用者
 - `u` (使用者) -- 檔案的擁有者
 - `g` (群組) -- 擁有檔案的群組
 - `o` (其他) -- 其他使用者 (若未指定參數，變更會套用至所有類別)
2. 用於刪除 (`-`)、設定 (`=`) 或插入 (`+`) 的字元
3. 縮寫
 - `r` — 讀取
 - `w` -- 寫入
 - `x` -- 執行

4. 檔案名稱，若有多個檔案名稱則以空格來分隔

例如，如果 [範例 27.2, "顯示目錄許可權的範例輸出" \[365\]](#) 中的使用者 tux 也想授與目錄 ProjectData 的寫入 (w) 存取權給其他人，他可以使用指令 `chmod o+w ProjectData` 來達到目的。

不過，如果他想拒絕自己以外，所有使用者的寫入許可權，他可以輸入指令 `chmod go-w ProjectData`，便可達到目的。要禁止所有使用者新增檔案至 ProjectData 資料夾，可輸入 `chmod -w ProjectData`。現在，即使是擁有者也無法寫入檔案；擁有者必須先重建寫入許可權，才可以寫入檔案。

變更擁有權許可權

其他控制檔案系統元件擁有權與許可權的重要指令是 `chown` (變更擁有者) 和 `chgrp` (變更群組)。指令 `chown` 可將檔案擁有權移交給其他使用者。不過，只允許 root 執行此變更。

假設 [範例 27.2, "顯示目錄許可權的範例輸出" \[365\]](#) 的檔案 Roadmap 不應該再屬於 tux，而是屬於使用者 geeko，則 root 應該輸入 `chown geeko Roadmap`。

`chgrp` 會變更檔案的群組擁有權。不過，檔案的擁有者必須是新群組的成員。使用此方法，只要 [範例 27.1, "顯示檔案許可權的範例輸出" \[364\]](#) 的使用者 tux 是新群組的成員，便可以使用指令 `chgrp project4 ProjectData`，將所擁有檔案 ProjectData 的群組切換成 project4。

27.2.3 setuid 位元

在某些狀況下，存取許可權可能受限太多。因此，Linux 擁有其他的設定，可以讓目前使用者與群組可以識別特定動作的暫時變更。例如，`passwd` 程式通常需要開機許可權來存取 `/etc/passwd`。此檔案包含一些重要的資訊，例如，使用者的主目錄，以及使用者與群組的 ID。因此，一般使用者將無法變更 `passwd`，因為授與所有使用者直接存取此檔案是非常危險的。此問題的可行解決方案是 `setuid` 機制。`setuid` (設定使用者 ID) 是一種特殊檔案屬性，可以指示系統根據特定使用者 ID 的標示來執行程式。請考慮使用 `passwd` 指令：

```
-rwsr-xr-x  1 root shadow 80036 2004-10-02 11:08 /usr/bin/passwd
```

您會看到 `s`，表示已針對使用者許可權設定 `setuid` 位元。藉由 `setuid` 位元，所有啟動 `passwd` 指令的使用者，皆能以 `root` 來執行。

27.2.4 `setgid` 位元

`setuid` 位元適用於使用者。不過，群組也有對等的屬性：`setgid` 位元。設定此位元的程式會以儲存時指定的群組 ID 來執行，無論啟動它的使用者為何。因此，在附帶 `setgid` 位元的目錄中，所有新建的檔案和子目錄會指定至目錄所屬的群組。請思考以下範例目錄：

```
drwxrws--- 2 tux archive 48 Nov 19 17:12
  backup
```

您會看到 `s`，表示已針對群組許可權設定 `setgid` 位元。目錄的擁有者與 `archive` 群組的成員可存取此目錄。非此群組成員的使用者會「對應」到個別群組。所有已寫入檔案的有效群組 ID 將會是 `archive`。例如，以 `archive` 群組 ID 執行的備份程式，即使沒有 `root` 權限，仍然能夠存取此目錄。

27.2.5 黏貼位元

另外還有黏貼位元。它屬於可執行程式或目錄結果是不一樣的。如果它屬於程式，用此方法標示的檔案會載入 `RAM`，每次使用時就不需要從硬碟取得該程式。此屬性很少用，因為現在硬碟速度夠快了。如果此位元指定至目錄，它會禁止使用者刪除其他人的檔案。典型範例包括 `/tmp` 和 `/var/tmp` 目錄：

```
drwxrwxrwt 2 root root 1160 2002-11-19 17:15 /tmp
```

27.2.6 存取控制清單

除 Linux 檔案系統物件外，傳統許可權概念(例如檔案或目錄)可以透過 `ACL`(存取控制清單)來擴充。它們允許將檔案系統物件的許可權，指定給原始擁有者或所屬群組以外的使用者或群組。

使用簡單的 `ls -l` 指令，可以偵測延伸存取許可權的檔案或目錄：

```
-rw-r--r--+ 1 tux project3 14197 Jun 21 15:03 Roadmap
```

Roadmap 的擁有者是 tux，而後者屬於 project3 群組。tux 擁有此檔案的寫入和讀取存取權。群組和其他使用者都有讀取存取權。區分此檔案與沒有 ACL 的檔案，唯一的不同是擁有許可權位元的欄中有多出來的 +。

執行 `getfacl Roadmap` 可取得 ACL 的詳細資訊：

```
# file: Roadmap
# owner: tux
# group: project3
user::rw-
user:jane:rw-          effective: r--
group::r--
group:djungle:rw-      effective: r--
mask::r--
other::---
```

輸出的前三行資訊，使用 `ls -l` 便能取得。這幾行僅敘述檔案名稱、擁有者和所屬群組。4 到 9 行記錄 ACL 項目。傳統存取許可權代表使用 ACL 時，可能的子集合。此 ACL 範例中，授與讀取和寫入存取權給檔案擁有者和使用者 jane (第 4 和 5 行)。傳統概念已經擴充，允許額外的使用者存取。群組存取權也適用相同的處理方式。所屬群組保存讀取許可權 (第 6 行) 和群組 djungle 具有讀取和寫入許可權。第 8 行的 mask 項目會減低使用者 jane 和群組 djungle 的有效許可權，只剩下讀取權限。其他使用者和群組沒有檔案的任何存取權 (第 9 行)。

在此僅提供最基本的資訊。在 [章 24, Linux 存取控制清單 \[319\]](#) 可以找到更多 ACL 的相關詳細資訊。

27.3 重要的 Linux 指令

此段落深入說明 SUSE Linux 系統的最重要指令。本章節列示很多指令。除了列示個別指令、參數之外，會在適當之處，介紹典型的範例應用程式。若要深入瞭解各種指令，請使用 `man` 後加上指令名稱來取得並使用手冊頁，例如，`man ls`。

在 `man` 頁面中，使用 `[PgUp]` 和 `[PgDn]`，上下移動。使用 `[Home]` 和 `[End]`，在文件的開頭和結尾之間移動。按 `[Q]` 結束此檢視模式。使用 `man man`，可詳細瞭解 `man` 指令本身的資訊。

在以下的介紹中，個別指令元件會以不同的字體表示。實際的指令以及它的必要選項，永遠會列印成 `command option`。不要求一定要放在 [方括號] 中的規格或參數。

視您的需要調整設定。如果沒有名稱為 `file` 的檔案存在，就不需要寫入 `ls file`。您通常可以結合數個參數，例如使用 `ls -la` 而不是 `ls -l -a`。

27.3.1 檔案指令

下節列出最重要的檔案管理指令。其中涵蓋一般檔案管理以及檔案系統 ACL 的操作。

檔案管理

ls [選項] [檔案]

如果您執行 `ls` 而未加上任何其他參數，程式會以簡要格式，列示所在目錄的內容。

-l
詳細清單

-a
顯示隱藏的檔案

cp [選項] 來源 目標

複製來源到目標。

-i
在覆寫現有目標 之前，等候確認 (若有需要)

-r
循環複製 (包括子目錄)

mv [選項] 來源 目標

複製來源到目標，然後刪除原始來源。

-b

移動之前，先建立來源的備份副本

-i

在覆寫現有的 `targetfile` 之前，等候確認 (若有需要)

rm [選項] 檔案

從檔案系統移除指定的檔案。除非使用選項 `-r`，否則 `rm` 無法移除目錄。

-r

刪除任何現有的子目錄

-i

刪除每一個檔案前，等候確認。

ln [選項] 來源 目標

建立從來源到目標的內部連結。通常，這種連結會直接指向同一檔案系統上的來源。不過，如果執行 `ln` 時使用 `-s` 選項，它會建立符號連結，只會指向來源所在的目錄位置，提供跨檔案系統的連結功能。

-s

建立符號連結

cd [選項] [目錄]

切換目前的目錄。`cd` 未加任何參數可以切換到使用者的主目錄。

mkdir [選項] 目錄

建立新目錄。

rmdir [選項] 目錄

如果指定的目錄已經是空的，則會刪除該目錄。

chown [選項] 使用者名稱[:[群組]] 檔案

將檔案的擁有權轉移給具有指定的使用者名稱的使用者。

-R

變更所有子目錄中的檔案與目錄

chgrp [選項] 群組名稱 檔案

將指定檔案的群組擁有權，移交至指定的群組。如果成員屬於目前群組和新群組，則檔案擁有者只可以變更群組擁有權。

chmod [選項] mode 檔案

變更存取許可權。

mode 參數有三個部分：group (群組)、access (存取) 和 access type (存取類型)。group 可接受下列字元：

u

使用者

g

群組

o

其他

至於 access，使用 + 可授與存取權，使用 - 則可拒絕授與權限。

access type 將提供下列控制選項：

r

讀取

w

寫入

x

執行 -- 可執行檔案，或是變更成目錄

s

Setuid 位元 -- 應用程式或程式可以依照檔案擁有者方式，進行啟動

做為替代方法，可以使用數字程式碼。此程式碼的四個位數組成值 4、2 和 1 的總合 -- 二進位遮罩的十進位結果。第一個位數會設定使用者 ID (SUID) (4)，設定群組 ID (2) 以及黏貼 (1) 位元。第二個位數定義檔案擁有者的許可權。第三個位數定義群組成員的權限，最後的位數會設定所有其他使用者的許可權。讀取許可權設成 4，寫入許可權設成 2，而執行檔案許可權設成 1。檔案的擁有者一般的執行檔案許可權是 6 或 7。

gzip [參數] 檔案

此程式會使用複雜算術演算法來壓縮檔案內容。用此方法壓縮的檔案，其副檔名是 .gz，而且使用前，必須解壓縮。要壓縮多個檔案或整個目錄，請使用 tar 指令。

-d

解壓縮包裝的 gzip 檔案，如此它們會恢復成原始大小而且可以正常處理 (和指令 gunzip 類似)。

tar 選項 **archive** 檔案

tar 將一或多個檔案放到歸檔。壓縮是選擇性的，tar 是相當複雜的指令，有多個選項可用。最常用的選項是：

-f

通常用來將輸出寫入檔案而不是螢幕

-c

建立新的 tar 歸檔

-r

新增檔案至現有的歸檔

-t

輸出歸檔的內容

u

但是只有在檔案比已包含在歸檔中的檔案更新時，才新增檔案

-x

解開歸檔的檔案 (擷取)

-z

使用 `gzip` 包裝產生的歸檔

-j

使用 `bzip2` 壓縮產生的歸檔

-v

列示處理的檔案

由 `tar` 建立，且副檔名是 `.tar` 的歸檔檔案。如果 `tar` 封存也使用 `gzip` 壓縮，則副檔名是 `.tgz` 或 `.tar.gz`。如果它使用 `bzip2` 壓縮，則副檔名是 `.tar.bz2`。您可以在 [節 27.1.8, "歸檔和資料壓縮" \[361\]](#) 找到應用程式範例。

locate 型式

這個指令只有在您已經安裝 `findutils-locate` 套件時才可使用。`locate` 指令可以尋找指定檔案的所在目錄。如有需要，請使用萬用字元來指定檔案名稱。程式執行速度非常快，因為它使用針對這個目的而建立的資料庫 (不必搜尋整個檔案系統)。不過這樣快速的方式，也因此產生了缺點：`locate` 找不到本身資料庫最後更新之後建立的任何檔案。資料庫可以由 `root` 使用 `updatedb` 來產生。

updatedb [選項]

此指令會更新 `locate` 所使用的資料庫。要在現有目錄包括檔案，請以 `root` 身份執行程式。加上 `&`，將它放到背景也是好的方法，如此您可以立即繼續在同一個指令行工作 (`updatedb &`)。這個指令通常當成每日 `cron` 工作 (請參閱 `cron.daily`) 來執行。

find [選項]

使用 `find`，搜尋指定目錄中的檔案。第一個引數會指定從什麼目錄開始搜尋。選項 `-name` 的後面必須有一個搜尋字串，搜尋字串也可以包含萬用字元。和使用資料庫進行搜尋的 `locate` 不同，`find` 會掃描實際目錄。

用於存取檔案內容的指令

cat [選項] 檔案

`cat` 指令可在不中斷的前提下顯示檔案的內容、列印整個內容至螢幕。

-n

在輸出的左邊界加上編號

less [選項] 檔案

此指令可以用來瀏覽指定檔案的內容。使用 `PgUp` 和 `PgDn` 向上或向下捲動螢幕的一半頁面，或者使用 `空格` 鍵捲動整個螢幕頁面。使用 `Home` 和 `End` 移至檔案的開頭或結尾。按 `Q` 結束程式。

grep [選項] **searchstring** 檔案

`grep` 指令可以在指定檔案中搜尋特定的搜尋字串。如果找到搜尋字串，指令會顯示一行，其中會有找到的 `searchstring` 以及檔案名稱。

-i

忽略大小寫

-H

只顯示相關檔案的名稱，不是文字行

-n

另外顯示發現符合資料的行數

-l

只列示 `searchstring` 不存在的檔案

diff [選項] **file1 file2**

`diff` 指令會比較任何兩個檔案的內容。程式產生的輸出，會列示不相符的所有行。只需要傳送程式變動的地方，而不是整個原始程式碼的程式設計人員經常使用此指令。

-q

只報告兩個檔案是否不同

u

產生"制式"差異，使輸出更方便閱讀

檔案系統

mount [選項] [設備] **mountpoint**

此指令可以用來裝載任何資料媒體，例如硬碟、CD-ROM 光碟機以及其他磁碟機至 Linux 檔案系統的目錄。

-r

唯讀裝載

-t filesystem

指定檔案系統，通常 `ext2` 代表 Linux 硬碟、`msdos` 代表 MS-DOS 媒體、`vfat` 代表 Windows 檔案系統、`iso9660` 代表 CD

至於未定義在檔案 `/etc/fstab` 中的硬碟，也必須指定設備類型。在此狀況下，只有 `root` 可以裝載它。如果要讓其他使用者也能裝載該檔案系統，請在 `/etc/fstab` 的適當行，輸入選項 `user` (用逗點分開) 並儲存所做的變更。可在 `mount(1)` `man` 頁面取得更多資訊。

umount [選項] **mountpoint**

此指令會取消裝載檔案系統已裝載的磁碟機。要防止資料遺失，從磁碟機取出抽取式資料媒體前，執行此指令。一般情況下，只允許 `root` 執行 `mount` 與 `umount` 這兩個指令。要讓其他使用者執行這些指令，請編輯 `/etc/fstab` 檔案，為相關的磁碟機指定選項 `user`。

27.3.2 系統指令

下節列出一些擷取系統資訊與程序以及網路控制時最重要的指令。

系統資訊

df [選項] [目錄]

df (磁碟可用空間) 指令，在未加上任何選項時，會顯示所有磁碟空間的資訊、目前使用的磁碟空間以及所有裝載磁碟機的可用空間。如果指定目錄，只會顯示該目錄所在磁碟機的資訊。

-h

顯示佔用的區塊數目 (GB、MB 或 KB) -- 以人類可判讀的格式。

-T

檔案系統的類型 (ext2、nfs 等等)

du [選項] [路徑]

在執行時未加上任何參數，此指令會顯示目前目錄的檔案和子目錄所佔用的總磁碟空間。

-a

顯示每一個別檔案的大小

-h

人類可判讀之格式的輸出

-s

只顯示計算的總共大小

free [選項]

指令 **free** 會顯示關於 RAM 以及交換空間使用的相關資訊，顯示二者類別的總計以及使用總數。請參閱 [節 30.1.6, "free 指令" \[424\]](#) 取得更多資訊。

-b

以位元組為單位的輸出

-k

以 KB 為單位的輸出

-m

以 MB 為單位的輸出

date [選項]

此簡單程式會顯示目前的系統時間。如果以 `root` 的身分執行，它也可以用來變更系統時間。可以在 `date(1)` man 頁面取得程式的詳細資料。

程序

top [選項]

`top` 提供目前執行程序的快速概觀。按 `[H]` 存取頁面，簡略說明主要選項來自訂程式。

ps [選項] [程序 ID]

如果執行時不加任何選項，此指令會顯示由您啟動的程式或程序的相關表格。此指令的選項前面不會加上連字號。

aux

顯示所有程序的詳細清單，與擁有者無關

kill [選項] 程序 ID

不幸地，有時候程式無法以正常方式終止。大部份情況下，您仍然可以執行 `kill` 指令，指定相關的程序 ID，以停止正在執行的程式 (請參閱 `top` 與 `ps`)。 `kill` 會傳送 *TERM* 訊號，指示該程式關閉自己。如果此指令沒有作用，可以使用以下參數：

-9

傳送 *KILL* 訊號而非 *TERM* 訊號，在大部份情況下會結束指定的程序。

killall [選項] processname

此指令類似 `kill`，但是使用處理程序名稱 (而非處理程序 ID) 做為引數，刪除所有該名稱的程序。

網路

ping [選項] 主機名稱或 IP 位址

ping 指令是標準工具，用於測試 TCP/IP 網路的基本功能。它會傳送小的資料封包至目的地主機，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。

-c 數字

決定要傳送的封包總數，並在分派後結束 (依預設，沒有限制設定)

-f

flood ping: 儘可能傳送很多資料封包，一種常用方法，保留給 root 來測試網路

-i 數值

指定兩個資料封包之間的時間間隔 (秒) (預設值: 1 秒鐘)

nslookup

網域名稱系統會將網域名稱解析成 IP 位址。使用此工具，傳送查詢至名稱伺服器 (DNS 伺服器)。

telnet [選項] 主機名稱或 IP 位址 [連接埠]

Telnet 實際上是一種網際網路通訊協定，供您透過網路在遠端主機上工作。Telnet 也是使用這種通訊協定的 Linux 程式名稱，用來在遠端電腦進行作業。

警告

不要在第三方可以"監聽"的網路上使用 telnet。特別是在網際網路上，使用加密傳送方法，例如 ssh，防止密碼被惡意使用的危險 (請參閱 ssh 的 man 頁面)。

其他

passwd [選項] [使用者名稱]

使用者可以使用此指令，在任何時候變更他們自己的密碼。管理員 root 可以使用此指令，變更系統上任何使用者的密碼。

su [選項] [使用者名稱]

`su` 指令可以從執行中的工作階段改用不同的使用者名稱登入。指定使用者名稱和相關密碼。將不會要求 `root` 的密碼，因為 `root` 已授權為可以使用任何使用者的身份。使用指令時如果未指定使用者名稱，將提示您輸入 `root` 密碼並變更為超級使用者 (`root`)。

-

使用 `su -`，為不同使用者啟動登入外圍程序。

halt [選項]

要避免資料遺失，您應該永遠使用此程式來關閉系統。

reboot [選項]

與 `halt` 功能相同，只是系統會立即重新啟動。

clear

此指令會清除主控台的可見區域。它沒有選項。

27.3.3 若需詳細資訊

本章節列示很多指令。如需關於其他指令或更詳細的資訊，建議參閱 O'Reilly 出版的《*Linux in a Nutshell*》。

27.4 vi 編輯器

許多的系統管理工作及程式設計仍然會用到文字編輯器。在 Unix 的世界中，`vi` 作為編輯器的表現十分突出，因為它提供了便利的編輯功能，而且支援滑鼠因而較其他編輯器更符合人體工學。

27.4.1 運作模式

基本上，vi 使用三種運作模式：*插入模式*、*指令模式*及*延伸模式*。按鍵會因為模式而有不同的功能。啟動時，vi 通常設定為「*指令*」模式。第一件要學習的事就是如何切換這些模式：

指令模式切換到插入模式

方法有很多種，包括輸入 **A** 為附加、**I** 為插入或 **O** 為在目前的行中插入新行。

插入模式切換到指令模式

按 **Esc** 鍵便可離開「*插入*」模式。vi 是無法在「*插入*」模式中終止的，因此習慣按 **Esc** 鍵是很重要的。

指令模式切換到延伸模式

vi 的「*延伸*」模式可以藉由輸入冒號 (:) 來啟動。*延伸* 或 *ex* 模式類似獨立的命令行編輯器，可用來處理各種簡單與更複雜的工作。

延伸模式切換到指令模式

在*延伸*模式中執行完指令後，編輯器會自動回到*指令*模式。如果您決定不要執行「*延伸*」模式中的任何指令，請用 **<** 鍵刪除冒號。編輯器便會回到*指令*模式。

您無法直接從*插入*模式切換到*延伸*模式，而不先切換到*指令*模式。

vi 像其他的編輯器一樣，都有自己的終止程式的程序。您無法在「*插入*」模式中終止 vi。首先，按 **Esc** 鍵離開*插入*模式。然後，您會有兩種選項：

1. **不儲存就離開**：若要終止編輯器而不儲存變更，請在「*指令*」模式中輸入 **: - Q - !**。驚歎號 (!) 會讓 vi 忽略任何變更。
2. **儲存並離開**：有許多方法可以儲存您的變更並終止編輯器。在「*指令*」模式中，使用 **Shift + Z + Z**。若要使用「*延伸*」模式離開程式並儲存所有的變更，請輸入 **: - W - Q**。在*延伸*模式中，w 代表寫入，而 q 代表結束。

27.4.2 使用 vi

vi 可以用來做為一般編輯器。在「插入」模式中，輸入文字，然後用 `<—` 鍵與 `Del` 鍵刪除文字。使用方向鍵來移動游標。

不過，這些控制鍵常會造成問題，因為有很多種終端機使用特殊鍵碼。這就是指令模式派上用場之處。按 `Esc` 鍵從插入模式切換到指令模式。在指令模式中，請用 `H`、`J`、`K` 及 `L` 等鍵來移動游標。這些鍵有下列功能：

`H`
向左移動一個字元

`J`
向下移動一行

`K`
向上移動一行

`L`
向右移動一個字元

指令模式中的指令都允許不同的變化。若要執行一個指令數次，只要在輸入實際的指令之前，輸入要重複的次數即可。例如，輸入 `5L` 來將游標向左移動五個字元。

表格 27.1, "vi 編輯器的簡單指令" [381] 顯示部份重要指令。此清單並不完整。節 27.4.3, "若需詳細資訊" [382] 中的文件有更完整的清單

表格 27.1 vi 編輯器的簡單指令

<code>Esc</code>	變更至指令模式
<code>I</code>	變更至插入模式 (字元顯示在目前的游標位置)
<code>A</code>	變更至插入模式 (字元會插入目前的游標位置之後)
<code>Shift</code> + <code>A</code>	變更至插入模式 (字元會新增至行尾)
<code>Shift</code> + <code>R</code>	變更至取代模式 (覆寫舊文字)

R	取代游標下的字元
O	變更至插入模式 (新行會插入目前行的後面)
Shift + O	變更至插入模式 (新行會插入目前行的前面)
X	刪除目前的字元
D - D	刪除目前的行
D - W	刪除至目前文字的最後
C - W	變更至插入模式 (目前文字的其他部份會被您輸入的下一個資料覆寫)
U	復原上次指令
Ctrl + R	重做已復原的變更
Shift + J	將以下行與目前行結合
.	重複上次的指令

27.4.3 若需詳細資訊

vi 支援許多指令。它讓您可以使用巨集、捷徑、具名緩衝區及許多其他有用的功能。不同選項的詳細說明已超出本手冊的範圍。SUSE Linux 會隨附 vim (加強的 vi)，也就是 vi 的加強版。有數個此應用程式的資訊來源：

- vimtutor 是 vim 的互動式教學課程。
- 在 vim 中，請輸入 :help 指令以取得很多的說明主題。
- 關於 vim 的書籍可以在線上取得，網址為 <http://www.truth.sk/vim/vimbook-OPL.pdf>。
- vim 計劃的網頁位於 <http://www.vim.org>，網站中會有所有的新聞、郵件清單及其他的文件。

- 網際網路也有不少的 vim 來源：<http://www.selflinux.org/selflinux/html/vim.html>、<http://www.linuxgazette.com/node/view/9039> 及 http://www.apmaths.uwo.ca/~xli/vim/vim_tutorial.html。請參閱 <http://linux-universe.com/HOWTO/Vim-HOWTO/vim-tutorial.html> 取得更多有關教學課程的連結。

重要: VIM 授權

vim 是「慈善軟體」，意即作者不索取任何的軟體費用，但鼓勵您以捐款贊助，來支持非營利的專案計畫。此計劃懇求您幫助烏干達的貧困孩童。更多的資訊可在線上取得，網址為 <http://iccf-holland.org/index.html>、<http://www.vim.org/iccf/> 及 <http://www.iccf.nl/>。

啟動及設定 Linux 系統

啟動 Linux 系統需要各種不同的元件。本章將概略說明一些基本原則，並特別介紹與啟動相關的元件。本章也將說明 Runlevel 的觀念以及使用 `sysconfig` 進行 SUSE 系統組態。

28.1 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 轉到開機載入器。
2. **開機載入器** 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器會接管開機程序。開機載入器執行的指令，決定其他部份的開機程序。因此，第一個硬碟的前 512 個位元組是主開機紀錄 (MBR)。然後，開機載入器會將控制傳送到實際作業系統，在這裡是指 Linux 核心。Linux 開機載入器的 GRUB 詳細資訊，請參閱 [章 29](#)，[開機載入器 \[399\]](#)。
3. **核心和初始 RAM 檔案系統** 為了送出系統控制，開機載入器會將核心和初始 RAM 檔案系統 (initramfs, initial RAM-based file system) 兩者都載入記憶體。另外，核心可以直接使用 initramfs 的內容。init ramfs 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。舊版的

SUSE Linux 中，這些工作分別是由 `initrd` 和 `linuxrc` 處理。如需有關 `initramfs` 的詳細資訊，請參閱 節 28.1.1, "`initramfs`" [386]。

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，例如提供需要的檔案系統核心功能，並且提供大量儲存控制器設備驅動程式。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 節 28.1.2, "`initramfs` 上的 `init`" [387]。
5. **`init`** `init` 透過提供的不同功能來負責系統實際開機的不同層級，`init` 相關資訊請參閱 節 28.2, "`init` 程序" [388]。

28.1.1 `initramfs`

`initramfs` 是一個小型檔案系統，核心可將此系統載入 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。

在實際根目錄檔案系統能夠裝載以及實際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，所以在整個開機過程中，`initramfs` 是為可用的。如此一來，才可以處理開機期間產生的所有 Hotplug 事件。

如果您要在已完成安裝的系統上變更硬體 (硬碟)，且必須在核心中安裝開機時此硬體所需的驅動程式，那麼您就必須更新 `initramfs`。更新 `initramfs` 會採用與更新 `initrd` (`initramfs` 的前身) 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 中，`/etc/sysconfig/kernel` 中的 `INITRD_MODULES` 變數可以指定要載入的模組。安裝後，此變數會自動設成正確的值。模組會確實依據它們在 `INITRD_MODULES` 中出現的順序來載入。如果使用到多個 SCSI 驅動程式時，這點尤其重要，因為不這麼做，硬碟名稱會有所變更。嚴格來說，載入那些存取根目錄檔案系統所需的驅動程式便足夠了。然而，因為後續載入可能會有問題，所以所有安裝所需的 SCSI 驅動程式都將透過 `initramfs` 或 `initrd` 載入。

重要: 更新 `initramfs` 或 `initrd`

開機載入器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。

28.1.2 `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。根據您實際系統組態，`init` 負責下列工作。

載入核心模組

根據硬體組態，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)。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。

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。RAID 相關資訊請參閱 [節 2.3, "軟體 RAID 組態" \[62\]](#)。LVM 相關資訊請參閱 [節 2.2, "LVM 組態" \[56\]](#)。

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。

在初始開機時呼叫 `init` 為安裝過程的一部份，該工作不同於先前所述。

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，必須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。

啟始硬體辨識並載入適當核心模組

如同在 [節 28.1.1, "initramfs" \[386\]](#) 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。這些值稍後會寫入 `/etc/sysconfig/kernel` 檔案中的 `INITRD_MODULES`，讓後續開機程序能夠使用自訂 `initrd`。安裝程序進行期間，`init` 會載入此模組。

載入安裝系統或救援系統

在硬體妥善識別及載入適當驅動程式後，`init` 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。

28.2 `init` 程序

`init` 程式的程序編號是 1，負責以要求的方式啟始系統。因此，`init` 在此扮演一個特別的角色。它是由核心直接啟動並拒絕訊息 9，因為這個訊息通常會刪除程序。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 [節 28.2.1, "Runlevel" \[388\]](#))。這個檔案還會指定每一個層次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。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 [節 28.2.2, "Init 程序檔" \[391\]](#))。

啟動系統和關閉系統的整個程序是由 `init` 維護。依此觀點，核心可以視為背景程序，它的工作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。

28.2.1 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 28.1, "可用的 Runlevel" \[388\]](#)。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)。非經由核心直接評估的所有參數，都會傳送給 `init`。

表格 28.1 可用的 *Runlevel*

Runlevel	說明
0	系統暫停

Runlevel	說明
S	單一使用者模式；從開機提示，只能使用使用美式鍵盤對應
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	未使用
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 /usr 分割區的 Runlevel 2

如果系統透過 NFS 裝載 /usr 分割區，則不應該使用 Runlevel 2。/usr 目錄保存有系統正常作業所不可或缺的重要程式。因為 Runlevel 2 (本地多重使用者模式，無遠端網路) 無法使用 NFS 服務，所以系統在很多方面都會受到嚴格限制。

若要在系統執行時變更 runlevel，請輸入 `init` 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 區域中最重要的指令摘要。

init 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工作。

init 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。

init 5

啟用圖形環境。您可以選用 GNOME 或 KDE 桌面，或是任何視窗管理員。

init 0 或 shutdown -h now
暫停系統。

init 6 或 shutdown -r now
暫停系統後重新開機

所有 SUSE Linux 標準安裝中，預設都使用 runlevel 5。系統會提示使用者使用圖形介面登入。如果預設 runlevel 是 3，必須如 [章 35, X Window System \[473\]](#) 所述，適當設定 X 視窗系統，runlevel 才可以切換至 5。這個動作完成之後，請輸入 `init 5`，檢查系統是否會依照要求的方式運作。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，編輯 `/etc/inittab` 時要十分小心，並且永遠要備份完整的版本。若要修復損毀，試著在開機提示的核心名稱後面輸入 `init=/bin/sh`，直接開機至外圍程序。接著，使用指令 `mount -o remount,rw /` 讓您的根目錄檔案系統變為可寫入狀態，然後，使用 `cp` 指令，以備份的版本取代 `/etc/inittab`。若要預防檔案系統發生錯誤，請在重新開機前，使用以下指令將根目錄檔案系統變更為唯讀：`mount -o remount,ro /`。

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 `init 5`，要求 `init` 變更為不同的 runlevel。
2. `init` 會洽詢它的組態檔 (`/etc/inittab`) 然後決定是否以新的 runlevel 為參數來啟動 `/etc/init.d/rc`。
3. 現在 `rc` 會呼叫目前 runlevel 的所有停止程序檔，但只限新的 runlevel 沒有啟動的程序檔。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` 後面的號碼指定啟動順序，因為還有一些相關因素要考量。

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`。啟動順序的相同程序，也會適用於此。

變更成與目前 `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`。

28.2.2 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。請在 `/etc/inittab` 中定義這些要執行的程序檔。

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`。也可在這裡找到變更 `runlevel` 的程序檔，不過要從其中一個子目錄（`/etc/init.d/rc0.d` 到 `/etc/init.d/rc6.d`），透過符號連結來呼叫。這樣的安排是為了明確，以避免程序檔用於多個 `runlevel` 時，造成重複情況。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 28.2, "可能的 `init` 程序檔選項" \[391\]](#) 都有說明。直接由 `init` 執行的程序檔沒有這些連結。需要時，可從 `runlevel` 獨立執行這些選項。

表格 28.2 可能的 `init` 程序檔選項

選項	說明
<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。

選項	說明
<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。

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。

開機

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 `pts` 檔案系統，啟動 `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。上述程序檔觸發的動作(例如，執行一些子程序檔)完成後，`blogd` 精靈就會停止，並將螢幕的所有輸出寫入 `/var/log/boot.msg` 記錄檔，但這只發生在 `/var` 裝載成讀寫時。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。

程序檔 `boot` 也負責啟動 `/etc/init.d/boot.d` 中，名稱開頭是 `s` 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。最後執行的是程序檔 `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`。

boot.setup

從單一使用者模式變更成任何其他 `runlevel` 時會執行此程序檔，它負責一些基本設定，例如鍵盤配置和虛擬主控台的啟始化。

halt

只有進入 runlevel 0 或 6 時，才會執行此程序檔。在此，它是以 暫停 或 重新開機 的方式執行。系統是否關閉或重新啟動，取決於呼叫 halt 的方式。

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。

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、命名以及組織自訂程序檔的指示，請參閱 LSB 的規格和 init、init.d 以及 insserv 的 man 頁面。另外請查閱 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。關於 init 程序檔的一些有用資訊，請參閱 [節 28.2.1, "Runlevel" \[388\]](#)。

若要為指定的程式或服務建立自訂 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，應該要編輯。請參閱 [範例 28.1, "迷你 INIT INFO 區塊" \[393\]](#)。

範例 28.1 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。在 Required-Start: 和 Required-Stop: 這兩行，在服務本身啟動或停止前，指定要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。在

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。

若要建立由 runlevel 目錄 (/etc /init. d/rc?. d/) 對應到 /etc /init. d/ 相關程序檔的連結，請輸入指令 `insserv new-script-name`。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 (請參閱 [節 28.2.3, "使用 YaST 設定系統服務 \(Runlevel\)" \[394\]](#))。

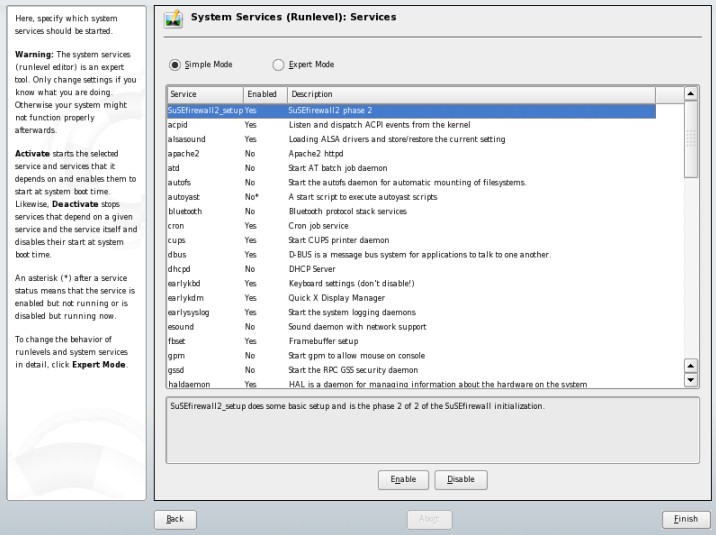
如果 /etc /init. d/ 裡已經有程序檔，應該整合至現有的 runlevel 配置，使用 insserv 在 runlevel 目錄中立即建立連結，或者在 YaST 的 runlevel 編輯器中啟用對應的服務。下次重新啟動時，將會套用您所做的變更 - 新服務會自動啟動。

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。

28.2.3 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。

圖形 28.1 系統服務 (Runlevel)



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 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)。

您可以在此 YaST 對話方塊中選取其中一個 Runlevel 選項 (如 [表格 28.1, "可用的 Runlevel" \[388\]](#) 所列) 做為新的預設值。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 一開始並未定義，以便建立自訂 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。

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。

警告: 錯誤的 **Runlevel** 設定可能會造成系統損害

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

28.3 透過 `/etc/sysconfig` 設定系統

`/etc/sysconfig` 中的組態檔是控制 SUSE Linux 主要組態的檔案。`/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。很多其他系統組態檔是根據 `/etc/sysconfig` 的設定而產生的。此工作是由 **SUSEconfig** 執行。例如，如果您變更網路組態，**SUSEconfig** 就可能也會變更檔案 `/etc/host.conf`，因為這是與網路組態相關的一個檔案。這種概念讓您可以對組態進行基本變更，不必重新啟動系統。

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。

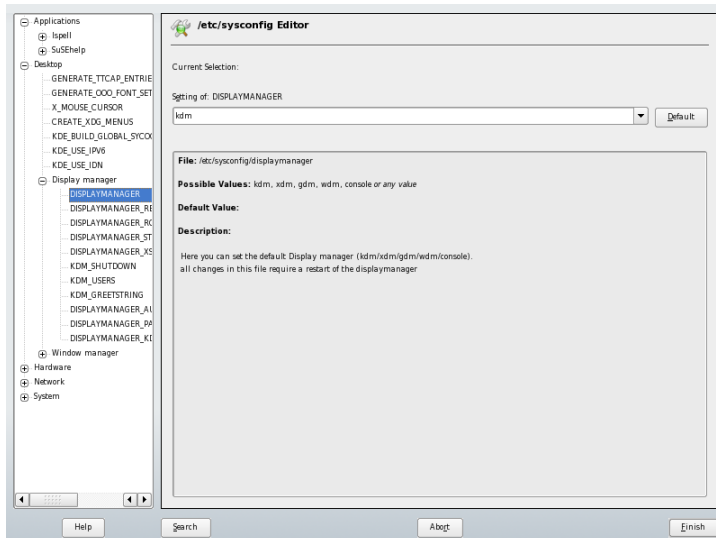
28.3.1 使用 **YaST Sysconfig** 編輯器變更系統組態

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 28.2 使用 *sysconfig* 編輯器的系統組態



YaST Sysconfig 對話方塊分為三個部分。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。此對話方塊還會提供變更變數後執行的組態程序檔，以及變更組態後新啟動服務的相關資訊。YaST 會提示您確認變更，當您選取完成 離開對話方塊後，會通知您要執行的程序檔為何。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。

28.3.2 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 使用者身分登入。
- 2 使用 `init 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 使用如 `init default_runlevel` 指令使系統組態還原為先前的 **Runlevel**。接著會以系統預設的 **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

若要使用 `SUSEconfig` 關閉自動系統組態，將 `/etc/sysconfig/suseconfig` 的 `ENABLE_SUSECONFIG` 設為 `no`。如果您想使用 **SUSE** 安裝支援，那麼請勿停用 `SUSEconfig`。也可以停用部份自動組態。

開機載入器

本章將描述如何設定 GRUB，即 SUSE Linux 所使用的開機載入器。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。

本章節著重於開機管理以及開機載入器 GRUB 的組態。整個開機程序簡述於 [章 28, 啟動及設定 Linux 系統 \[385\]](#)。開機載入器是機器 (BIOS) 與作業系統之間的介面 (SUSE Linux)。開機載入器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。它們通常具有開機載入器程式，在這個例子中為 GRUB。接下來的 64 位元組提供多達四個分割區表格的空間 (請參閱 "分割區類型" (章 1, 使用 YaST 安裝, ↑啟動))。分割區表格包含硬碟的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。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，BIOS 和所有個人電腦作業系統都會被視為無效。

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，直到設定檔案系統之後，才會寫入記錄。因此，Linux 分割區無法自行開機，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)。

29.1 開機管理

在最簡單的案例中 -- 如果只有一個作業系統安裝在電腦上 -- 開機管理會如上述般進行。如果多個作業系統安裝在電腦上，可以使用以下選項：

從外部媒體啟動其他系統

其中一個作業系統可以從硬碟開機。其他作業系統是透過安裝在外部媒體的開機管理員開機的 (磁片、USB 儲存媒體)。

在 MBR 安裝開機管理員

開機管理員可以在一部電腦上進行同時安裝並交替使用多個系統。使用者可以在開機程序期間選取要開機的系統。若要變更至其他系統，必須重新啟動電腦。只有選取的開機管理員與安裝的作業系統相容時，才有可能。GRUB，這是 SUSE Linux 使用的開機管理員。

29.2 選取開機載入器

根據預設，SUSE Linux 會使用 GRUB 開機載入器。不過，有時候以及在特殊軟、硬體的配合，LILO 可能更合適。如果您更新使用 LILO 的舊版 SUSE Linux，則會安裝 LILO。

如需關於安裝和設定 LILO 的資訊，請參閱「支援資料庫」中的關鍵字 LILO 的部份 `/usr/share/doc/packages/lilo`。

29.3 使用 GRUB 開機

GRUB (Grand Unified Bootloader) 包含兩個階段。第一個階段包含 512 個位元組並寫入 MBR 或硬碟分割區或磁片的開機磁區。接下來，會載入第二個階段。此

階段包含實際的程式碼。第一個階段的唯一工作是載入開機載入器的第二個階段。

stage2 可以存取檔案系統。目前支援，Ext2、Ext3、ReiserFS、Minix 和 Winodws 使用的 DOS FAT 檔案系統。對於 JFS、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 (menu.lst) 所做的變更，將不再需要重新安裝開機管理員。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 開機的分割區或作業系統，所有的相關資訊。沒有此資訊，系統控制權便無法傳送給作業系統。

/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器時需要的參數和選項。

GRUB 有多種方法可以控制。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，開機前可以變更所有開機參數。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 "[在開機程序期間編輯功能表項目](#)" [405])。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器組態中不存在的項目，啟動安裝的作業系統。

GRUB 外圍程序在安裝的系統中，提供 GRUB 的模組。它可以在實際套用之前，安裝 GRUB 或測試新設定。請參閱 [節 29.3.4, "GRUB 外圍程序"](#) [408]。

29.3.1 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 開機載入器來修改 GRUB 組態，如 [節 29.4, "使用 YaST 設定開機載入器" \[410\]](#) 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。

要識別功能表概觀中的功能表項目，為每一個項目指定標題。關鍵字標題後面的文字 (包括任何空格) 會在功能表中顯示成可選取的選項。下一個標題的所有指令，會在選取此功能表項目時執行。

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

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

GRUB 中的設備名稱會在 ["硬碟和分割區的命名慣例" \[403\]](#) 說明。上述範例指定第一個硬碟中，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。

如果核心沒有內建的驅動程式可存取根分割區，`initrd` 必須以個別的 GRUB 指令來指定，它唯一的引數是 `initrd` 檔案的路徑。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 GRUB 設備或 GRUB 設備上的分割區。此設備是用於所有核心、`initrd` 或其它檔案路徑，就檔案路徑而言，直到下一個 `root` 指令下達時，才會明確指定設備。此指令不是用於安裝時產生的 `menu.lst` 檔案。它只是利於手動編輯。

`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** 來開機，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，將其中一個定義為預設項目。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開機。`timeout` 和 `default` 通常在功能表項目前面。["範例功能表檔案" \[404\]](#) 會敘述一個範例檔案。

硬碟和分割區的命名慣例

硬碟和分割區使用的命名慣例 **GRUB** 與一般 **Linux** 設備使用的命名慣例不同。在 **GRUB**，分割區的編號會從 0 開始。因此，`(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/hda1`。

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：

<code>(hd0, 0)</code>	第一個硬碟的主分割區
<code>(hd0, 1)</code>	第二個主分割區
<code>(hd0, 2)</code>	第三個主分割區
<code>(hd0, 3)</code>	第四個主分割區 (一般是擴充分割區)
<code>(hd0, 4)</code>	第一個邏輯分割區
<code>(hd0, 5)</code>	第二個邏輯分割區

GRUB 不會區分 **IDE**、**SCSI** 和 **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 **GRUB** 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 [節 29.3.2, "檔案 device.map" \[407\]](#)。

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核心：

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

範例功能表檔案

以下範例顯示 GRUB 功能表檔案的結構。範例安裝包含 /dev/hda5 下的 Linux 開機分割區、/dev/hda7 下的根分割區，以及 /dev/hda1 下的 Windows 安裝。

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

title linux
    kernel (hd0,4)/vmlinuz root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd

title windows
    chainloader(hd0,0)+1

title floppy
    chainloader(fd0)+1

title failsafe
    kernel (hd0,4)/vmlinuz.shipped root=/dev/hda7 ide=nodma \
    apm=off acpi=off vga=normal nosmp maxcpus=0 3
    initrd (hd0,4)/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` 是位在 /dev/hda5。

color white/blue black/light-gray

色彩配置：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訂的 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。若要關閉自動開機，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。個別作業系統的章節是由 `title` 開始。

- 第一個項目 (`title linux`) 負責啟動 SUSE Linux。核心 (`vmlinuz`) 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/hda7/`) 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開機 (`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。

若有需要可自行變更功能表檔案。GRUB 會在下次開機時使用修改的設定。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 ["在開機程序期間編輯功能表項目"](#)^[405]。

在開機程序期間編輯功能表項目

在圖形化 GRUB 開機功能表中，可以使用方向鍵選取要開機的作業系統。如果選取 Linux 系統，可以在開機提示時輸入其他開機參數。要完全直接編輯個別功能表，按 `[Esc]` 結束開頭顯示畫面，然後按 `[E]`。用此方式所做的變更，只會套用至目前的開機程序，不會永久變更。

重要：開機程序期間的鍵盤配置

US 鍵盤配置是開機時唯一可以使用的鍵盤配置。

啟動編輯模式之後，使用方向鍵選取功能表項目來編輯組態。要讓組態變成可以編輯，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。

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添加至現有的行，以空格分隔：

```
title linux
    kernel (hd0,0)/vmlinuz root=/dev/hda3 additional parameter
    initrd (hd0,0)/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 開機載入器模組做此變更。將新參數添加至現有的行，由空格分開。

使用萬用字元以選取開機核心

特別是在開發或使用自訂核心時，您必須變更 `menu.lst` 中的項目，或是編輯指令行以反映目前核心及 `initrd` 的檔案名稱。若要簡化此程序，請使用萬用字元以動態更新 GRUB 的核心清單。接著會將所有符合特定型式的核心影像都自動加入可開機影像的清單。請注意，未提供此功能的技術支援。

在 `menu.lst` 中輸入其他的功能表項目以啟動萬用字元選項。為了使用上的便利，所有的核心和 `initrd` 影像必須具有共同的基礎名稱以及符合核心與其相關 `initrd` 的識別碼。請思考以下設定：

```
initrd-default
initrd-test
vmlinuz-default
vmlinuz-test
```

在此狀況中，您可以在一個 GRUB 組態中同時新增兩個開機影像。若要取得功能表項目 `linux-default` 與 `linux-test`，將需要 `menu.lst` 中的下列項目：

```
title linux-*
    wildcard (hd0,4)/vmlinuz-*
    kernel (hd0,4)/vmlinuz-* root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd-*
```

在此範例中，GRUB 會搜尋分割區 (hd0,4) 以尋找符合萬用字元的項目。這些項目會用來產生新的 GRUB 功能表項目。在前述的範例中，GRUB 的行為方式將如同下列項目存在於 `menu.lst` 一般：

```
title linux-default
    wildcard (hd0,4)/vmlinuz-default
    kernel (hd0,4)/vmlinuz-default root=/dev/hda7 vga=791
    initrd (hd0,4)/initrd-default
title linux-test
    wildcard (hd0,4)/vmlinuz-test
```

```
kernel (hd0,4)/vmlinuz-test root=/dev/hda7 vga=791
initrd (hd0,4)/initrd-test
```

如果所使用的檔案名稱不一致，或是如果其中一個已展開的檔案 (例如，initrd 影像) 已遺失，此組態就會發生問題。

29.3.2 檔案 device.map

檔案 device.map 會將 GRUB 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 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/hda
(hd1) /dev/sda
```

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。如果您在開機時發生問題，請檢查此檔案中的順序是否與 BIOS 中的順序一致，然後在需要時使用 [節 29.3.4, "GRUB 外圍程序" \[408\]](#) 中所描述的 GRUB 外圍程序來修改它。啟動 Linux 系統之後，檔案 device.map 可以透過 YaST 開機載入器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
```

29.3.3 檔案 /etc/grub.conf

除了 menu.lst 和 device.map 之外，第三個重要的 GRUB 組態檔是 /etc/grub.conf。此檔案包含指令 grub 正確安裝開機載入器時，需要的參數和選項：

```
root (hd0,4)
install /grub/stage1 d (hd0) /grub/stage2 0x8000 (hd0,4)/grub/menu.lst
quit
```

個別項目的意義如下：

root (hd0,4)

此指令告訴 GRUB 將以下指令套用至第一個硬碟的第一個邏輯分割區 (開機檔案的位置)。

nstall 參數

指令 grub 應該與參數 install 一起執行。開機載入器的 stage1 應該安裝在第一個硬碟的 MBR (/grub/stage1 d (hd0))。stage2 應該載入記憶體位址 0x8000 (/grub/stage2 0x8000)。最後項目 ((hd0, 4) /grub/menu.lst) 告訴 GRUB 到什麼地方尋找功能表檔案。

29.3.4 GRUB 外圍程序

GRUB 實際上有兩個版本：當成開機載入器以及當成 /usr/sbin/grub 的一般 Linux 程式。此程式稱為 *GRUB 外圍程序*。將 GRUB 安裝成硬碟或磁片上的開機載入器，這種功能是以指令 install 和 setup 的形式，整合在 GRUB。Linux 載入時，可以在 GRUB 外圍程序使用此指令。

不過，指令 setup 和 install 也可以在 Linux 啟動前的開機程序中使用。這可利於修復無法再開機的缺陷系統，因為開機載入器錯誤的組態檔，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。只要使用像 menu.lst 的語法，輸入實驗性組態檔就可以了。測試此項目的功能，無需變更現有的組態檔。例如，要測試新核心，輸入指令 kernel 以及新核心的路徑。如果開機程序失敗，下次啟動時，可以繼續使用未受損的 menu.lst。同樣地，指令行介面也可以用來輸入正確的參數，即使 menu.lst 檔案錯誤，也可以啟動系統。正在執行的系統中，正確的參數可以輸入在 menu.lst，讓系統變成可永久開機。

GRUB 設備與 Linux 設備名稱的對應，只有將 GRUB 外圍程序當成 Linux 程式執行時，才會發生關係 (輸入 grub，如 [節 29.3.2, "檔案 device.map" \[407\]](#) 所述一般)。基於此目的，程式會讀取檔案 device.map。如需更多資訊，請參閱 [節 29.3.2, "檔案 device.map" \[407\]](#)。

29.3.5 設定開機密碼

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)。要封鎖此類型的存取或者防止使用者啟動特定作業系統，可以設定開機密碼。

重要：開機密碼和開頭顯示畫面

如果您在 GRUB 使用開機密碼，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 在根提示時，輸入 `grub`。
- 2 在 GRUB 外圍程序加密密碼：

```
grub> md5crypt
Password: ****
Encrypted: $1$lS2dv/$JOYcdxIn7CJk9xShzzJVw/
```

- 3 將加密字串貼到檔案 `menu.lst` 的全域段落：

```
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** 並輸入密碼後的開機提示時執行。不過，使用者仍然可以從開機功能表，啟動所有作業系統。

- 4 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開機的 `menu.lst` 每一個段落。例如：

```
title linux
kernel (hd0,4)/vmlinuz root=/dev/hda7 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，會顯示以下錯誤：

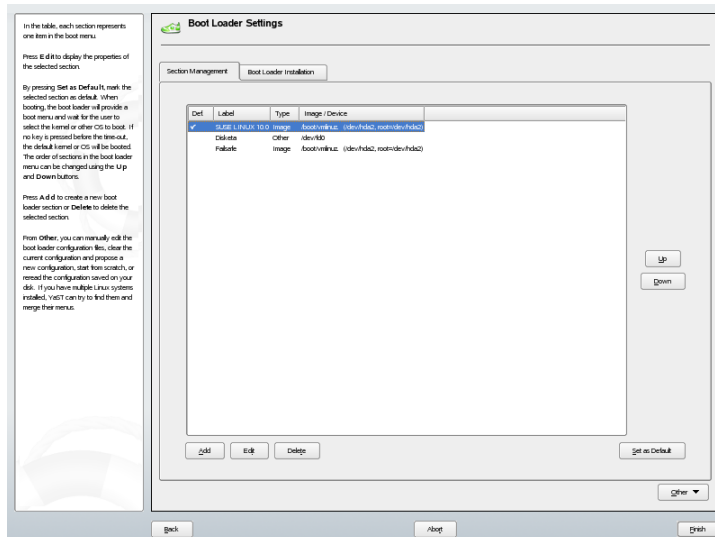
```
Error 32: Must be authenticated
```

按 **Enter** 進入功能表。然後按 **P**，取得密碼提示。輸入密碼並按 **Enter** 之後，選取的作業系統 (本案例為 **Linux**) 應該會啟動。

29.4 使用 YaST 設定開機載入器

在 SUSE Linux 系統中設定開機載入器最簡單的方法就是使用 YaST 模組。在「YaST 控制中心」選取 **系統 → 開機載入器組態**。系統目前的開機載入器組態會顯示出來，供您進行必要的變更。請參閱 [圖形 29.1, "使用 YaST 設定開機載入器" \[410\]](#)。

圖形 29.1 使用 YaST 設定開機載入器



主要視窗包含 2 個索引標籤：

磁碟區管理

使用這個索引標籤可以編輯、變更和刪除單一作業系統的開機載入器磁碟區。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**。如果您不想使用某個現有選項，請先選取再按一下 **刪除**。如果不熟開機載入器選項，請先閱讀 [節 29.3, "使用 GRUB 開機" \[400\]](#)。

開機載入器安裝

使用這個索引標籤可以檢視和變更與類型、位置和其他開機載入器相關的設定。

29.4.1 開機載入程式類型

開機載入器類型可在 *開機載入器安裝* 索引標籤設定。SUSE Linux 會使用預設的 GRUB 開機載入器。若要使用 LILO，請繼續下列步驟：

過程 29.2 變更開機載入器類型

- 1 開啟索引標籤 *開機載入器安裝*。
- 2 在 *類型* 面板，按一下 *開機載入器* 功能表然後選 *LILO*。
- 3 從快顯示功能表選擇下列其中一個操作：

建議新組態

YaST 建議新的組態

轉換目前的組態

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

使用此選項來寫入自訂組態。安裝 SUSE Linux 時不能使用這個動作。

讀取儲存在硬碟的組態

使用此選項載入 `/etc/lilo.conf`。安裝 SUSE Linux 時不能使用這個動作。

- 4 按一下 *確定* 以儲存變更。
- 5 完成後，按一下主對話方塊視窗的 *完成* 以啟動變更。

轉換後，舊的 GRUB 組態會儲存在硬碟。若要使用它，只要將開機載入器類型變更回 GRUB 然後從快顯示功能表選擇 *回復轉換前儲存的組態*。只能在已安裝好的系統使用這個操作。

注：自訂開機載入器

如果要使用 GRUB 或 LILO 以外的開機載入器，按一下選項 *不要安裝任何開機載入器*。在選擇此選項前，請先詳細閱讀有關開機載入器的文件。

29.4.2 開機載入程式位置

可能會需要變更開機載入器的位置。YaST 模組可以幫助您。

過程 29.3 變更開機載入器位置

- 1 若要變更開機載入器的位置，按一下 **開機載入器安裝** 索引標籤，然後從 **開機載入器位置** 功能表選擇下列其中一個選項：

/dev/hdX 的主開機記錄

主要開機磁區記錄。當 SUSE 決定使用這個放是開機時，建議您使用這個選項。X 表示為硬碟，其他的字 a、b、c、d：

```
hda => ide0 master
hdb => ide0 slave
hdc => ide1 master
hdd => ide1 slave
```

開機分割區的開機磁區 /dev/hdXY

/boot 分割區的開機磁區這個選項為預設值，如果在硬碟上安裝多個作業系統。Y 表示分割區，例如 1、2、3、4、5 等等。輸入的項目可能會如下所示：

```
/dev/hda1
```

Root 分割區的開機磁區 /dev/hdXY

/(root) 分割區的開機磁區也可使用這個選項，如果在硬碟上安裝多個作業系統，且要繼續使用舊的開機管理員。

其他

這個選項可讓您指定開機載入器的位置。

- 2 按一下 **完成** 以啟用變更。

29.4.3 預設系統

若要變更預設系統，請按照下列步驟進行：

過程 29.4 設定預設系統

- 1 開啟 **磁碟區管理** 索引標籤。

- 2 請使用滑鼠從清單選取想要的系統或按一下 *up* 或 *down*。
- 3 按一下 *設定為預設*。
- 4 按一下 *完成* 以啟用這些變更。

29.4.4 開機載入器逾時

開機載入器不會立即啟動預設系統。逾時期間，您可以停止啟動預設系統，變更要啟動的系統或寫入核心參數。若要增加或減少開機載入器逾時時間，請繼續下列步驟：

過程 29.5 變更開機載入器逾時

- 1 開啟 *開機載入器安裝 索引標籤*。
- 2 按一下 *開機載入器選項*。
- 3 按一下 *顯示開機功能表*。
- 4 在 *開機功能表*，輸入新的值或使用滑鼠按住適當的箭號，或使用鍵盤上的箭號，變更 *開機載入器逾時* 的值。
- 5 按一下 *確定*。
- 6 按一下 *完成* 以啟用變更。

您可以按一下 *逾時後繼續開機* 決定是否永久顯示開機功能表而不需倒數。

29.4.5 安全性設定

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

過程 29.6 設定開機載入器密碼

- 1 開啟 *開機載入器安裝 索引標籤*。

- 2 按一下 **開機載入器選項**。
- 3 在 **密碼保護** 勾選 **使用密碼保護開機載入器**，設定您的密碼。
- 4 按一下 **確定**。
- 5 按一下 **完成** 以啟用變更。

29.4.6 磁碟順序

如果您的電腦有多個硬碟，您可以指定機器的 BIOS 設定中指定的磁碟開機順序 (請參閱 [節 29.3.2, "檔案 device.map" \[407\]](#))。若要執行，請繼續下列程序：

過程 29.7 設定磁碟順序

- 1 開啟 **開機載入器安裝 索引標籤**。
- 2 按一下 **開機載入器詳細資料**。
- 3 如果列出超過一個磁碟，選擇一個磁碟然後按一下 *Up* 或 *Down* 重新排列顯示的磁碟。
- 4 按一下 **確定** 以儲存變更。
- 5 按一下 **完成** 以啟用變更。

使用這個模組，您也可以使用標準程式碼 (啟動作用中的分割區) 取代主開機記錄。按一下 **硬碟系統區域更新中的 使用標準程式碼取代 MBR**。您也可以按一下同一個窗格的 **啟動開機載入器分割區** 以啟動包含開機載入器的分割區。按一下 **完成** 以啟用變更。

29.5 解除安裝 Linux 開機載入器

YaST 可以用來解除安裝 Linux 開機載入器，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，然後在需要時還原，覆寫。

要解除安裝 GRUB，請開啟 YaST 開機載入器模組 (系統 → 開機載入器組態)。在第一個對話方塊，選取 重設 → 還原硬碟的 MBR，然後使用完成來結束對話方塊。在 MBR 中，原始的 MBR 資料會覆寫 GRUB。

29.6 建立開機 CD

如果使用開機管理員來啟動系統時發生問題，或者無法在硬碟或磁片的 MBR 上安裝開機管理員，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。

要使用 GRUB 建立可開機的 CD-ROM 只需要一個特殊形式的 *stage2*，它叫做 *stage2_eltorito*，還可以選用自訂的 *menu.lst*。不需要典型的檔案 *stage1* 和 *stage2*。

建立 ISO 影像要使用的目錄，例如，您可以使用 `cd /tmp` 和 `mkdir iso` 等指令。同時，使用 `mkdir -p iso/boot/grub` 指令可建立一個 GRUB 子目錄。將檔案 *stage2_eltorito* 複製到目錄 *grub*：

```
cp /usr/lib/grub/stage2_eltorito iso/boot/grub
```

另外，將核心 (*/boot/vmlinuz*)、*initrd* (*/boot/initrd*) 以及檔案 */boot/message* 複製到 *iso/boot/*：

```
cp /boot/vmlinuz iso/boot/  
cp /boot/initrd iso/boot/  
cp /boot/message iso/boot/
```

若要讓它們可供 GURB 使用，請將檔案 *menu.lst* 複製到 *iso/boot*，然後調整路徑項目，使它們指向 CD-ROM 光碟機。方法是以 CD-ROM 光碟機的裝置名稱，也就是 (*cd*) 來取代路徑名稱中的硬碟裝置名稱，也就是格式中所列的 (*hd**)：

```
gfxmenu (cd)/boot/message  
timeout 8  
default 0  
  
title Linux  
    kernel (cd)/boot/vmlinuz root=/dev/hda5 vga=794 resume=/dev/hda1  
splash=verbose showopts  
    initrd (cd)/boot/initrd
```

最後，使用以下指令建立 ISO 影像：

```
mkisofs -R -b boot/grub/stage2_eltorito -no-emul-boot \
-boot-load-size 4 -boot-info-table -o grub.iso iso
```

使用您偏好的公用程式，將產生的檔案 `grub.iso` 燒錄至 CD。

29.7 圖形化 GRUB 畫面

從 SUSE Linux 7.2 開始，如果使用「`vga=<value>`」選項作為核心參數，則圖形化 SUSE 畫面會顯示在第一個主控台。如果您使用 YaST 來安裝，則會依照選取的解析度與顯示卡來自動啟動此選項。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。
若要再次啟動它，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` 到開機載入器的組態。[章 29, 開機載入器 \[399\]](#) 能提供更多有關這方面的資訊。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 *framebuffer* 支援中的 *使用開機顯示畫面而非開機標幟* 選項。

提示

在核心中關閉框架緩衝區支援會自動關閉開機顯示畫面。如果您使用自訂核心來執行，SUSE 將不會為您的系統提供任何支援。

29.8 疑難排解

此章節列示使用 GRUB 開機時，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「支援資料庫」中討論，網址是 <http://portal.suse.de/sdb/en/index.html>。如果您的特殊問題未包含在此清單中，使用「支援資料庫」的搜尋對話方塊 (網址是 <https://portal.suse.com/PM/page/search.pm>) 來搜尋像 *GRUB*、*boot* 以及 *boot loader* 的關鍵字。

GRUB 和 XFS

XFS 在分割區開機區塊中，不會保留空間給 `stage1`。因此，不要將 XFS 分割區指定成開機載入器的位置。建立不是以 XFS 格式化的個別開機分割區，就可以解決此問題。

GRUB 和 JFS

雖然技術上可行，不過 GRUB 和 JFS 的組合仍有問題。在此案例中，建立個別開機分割區 (`/boot`) 並以 Ext2 格式化。在此分割區安裝 GRUB。

GRUB 報告 GRUB 位置錯誤

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。如果發生此狀況，使用 LILO 或更新 BIOS。如需關於 LILO 的安裝、組態和維護的詳細資訊，請參閱「支援資料庫」中的關鍵字 LILO。

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。開機載入器的 *stage1* 可以正確找到和載入，不過 *stage2* 則找不到。將新磁碟註冊在 BIOS 就可以解決此問題。

包含 IDE 和 SCSI 硬碟的系統不會啟動

安裝時，YaST 可能已經判斷硬碟的開機順序錯誤 (而且您尚未更正它)。例如，GRUB 可以將 `/dev/hda` 當成 `hd0` 而將 `/dev/sda` 當成 `hd1`，即使 BIOS 中的開機順序是其他方式 (SCSI 在 IDE 前面)。

發生此狀況，在開機程序時，透過 GRUB 指令行的協助來更正硬碟。系統啟動之後，編輯檔案 `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，重新安裝開機載入器：

```
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(hdl,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，必須為 chainloader 指定第二個硬碟。

29.9 若需詳細資訊

如需關於 GRUB 更進一步的資訊，請參閱 <http://www.gnu.org/software/grub/>。如果您的機器上裝有 texinfo，輸入 `info grub` 可以檢視外圍程序中的 GRUB 資訊頁面。您也可以「支援資料庫」搜尋關鍵字「GRUB」，取得關於特殊問題的相關資訊，網址是 <http://portal.suse.de/sdb/en/index.html>。

SUSE Linux 的特殊功能

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)。

30.1 特殊軟體套件的資訊

程式 `bash`、`cron`、`logrotate`、`locate`、`ulimit` 和 `free` 以及檔案 `resolv.conf`，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。

30.1.1 Bash 套件與 `/etc/profile`

Bash 是 SUSE Linux 的預設外圍程序。如果以它做為登入外圍程序，可以讀取多種起始化檔案。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`

使用者可以在 `~/.profile` 或 `~/.bashrc` 中建立自訂設定。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。

30.1.2 cron 套件

如果您要按照預先定義的時間在背景中自動定期執行指令，可以使用傳統工具 `cron`。`cron` 是由特殊格式的時間表格所驅動的。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。請在時間表格之後，輸入應該直接執行指令的使用者名稱。在 [範例 30.1, "/etc/crontab 中的項目" \[420\]](#) 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 30.1 `/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，其指令會由 `/usr/lib/cron/run-crons` 所控制。`/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。

如果要按照自訂時間，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 30.2, "/etc/crontab: 移除時間戳記檔案" \[421\]](#)，它可以在整點前移除 `hourly`、在每天的 2:14 a.m. 移除 `daily` 等)。

範例 30.2 /etc/crontab: 移除時間戳記檔案

```
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
```

每日的系統維護工作因為明確的理由，已經配送到不同的程序檔。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。

30.1.3 記錄檔：logrotate 套件

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。

組態

使用 `/etc/logrotate.conf` 檔案來設定 logrotate。尤其 `include` 規格主要是設定其他要讀取的檔案。SUSE Linux 能確保產生記錄檔的程式，會將個別的組態檔安裝在 `/etc/logrotate.d` 中。例如，套件 `apache2` (`/etc/logrotate.d/apache2`) 和 `syslogd` (`/etc/logrotate.d/syslog`) 中的這類程式。

範例 30.3 */etc/logrotate.conf* 的範例

```
# 請參閱 "man logrotate" 以尋求詳細資訊
# 每週輪換記錄檔案
weekly

# 保留四週的記錄備份
rotate 4

# 在週期記錄檔案之後，建立新的（空的）記錄檔
create

# 如果您想要壓縮記錄檔的話，請不要加上註解符號
#compress

# RPM 套件會將記錄檔的週期資訊放到此目錄中
include /etc/logrotate.d

# 沒有套件擁有 lastlog 或 wtmp - 在此我們將輪流使用
#/var/log/wtmp {
#     monthly
#     create 0664 root utmp
#     rotate 1
# }
```

系統專用的記錄檔也可以在此處設定。

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。

30.1.4 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，請安裝套件 `find-locate`。updatedb 程序會在每晚自行啟動，或在啟動系統後的 15 分鐘左右啟動。

30.1.5 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 30.1](#), "`ulimit`: 設定使用者的資源" [423] 中所列的選項。

表格 30.1 *ulimit*: 設定使用者的資源

<code>-m</code>	實體記憶體的最大大小
<code>-v</code>	虛擬記憶體的最大大小
<code>-s</code>	堆疊的最大大小
<code>-c</code>	核心檔案的最大大小
<code>-a</code>	限制集的顯示

您可以在 `/etc/profile` 中做整個系統的設定。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 30.4 *ulimit*: 在 `~/.bashrc` 中的設定

```
# 限制實體記憶體
ulimit -m 98304

# 限制虛擬記憶體
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` 指令。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。

30.1.6 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`)。

核心也可以有其它的快取，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。

30.1.7 /etc/resolv.conf 檔案

領域名稱解析會透過 `/etc/resolv.conf` 檔案來處理。請參照 [章 40](#)，[領域名稱系統 \[547\]](#)。

只有 `/sbin/modify_resolvconf` 程序檔可以更新此檔案，其它的程式沒有權限可以直接修改 `/etc/resolv.conf`。能保證系統的網路組態與相關檔案都保持在一致狀態的唯一方法，就是強制執行此規則。

30.1.8 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` 頁面的快速概觀，這些都會提供更深入詳盡的說明。`info` 是 GNU 的超文字系統。您可以輸入 `info info` 來讀取此系統的介紹。您可以輸入 `emacs -f info` 或直接在主控台中使用 `info`，以便使用 Emacs 檢視資訊頁面。您也可以使用 `tkinfo`、`xinfo` 或 SUSE 說明系統檢視 `info` 頁面。

30.1.9 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，以及取得自訂或預設組態的設定。起始化檔案 `~/.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。如果要自訂程式，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` 檔案為自訂檔案。如果使用者是使用 Emacs 中的自訂選項來進行設定，這些設定會儲存至 `~/.gnu-emacs-custom` 中。

透過 SUSE Linux，emacs 套件可將 `site-start.el` 檔案安裝至 `/usr/share/emacs/site-lisp` 目錄中。`site-start.el` 檔案會在起始化檔案 `~/.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。本機系統管理員可在 `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` (通常會安裝)：使用 X11 支援的程式。
- `emacs-nox`：不使用 X11 支援的程式。
- `emacs-info`：線上的 `info` 格式文件。
- `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。

- 需要時可加以安裝的多個附加套件：emacs-auctex (用於 LaTeX)、psgml (用於 SGML 與 XML)、gnuserv (用於用戶端和伺服器操作) 以及其他等等。

30.2 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]**。

30.3 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
/usr/X11R6/lib/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/X11R6/lib/X11/app-defaults/XTerm
/usr/share/emacs/<版本>/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 SUSE Linux 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 (多重鍵)。請參閱 `/usr/X11R6/lib/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: 若需詳細資訊

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>。

30.4 語言與國家專用的設定

SUSE Linux 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。依預設，SUSE Linux 僅會設定 `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。

其他的變數可以透過 YaST `sysconfig` 編輯器來設定 (請參閱 [節 28.3.1, "使用 YaST Sysconfig 編輯器變更系統組態" \[396\]](#))。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>]]
```

30.4.1 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/codlstpl/en_listpl.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。描述檔包含在 `glibc-i18ndata` 套件中。`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。

30.4.2 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 `Bokmål` (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。

30.4.3 若需詳細資訊

- *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

CUPS 是 SUSE Linux 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。基於相容性的原因，LPRng 包括在 SUSE Linux 中。

印表機可藉由介面 (例如 USB 或網路) 和印表機語言加以區分。在購買印表機時，請確認硬體和適當的印表機語言可支援印表機的介面。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。此語言已經很舊且非常有效率。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 `hpijs` 驅動程式之外，目前沒有開發 Linux 驅動程式並提供 Linux 散發者使用的印表機製造商得到開放原始碼授權。這些印表機大部分屬於中等價位。

專用印表機 (通常為 GDI 印表機)

專用印表機通常僅可使用一或數個 Windows 驅動程式。這些印表機不支援任何常見印表機語言，它們所使用的印表機語言在發行新版的機型時會有變更。請參閱 節 31.7.1, "沒有標準印表機語言模式支援的印表機" [445] 取得更多資訊。

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://cdb.suse.de/--SUSE Linux> 印表機資料庫
- <http://www.linuxprinting.org/>—LinuxPrinting.org 印表機資料庫
- <http://www.cs.wisc.edu/~ghost/>—Ghostscript 網頁
- `/usr/share/doc/packages/ghostscript/catalog.devices`—包括的驅動程式

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，目前被評比為"完全支援"的印表機，在最新的 SUSE Linux 版本發行之後，可能就不在此狀態。因此，資料庫不一定能指出正確狀態，而僅提供估計值。

31.1 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)。

每個印表機皆存在一個專用的印表機佇列。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。

過濾器會將使用者想要列印的資料 (ASCII、PostScript、PDF、JPEG 等) 轉換成印表機特定的資料 (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。如果您是使用非 PostScript 印表機，過

濾器系統會將資料轉換為使用 Ghostscript 的印表機特定資料。這將需要印表機適用的 Ghostscript 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。

31.2 連線印表機的方法和通訊協定

有各種方法可將印表機連線到系統。CUPS 列印系統的組態無法辨識本地印表機和透過網路連線到系統的印表機。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線。CUPS 支援序列埠、USB、並列埠和 SCSI 連線。如需關於印表機連線的詳細資訊，請閱讀「支援資料庫」文章 *CUPS in a Nutshell*，網址為 <http://portal.suse.com>。可在搜尋對話方塊中輸入 *cups* 以找尋文章。

警告：纜線連線到機器

在將印表機連線到機器時，請不要忘記只有 USB 設備可在操作中插上和拔除。在變更其他連線之前，系統必須關閉。

31.3 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 SUSE Linux 安裝期間，會預先安裝許多 PPD 檔案，以便即使在沒有 PostScript 支援的情況下仍能使用印表機。

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 [節 31.6.3, "各種套件中的 PPD 檔案" \[442\]](#) 和 [節 31.7.2, "PostScript 印表機沒有可用的 PPD 檔案" \[445\]](#)。

新的 PPD 檔案可儲存在目錄 `/usr/share/cups/model/` 中，或以 YaST (請參閱 ["手動組態" \[435\]](#)) 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，此種安裝會導致遺失 SUSE Linux 所提供的支援，其次，印表機指令可能會

以不同方式運作，使系統將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。

31.4 設定印表機

在將印表機連線到電腦並安裝軟體之後，請在系統中安裝印表機。這應該使用 SUSE Linux 隨附的工具來執行。因為 SUSE Linux 非常強調安全性，協力廠商工具通常難以應付安全性限制，結果產生的麻煩多過好處。有關疑難排解的更多資訊，請參閱 [節 31.6.1, "CUPS 伺服器與防火牆" \[440\]](#) 和 [節 31.6.2, "CUPS 列印服務中的變更" \[441\]](#)。

31.4.1 本地印表機

在您登入時，如果偵測到無法辨識的本地印表機，YaST 會開始設定它。所用的對話方塊，與下列組態描述相同。

若要設定印表機，請選擇 YaST 控制中心裡的 **硬體** → **印表機**。這樣會開啟主要印表機的設定視窗，所偵測到的設備會列在視窗上方。下方會列出目前已設定的所有佇列。如果未偵測到您的印表機，請以手動方式設定。

重要

如果 **印表機** 項目在 YaST 控制中心內沒有提供，則最有可能是未安裝 `yast2-printer` 套件。若要解決此問題，請安裝 `yast2-printer` 套件，然後重新啟動 YaST。

自動組態

如果並列埠或 USB 埠可以自動設定，並偵測到連線的印表機，YaST 就能夠自動設定印表機。印表機資料庫還必須包含硬體自動偵測時 YaST 擷取的印表機 ID 字串。如果硬體 ID 與型號指定不同，請以手動方式選擇型號。

若要確保一切功能正常，每個組態都要用 YaST 的列印測試功能檢測。YaST 的測試頁也提供受測組態的重要資訊。

手動組態

如果不符合自動組態的需求，或是您要自訂安裝，請以手動方式設定印表機。視自動偵測的成功程度和在資料庫中找到的印表機型號資訊多寡，YaST 可能可以決定正確的自動設定，或至少做出合理的預先選擇。

下列參數必須設定：

硬體連線 (連接埠)

硬體連線的組態端視 YaST 能否在硬體自動偵測時找到印表機。如果 YaST 能夠自動偵測到印表機的型號，就可以假設印表機連線在硬體層次上工作正常，因此不需要變更這方面的設定。如果 YaST 無法自動偵測到印表機的型號，硬體層次就可能有些連線的問題。在此情況下，需要以手動方式設定連線。

在 *印表機組態* 對話方塊按下 *設定* 啟動手動設定工作流程。在這裡選取 *印表機類型* (例如 USB 印表機)，按下 *下一步*，輸入 *印表機連線* 並選取裝置。

佇列名稱

發出列印指令時會用到佇列名稱。這個名稱應該十分簡短，並且只能包含小寫字母和數字。在下個對話方塊輸入 *列印名稱* (*佇列名稱*)。

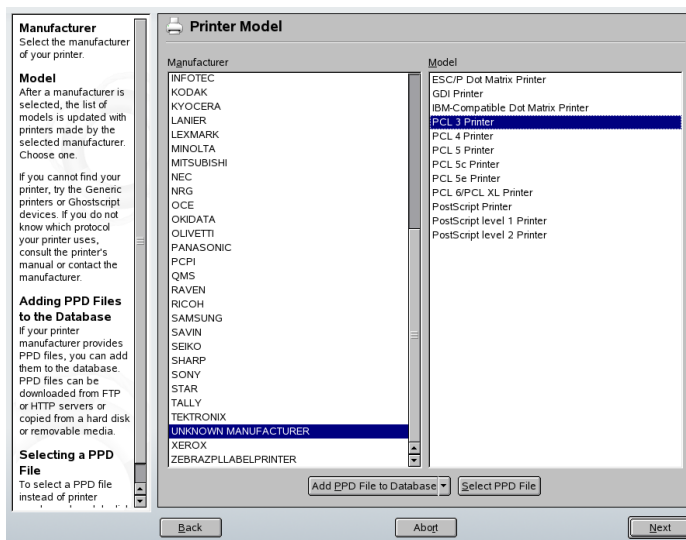
印表機型號和 PPD 檔案

所有印表機專用的參數，像是要使用的 Ghostscript 驅動程式和驅動程式的印表機篩選器參數，都儲存在 PPD (PostScript 印表機描述，PostScript Printer Description) 檔案中。若需更多 PPD 檔案的詳細資訊，請參閱 [節 31.3, "安裝軟體" \[433\]](#)。

很多印表機型號都有數種 PPD 檔案可用。例如，如果數種 Ghostscript 驅動程式都可搭配特定機種使用。您在下個對話方塊選擇一個製造商和一種型號時 (*印表機型號*)，YaST 會選擇與該印表機對應的 PPD 檔案。如果該型號有數種 PPD 檔案可用，YaST 會預設為其中一種 (通常標示為建議)。您可以在下個對話方塊使用 *編輯* 變更選取的 PPD 檔案。

至於非 PostScript 的型號，所有印表機專用的資料都由 Ghostscript 驅動程式產生。因此，印表機組態是決定輸出品質最重要的一項因素。所選擇之 Ghostscript 驅動程式 (PPD 檔案) 種類和為其指定的選項都會影響到列印成品。需要時可在選取 *編輯* 後，變更其他選項 (PPD 檔案中所具備者)。

圖形 31.1 選擇印表機型號



每次都要列印測試頁，來檢查您的設定是否如預期般有效。例如，如果輸出時出現問題，而且有幾頁幾乎空白，您應該移除所有紙張，然後停止 YaST 的測試，停止列印。

如果印表機資料庫不包括您的型號項目，您可以選取 *新增 PPD 檔案至資料庫* 以新增新 PPD 檔案，或使用一般 PPD 檔案集合，讓印表機使用一種標準印表機語言來工作。若要這麼做，請選取 *未知的廠商* 為您的印表機製造商。

進階設定

正常情況下，您不需要變更這些設定中的任何項目。

31.4.2 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。此處有一些關於這些通訊協定的詳細資訊：

socket

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線。經常使用的插槽連接埠號碼為 9100 或 35。設備 URI 的範例為
socket: //host-printer: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 服務的连接埠號碼為 515。設備 URI 的範例為 lpd: //host-printer/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 的连接埠號碼為 631。設備 URI 的範例為 ipp: //host-printer/ps 與
ipp: //host-cupsserver/printers/ps。

SMB (Windows 共用)

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

使用 YaST 設定網路中的 CUPS

應該使用 YaST 設定網路印表機。YaST 幫助建立組態，且是處理在 CUPS 安全性限制的最好工具 (請參閱 [節 31.6.2, "CUPS 列印服務中的變更" \[441\]](#))。如需在網路上安裝 CUPS 的說明，請參閱 <http://portal.suse.com> 上「支援資料庫」的文章 *Nutshell 中的 CUPS (CUPS in a Nutshell)*。

其他 (未偵測到) 和按一下 設定。如果沒有網路管理員的其他指示，請嘗試選項 直接列印至網路印表機 和根據本機需求處理。

以指令行工具設定

或者，也可以使用指令行工具 (例如，`lpadmin` 與 `lpoptions`) 設定 CUPS。您需要包含後端 (如 USB 和參數 `/dev/usb/lp0`) 的裝置 URI (Uniform Resource Locator，一致資源定址器)。例如，完整的 URI 可為 `parallel: /dev/lp0` (連線到第一並列埠的印表機) 或 `usb: /dev/usb/lp0` (第一個偵測到連線到 USB 埠的印表機)。

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。若要使用下列與法新增印表機佇列：

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

然後裝置 (`-v`) 將會變為可用 `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 檔案以及裝置名稱。

請勿使用 `-E` 做為第一選項。對於所有 CUPS 指令，第一個引數 `-E` 設定使用加密連線。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://192.168.1.0:9100/ -P \
/usr/share/cups/model/Postscript-levell.ppd.gz -E
```

有關 `lpadmin` 更多的選項，請參閱 `lpadmin(1)` 一文。

在系統安裝期間，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
```

前面的星號(*)表示預設選項已經啟動。

2 以 lpadmin 變更選項：

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

3 檢查新設定：

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

```
解析度 / 輸出解析度: 150dpi 300dpi *600dpi
```

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

31.5 應用程式的組態

應用程式與指令行工具一樣，都依賴現有印表機佇列。通常並不需要為了特別的應用程式而重新設定印表機，因為您應該可以使用既有的佇列，由應用程式進行列印。

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`。若要讓它與 KDE 程式一起作業，請啟用 *由外部程式列印*。否則請勿輸入列印指令。

工具如 xpp 和 KDE 程式 kprinter 提供圖形化介面，可從不同佇列選取並設定 PPD 檔案可使用 CUPS 標準選項和印表機特定選項。您可以把 kprinter 當做非 KDE 應用程式的標準列印介面，方法是在這些應用程式的列印對話方塊中指定 kprinter 或 kprinter--stdin 做為列印指令。應用程式自己的動作會決定選擇這兩個指令中的哪一個。如果設定正確，應用程式會在發出列印工作時呼叫 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。

31.6 SUSE Linux 的特殊功能

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

31.6.1 CUPS 伺服器與防火牆

有數種方式可將 CUPS 設為網路伺服器的用戶端。

1. 對於網路伺服器上的每一個佇列，您可以設定一個本地佇列，透過它將所有工作轉送給對應網路伺服器。通常不建議使用此方法，因為只要網路伺服器的組態變更，就必須重新設定所有用戶端機器。
2. 列印工作也可直接轉送給一台網路伺服器。對於此類型的組態，請勿執行本地的 CUPS 精靈。lp 或其他程式的對應程式庫呼叫可將工作直接傳送給網路伺服器。但是，如果您也想在本地印表機列印，則此組態無法運作。
3. CUPS 精靈可聆聽其他網路伺服器傳送來宣告可用佇列的 IPP 廣播封包。若要使用此方法，必須對內送封包開啟連接埠 631/UDP。

這是透過遠端 CUPS 伺服器列印的最佳 CUPS 組態。但是，這有一個風險，攻擊者使用佇列傳送 IPP 廣播，而本地精靈存取到偽造的佇列。如果這樣，則佇列會與本地伺服器上另一個佇列以相同名稱顯示，工作的擁有者會相信工作已傳送到本地伺服器，而事實上卻是傳送到攻擊者的伺服器。

YaST 可以掃描所有的網路主機以查看它們是否提供此服務，以及傾聽 IPP 廣播，以便找到 CUPS 伺服器。第二個方法也可在系統安裝期間用來尋找 CUPS 伺服器以提供建議。它需要為內送封包開啟 631/UDP 埠。使用第二種方法開啟

連接埠以設定遠端佇列的存取可能會造成安全上的風險，因為攻擊者有可能會廣播使用者可能接受的伺服器。

建議對話方塊中所顯示的防火牆預設值，是在任何介面上拒絕 IPP 廣播。因此，第二個偵測遠端佇列的方法和第三個存取遠端佇列的方法均沒有作用。所以，必須將一個介面標示為 `internal` (預設開啟連接埠) 或明確地開啟 `external` 介面的連接埠，來修改防火牆組態。為了安全，預設不會開放任何連接埠。

建議的防火牆組態必須修改為在安裝期間啟用 CUPS 以偵測遠端佇列，以及在一般操作期間從本地系統存取不同的遠端伺服器。或者，使用者可主動掃描本地網路主機或手動設定所有佇列，以偵測 CUPS 伺服器。但是，因為本節一開始提到的，不建議使用這個方法。

31.6.2 CUPS 列印服務中的變更

這些變更一開始是針對 SUSE Linux 9.1 所套用。

cupsd 以使用者 lp 執行

啟動時，`cupsd` 從使用者 `root` 變更為使用者 `lp`。如此可提供較高層級的安全性，因為 CUPS 列印服務不會以未受限制的權限執行，而只會以列印服務所需的權限執行。

不過，無法透過 `/etc/shadow` 執行驗證 (密碼檢查)，因為 `lp` 沒有 `/etc/shadow` 的存取權。而必須使用透過 `/etc/cups/passwd.md5` 的 CUPS 特定驗證。因此，具有 CUPS 管理群組 `sys` 和 CUPS 密碼的 CUPS 管理員必須在 `/etc/cups/passwd.md5` 中輸入。要這樣做，請以 `root` 身分輸入：

```
lppasswd -g sys -a CUPS-admin-name
```

如果要使用管理網頁前端 (CPUS) 或印表機管理工具 (KDE)，這個設定是必要的。

當 `cupsd` 以 `lp` 身分執行時，無法產生 `/etc/printcap`，因為 `lp` 不被允許在 `/etc/` 中建立檔案。因此，`cupsd` 會產生 `/etc/cups/printcap`。為了要確保僅可從 `/etc/printcap` 讀取佇列名稱的應用程式繼續正常運作，`/etc/printcap` 是指向 `/etc/cups/printcap` 的符號連結。

當 cupsd 以 lp 身分執行，無法開啟連接埠 631。因此，cupsd 無法以 rccups reload 重新載入。請使用 rccups restart。

BrowseAllow 和 BrowseDeny 的通用功能

BrowseAllow 和 BrowseDeny 設定的存取權限可套用於傳送給 cupsd 的所有類型套件。/etc/cups/cupsd.conf 中的預設設定值如下：

```
BrowseAllow @LOCAL
BrowseDeny All
```

以及

```
<Location />
  Order Deny, Allow
  Deny From All
  Allow From 127.0.0.1
  Allow From 127.0.0.2
  Allow From @LOCAL
</Location>
```

利用此方法，僅 LOCAL 主機可存取 CUPS 伺服器上的 cupsd。LOCAL 主機是指屬於非 PPP 介面的 IP 位址之主機 (是沒有設定 IFF_POINTOPOINT 旗標的介面)，而其 IP 位址和 CUPS 伺服器屬於相同的網路。來自其他伺服器的封包會立刻被拒絕。

預設會啟用 cupsd

在標準安裝中，會自動啟用 cupsd，以便不需其他手動作業即可存取 CUPS 網路伺服器的佇列。在 "[cupsd 以使用者 lp 執行](#)" [441] 與 "[BrowseAllow 和 BrowseDeny 的通用功能](#)" [442] 的項目是本功能的重要先決條件。如果沒有，則自動啟用 cupsd 的安全性將不足。

31.6.3 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 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。當您從廠商和型號清單中選取印表機時，會接收到相符廠商和型號的 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 `cups-drivers-stp` 套件中的 Gimp-Print 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。適合的 Foomatic PPD 檔案含有項目 `*NickName:... Foomatic /Ghostscript driver` 與 `*cupsFilter:... foomatic-rip`。這些 PPD 檔案位於 `cups-drivers` 套件中。

YaST 比較偏好 Foomatic PPD 檔案，如果 Foomatic PPD 檔案含有項目 `*NickName:... Foomatic ...` (建議) 符合印表機型號，而 `manufacturer-PPDs` 套件未包含更適合的 PPD 檔案。

cups-drivers-stp 套件中 Gimp-Print PPD 的檔案

除了 `foomatic-rip` 之外，來自 Gimp-Print 的 CUPS 過濾器 `rastertoprinter` 也可以用在許多非 PostScript 印表機上。此過濾器和適合的 Gimp-Print PPD 檔案可在 `cups-drivers-stp` 套件中找到。Gimp-Print PPD

檔案位於 `/usr/share/cups/model/stp/` 中，並含有項目 `*NickName:... CUPS+Gimp-Print` 與 `*cupsFilter:... rastertoprinter`。

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 印表機應該使用印表機製造商適當的 PPD 檔案設定，因為這個檔案可啟動 PostScript 印表機所有的功能。YaST 偏好 `manufacturer-PPDs` 套件的 PPD 檔案，如果符合下列條件：

- 在硬體偵測期間決定的廠商和型號符合 `manufacturer-PPDs` 套件中 PPD 檔案內的廠商和型號。
- `manufacturer-PPDs` 套件中的 PPD 檔案是唯一適合印表機型號的 PPD 檔案，或是含有也符合印表機型號的 `*NickName:... Foomatic/Postscript` (建議) 項目的 Foomatic PPD 檔案。

同時，在下列情況中，YaST 不使用任何來自 `manufacturer-PPDs` 套件的 PPD 檔案：

- 來自 `manufacturer-PPDs` 套件的 PPD 檔案不符合廠商和型號。如果 `manufacturer-PPDs` 套件對相似型號僅包含一個 PPD 檔案會發生此狀況，例如，型號系列中個別型號沒有各自的 PPD 檔案，但是在 PPD 檔案中以類似 `Funprinter 1000 series` 的格式指定型號名稱。
- 不建議使用該 Foomatic PostScript PPD 檔案。這可能是因為印表機型號無法在 PostScript 模式下有效率的操作，例如，此模式可能不信任該印表機，因為記憶體太少或印表機處理器太弱而使得速度太慢。此外，印表機可能預設不支援 PostScript，例如因為只有選用模組提供 PostScript。

如果來自 `manufacturer-PPDs` 套件的 PPD 檔案不適合 PostScript 印表機，但是 YaST 基於上述因素無法加以設定，請在 YaST 中手動選取對應印表機型號。

31.7 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。

31.7.1 沒有標準印表機語言模式支援的印表機

不支援任何常用印表機語言並僅能以特殊控制序列處理的印表機稱為 *GDI* 印表機。這些印表機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。實際問題不在於程式設計介面，而是 *GDI* 印表機僅可使用對應印表機型號的專用印表機語言處理。

部分印表機可切換到 *GDI* 模式或標準印表機語言來操作。部分製造廠商提供其 *GDI* 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。

31.7.2 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。然後使用 `cupstestppd` 公用程式檢查 PPD 檔案是否符合 "Adobe PostScript Printer Description File Format Specification 版本 4.3"。如果公用程式傳回 "FAIL"，表示 PPD 檔案的錯誤很嚴重，可能會導致重大問題。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。

31.7.3 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 address:378 (hexadecimal)
- Interrupt:irrelevant
- Mode:Normal, SPP, or Output Only
- DMA:disabled

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 31.1, "/etc/modprobe.conf: 第一並列埠的中斷模式" \[446\]](#) 中所示的項目啟用。在啟用中斷模式之前，請檢查檔案 `/proc/interrupts` 以了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 31.1 `/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
```

31.7.4 網路印表機連線方式

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 連線：

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

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

以使用者 root 的身分，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 31.2, "來自 lpd 的錯誤訊息" \[447\]](#) 中的回應，問題可能是因為遠端 lpd 所造成。

範例 31.2 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。

```
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 31.3, "來自 CUPS 網路伺服器的廣播" \[447\]](#) 中所示。

範例 31.3 來自 CUPS 網路伺服器的廣播

```
ipp: //host.domain: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 連線：

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

如果無法建立對 cupsd 的連線，則 cupsd 可能不是作用中或可能有基本的網路問題。假設對應的 cupsd 是作用中的，而且主機可以接受查詢，則 lpstat -h host -l -t 會傳回在 主機上所有佇列的狀態報告。

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 節 31.4.2, "網路印表機" [436]。

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線)。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 位址 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 來源連接埠-目的連接埠 IP 位址。這可能會花費一些時間。如需進一步資訊，請參閱 nmap man 頁面。

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。

31.7.5 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。

31.7.6 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。除去問題的起因之後，系統管理者必須以指令 `/usr/bin/enable` 重新啟動列印。

31.7.7 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 (也就是，傳送給印表機)，請使用像 `lpstat -h print-server -o` 這類的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 print-server queue-jobnumber
```

31.7.8 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 列印伺服器 -o` 檢查目前列印中的佇列。以 `cancel 佇列-工作編號` (或 `cancel -h 列印伺服器 佇列-工作編號`) 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，對於連接到並列埠的印表機，可使用指令 `fuser -k /dev/lp0` 來終止仍然在存取印表機 (更精確的說是並列埠) 的所有程序。
- 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。

31.7.9 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。

31.7.10 若需詳細資訊

許多特定問題的解決方法在「支援資料庫」中都有說明。若您碰到印表機的問題，請參閱支援資料庫內的文章 *安裝印表機 (Installing a Printer)* 和 *SUSE Linux 9.2 的印表機組態 (Printer Configuration from SUSE Linux 9.2)*。您用關鍵字印表機來搜尋即可找到。

熱插拔系統

熱插拔系統可以控制電腦中大部份設備的起始化。熱插拔系統不僅可用於作業期間插入和移除的設備，並可用於系統啟動時所偵測到的所有設備。它與 `sysfs` 檔案系統及 `udev` 密切配合，在 [章 33, 使用 `udev` 的動態裝置節點 \[457\]](#) 中描述了相關的資訊。

直到核心啟動時，只會起始必要的設備，例如匯流排系統、開機磁碟或鍵盤。核心會為所有偵測到的設備觸發熱插拔事件。`udev` 精靈會監聽這些事件，並執行 `udev` 來建立裝置節點和設定裝置。如果是無法自動偵測的裝置 (例如舊的 ISA 卡)，則會使用靜態組態。

除了少數的歷史特例，大部份的設備只要一提供存取，就會立即起始化，不論是在系統開機或熱插拔設備時。在起始化期間，會使用核心來註冊介面。此註冊將會接著觸發熱插拔事件，以對各自的介面進行自動設定。

在舊版的 SUSE Linux 中，使用組態資料的靜態集做為起始設備的基礎。所有熱插拔事件都是由名為代辦的不同程序檔來處理。在這個版本的 SUSE Linux 中，熱插拔子系統已整合至 `udev`，且由 `udev` 規則提供舊版熱插拔代辦的功能。

您可在 `/etc/sysconfig/hotplug` 中，找到熱插拔子系統的一般設定。所有變數已加上註解。一般設備組態是根據 `/etc/udev/rules.d` 中找到的相符規則來進行 (請參閱 [章 33, 使用 `udev` 的動態裝置節點 \[457\]](#))。特定設備的組態檔位於 `/etc/sysconfig/hardware`。舊版 SUSE Linux 使用的熱插拔事件回呼 `/proc/sys/kernel/hotplug` 通常是空的，因為 `udev` 會經由網路連結插槽收到熱插拔訊息。

32.1 設備和介面

熱插拔系統不僅會設定設備，也會設定介面。設備通常連接到匯流排，並提供介面所需的功能。介面代表設備的整個子集合或部份子集合的使用者能看到的大綱。設備通常需要設備驅動程式 (其形式為核心模組) 才能正確運作。此外，可能還需要某些高層次驅動程式，才能為使用者提供介面。介面大部份是由 `udev` 所建立的設備節點所代表。設備與介面的區別，對於瞭解整體概念很重要。

在 `sysfs` 檔案系統中所輸入的設備可以在 `/sys/devices` 之下找到。介面是位在 `/sys/class` 或 `/sys/block` 底下。`sysfs` 中的所有介面應該有一個所屬設備的連結。不過，還是有部份設備不會自動新增此連結。沒有該連結，就無法知道此介面屬於哪一個設備，而且將無法找到合適的組態。

設備會透過設備描述來處理。這可以是 `sysfs` 的設備路徑 (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0`)、連接點的描述 (`bus-pci-0000:02:00.0`)、個別 ID (`id-32311AE03FB82538`) 或其他類似的東西。在過去，介面是透過其名稱來定址。這些名稱代表現有設備的簡單編號，而且當新增或移除設備時有可能會變更。

介面也可以用相關設備的描述來定址。通常，內容會指出描述是指設備本身或它的介面。設備、介面和描述的典型範例包括：

PCI 網路卡

連接 PCI 匯流排的設備 (`/sys/devices/pci0000:00/0000:00:1e.0/0000:02:00.0` 或 `bus-pci-0000:02:00.0`)，且具有網路介面 (`eth0`, `id-00:0d:60:7f:0b:22` 或 `bus-pci-0000:02:00.0`)。網路介面是由網路服務所使用，或者網路介面會連線至虛擬網路設備，例如通道或 VLAN，因此會有介面。

PCI SCSI 控制器

設備 (`/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0` 或 `bus-scsi-1:0:0:0`)，以匯流排 (`/sys/class/scsi_host/host1`) 的形式，供多個實體設備介面使用。

SCSI 硬碟

設備 (`/sys/devices/pci0000:20/0000:20:01.1/host1/1:0:0:0` 或 `bus-scsi-1:0:0:0`)，包含多個介面 (`/sys/block/sda*`)。

32.2 熱插拔事件

每一個設備和介面都有一個關聯的熱插拔事件，它是由 `udev` 所處理。熱插拔事件是在建立或移除設備的連結，或是驅動程式註冊為介面或刪除介面時，由核心所觸發。從 SUSE Linux 9.3 開始，`udev` 會接收和處理熱插拔事件。`udev` 會直接監聽來自核心的網路連結訊息，否則就必須在 `/proc/sys/kernel/hotplug` 中指定 `/sbin/udevsend`。`udev` 可根據一組規則來設定設備 (請參閱 [章 33, 使用 `udev` 的動態裝置節點 \[457\]](#))。

32.3 熱插拔設備組態

自 SUSE Linux 10.0 起，已經棄用熱插拔代辦。現在，所有設備組態都應該透過 `udev` 規則來完成。`udev` 提供相容性規則來呼叫現有的自訂代辦。不過，您應該考量將自訂代辦轉換成 `udev` 規則。

熱插拔代辦是一個執行檔，執行事件合適的動作。設備事件的代辦程式位在 `/etc/hotplug.d/事件名稱與/etc/hotplug.d/default`。將會依字母順序執行這些以 `.hotplug` 為字尾目錄中的所有程式。

載入核心模組通常有助於進行設備組態。在部分情況下，必須呼叫其他指令才能正確進行設報組態。在 SUSE Linux 中，這通常是由 `udev` 規則來處理。不過，如果需要自訂設備組態，則會透過 `/sbin/hwup` 或 `/sbin/hwdown` 來完成設備組態。這些程式會在目錄 `/etc/sysconfig/hardware` 中，搜尋設備合適的組態，然後套用它。例如，若要防止起始特定設備，可利用適當的名稱建立組態檔，並將啟動模式設為手動或關閉。如果 `/sbin/hwup` 找不到任何組態，它就會尋找環境變數 `MODALIAS`。如果它存在，`modprobe` 就會自動載入相對應的模組。`MODALIAS` 變數是由核心熱插拔事件針對需要載入模組的設備而自動產生。如需更多資訊，請參閱 [節 32.4, "自動化模組載入" \[454\]](#)。如需關於 `/sbin/hwup` 的詳細資訊，請參閱檔案 `/usr/share/doc/packages/sysconfig/README` 以及 `man hwup` 的手冊頁面。

在呼叫介面代辦程式前，`udev` 通常會先產生一個系統可以存取的設備節點。`udev` 可以指定持久性名稱給介面。如需詳細資料，請參閱 [章 33, 使用 `udev` 的動態裝置節點 \[457\]](#)。然後就會根據個別的 `udev` 規則來設定介面本身。部份介面的程序敘述如下。

32.3.1 啟動網路介面

網路介面是使用 `/sbin/ifup` 起始，而使用 `/sbin/ifdown` 停止的。`/usr/share/doc/packages/sysconfig/README` 檔案以及 `ifup` `man` 頁面提供詳細資訊。

如果電腦有多個不同驅動程式的網路設備，則在系統啟動時，如果另一個驅動程式載入速度比較快，則可能變更介面的指定。SUSE Linux 嘗試使編號永久可用 — 設備可保留組態期間指定的介面名稱。這個指定動作是透過 `udev` 規則來完成的。稍後若要變更指定，則必須變更 `udev` 規則。

不過，最好的解決方案就是使用永久的介面指定。您可以在組態檔案中指定個別介面的名稱。如需此方法的詳細資訊，請參閱檔案 `/usr/share/doc/packages/sysconfig/README`。從 SUSE Linux 9.3 開始，`udev` 也處理網路介面，雖然這些都不是設備節點。這將使永久介面名稱的使用更加標準化。

32.3.2 啟動儲存設備

必須裝載儲存設備的介面，才能存取它們。此可以完全自動化或預先設定。此外，SUSE Linux 可區分系統和使用者設備。只有在 `/etc/fstab` 中建立項目，才能自動裝載系統設備。預設會透過 `hal` 來處理使用者設備。如果需要使用者設備的不同組態，就可以將這些設備輸入 `/etc/fstab`。此外，也可以修改在 `hal` 中處理此設備的方式。如需有關 `hal` 的詳細資訊，請參閱 `/usr/share/doc/packages/hal/hal-spec.html`。

建議使用持久性設備名稱，因為傳統設備名稱可能會因起始順序而變動。如需關於持久性設備名稱的詳細資訊，請參閱 [章 33, 使用 udev 的動態裝置節點\[457\]](#) 章節。

32.4 自動化模組載入

如果 `/sbin/hwup` 無法偵測組態檔，`modprobe` 就會根據環境變數 `MODALIAS` 的內容來搜尋相對應的模組。此環境變數是由相對應的熱插拔事件的核心所產生。若要使用非核心標準驅動程式的驅動程式，則應該在 `/etc/sysconfig/hardware` 中建立適當的硬體組態檔。

32.5 開機程序檔冷插拔

`boot.coldplug` 負責起始開機期間尚未設定的所有設備。它會為指定為 `/etc/sysconfig/hardware/hwcfg-static-*` 的每一個靜態設備組態，呼叫 `hwup`。之後它就會重播 `/lib/klibc/events` 儲存的所有事件，以起始所有設備。

32.6 錯誤分析

32.6.1 日誌檔

除非另有指定，`hotplug` 只會傳送一些重要訊息至 `syslog`。若要取得詳細資訊，請將檔案 `/etc/sysconfig/hotplug` 中的變數 `HOTPLUG_DEBUG`，設定成 `yes`。如果將此變數設定成值 `max`，會為所有熱插拔程序檔，記錄每一個外圍程序指令。這表示在 `/var/log/messages` 中，用來儲存所有訊息的 `syslog` 會變得相當大。因為在開機程序中，直到啟動 `hotplug` 和 `coldplug` 之後，才會啟動 `syslog`，因此不會記錄第一個訊息。如果這些訊息對於您很重要，請透過變數 `HOTPLUG_SYSLOG` 指定其他日誌檔。如需關於此主題的資訊，請參閱 `/etc/sysconfig/hotplug`。

32.6.2 開機問題

如果在開機程序期間電腦當機，在開機提示輸入 `NOHOTPLUG=yes` 或 `NOCOLDPLUG=yes`，可以關閉 `hotplug` 或 `coldplug`。因為熱插拔停用的緣故，核心不會發出任何熱插拔事件。在執行中的系統，您可以輸入指令 `/etc/init.d/boot.hotplugstart`，啟動熱插拔。所有在該時間產生的事件，會發出並予以處理。若要拒絕排入佇列的事件，可以先在 `/proc/sys/kernel/hotplug` 中輸入 `/bin/true`，以後再將項目重設成 `/sbin/hotplug`。因為停用冷插拔之故，所以不會套用靜態組態。要套用靜態組態，您可以稍後輸入 `/etc/init.d/boot.coldplug start`。

要查明 `hotplug` 載入的特殊模組是否發生問題，請在啟動提示輸入 `HOTPLUG_TRACE=<N>`。所有要載入的模組名稱，在實際載入的 N 秒後，會列示在螢幕上。進行此作業時，您無法介入。

32.6.3 事件記錄器

`udev` 規則會針對每個事件執行 `/sbin/hotplugeventrecorder` 程序檔。如果 `/events` 目錄存在，所有的熱插拔的事件都會在此目錄中儲存成個別檔案。因此，可以針對測試用途來重新產生事件。如果此目錄不存在，不會記錄任何資料。

使用 udev 的動態裝置節點

Linux 核心 2.6 為動態裝置目錄 `/dev` 引進新的使用者空間解決方案，並提供一致的裝置指定：`udev`。它僅提供實際出現的裝置檔案。它所建立或移除的裝置節點檔案通常是位於 `/dev` 目錄，而且它能夠重新命名網路介面。先前以動態 `/dev` 搭配 `devfs` 的實作已由 `udev` 取代。

傳統上，裝置節點會儲存在 Linux 系統上的 `/dev` 目錄中。每個裝置的可能類型都會有一個節點，無論其是否實際存在系統中。因此，此目錄包含了數千個未使用的檔案。必須先利用特殊應用程式來建立相對應的節點，才能使用新增的子系統或核心裝置。`devfs` 檔案系統帶來了一個重大的加強功能，因為現在只有當裝置真正存在且核心知道有這個裝置時，才會在 `/dev` 中給予一個裝置節點。

`udev` 引進一個新的建立裝置節點的方法。核心會將它的內部狀態匯出至 `sysfs`，而且每當核心識別到裝置時，它就會更新 `sysfs` 中的資訊，並將事件傳送到使用者空間。利用 `sysfs` 提供的資訊，`udev` 就可以比對簡單的規則語法與所提供的裝置屬性，然後建立或移除相對應的裝置節點。

使用者不需要為新裝置建立任何 `udev` 規則。如果連結了某個裝置，則會自動建立適當的裝置節點。不過，規則讓您得以定義裝置命名原則。這不僅讓您便於用易記的名稱來取代難以理解的裝置名稱，也便於讓已連接相同類型的兩個裝置能同時有一致的裝置名稱。

假設您有兩部透過 USB 連接的印表機，一部是高品質的彩色雷射印表機，另一部是黑白噴墨印表機。它們會顯示為 `/dev/usb/lpX`，其中的 X 是根據其連接的順序來編號。使用 `udev` 建立自訂 `udev` 規則，將一部印表機命名為 `/dev/colorlaser`，而另一部命名為 `/dev/inkprinter`。因為這些裝置節點是由

udev 根據裝置特性而建立，所以無論連線順序或狀態為何，這些裝置節點都一定會指向正確的裝置。

33.1 建立規則

在 udev 於 /dev 中建立裝置節點之前，會按照字母順序以字尾 .rules 讀取 /etc/udev/rules.d 中的所有檔案。會先使用符合裝置的第一個規則，即使其他規則也符合條件。井字符號 (#) 代表註解。規則採用以下格式：

```
key, [key,...] NAME [, SYMLINK]
```

至少必須指定一個關鍵字，因為規則被指定至裝置時，會以這些關鍵字為基礎。也必須指定一個名稱。建立在 /dev 中的裝置節點會使用此名稱。選擇性的符號連結參數讓節點可以建立在其他地方。印表機的規則可以採用以下格式：

```
BUS=="usb", SYSFS{serial}=="12345", NAME="lp_hp", SYMLINK+="printers/hp"
```

在此範例中，有兩個關鍵字，BUS 與 SYSFS{serial}。udev 會將此序號與連結到 USB 匯流排的裝置序號相比較。若要指定 lp_hp 名稱給 /dev 目錄中的裝置，所有的關鍵字都必須一致。此外，還會建立參照裝置節點的 /dev/printers/hp 符號連結。同時還會自動建立 printers 目錄。然後，列印工作便可以傳送到 /dev/printers/hp 或 /dev/lp_hp 中。

33.2 保留字元替換

NAME 和 SYMLINK 這兩個參數允許使用保留字元來替換特殊值。用一個簡單的範例來說明該程序：

```
BUS=="usb", SYSFS{vendor}=="abc", SYSFS{model}=="xyz", NAME="camera%n"
```

相機裝置的數字，例如 camera0 或 camera1，會替代 NAME 中的 %n 運算子。另一個有用的運算子是 %k，會被核心的標準裝置名稱所取代，例如 hda1。您也可以使用 udev 規則呼叫外部程式，並使用以 NAME 和 SYMLINK 值傳回的字串。udev 線上文件說明可能的保留字元的完整清單。

33.3 關鍵字的樣式比對

在 `udev` 規則的關鍵字中，可以使用萬用字元來尋找符合的內容。例如，`*` 字元就可以用來做為任何字元的保留字元，或 `?` 字元可以剛好做為一個字元的保留字元。

```
KERNEL="ts *", NAME="input/%k"
```

此規則會將標準目錄中的標準核心名稱指派給其指定的開頭字母為 `"ts"` 的裝置。如需關於在 `udev` 規則中使用樣式比對的詳細資訊，可以在線上文件 `udev` 中找到。

33.4 關鍵字選擇

為了要以唯一方式識別裝置並區分多個裝置，有效的 `udev` 規則必須有唯一的屬性。這裡有一些標準關鍵字的範例：

SUBSYSTEM

子系統，裝置是它的一部份

BUS

裝置匯流排類型

KERNEL

核心使用的裝置名稱

ID

匯流排上的裝置編號 (例如，PCI 匯流排 ID)

SYSFS{...}

`sysfs` 裝置屬性，例如標籤、廠商或序號

`SUBSYSTEM` 與 `ID` 這兩個關鍵字可能很有用，但常使用的卻是 `BUS`、`KERNEL` 及 `SYSFS{...}`。`udev` 組態也會提供呼叫外部程序碼的關鍵字，並評估其結果。請在 `udev` 線上文件中尋找關於此的詳細資訊。

`sysfs` 檔案系統會以目錄樹的方式，來提供硬體的相關資訊。每個檔案通常只包含一個項目的資訊，例如裝置名稱、廠商或序號。這些檔案每個都可以用來與關鍵字比對。不過，若要在一個規則中使用多個 `SYSFS` 關鍵字，您只能在相

同目錄中使用檔案做為關鍵字值。udevinfo 工具可以協助找到有用的唯一關鍵字值。

您必須找到一個參照相關裝置及包含 dev 檔案的 /sys 子目錄。這些目錄都位於 /sys/block 或 /sys/class 中。如果該裝置的裝置節點已經存在，udevinfo 可以為您找到正確的子目錄。udevinfo -q path -n /dev/sda 命令會輸出 /block/sda。這表示需要的目錄為 /sys/block/sda。現在用 udevinfo -a -p /sys/block/sda 指令來呼叫 udevinfo。也可以將兩個指令結合在一起，如 udevinfo -a -p `udevinfo -q path -n /dev/sda`。以下為輸出的摘要：

```
BUS=="scsi "  
ID=="0:0:0:0 "  
SYSFS{detach_state}=="0 "  
SYSFS{type}=="0 "  
SYSFS{max_sectors}=="240 "  
SYSFS{device_blocked}=="0 "  
SYSFS{queue_depth}=="1 "  
SYSFS{scsi_level}=="3 "  
SYSFS{vendor}==" "  
SYSFS{model}=="USB 2.0M DSC "  
SYSFS{rev}=="1.00 "  
SYSFS{online}=="1 "
```

請從輸出的資訊中，尋找不會變更的適合關鍵字。請記住，您無法在一個規則中使用來自不同目錄的關鍵字。

33.5 大容量儲存裝置的一致名稱

SUSE Linux 隨附預先定義的規則，能將相同的指定，指派至硬碟與其他儲存裝置，不論是要以何種順序來起始它們。利用 udev 隨附的小型協助程式，就可以讀取唯一的裝置屬性 (例如硬體序號、UUID 或檔案系統標籤)。協助程式可讓 udev 規則處理使用特殊的裝置資訊。在簡化範例中，第一個規則會匯入由 SCSI 裝置在 udev 環境中收集的值。第二個規則會使用已匯入的值來建立持續性的符號連結。

```
KERNEL="sd*[!0-9]", IMPORT="/sbin/scsi_id -g -x -s $p -d %N"  
KERNEL="sd*[!0-9]", SYMLINK+="${env{ID_TYPE}}/by-id/${env{ID_BUS}}-${env{ID_SERIAL}}"
```

當載入大容量儲存裝置的驅動程式後，它便會向核心註冊所有可用的硬碟。它們都會觸發呼叫 udev 的 hotplug 區塊事件。接著 udev 會讀取規則來判斷是否需要建立符號連結。

如果是透過 `initrd` 來載入驅動程式，則會遺失 `hotplug` 事件。不過，所有的資訊會儲存在 `sysfs` 中。`udevstart` 公用程式會在 `/sys/block` 與 `/sys/class` 中找到所有的裝置檔案，並啟動 `udev`。

還有一個 `boot.udev` 啟動程序檔，能在開機程序期間，重新建立所有的裝置節點。不過，啟動程序檔必須透過 `YaSTrunlevel` 編輯器來啟動，或用 `insserv boot.udev` 指令來啟動。

Linux 的檔案系統

Linux 支援多種不同的檔案系統。本章節簡要介紹最常用的 Linux 檔案系統、詳盡闡述它們的設計概念、優點和應用範圍。同時並提供 Linux 部份關於 LFS (大型檔案支援) 的資訊。

34.1 詞彙

中繼資料

一種檔案系統--可確保磁碟上的所有資料都能適當組織而且可以存取的內部資料結構。基本上，它是"關於資料的資料。"幾乎每一種檔案系統都有自己的中繼資料結構，這也是檔案系統顯示不同效能特性的部份原因。它對於維護中繼資料的完整極為重要，因為要不是如此，檔案系統上所有資料便無法存取。

inode

Inode 包含檔案的各種資訊，包括大小、連結數目、建立日期和時間、修改、存取，以及實際儲存檔案內容的磁碟區塊的指標。

日誌

在檔案系統的內容中，日誌是一種磁碟上的結構，包含檔案系統會在其中儲存對檔案系統的中繼資料進行何種變更的一種記錄。日誌處理大大降低 Linux 的復原時間，因為它放棄系統啟動時檢查整個檔案系統的冗長搜尋程序。相反地，只會 QA 日誌。

34.2 Linux 的主要檔案系統

不像兩、三年前，選擇 Linux 的檔案系統不再是幾秒鐘的事了 (Ext2 或 ReiserFS?)。從 2.4 開始的核心，提供各種檔案系統可供選擇。下面是這些檔案系統基本工作方式以及其優點的概觀。

世上沒有符合所有應用程式類型的檔案系統，這一點很重要，要牢記於心。每一種檔案系統都有自己特殊的優、缺點，必須考慮在內。即使最尖端的檔案系統，也無法取代合理的備份策略。

資料整合性和資料一致性這兩個詞彙，用在本章節時，不是指使用者空間資料的一致性 (應用程式寫入檔案中的資料)。此資料是否一致，必須由應用程式本身控制。

重要: 設定檔案系統

除非在本章節中提及，否則設定或變更分割以及檔案系統的一切步驟，都可以使用 YaST 模組來執行。

34.2.1 ReiserFS

ReiserFS 在 2.4 核心版本是正式的重要功能，而從 SUSE Linux 6.4 版的 2.2.x SUSE 核心的核心修補程式就已經可以使用。ReiserFS 是由 Hans Reiser 和 Namesys 開發小組設計。ReiserFS 已經證實是舊有 Ext2 的強大替代方案。它的關鍵價值是較佳的磁碟空間利用、較好的磁碟存取效能以及快速損毀復原。

詳細說明 ReiserFS 的長處，包括：

更佳的磁碟空間利用

在 ReiserFS，所有資料按照稱為 B^* -平衡樹的結構整理的。樹狀結構提供更佳的磁碟空間利用，因為小的檔案可以直接儲存在 B^* 樹葉節點，而不是儲存在別處，而且只維護實際磁碟位置的指標。此外，未以 1 或 4 kB 的區塊配置儲存體，而是按需要的正確大小。另一項優點則依賴 inode 的動態配置。這樣會使得檔案系統比傳統的檔案系統更有彈性，例如在 Ext2，inode 密度必須在檔案系統建立期間指定。

最佳的磁碟存取效能

至於小的檔案，檔案資料和「stat_data」(inode) 資訊兩者通常儲存在一起。它們可以使用單一磁碟 I/O 作業來讀取，表示只需要存取一次磁碟，便能擷取所有需要的資訊。

快速損毀復原

使用日誌來追蹤最新的中繼資料變更，只要幾秒便能檢查檔案系統，即使很大的檔案系統也沒問題。

透過資料日誌的可靠性

ReiserFS 也支援資料日誌，以及 ordered 資料模式，此模式類似 Ext3 章節中(節 34.2.3, "Ext3" [466]) 描述的概念。預設模式為 data=ordered 可以確保資料和中繼資料的完整性，但是日誌僅適用中繼資料。

34.2.2 Ext2

Ext2 的起源要回到 Linux 歷史的古早年代。它的前輩 - 延伸檔案系統，是在 1992 年 4 月落實並整合至 Linux 0.96c。延伸檔案系統已經過多次修改，而到了 Ext2，成為多年來最受歡迎的 Linux 檔案系統。有了日誌檔案系統的建立以及其驚人的快速回復時間，Ext2 就變得不那麼重要了。

簡短的 Ext2 功能摘要可以協助了解為什麼它 (以及在某些領域) 是很多 Linux 使用者最喜愛的 Linux 檔案系統。

穩固性

Ext2 經過多次改良和密集測試，已經算是「老前輩」了。這可能是為什麼人們通常稱它堅如磐石的原因。在檔案系統無法完全取消裝載而導致系統中斷後，e2fsck 會開始分析檔案系統資料。中繼資料會進入一致性狀態而且待處理的檔案或資料區塊會寫入指定的目錄(稱為 lost+found)。與日誌檔案系統相比，e2fsck 會分析整個檔案系統，不只是中繼資料最近修改的位元而已。這比檢查日誌檔案系統的記錄資料，要花費更多時間。按照檔案系統大小，此程序會花半小時或更長的時間。因此，不要為任何需要高可用性的伺服器選擇 Ext2。不過，因為 Ext2 不會維護日誌，而且使用相當少的記憶體，因此有時候比其他檔案系統較快速一些。

升級容易

Ext3 以 Ext2 的程式碼做為強大的基礎，因此可以成為眾人喝采的下一代檔案系統。它的可靠性和穩固性，巧妙地結合了日誌檔案系統的優點。

34.2.3 Ext3

Ext3 是由 Stephen Tweedie 設計。不像其他所有下一代檔案系統，Ext3 不依循全新的設計原則。它是以 Ext2 為基礎。這兩個檔案系統彼此關係十分密切。Ext3 檔案系統可以輕易地在 Ext2 檔案系統的最上層建立。Ext2 和 Ext3 最重要的差別是 Ext3 支援日誌處理。簡而言之，Ext3 提供三個主要優點：

可輕易從 Ext2 升級，並具有很高的可靠性

因為 Ext3 是以 Ext2 程式碼為基礎，而且共用它的磁碟格式以及中繼資料格式，所以從 Ext2 升級至 Ext3 十分簡單。不像轉換至其他日誌檔案系統 (例如 ReiserFS、JFS 或 XFS) 那麼冗長乏味 (備份整個檔案系統，然後從頭建立)，轉換至 Ext3 只是數分鐘的事。它也非常安全，因為重新建立整個檔案系統，不保證萬無一失。思考一下現有 Ext2 系統等候升級至日誌檔案系統的數量，您就可以輕易地發現為什麼 Ext3 對很多系統管理員具有一定的重要性。從 Ext3 降級至 Ext2 就和升級一樣容易。只要乾淨取消掛載 Ext3 檔案系統，然後重新掛載成 Ext2 檔案系統就可以了。

可靠性和效能

其他日誌檔案系統，有些會依照「僅中繼資料」日誌方法。這表示您的中繼資料永遠會保存在一致的狀態，但是同樣地無法自動保證檔案系統資料本身。Ext3 的設計是妥善管理中繼資料和資料二者。「管理」的程度可以自訂。在 `data=journal` 模式啟用 Ext3，可提供最大的安全性 (資料整合性)，不過因為中繼資料和資料是記錄為日誌，所以系統速度會減慢。較新的方法是使用 `data=ordered` 模式，可以確定資料和中繼資料整合性，不過限中繼資料使用日誌處理。檔案系統驅動程式會收集所有對應至某一中繼資料更新的所有資料區塊。更新中繼資料前，這些資料區塊會寫入硬碟。如此一來便可以達到中繼資料和資料的一致性，不會犧牲效能。第三個要使用的選項是 `data=writeback`，允許在其中繼資料已經提交至日誌後，將資料寫入主要檔案系統。一般認為此選項的效能最好。不過，它可以允許在損毀和復原舊資料後，重新顯示舊資料，同時又維護內部檔案系統整合性。除非您另有其他指定，否則 Ext3 是預設與 `data=ordered` 一起執行的。

34.2.4 將 Ext2 檔案系統轉換成 Ext3

從 Ext2 轉換成 Ext3，涉及兩個個別步驟：

建立日誌

以 `root` 登入，然後執行 `tune2fs -j`。這樣會以預設參數建立 Ext3 日誌。要由自己決定日誌大小以及它所在的位置，執行 `tune2fs -J` 取代想要的日誌選項 `size=` 和 `device=`。關於 `tune2fs` 程式的詳細資訊，可以從它的手冊頁面 (`tune2fs(8)`) 取得。

在 `/etc/fstab` 指定檔案系統類型

要確定 Ext3 檔案系統可以依此識別，請編輯檔案 `/etc/fstab`，將指定給相應分割區的檔案系統類型從 `ext2` 變更成 `ext3`。所做的變更會在下次啟動時生效。

根目錄使用 `ext3`

要啟動根目錄檔案設定為 Ext3 分割區，請在 `initrd` 中包括模組 `ext3` 和 `jbd`。要執行此步驟，請編輯檔案 `/etc/sysconfig/kernel`，將這將這兩個模組包含在 `INITRD_MODULES`，再執行指令 `mkinitrd`。

34.2.5 Reiser4

核心 2.6 發行後，有其他成員加入日誌檔案系統系列：**Reiser4**。Reiser4 與前一代產品 ReiserFS (版本 3.6) 全然不同。它導入調整檔案系統功能插件的概念，以及更詳細的安全性概念。

詳細的安全性概念

在設計 Reiser4 方面，它的開發人員強調安全性相關功能的實作。因此 Reiser4 會隨附一組專用的安全性插件。最重要的部分是導入檔案「項目」的概念。目前每個檔案都會定義檔案存取控制。如果大型檔案的資訊與多個使用者、群組或應用程式相關，則包含所有相關人員的存取權會相當不明確。在 Reiser4 中，您可以將那些檔案分成較小部分的（「項目」）。可以分別設定每個項目和每位使用者的存取權，如此可擁有更明確的檔案安全性管理。完整的例子會是 `/etc/passwd`。以前，只有 `root` 可以讀取和編輯檔案，而非 `root` 使用者僅有讀取該檔案的權限。使用 Reiser4 的項目概念，您可以將該檔案分成一組項目（每位使用者一個項目），讓使用者或應用程式修該他們自己的資料，但是不允許存取其他使用者的資料。此概念兼顧了安全性和彈性。

透過插件的擴充性

檔案系統會使用到的一般檔案系統功能和外部功能，Reiser 4 會以插件的方式執行。這些插件可以輕鬆地新增到基本系統。您不再需要重新編譯核心或重新格式化硬碟，就可以新增功能到您的檔案系統。

透過延緩的配置獲得較佳的檔案系統配置

Reiser4 和 XFS 一樣，支援延緩的配置。請參閱 [節 34.2.7, "XFS" \[468\]](#)。使用延緩的配置，就算是中繼資料，也可以產生較佳的整體配置。

34.2.6 JFS

JFS，是由 IBM 開發的 *日誌檔案系統*。JFS Linux 連接埠的第一個 beta 版本，在 2000 年夏天進入 Linux 社群。1.0.0 版在 2001 年問世。JFS 經過量身訂製後，可以符合以效能為最終目標的高運能伺服器的環境需求。JFS 是完整的 64 位元檔案系統，可支援大型檔案以及分割區，這是它用在伺服器環境的另一個原因。

仔細觀察 JFS 就知道為什麼這個檔案系統是您的 Linux 伺服器不錯的選擇：

有效率的日誌處理

JFS 依照「僅中繼資料」方法。不需要密集檢查，只會檢查由最近檔案系統活動所產生的中繼資料變更，節省大量的復原時間。需要多個並行記錄項目的並行作業，可以結合成一個群組，大大降低透過過多重寫入作業，所造成的檔案系統效能損失。

有效率的目錄組織

JFS 保存兩種不同的目錄組織。對於小型目錄，它允許目錄的內容直接儲存至它的 inode。對於大型目錄，它使用 B⁺樹，大大改善目錄管理。

透過動態 inode 配置取得較佳的空間利用

以 Ext2 為例，您必須事先定義 inode 密度 (管理資訊佔用的空間)，如此限制檔案系統的檔案或目錄數量上限。JFS 讓您不用擔心這些 -- 它動態配置 inode 空間，在不需要的時候會釋放 inode 空間。

34.2.7 XFS

1990 年代早期，SGI 開始對原先要當成 IRIX OS 的檔案系統 XFS 進行研發。XFS 背後的想法是建立高效能 64 位元日誌檔案系統，配合今日嚴格的計算挑戰。XFS 對於操控大型檔案以及執行高階硬體，具備良好功能。不過，XFS 還是有一個缺點。和 ReiserFS 一樣，XFS 在專注於中繼資料整合性，不重視資料整合性。

以下是 XFS 關鍵特性的快速回顧，說明為什麼它可在高階計算中，對其他日誌檔案系統顯現強大的競爭能力。

利用配置群組，取得高擴充性

建立 XFS 檔案系統時，檔案系統所屬的區塊裝置，會分割成 8 或更多等同大小的線性區域。這些稱為**配置群組**。每一個配置群組管理自己的 inode 以及可用的磁碟空間。事實上，配置群組可以看成是檔案系統中的檔案系統。因為配置群組彼此各自獨立，所以核心可以同時處理一個以上的配置群組。此特性是 XFS 具有優良擴充性的關鍵。當然，獨立配置群組的概念符合多處理器系統的需求。

透過有效率的磁碟空間管理取得高效能

可用空間和 inode 是由配置群組裡面的 B^+ 樹處理。使用 B^+ 樹可大大改良 XFS 的效能和擴充性。XFS 會使用**延緩的配置**。它會將程序分成兩個部分來處理配置。待處理的交易會儲存在 RAM 並保留適當的空間。XFS 仍然沒有決定資料到底要儲存在什麼地方 (提到檔案系統區塊的時候)。此決策會盡量延緩至最後時刻。部分暫時資料永遠不會儲存至磁碟，因為當決定 XFS 決定實際儲存位置時，它可能已經過時了。因此 XFS 會增加寫入效能並降低檔案系統零散化。因為延緩的配置會造成寫入事件比其他檔案系統要來得少，因此寫入情況較嚴重時，發生損毀之後，有可能會遺失資料。

預先配置來避免檔案系統零散化

寫入資料至檔案系統前，XFS 會**保留** (預先配置) 檔案需要的可用空間。因此，可大幅降低檔案系統零散化。因為檔案的內容是分佈在檔案系統中，所以效能就會提高。

34.3 其他支援的檔案系統

表格 34.1, "[Linux 的檔案系統類型](#)" [469] 彙整 Linux 支援的其他檔案系統。支援它們主要是確定不同媒體或外來作業系統中，資料交換的相容性。

表格 34.1 *Linux 的檔案系統類型*

<code>cramfs</code>	壓縮的 ROM 檔案系統：ROM 的壓縮唯讀檔案系統。
<code>hpf fs</code>	高效能檔案系統：IBM OS/2 標準檔案系統 -- 只在唯讀模式下支援。
<code>iso9660</code>	CD-ROM 的標準檔案系統。

minix	源自作業系統學術研究專案的檔案系統，是 Linux 使用的第一個檔案系統。現在，它是磁片的檔案系統。
msdos	<i>fat</i> 最早源自 DOS 的檔案系統，現在各種作業系統均使用之。
ncpfs	透過網路掛載 Novell 卷冊的檔案系統。
nfs	網路檔案系統：使用這種檔案系統，資料可以儲存在網路的任何機器，而且可以經由授權從網路存取。
smbfs	一些產品 (例如 Windows) 使用伺服器訊息區塊透過網路存取檔案。
sysv	用於 SCO UNIX、Xenix 和 Coherent (個人電腦的商用 UNIX 系統)。
ufs	用於 BSD、SunOS 和 NeXTstep。僅支援唯讀模式。
umsdos	<i>MSDOS 的 UNIX</i> ：套用在一般 <i>fat</i> 檔案系統之上，建立特殊檔案來達到 UNIX 功能 (許可權、連結、長檔案名稱)。
vfat	虛擬 <i>FAT</i> ： <i>fat</i> 檔案系統的副檔名 (支援長檔名)。
ntfs	<i>Windows NT</i> 檔案系統，唯讀。

34.4 Linux 的大型檔案支援

一開始，Linux 支援的檔案大小最多是 2 GB。在多媒體引爆之前，而且只要沒有人試著在 Linux 操控大型資料庫，這已經夠用了。當應用程式必須使用的一組新介面時，修改核心和 C 程式庫以支援超過 2 GB 的檔案大小，對於伺服器計算變得越來越重要。現在，幾乎所有主要檔案系統提供 LFS 支援，它讓您可以執行高階運算。[表格 34.2, "檔案系統的大小上限 \(磁碟上格式\)" \[471\]](#) 提供您 Linux 檔案和檔案系統目前限制的概觀。

表格 34.2 檔案系統的大小上限 (磁碟上格式)

檔案系統	檔案大小 (位元組)	檔案系統大小 (位元組)
Ext2 或 Ext3 (1 kB 區塊大小)	2^{34} (16 GB)	2^{41} (2 TB)
Ext2 或 Ext3 (2 kB 區塊大小)	2^{38} (256 GB)	2^{43} (8 TB)
Ext2 或 Ext3 (4 kB 區塊大小)	2^{41} (2 TB)	2^{44} (16 TB)
Ext2 或 Ext3 (8 kB 區塊大小) (含 8 kB 頁面的系統，例如 Alpha)	2^{46} (64 TB)	2^{45} (32 TB)
ReiserFS v3	2^{46} (64 GB)	2^{45} (32 TB)
XFS	2^{63} (8 EB)	2^{63} (8 EB)
JFS (512 位元組區塊大小)	2^{63} (8 EB)	2^{49} (512 TB)
JFS (4 kB 區塊大小)	2^{63} (8 EB)	2^{52} (4 PB)
NFSv2 (用戶端)	2^{31} (2 GB)	2^{63} (8 EB)
NFSv3 (用戶端)	2^{63} (8 EB)	2^{63} (8 EB)

重要: Linux 核心限制

表格 34.2, "檔案系統的大小上限 (磁碟上格式)" [471] 敘述磁碟上格式的限制。2.6 核心採用自己的檔案大小限制以及所處理的檔案系統大小限制。限制如下:

檔案大小

在 32 位元系統，檔案不得超過 2 TB (2^{41} 位元組)。

檔案系統大小

檔案系統最大可以達 2^{73} 位元組。不過，此限制仍然跟不上目前可用的硬體。

34.5 若需詳細資訊

上述的每一個檔案系統專案，會維護自己的首頁，在其中可以找出郵件清單資訊、進一步文件以及常見問題。

- <http://e2fsprogs.sourceforge.net/>
- <http://www.zipworld.com.au/~akpm/linux/ext3/>
- <http://www.namesys.com/>
- <http://oss.software.ibm.com/developerworks/opensource/jfs/>
- <http://oss.sgi.com/projects/xfs/>

關於 Linux 系統深入的多部份教學課程，可以在 *IBM developerWorks* 找到：

<http://www-106.ibm.com/developerworks/library/l-fs.html>。

關於 Linux 不同日誌檔案系統的比較，請參閱 Juan I. Santos Florido 在 *Linuxgazette* 的文章：<http://www.linuxgazette.com/issue55/florido.html>。

對於深入分析 Linux 的 LFS 感興趣的人，應該參閱 Andreas Jaeger 的 LFS 網站：

http://www.suse.de/~aj/linux_lfs.html。

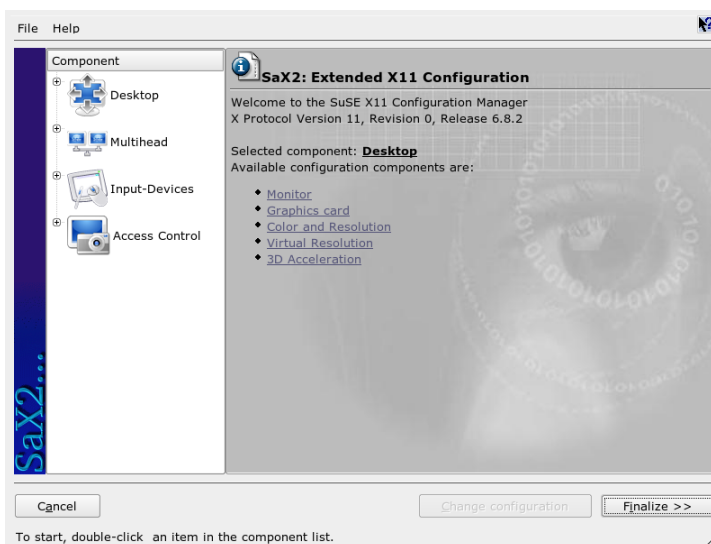
X Window System

X Window System (X11) 是 UNIX 中既成現實標準的圖形化使用者介面。X 採網路架構，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 或網際網路) 連線的其他主機上顯示。本章說明 X Window System 環境的設定和最佳化，提供有關在前端使用 SUSE Linux 的後端資訊和說明 OpenGL 和 3D 的組態。

35.1 使用 SaX2 來設定 X11

圖形使用者介面 (或 X 伺服器) 可處理軟硬體間的通訊。如 KDE 與 GNOME 等桌面以及許多視窗管理員，都可以使用 X 伺服器與使用者互動。圖形使用者介面會在安裝期間初次設定。稍後若要變更設定，請從 YaST 控制中心使用各自的模組，或從指令行以 `sax2` 指令手動執行 SaX2。SaX2 主視窗會為 YaST 控制中心中的個別模組提供一般防護。

圖形 35.1 SaX2 的主視窗



在左瀏覽列中，有六個項目，每個都分別顯示 YaST 控制中心中的組態對話方塊。請在 章 3, 使用 YaST 進行系統組態 (↑啟動) 中尋找下列提到的章節。

顯示器

如需顯示器與圖形介面卡的描述，請參閱 節 3.11.1, "圖形介面卡與顯示器內容" (章 3, 使用 YaST 進行系統組態, ↑啟動)。

滑鼠

如需圖形環境中的滑鼠組態描述，請參閱 節 3.11.2, "滑鼠內容" (章 3, 使用 YaST 進行系統組態, ↑啟動)。

鍵盤

如需圖形環境中的鍵盤組態描述，請參閱 節 3.11.3, "鍵盤內容" (章 3, 使用 YaST 進行系統組態, ↑啟動)。

圖形板

如需圖形板組態的描述，請參閱 節 3.11.4, "圖形板內容" (章 3, 使用 YaST 進行系統組態, ↑啟動)。

觸碰式螢幕

如需觸碰式螢幕組態的描述，請參閱 節 3.11.5, "觸碰式螢幕內容" (章 3, 使用 *YaST* 進行系統組態, ↑啟動)。

VNC

如需 VNC 組態的描述，請參閱 節 3.11.6, "遠端存取內容" (章 3, 使用 *YaST* 進行系統組態, ↑啟動)。

35.2 X 組態最佳化

X.Org 是一個由「開放原始碼」(Open Source) 實作的 X Window System。它是由 X.Org Foundation 進一步發展而來，該組織還負責開發 X Window System 的新技術與標準。

為了使可用的硬體，包括滑鼠、圖形卡、監視器以及鍵盤發揮最大效能，可以透過手動組態來達到最佳化。以下說明最佳化的一些層面。如需 X Window System 組態的詳細資訊，請參閱目錄 `/usr/share/doc/packages/Xorg` 中的各個檔案以及 `man xorg.conf`。

警告

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。錯誤的組態會對硬體造成無法挽回的損害 (固定頻率的監視器特別是如此)。本書的作者以及 SUSE Linux 無法對可能的損害負起任何責任。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。

SaX 和 `xorgconfig` 程式將建立 `xorg.conf` 檔，預設的存放位置是 `/etc/X11`。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。

下列段落描述組態檔 `/etc/X11/xorg.conf` 的結構。它由多個段落組成，每一個處理組態的某個層面。每一個段落以關鍵字 `Section` <指定> 開始，並以 `EndSection` 結束。段落包含下列格式：

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

提供的段落類型列出在 表格 35.1, "/etc/X11/xorg.conf 中的段落" [476] 中。

表格 35.1 */etc/X11/xorg.conf 中的段落*

類型	意義
Files (檔案)	這個段落描述字型與 RGB 顏色表的使用路徑。
ServerFlags (伺服器標誌)	一般切換設定於此。
InputDevice (輸入設備)	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 (磁碟機) 的重要參數以及定義 Protocol (通訊協定) 與 Device (設備) 的選項。
顯示器	描述使用的顯示器。這個段落的個別元素是名稱 (將在 Screen (螢幕) 定義中討論)、bandwidth (頻寬) 以及同步頻率限制 (HorizSync (水平同步) 和 VertRefresh (垂直更新))。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。這樣可以防止不小心將太高的頻率送往顯示器。
Modes (模式)	特定螢幕解析度的 modeline (模式行) 參數儲存於此。這些參數可以由 SaX2 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。請在 HOWTO 檔 /usr/share/doc/howto/en/XFree86-Video-Timings-HOWTO.gz 中找出每一個個別數值的意義之詳細資料。
Device (設備)	這個段落定義特定圖形卡。它是以本身描述名稱敘述。
Screen (螢幕)	這個段落將 Monitor (螢幕) 和 Device (裝置) 組合成 X.Org 的必需設定。在 Display (顯示) 子段落中, 指定虛擬畫面的大小 (Virtual (虛擬)), ViewPort (檢視埠) 和 Modes (模式) 可與該畫面一起使用。

類型	意義
ServerLayout (伺服器配置)	這個段落定義單一或多重顯示的組態。這個段落連結輸入裝置 InputDevice (輸入裝置) 和顯示設備 Screen (螢幕)。

Monitor (顯示器)、Device (設備) 和 Screen (螢幕) 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 (顯示器) 和 Device (設備) 段落。也可能有多個 Screen (螢幕) 段落。下列 ServerLayout (伺服器配置) 段落決定使用哪一個。

35.2.1 螢幕段落

首先，請仔細觀察螢幕段落，它由一個顯示器以及一個設備段落組成，以決定要使用的解析度與色彩深度。螢幕段落可能如同 [範例 35.1, "/etc/X11/xorg.conf 檔的螢幕段落" \[477\]](#) 一般。

範例 35.1 `/etc/X11/xorg.conf` 檔的螢幕段落

```
Section "Screen"
    DefaultDepth 16
    SubSection "Display"
        Depth 16
        Modes "1152x864" "1024x768" "800x600"
        Virtual 1152x864
    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 ]"
    Monitor "Monitor[ 0 ]"
EndSection
```

Identifier (識別碼) (此處為 Screen[0]) 行對這個段落提供一個定義的名稱，以便對下列 ServerLayout (伺服器配置) 段落提供唯一性參照。Device (設備) 和 Monitor (顯示器) 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 (設備) 以及 Monitor (顯示器) 段落連結。以下詳細討論這些段落。

如果伺服器未以特定的色彩深度啟動的話，請使用 DefaultDepth (預設深度) 設定來選取伺服器使用的色彩深度。每一個色彩深度有一個 Display (顯示器) 子段落。關鍵字 Depth (深度) 指定這個子段落的有效色彩深度。Depth (深度) 的可用值有 8、15、16 和 24。不是所有的 X 伺服器模組都支援所有這些值。

色彩深度之後，是一組 Modes (模式) 段落中設定的解析度清單。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 (模式) 段落搜尋適合的 Modeline (模式行)。Modeline (模式行) 同時由顯示器與圖形卡的功能決定。Monitor (顯示器) 設定值將決定 Modeline (模式行) 的結果。

第一個找到的解析度是 Default mode (預設模式)。按 **Ctrl** + **Alt** + **+** (數字鍵盤) 可以切換到清單中右邊的下一個解析度。按 **Ctrl** + **Alt** + **-** (數字鍵盤) 可以向左切換。您可以在 X 執行中變更解析度。

Display (顯示器) 子段落最後一行的 Depth 16 (深度 16) 指虛擬螢幕的大小。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。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 16 MB 視訊 RAM，虛擬螢幕可用佔用 4096x4096 像素，8 位元色彩深度。不過，特別是加速卡，不建議將所有記憶體用於虛擬螢幕，因為卡上的這個記憶體還要用於各種字型與圖形的快取處理。

35.2.2 設備段落

設備段落描述特定的圖形卡。xorg.conf 中的設備項目數沒有限制，但每一個設備的名稱要使用關鍵字 Identifier (識別碼) 來區別。如果有安裝超過一個以上的顯示卡，會遵照規則，以號碼來排序段落。第一個稱為 Device[0] (設備[0])，第二個稱為 Device[1] (設備[1])，以下類推。以下檔案顯示

一個使用 Matrox Millennium PCI 圖形卡的電腦，其 Device（設備）段落的例外情況：

```
Section "Device"
    BoardName      "MGA2064W"
    BusID          "0:19:0"
    Driver         "mga"
    Identifier     "Device[0]"
    VendorName     "Matrox"
    Option         "sw_cursor"
EndSection
```

如果您使用 SaX2 進行組態，設備段落看起來應該和上述範例類似。Driver（驅動程式）和 BusID（匯流排 ID）都由電腦中安裝的硬體決定，並由 SaX2 自動偵測。BusID（匯流排 ID）定義安裝圖形卡的 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。

您可以使用 Driver（驅動程式）參數來指定這個圖形卡使用的驅動程式。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，這是目錄 `/usr/X11R6/lib/modules/drivers`。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。設備段落中的選項 `sw_cursor` 便是一個範例。它可以關閉硬體滑鼠游標並描述使用軟體的滑鼠指標。視驅動程式模組而定，各有不同的各種選項，可以在 `/usr/X11R6/lib/X11/doc` 目錄內驅種程式模組的描述檔案中找到它們。通常，也可以在手冊頁找到有效的選項（`man xorg.conf` 和 `man X.Org`）。

35.2.3 顯示器和模式段落

和 Device（設備）段落類似，Monitor（顯示器）和 Modes（模式）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（顯示器）段落。伺服器配置段落指定相關的 Monitor（顯示器）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（顯示器）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。顯示器的內容，特別是容許頻率，儲存在 Monitor（顯示器）段落中。

警告

除非您對顯示器與圖形卡功能有深度的瞭解，否則請不要變更模式行，以免造成顯示器嚴重損害。

想自行開發顯示器描述的人員，應該非常熟悉 `/usr/X11/lib/X11/doc` 說明文件的內容。特別值得一提的是涵蓋視訊模式的段落。該段落詳細說明硬體功能以及如何建立模式行。

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 直接從顯示器讀取容許頻率以及最佳解析度值，如 SaX2 組態段落中的說明那樣。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。這樣幾乎所有的圖形卡與顯示器組合都可以運作。

35.3 安裝與設定字型

SUSE Linux 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 [節 35.3.2, "X11 核心字型" \[484\]](#))。要啟動字型的使用，它的安裝目錄必須是 `/etc/fonts/fonts.conf` 中設定的目錄的子目錄 (請參閱 [節 35.3.1, "Xft" \[481\]](#))。

字型檔案必須手動方式 (以 root 身份) 複製到適當的目錄，例如 `/usr/X11R6/lib/X11/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。然後，請執行 `SUSEconfig--module fonts`。

`SUSEconfig--module fonts` 會執程序檔 `/usr/sbin/fonts-config`，來處理字型的組態。如需這個程序檔的執行內容，請參閱程序檔的手冊頁 (`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以安裝至任何目錄。只有 CID-keyed 字型的程序稍有不同。相關資訊請參閱 [節 35.3.3, "CID-Keyed 字型" \[485\]](#)。

X.Org 含有兩種完全不同的字型系統：舊式 *X11* 核心字型系統和新設計的 *Xft* 與 *fontconfig* 系統。以下數節簡短描述這兩種系統。

35.3.1 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。多種語言文字的正確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，KDE 和 GNOME 二個桌面環境、Mozilla 以及其他許多應用程式已經預設使用 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 的內容是由全域組態檔 `/etc/fonts/fonts.conf` 和使用者專屬組態檔 `~/.fonts.conf` 控制。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` 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 (`:outline=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:outline=true" family style weight
```

這個指令的輸出看起來如下：

```
FreeSansBold.ttf: FreeSans:style=Bold:weight=200
FreeMonoBoldOblique.ttf: FreeMono:style=BoldOblique:weight=200
FreeSerif.ttf: FreeSerif:style=Medium:weight=80
FreeSerifBoldItalic.ttf: FreeSerif:style=BoldItalic:weight=200
FreeSansOblique.ttf: FreeSans:style=Oblique:weight=80
FreeSerifItalic.ttf: FreeSerif:style=Italic:weight=80
FreeMonoOblique.ttf: FreeMono:style=Oblique:weight=80
FreeMono.ttf: FreeMono:style=Medium:weight=80
FreeSans.ttf: FreeSans:style=Medium:weight=80
FreeSerifBold.ttf: FreeSerif:style=Bold:weight=200
FreeSansBoldOblique.ttf: FreeSans:style=BoldOblique:weight=200
FreeMonoBold.ttf: FreeMono:style=Bold:weight=200
```

可以使用 `fc-list` 來查詢的重要參數：

表格 35.2 *fc-list* 的參數

參數	意義及可用值
<code>family</code> (系列)	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)、Heavy (濃)。
<code>lang</code> (語言)	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。
<code>weight</code> (粗細)	字型粗細，例如，80 表示一般，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> (檔案)	字型檔案的名稱。
<code>outline</code> (外框)	true (真) 表示外框字型，false (偽) 表示其他字型。

參數	意義及可用值
<code>scalable</code> (縮放)	<code>true</code> (真) 表示縮放字型, <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, <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。

35.3.2 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, 以及 CID-keyed 字型。Unicode 字型的支援已經有一段時間。X11 核心字型系統最早於 1987 年針對 X11R1 的黑白點陣字型處理而開發。以上所述的擴充是在之後新增的。

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。Unicode 字型的處理速度較慢而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。它已經過時, 而且無法以現有方法擴充。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 `xset fp 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。最簡單的達成方法是輸入 `su` 與 `root` 密碼來取得 `root` 權限。`su` 可以將啟動 X 伺服器的使用者的存取權限轉移給 `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。

依預設，SUSE Linux 使用 UTF-8 語言環境。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 附隨的 Unicode 字型絕大部份都具有歐洲語言 (舊編號方式為 `iso-8859-*`) 所需的文字符號。

35.3.3 CID-Keyed 字型

和其他字型類型相反，您無法將 CID-keyed 字型安裝在任一目錄。CID-keyed 字型必須安裝在 `/usr/share/ghostscript/Resource/CIDFont`。這和 `Xft` 與 `fontconfig` 無關，但是對 Ghostscript 和 X11 核心字型系統卻是必要的。

提示

請參閱 <http://www.xfree86.org/current/fonts.html> 來取得 X11 的字型詳細資訊。

35.4 OpenGL—3D 組態

35.4.1 硬體支援

SUSE Linux 包括數個 3D 硬體支援程式的 OpenGL 驅動程式。[表格 35.3, "支援的 3D 硬體" \[485\]](#) 將會提供概觀介紹。

表格 35.3 支援的 3D 硬體

OpenGL 驅動程式	支援的硬體
nVidia	nVidia 晶片：除了 Riva 128(ZX) 以外的所有晶片

OpenGL 驅動程式 支援的硬體

DRI	3Dfx Voodoo Banshee, 3Dfx Voodoo-3/4/5, Intel i810/i815/i830M, Intel 845G/852GM/855GM/865G/915, Matrox G200/G400/G450/G550, ATI Rage 128(Pro)/Radeon (高達 9250)

如果您第一次安裝 YaST，而且 YaST 偵測到 3D 支援，3D 加速功能就可以在安裝時啟動。要使用 nVidia 繪圖晶片，必須先安裝 nVidia 驅動程式。要這樣做，請在 YOU (YaST Online Update) 中選取 nVidia 驅動程式。因為授權限制，nVidia 驅動程式未包含在配送中。

如果執行的是更新而不是全新安裝，或是需要設定 3Dfx 附加圖形介面卡 (Voodoo Graphics 或 Voodoo-2)，那麼設定 3D 硬體支援的程序是不一樣的。這要視使用的 OpenGL 驅動程式而定。詳細資訊請見下一章節：

35.4.2 OpenGL 驅動程式

OpenGL 驅動程式 nVidia 和 DRI 可以透過 SaX2 輕輕鬆鬆設定。要使用 nVidia 介面卡，必須先安裝 nVidia 驅動程式。輸入指令 `3Ddi ag` 以確認 nVidia 或 DRI 組態是否正確。

基於安全因素，只有屬於 `video` 群組的使用者才有權存取 3D 硬體。因此，請確定所有本地使用者皆是此群組的成員。否則，OpenGL 應用程式會使用速度緩慢的軟體轉譯後援 OpenGL 驅動程式。使用指令 `id` 檢查目前使用者是否屬於 `video` 群組。如果不是，請用 YaST 將使用者新增至群組。

35.4.3 診斷工具 3Ddiag

您可以使用診斷工具 3Ddiag 來驗證 SUSE Linux 的 3D 組態。這個指令行工具必須使用終端機啟動。輸入 `3Ddi ag -h` 以列出 3Ddiag 的可能選項。

要檢驗 X.Org 組態，工具會檢查是否已安裝 3D 支援所需要的套件，以及檢查是否使用正確的 OpenGL 程式庫和 GLX 擴充。如果您收到錯誤訊息，請依照診斷工具 3Ddiag 的指示進行操作。如果一切無誤，您只會在畫面上看到完成訊息。

35.4.4 OpenGL 測試公用程式

要測試 OpenGL，程式 `glxgears` 以及 `tuxracer` 和 `armagetron` 等遊戲 (相同名稱的套件) 十分有用。如果已啟動 3D 支援，在很新的電腦上應該可以玩的很順暢。沒有 3D 支援，這些遊戲執行速度會相當慢 (投影片效果)。請使用 `glxinfo` 指令來檢驗 3D 是否在作用中，如果在作用中，就會輸出一行字，其中包括 `direct rendering: YES`。

35.4.5 疑難排解

如果 OpenGL 3D 測試結果為否定 (遊戲無法流暢的運作)，請使用診斷工具 3Ddiag 確認組態沒有錯誤存在 (錯誤訊息)。如果更正這些錯誤仍然沒有用，或者一直沒有顯示失敗訊息，請查閱 X.Org 日誌檔。

您往往會在 X.Org 檔案 `/var/log/Xorg.0.log` 中發現 DRI 已停用這一行字。確實原因必須仔細檢查日誌檔才能找到，而且這項工作需要一些經驗才行。

在類似情況下，不會有任何組態錯誤，因為 3Ddiag 已經偵測過了。因此在此情況下，唯一選擇是使用 DRI 驅動程式的軟體轉譯後援軟體，該軟體不提供 3D 硬體支援功能。如果發生 OpenGL 表示錯誤或不穩定，您可以不用 3D 支援逕行操作。使用 `SaX2`，完全停用 3D 支援。

35.4.6 安裝支援

除了 DRI 驅動程式的軟體轉譯後援外，Linux 的所有 OpenGL 驅動程式仍在開發階段，因而尚屬實驗性質。因為對 Linux 的 3D 硬體加速需求量很高，所以配送中包括驅動程式。由於 OpenGL 驅動程式的實驗性質，所以 SUSE 無法為 3D 硬體加速的設定提供任何安裝支援，也無法對任何相關問題提供進一步的協助。圖形化使用者介面 (X Window 系統) 的基本組態不包括 3D 硬體加速組態。如果您遇到 3D 硬體加速問題，建議完全停用 3D 支援。

35.4.7 其他線上文件

如需關於 DRI 的資訊，請參閱 `/usr/X11R6/lib/X11/doc/README.DRI (xorg-x11-doc)`。可在 <http://ftp.suse.com/pub/suse/i386/supplementary/X/nvidia-installer-HOWTO.html> 找到 nvidia 驅動程式安裝相關資訊。

使用 PAM 驗證

Linux 在驗證程序中將 PAM (可插式驗證模組, Pluggable Authentication Modules) 當做使用者與應用程式之間的溝通層。PAM 模組適用於整個系統, 任何應用程式皆可要求。本章節說明模組驗證機制如何運作以及如何設定。

系統管理員與程式設計人員通常會想要限制系統某些部份的存取, 或是限制應用程式某些功能的使用。若未使用 PAM, 則每次引用新的驗證機制 (例如 LDAP 或 SAMBA) 時, 就必須調整應用程式。然而這個程序相當耗費時間, 而且容易產生錯誤。避免這些缺點的方法就是將應用程式與驗證機制區隔開來, 並將後者委託給集中管理的模組。如此一來每當需要新的驗證配置時, 就能夠調整或撰寫適當的 PAM 模組以供有問題的程式使用。

每個依賴 PAM 機制的程式都有自己的組態檔, 位置在 `/etc/pam.d/programname` 目錄。這些檔案是定義用來驗證的 PAM 模組。除此之外, 在 `/etc/security` 之下具有大部份 PAM 模組的全域組態檔, 它們定義出這些模組的精確行為 (這些範例包括 `pam_env.conf`、`pam_pwcheck.conf`、`pam_unix2.conf` 以及 `time.conf`)。每個應用程式使用 PAM 模組時, 其實就是呼叫一組 PAM 函數, 然後處理各種組態檔中的資訊, 並將結果傳回呼叫的應用程式。

36.1 PAM 組態檔的結構

在 PAM 組態檔中的每一行最多包含四個資料欄:

<模組類型> >控制旗標< 模組路徑> <選項>

PAM 模組是以堆疊的方式處理。不同模組的類型就有不同的目的，例如，一個模組檢查密碼，另一個驗證存取系統的位置，還有一個則讀取使用者特定的設定值。PAM 共有四種不同類型的模組：

auth

此類型模組的目的就是要檢查使用者的驗證性。傳統上是以查詢密碼的方式來檢查，但是這也可以使用晶片卡或透過生物測定 (指紋或虹彩掃描) 來完成。

account

此類型的模組會檢查使用者是否具有使用所要求服務的一般權限。舉例而言，執行這類檢查是要確保沒有人可以用使用者已過期的帳戶登入。

password

此類型模組的目的就是要能夠變更驗證權杖。在大部份情況下，這個權杖是密碼。

工作階段

此類型的模組是用來管理和設定使用者的工作階段。模組在驗證之前和之後啟動，以便在系統日誌中註冊登入嘗試，並設定使用者的特定環境 (郵件帳戶、主目錄、系統限制等等)。

第二個資料欄包含多種控制旗標，它們會影響已啟動模組的行為：

required

具有此旗標的模組必須成功地處理完成後，才能開始進行驗證。具有 `required` 旗標的模組失敗後，需待所有具有相同旗標的其他模組處理完畢，使用者才會收到關於驗證嘗試失敗的訊息。

requisite

具有此旗標的模組也必須成功地處理，方式與具有 `required` 旗標的模組相似。然而在失敗時，具有此旗標的模組會對使用者發出立即回應，而且不會再處理其他的模組。當成功時，就會接著處理其他的模組，像是任何具有 `required` 旗標的模組。`requisite` 旗標可以當做基本的過濾器，檢查是否具備正確驗證所必需的條件。

sufficient

具有此旗標的模組經成功處理後，呼叫應用程式會收到關於成功的立即訊息，而且不會再處理其他的模組 (假設具有 `required` 旗標的模組之前沒有

失敗)。具有 `sufficient` 旗標的模組若失敗並不會有直接的結果，這是因為任何後續的模組都是以個別的順序處理。

optional

具有此旗標的模組不論是成功或失敗都不會有任何直接的結果。這對於只想顯示訊息 (例如，通知使用者郵件已寄達)，但不想採取任何進一步動作的模組而言非常有用。

include

如果給予這個旗標，則指定為引數的檔案會在這個位置插入。

模組只要是位於 `/lib/security` 的預設目錄中即可，其路徑並不需要明確指定，(至於 SUSE Linux 所支援的所有 64 位元平台，目錄則為 `/lib64/security`)。第四個資料欄可能包含指定模組的選項，例如 `debug` (啟用除錯) 或 `nullok` (允許使用空密碼)。

36.2 sshd 的 PAM 組態

為了顯示 PAM 實際運作的理論，請將 `sshd` 的 PAM 組態當做實際範例：

範例 36.1 *sshd 的 PAM 組態*

```
##PAM-1.0
auth    include      common-auth
auth    required      pam_nologin.so
account include      common-account
password include      common-password
session include      common-session
# Enable the following line to get resmgr support for
# ssh sessions (see /usr/share/doc/packages/resmgr/README.SuSE)
#session optional    pam_resmgr.so fake_ttyname
```

應用程式的一般 PAM 組態 (在此例中為 `sshd`) 包含四個 `include` 陳述式，它們參照四個模組類型的組態檔：`common-auth`、`common-account`、`common-password` 以及 `common-session`。這四個檔案具有每個模組類型的預設組態。藉由包括它們而不是為每個 PAM 應用程式分別呼叫每個模組，這樣如果管理員變更預設值，就會自動取得更新的 PAM 組態。在以前，當 PAM 有變更或是安裝新應用程式時，必須為所有的應用程式手動調整所有的組態檔案。現在 PAM 組態是由中央的組態檔案組成，每個服務的 PAM 組態都會自動繼承所有的變更內容。

第一個 `include` 檔 (`common-auth`) 會呼叫 `auth` 類型的兩個模組：`pam_env` 與 `pam_unix2`。請參閱 [範例 36.2, "auth 區段的預設組態" \[492\]](#)。

範例 36.2 `auth` 區段的預設組態

```
auth    required    pam_env.so
auth    required    pam_unix2.so
```

第一個 `pam_env` 會載入 `/etc/security/pam_env.conf` 檔案，以根據此檔案的指定來設定環境變數。這可用來將 `DISPLAY` 變數設為正確的值，因為 `pam_env` 模組知道登入正在發生的位置。第一個模組 `pam_unix2` 會根據 `/etc/passwd` 與 `/etc/shadow` 檢查使用者的登入名稱與密碼。

在成功地呼叫 `common-auth` 中所指定的模組後，第三個模組會呼叫 `pam_nologin`，以檢查 `/etc/nologin` 是否存在。如果它存在，則除了 `root` 以外的使用者都不能登入。在處理完 `auth` 模組的整個堆疊後，`sshd` 才會收到關於登入是否成功的回應。假定堆疊的所有模組都有 `required` 控制旗標，則必須成功地處理完所有的模組後，`sshd` 才會收到成功結果的訊息。如果其中一個模組沒有成功，仍然會處理整個模組堆疊，並且會在此時通知 `sshd` 失敗的結果。

只要已成功處理 `auth` 類型的所有模組，就會處理另一個 `include` 陳述式，在此例為 [範例 36.3, "account 區段的預設組態" \[492\]](#) 中的陳述式。`common-account` 只包含一個模組：`pam_unix2`。如果 `pam_unix2` 傳回的結果是使用者存在，則 `sshd` 會收到此已成功的訊息，而且會接著處理下一個堆疊的模組 (`password`)，如 [範例 36.4, "password 區段的預設組態" \[492\]](#) 中所示。

範例 36.3 `account` 區段的預設組態

```
account required    pam_unix2.so
```

範例 36.4 `password` 區段的預設組態

```
password required    pam_pwcheck.so    nullok
password required    pam_unix2.so      nullok use_first_pass use_authtok
#password required    pam_make.so      /var/yp
```

此外，`sshd` 的 PAM 組態只有一個 `include` 陳述式參照 `password` 模組的預設組態，它是在 `common-password` 中。不論應用程式何時要求變更驗證權杖，這些模組皆必須順利完成 (控制旗標 `required`)。變更密碼或另一個驗證權杖時，必需做安全性檢查。這是以 `pam_pwcheck` 模組來達成。之後使用的 `pam_unix2` 模組會沿用任何 `pam_pwcheck` 的舊密碼或新密碼，因此使用者不必再進行驗證。這也使人無法規避 `pam_pwcheck` 所執行的檢查。當 `account` 或 `auth` 類

型前面的模組是設定成會針對過期密碼發出警告時，就應該使用 `password` 類型的模組。

範例 36.5 `session` 區段的預設組態

```
session required      pam_limits.so
session required      pam_unix2.so
```

最後一個步驟會呼叫與 `common-session` 檔案繫結在一起的 `session` 類型的模組，以便根據有問題的使用者之設定值來設定工作階段。雖然會已再次處理 `pam_unix2`，但是由於在此模組 `pam_unix2.conf` 的個別組態中指定了 `none` 選項，所以不會有實際的結果。`pam_limits` 模組會載入 `/etc/security/limits.conf` 檔案，它可以定義某些系統資源的使用限制。當使用者登出時，會第二次呼叫 `session` 模組。

36.3 PAM 模組的組態

有些 PAM 模組是可設定的。對應的組態檔位於 `/etc/security`。本小節簡單說明與 `sshd` 範例相關的組態檔——`pam_unix2.conf`、`pam_env.conf`、`pam_pwcheck.conf` 以及 `limits.conf`。

36.3.1 `pam_unix2.conf`

傳統的密碼驗證方法是由 PAM 模組 `pam_unix2` 所控制。它可以從 `/etc/passwd`、`/etc/shadow`、NIS 對應、NIS+ 表格、或從 LDAP 資料庫讀取所需的資料。設定個別應用程式本身的 PAM 選項或編輯 `/etc/security/pam_unix2.conf` 以進行全域設定，即可影響此模組的行為。[範例 36.6](#), "[pam_unix2.conf](#)" [493] 中顯示了模組最基本組態檔。

範例 36.6 `pam_unix2.conf`

```
auth:      nullok
account:
password:      nullok
session:      none
```

`auth` 與 `password` 模組類型的 `nullok` 選項是指定對應的帳戶類型可以使用空密碼。也允許使用者變更帳戶密碼。`session` 模組類型的 `none` 選項是指定

其本身不記錄訊息 (這是預設值)。如需其他組態選項的資訊，請參閱檔案本身的註解以及 `pam_unix2(8)` 的手冊頁。

36.3.2 `pam_env.conf`

這個檔案可用來定義使用者的標準環境，每當呼叫 `pam_env` 模組時就會設定該環境。使用它即可使用下列語法預設環境變數：

```
VARIABLE [DEFAULT=[value]] [OVERRIDE=[value]]
```

VARIABLE

要設定的環境變數名稱。

[DEFAULT=[value]]

管理員想要設定的預設值。

[OVERRIDE=[value]]

可以由 `pam_env` 查詢及設定的值，用以覆寫預設值。

每次執行遠端登入時調整的 `DISPLAY` 變數，就是使用 `pam_env` 的典型範例。這會顯示於 [範例 36.7, "pam_env.conf" \[494\]](#)。

範例 36.7 `pam_env.conf`

```
REMOTEHOST      DEFAULT=localhost OVERRIDE=@{PAM_RHOST}
DISPLAY         DEFAULT=${REMOTEHOST}:0.0 OVERRIDE=${DISPLAY}
```

第一行將 `REMOTEHOST` 變數的值設為 `localhost`，每當 `pam_env` 無法決定任何其他值時，就會使用這個值。`DISPLAY` 變數接著就會包含 `REMOTEHOST` 的值。如需詳細資訊，可以從 `/etc/security/pam_env.conf` 檔案中的註解取得。

36.3.3 `pam_pwcheck.conf`

此組態檔是供 `pam_pwcheck` 模組使用，該模組會從該檔案讀取所有 `password` 類型模組的選項。此檔案所儲存設定值的優先順序，高於個別應用程式的 `PAM` 設定值。如果尚未定義應用程式特定的設定值，應用程式就會使用全域設定值。

[範例 36.8, "pam_pwcheck.conf" \[495\]](#) 說明 `pam_pwcheck` 允許空的密碼和密碼的

修改。模組的更多選項，請見 `/etc/security/pam_pwcheck.conf` 檔案中的說明。

範例 36.8 `pam_pwcheck.conf`

```
password: nullok
```

36.3.4 `limits.conf`

系統限制可以以使用者或群組為單位，在 `limits.conf` 檔案中做設定，該檔案是由 `pam_limits` 模組讀取。該檔案允許您設定硬性限制，也就是完全不能超過，也可設定軟性限制，也就是可以暫時超過。若要瞭解關於語法以及可用選項的資訊，請參閱檔案中所包括的註解。

36.4 若需詳細資訊

在已安裝系統的 `/usr/share/doc/packages/pam` 目錄中，可以找到下列的其他文件：

README

在此目錄的最上層中，有一些一般性的 README 檔案。`modules` 子目錄中存有關於可用 PAM 模組的 README 檔案。

Linux-PAM 系統管理員指南

此文件包括系統管理員就 PAM 所應該知道的每件事。它討論很多主題，涵蓋組態檔的語法至 PAM 的安全性層面。該文件有三種格式：PDF 檔案、HTML 格式以及純文字。

Linux-PAM 模組撰寫者手冊

此文件是以開發人員的觀點將主題做成摘要，內含如何撰寫標準相容 PAM 模組的資訊。它具有三種格式：PDF 檔案、HTML 格式以及純文字。

Linux-PAM 應用程式開發人員指南

想要使用 PAM 程式庫的應用程式開發人員所需的全部內容都包括在此文件中。它具有三種格式：PDF 檔案、HTML 格式以及純文字。

Thorsten Kukuk 已開發一些適用於 SUSE Linux 的 PAM 模組，並提供一些可用的相關資訊，網址為：<http://www.suse.de/~kukuk/pam/>。

使用 Xen 的圖形化系統

Xen 可以在一部實體電腦上執行多個 Linux 系統。不同系統的硬體會以虛擬模式提供。本章將提供這項技術的可能性和限制性的說明。整個介紹會包括安裝、設定和執行 Xen。

虛擬機器通常需要模擬系統所需的硬體。缺點是模擬硬體會比實際硬體跑得慢。Xen 的實作方法不同。它限制模擬元件的數量到最小。若要達成這個目標，Xen 使用超虛擬。這項技術和虛擬機器很像，但是基本的硬體不同。所以，主機和外來作業系統會適用核心層級。使用者空間保持不變。Xen 使用 Hypervisor 和控制外來系統 (同時也稱作 domain-0) 控制硬體。這些提供虛擬區塊和網路裝置的所有需要。外來系統會使用這些虛擬區塊和網路裝置執行系統，且會連結到其他外來系統或本地網路。執行 Xen 的多個實體電腦會被設定成可用虛擬區塊和網路裝置，所以也可在執行時，將外來系統從一個硬體轉移到另一個。Xen 原來的開發目的是在一台電腦上執行最多 100 個外來系統，但是這個數量會根據執行外來系統的系統需求 (尤其是記憶體消耗) 而不同。

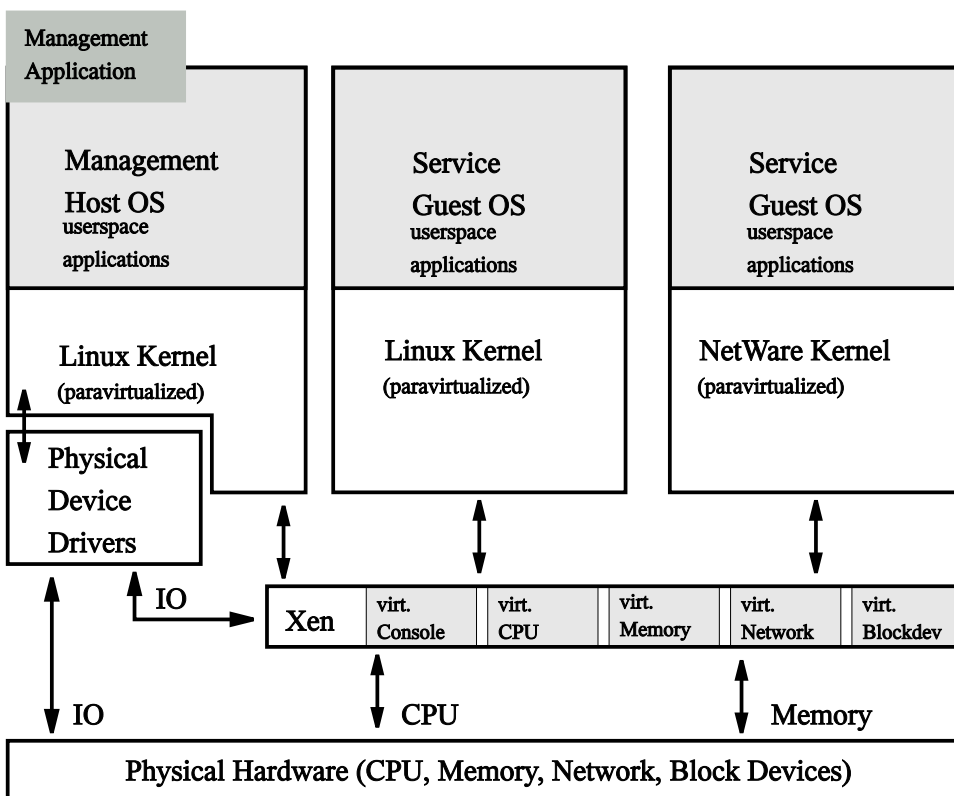
若要限制 CPU 的使用，Xen Hypervisor 提供三種不同的排程器。您也可在執行外來系統時，變更排程器，以變更執行中外來系統的優先順序。更高層級的做法，就是轉移外來系統，使其也可用來調整可用 CPU 電力。

Xen 圖形化系統對於支援的硬體也有某些缺點：

- 多個閉鎖來源驅動程式，如 Nvidia 或 ATI 無法正常運作。這些案例中，如果可能，您必須使用開放原始碼驅動程式，即使他們無法支援晶片的完整功能。使用 Xen 時，也無法支援某些 WLAN 晶片和 CardBus 橋接器。
- 在第二版，Xen 不支援 PAE (實體位址延伸)，這代表它不支援超過 4 GB 的記憶體。

- 它也不支援 ACPI。仰賴 ACPI 的電源管理和其他模式就無法正常運作。

圖形 37.1 Xen 概觀



37.1 Xen 安裝

安裝 Xen 的過程包含設定 domain-0 領域和安裝 Xen 用戶端。首先，請確認已安裝需要的套件。這些是 `python`、`bridge-utils`、`xen` 和 `kernel-xen` 套件。使用 SUSE 套件時，Xen 會被新增到 GRUB 組態。其他的案例中，在 `boot/grub/menu.lst` 建立項目。項目必須和下列類似：

```
title Xen2
    kernel (hd0,0)/boot/xen.gz dom0_mem=458752
    module (hd0,0)/boot/vmlinuz-xen <parameters>
    module (hd0,0)/boot/initrd-xen
```

使用包含 `/boot` 目錄的分割區取代 `(hd0,0)`。請參閱 [章 29, 開機載入器 \[399\]](#)。變更 `dom0 mem` 的數量以符合您的系統。系統記憶體的最大值是 -65536kB。使用正常啟動的 Linux 核心取代 `<parameters>`。然後重新以 Xen 模式開機。這將會以 Xen Hypervisor 啟動，並小幅變更 Linux 核心為執行多數硬體的 Domain-0。除了已提過的例外，其他應該都可正常運作。

37.2 領域安裝

外來領域安裝和設定包含下列幾個步驟。接下來，會安裝第一個外來領域，以完成建立第一個網路連結的所有不同任務。

若要安裝外來系統，您必須提供區塊裝置的 `root` 檔案系統或需要設定的檔案系統映像檔。若要稍後存取系統，使用虛擬主控台或設定外來系統的網路連接。安裝 SUSE Linux 到 YaST 支援的目錄。此外來系統的硬體需求和正常的 Linux 安裝類似。

領域可共享從其他領域唯讀安裝的檔案系統，如 `/usr` 或 `/opt`。不要共享寫入讀取的檔案系統。要與許多外來領域共享可寫入資料，可使用 NFS 或其他網路或叢集檔案系統。

警告: 啟動外來領域

啟動外來領域時，請確認安裝程式或控制 `domain-0` 不會安裝外來系統的檔案系統。

第一件事是建立檔案系統映像檔，是 Linux 外來系統安裝的位置。

- 1 若要建立名為 `guest1` 的空映像檔，請在目錄 `/var/tmp/` (4 GB 大) 中，使用下列指令：

```
dd if=/dev/zero of=/var/tmp/guest1 seek=1M bs=4096 count=1
```

- 2 該映像檔為一個大的、空的、沒有任何資訊的檔案。若要將檔案寫入到映像檔，需要檔案系統：

```
mkreiserfs -f /var/tmp/guest1
```

指令 `mkreiserfs` 會通知您這不是區塊特殊裝置，並要求確認。輸入 `Y` 然後按一下 `[Enter]` 繼續。

- 3 會在目錄執行實際的安裝。所以必須安裝檔案系統映像檔 `/var/tmp/guest1` 到目錄。

```
mkdir -p /var/tmp/dirinstall
mount -o loop /var/tmp/guest1 /var/tmp/dirinstall
```

重要

當您完成安裝後，再解安裝此檔案系統映像檔。安裝時，YaST 同時會安裝 `/proc` 檔案系統，所以也必須解安裝：

```
umount /var/tmp/dirinstall/proc
umount /var/tmp/dirinstall
```

37.2.1 使用 YaST 安裝外來領域

若要使用 YaST 安裝外來領域，您需要先準備新的外來領域的檔案系統映像檔。啟動 YaST 然後選擇 軟體 → 安裝至 Xen 的目錄。

目錄安裝的 YaST 模組有多個選項，請按照自己的需求設定：

- 目標目錄： `/var/tmp/dirinstall`

設定該選項到檔案系統映像檔的安裝點以供使用。預設值為接受。

- 第一次開機時執行 YaST 和 SUSEconfig： 是

設定該選項為 是。第一次啟動外來領域時，將會要求您提供 root 密碼和初次使用者。

- 建立映像檔： 否

這次建立的映像檔只是安裝目錄的 Tar 歸檔。這並不是很實用。

- 軟體

選取要使用的安裝類型。任何預設值應該會是好的開始。

按一下 下一步 開始安裝。根據套件的數量，安裝可能會花費一段時間。完成安裝後，必須移走 `tls` 程式庫：

```
mv /var/tmp/dirinstall/lib/tls /var/tmp/dirinstall/lib/tls.disabled
```


Xen使用 domain-0 其中一個核心，以啟動外來領域。若要在外來領域使用網路，必須讓外來領域可使用該核心模組。

```
cp -a /lib/modules /$(rpm -qf --qf %{VERSION}-%{RELEASE}-xen \
    /boot/vmlinuz-xen) /var/tmp/dirinstall/lib/modules
```

若要避免檔案系統錯誤，安裝後，必須解安裝檔案系統映像檔：

```
umount /var/tmp/dirinstall/proc
umount /var/tmp/dirinstall/
```

同時可能會建立 domain-0 和外來系統的特殊核心。主要的不同是在外來系統不需要硬體驅動程式。因為這些驅動程式為模組，不會在外來系統使用，SUSE 只傳送兩個任務的一個核心。

37.2.2 設定救援系統為外來領域

要快速取得執行中系統的快速方法是，重複使用已存在 root 檔案系統，如 SUSE Linux 救援系統。基本上，交換虛擬區塊的核心映像檔/裝置驅動程式和此映像檔中的網路裝置。若要快速完成任務，可至 /usr/share/doc/packages/xen/ 取得程序檔 mk-xen-rescue-img.sh。

使用救援方法建立 root 檔案系統的缺點是，不會有 RPM 資料庫，所以您將無法輕易的使用 RPM 新增套件。另一方面，檔案會很小，但包含大部分啟動網路所需的物件。

若要執行程序檔 mk-xen-rescue-img.sh，您至少需要包含救援映像檔的目錄和放置目標映像檔的目標位置。按照預設值，目錄會放置在開機 DVD 中的 /boot。

```
cd /usr/share/doc/packages/xen
./mk-xen-rescue-img.sh /media/dvd/boot /usr/local/xen 64
```

程序檔的第一個參數是救援映像檔的目錄。第二個參數是映像檔的目標位置。可選參數為新建外來領域的磁碟空間需求和要使用的核心版本。

然後程序檔會複製映像檔到新的位置，取代核心和多個核心模組，並停用系統的 t1s 目錄。最後一步，會在 /etc/xen/ 產生新的映像檔的組態檔。

37.3 設定 Xen 外來領域

如何設定外來領域的文件並不龐大。如何設定領域的大部分資訊可在範例組態檔 `/etc/xen/config` 找到。需要的選項會與預設值一起討論，或在範例組態中討論。要取得 [節 37.2.1, "使用 YaST 安裝外來領域" \[500\]](#) 詳細的安裝資訊，使用下列內容建立檔案 `/etc/xen/guest1`：

```
kernel = "/boot/vmlinuz-xen" ❶
ramdisk = "/boot/initrd-xen" ❷
memory = 128 ❸
name = "guest1" ❹
nics = "1" ❺
vif = [ 'mac=aa:cc:00:00:00:ab, bridge=xen-br0' ] ❻
disk = [ 'file: /var/tmp/guest1,hda1,w' ] ❼
root = "/dev/hda1 ro" ❽
extra = "3" ❾
```

- ❶ 輸入 `domain-0` 中到 Xen 核心的路徑。核心會稍後在外來系統執行。
- ❷ 選取包含 Xen 核心裝置驅動程式的初始 RAM 磁碟。沒有這個，通常無法正常執行核心，因為無法安裝 root 檔案系統。
- ❸ 定義外來系統應分享的記憶體量。如果系統的記憶體不足以供外來系統，則會失敗。
- ❹ 外來系統的名稱。
- ❺ 外來領域的虛擬網路介面數量。
- ❻ 虛擬網路介面組態，包括 MAC 位置 and 要連接的橋接器。
- ❼ 設定 Xen 外來領域可使用的虛擬區塊裝置。要使用實體區塊裝置，建立如 `[ 'phy:sdb1,hda1,w', 'phy:system/swap1,hda2,w' ]` 的項目。
- ❽ 設定核心的 root 裝置。對外來領域而言必須為虛擬裝置。
- ❾ 在此新增額外的核心參數。範例 3 表示外來領域會在 `runlevel 3` 啟動。

37.4 啟動和控制 Xen 領域

啟動外來領域前，Xen Hypervisor 必須有足夠的記憶體供新的外來領域使用。首先，檢查使用的記憶體數量：

```

xm list
Name                Id  Mem(MB)  CPU  State  Time(s)  Console
Domain-0            0    458     0   r----   181.8

```

如果是 512 MB 的電腦，Xen Hypervisor 會使用 64 MB 而 Domain-0 會用掉剩餘記憶體。若要釋放更多記憶體供新的外來領域，使用指令 `xm balloon`。若要設定 Domain-0 為 330 MB，以 `root` 身分，輸入下列：

```
xm balloon 0 330
```

在下一個 `xm list`，Domain-0 的記憶體使用會降至 330 MB。現在有足夠的記憶體，使用 128 MB 啟動外來領域。指令 `xm start guest1 -c` 啟動外來領域並連結啟動外來領域的主控台到目前終端機。如果是第一次啟動這個外來領域，使用 YaST 結束安裝。

您可以中斷或重新連結主控台和其它終端機。若要中斷，使用 `[Ctrl]+[I]`。若要重新連接，請先使用 `xm list` 檢查所需外來領域的 ID，然後使用 `xm console ID` 連接到該 ID。

Xen 的 `xm` 工具包含許多參數。輸入 `xm help`。表格 37.1, "xm 指令" [503] 檢視包含簡短說明的清單，它提供某些重要指令為起點。

表格 37.1 *xm 指令*

<code>xm help</code>	列印 <code>xm</code> 工具可使用的指令清單。
<code>xm console ID</code>	使用 ID <i>ID</i> ，連接到外來領域的第一主控台 (tty1)。
<code>xm balloon ID Mem</code>	使用 ID <i>ID</i> ，設定領域 的記憶體大小為 <i>Mem</i> MB。
<code>xm create domname [-c]</code>	使用組態檔 <i>domname</i> 啟動領域。選項 <code>-c</code> 連結目前終端機到新的外來領域的第一 tty。
<code>xm shutdown ID</code>	使用 ID <i>ID</i> ，正常關閉外來領域。
<code>xm destroy ID</code>	使用 ID <i>ID</i> ，立即結束外來領域。

<code>xm list</code>	列印執行中領域的清單，包括其 ID、記憶體和 CPU 時間值。
<code>xm info</code>	顯示有關 Xen 主機包括 CPU 和記憶體的資訊。

37.5 若需詳細資訊

更多有關 Xen 的資訊可至下列網站查訊：

- [file: /usr/share/doc/packages/xen/user/html/index.html](file:/usr/share/doc/packages/xen/user/html/index.html)—Xen 使用者的官方網站。它需要套件 `xen-doc-html`。
- [file: /usr/share/doc/packages/xen/interface/html/index.html](file:/usr/share/doc/packages/xen/interface/html/index.html)—更多技術介面文件。它也需要套件 `xen-doc-html`。
- [http: //www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html](http://www.cl.cam.ac.uk/Research/SRG/netos/xen/index.html)—Xen 首頁包含許多不同元件連結。
- [http: //lists.xensource.com/](http://lists.xensource.com/)—多個 Xen 的郵件清單。

部 IX. 服務

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。自訂的 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、數據機或其他裝置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。本章節僅討論基本機制及相關的網路組態檔。

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 38.1, "TCP/IP 通訊協定家族中的數種通訊協定" \[507\]](#) 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，整體上就是指「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。這些文件可從 <http://www.ietf.org/rfc.html> 線上取得。

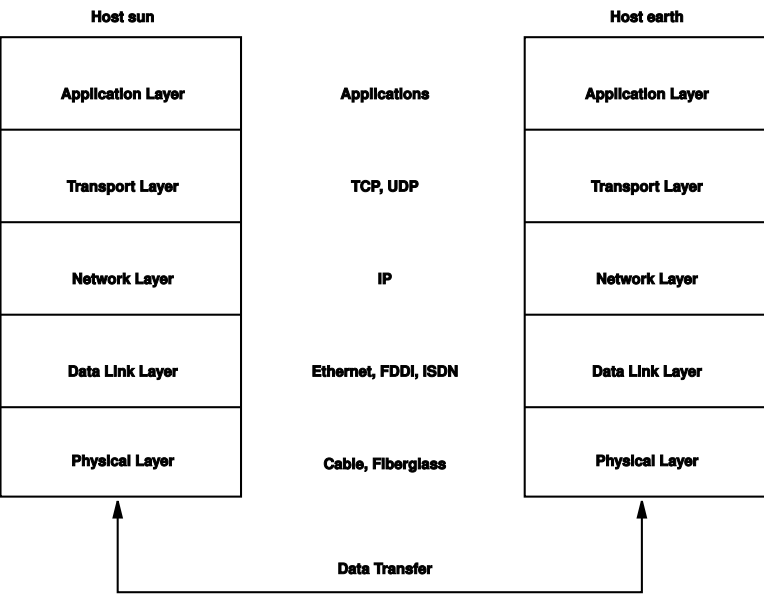
表格 38.1 TCP/IP 通訊協定家族中的數種通訊協定

協定	說明
TCP	傳輸控制通訊協定：以連線為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。資料以最初傳送的原始資料流格式，抵達到目的地主機的相關應用程式。TCP 會決定傳輸期間是否遺失任何資料，且沒有混亂。只要是資料順序很重要的地方，就會執行 TCP。

協定	說明
UDP	使用者 Datagram 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。不會保證資料在收件者端抵達的順序，而且可能遺失資料。 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：此通訊協定控制執行 IP 多重廣播時的機器行為。

如 [圖形 38.1, "TCP/IP 的簡化層模型" \[509\]](#) 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 38.1 TCP/IP 的簡化層模型

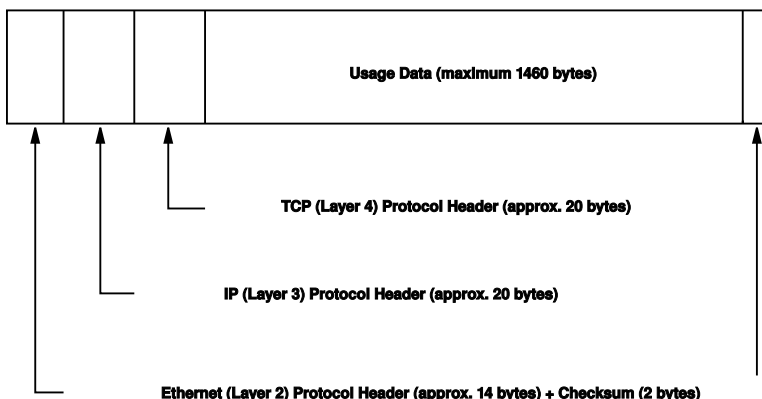


圖表提供每層的一或兩個範例。層的順序是依據抽象階層 (*abstraction level*)。最低層非常靠近硬體。不過，最上層對硬體而言幾乎是完全抽象的。每層都有自己的特殊功能。這些特殊功能通常隱含於其描述中。資料連結及實體層代表使用的實體網路 (如乙太網路)。

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題中。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 38.2, "TCP/IP 乙太網路封包" [510] 中的說明。proof sum 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 38.2 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。就如同洋蔥的層次一般，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。

38.1 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 [節 38.2, "IPv6 -- 下一代的網際網路" \[513\]](#)。

38.1.1 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 38.1, "寫入 IP 位址" \[510\]](#) 中的第二列所述。

範例 38.1 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 (無類別領域間路由，*classless interdomain routing*)。

38.1.2 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 38.2, "連結 IP 位址到網路遮罩" \[511\]](#)。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 38.2, "連結 IP 位址到網路遮罩" \[511\]](#) 中，第一個 24 位元的網路也可寫成 192.168.0.0/24。

範例 38.2 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
```

另一個範例：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 38.2 特定位址

位址類型	說明
基本的網路位址	這是網路遮罩「及」網路中的任何位址，如 <code>Result</code> 下的 範例 38.2, "連結 IP 位址到網路遮罩" [511] 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。因此上述範例會成為 192.168.0.255。此位址不能指定給任何主機。
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線。

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 架構的網路，有三種位址領域可以使用。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。這些位址領域在 RFC 1597 指定並列於 [表格 38.3, "私人 IP 位址領域" \[512\]](#) 中。

表格 38.3 私人 IP 位址領域

網路/網路遮罩	領域
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

38.2 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：即廣播位址與基本網路位址。

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。

38.2.1 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 [節 38.2.2, "定址類型與結構" \[515\]](#) 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**網路芳鄰尋找** (*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。這樣使用者就能輕鬆地存取數個網路，可以比擬為行動電話公司提供的國際漫遊服務：當您帶著手機出國時，電話只要進入通信區域，就會自動登入國外的服務，如此一來，使用同樣的電話號碼仍然可以找到您，而且您也可以撥打電話，就像在國內一樣。

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。

舊版相容性

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 [節 38.2.3, "IPv4 與 IPv6 的共存" \[518\]](#)。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。

透過多重廣播自訂量身訂做的服務

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**)。

38.2.2 定址類型與結構

如前面所提，目前的 IP 通訊協定欠缺兩個重要層面：IP 位址正在逐漸短缺，而設定網路及維護路由表的工作變得更加複雜而繁重。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。不過，每個位址僅允許一個這類的::。這類的簡略的標記法，顯示於 [範例 38.3, "範例 IPv6 位址" \[515\]](#) 中，其中三行都是代表相同的位址。

範例 38.3 範例 IPv6 位址

```
fe80 : 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 38.4, "指定字首長度的 IPv6 位址" [516] 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 38.4 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 38.4, "各種 IPv6 字首" [516] 中。

表格 38.4 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。目前而言，有以下位址空間：2001::/16 (製造品質位址空間) 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 [節 38.2.3, "IPv4 與 IPv6 的共存" \[518\]](#))，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尋找。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，這個工作是由主代辦執行的，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。

38.2.3 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 架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 [節 38.2.2, "定址類型與結構" \[515\]](#))。

在(全球) IPv4 網路中多少有些隔離的 IPv6 主機可以透過通道進行通訊：IPv6 封包會封裝成 IPv4 封包，在 IPv4 網路上移動。兩個 IPv4 主機之間的連線，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。

重要：6bone 方案

「過去」網際網路的核心，已經有透過通道連接的 IPv6 子網路的全域散佈網路。這就是 6bone 網路 (<http://www.6bone.net>)，這是 IPv6 測試環境，由想要開發及提供 IPv6 架構服務的程式人員及網際網路提供者使用，可獲得執行新通訊協定所需的經驗。詳細資訊請參閱專案的網際網路網站。

38.2.4 設定 IPv6

若要設定 IPv6，您通常不需要在個別工作站中做任何變更。不過，必須載入 IPv6 支援。如果要這樣做，請輸入 `modprobe ipv6` 做為 root。

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` 自動設定位址和路由的組態。

請參閱 `ifup` 的手冊頁，了解如何使用 `/etc/sysconfig/network` 檔案來設定不同類型的通道。

38.2.5 若需詳細資訊

上述概觀沒有完整地涵蓋 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.ngnet.it/e/cosa-ipv6.php>

有一系列內容完整、提供 IPv6 基本概念的介紹文章。有關主題的最佳入門。

<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。

<http://www.6bone.net/>

如果想要加入發掘 IPv6 網路的行列，請瀏覽此網站。

<http://www.ipv6.org/>

所有有關 IPv6 的入門資訊。

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)。

38.3 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 `earth.example.com`，以 `hostname.domain` 格式來表示。完整名稱，也就是完整領域名稱 (*Fully Qualified Domain Name, FQDN*)，是由主機名稱和領域名稱 (`example.com`) 組成的。後者也包含了「最上層領域」(*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領域名稱。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連線、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領域，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領域的名稱伺服器。有關最上層領域 *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*)，負責接收該領域的電子郵件。

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線，完全不需要手動設定名稱伺服器。撥接通訊協定在連線建立時會提供名稱伺服器位址。關於使用 SUSE Linux 存取名稱伺服器的組態，在 [章 40, 領域名稱系統 \[547\]](#) 中有詳細的說明。

`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領域。

38.4 使用 YaST 手動設定網路連線

Linux 可支援多種網路類型。大多數使用不同的裝置名稱和組態檔，會分佈在檔案系統的不同位置。要更了解手動網路組態的綜覽，請參閱 [節 38.5, "手動設定網路連線" \[531\]](#)。

安裝期間，YaST 可以自動設定所有偵測到的介面。安裝後，可隨時在安裝系統設定其他的硬體。下列章節，說明 SUSE Linux 支援的所有網路連結類型的網路組態。

38.4.1 使用 YaST 設定網路卡

啟動模組後，YaST 會顯示一般網路組態對話方塊。上半部顯示一個清單，列出所有尚未設定的網路卡。任何正確偵測的介面會在此列出名稱。無法偵測的裝置，可使用 *其他 (未偵測到)* 來設定，請參閱 ["手動設定未偵測到的網路卡" \[522\]](#)。對話方塊下半部顯示目前已設定組態的設備清單，與其網路類型和位址。您現在可以設定新的網路卡或變更現有的組態設定。

手動設定未偵測到的網路卡

無法偵測到的網路卡的組態設定 (列為 *其他* 的網路卡) 包括下列項目：

網路組態

從可用選項和組態名稱，設定介面的設備類型和組態名稱。有關組態名稱命名慣例的資訊，請參閱 `getcfg(8)` 手冊頁。

核心模組

硬體組態名稱 會指定 `/etc/sysconfig/hardware/hwcfg-*` 檔案的名稱，其中包含網路卡的硬體設定，這包含合適的核心模組名稱，以及所需的選項以初始化硬體。YaST 通常會建議有用的 PCMCIA 和 USB 硬體名稱。對於其他硬體來說，只有在網路卡使用網路名稱 0 設定時，`hwcfg-static-0` 才有意義。

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，從 *從清單選擇* 選擇網路卡型號。然後 YaST 會自動選取適當的網路卡核心模組。使用 *下一步* 結束對話方塊。

圖形 38.3 網路卡的組態

Here, set up your networking device. The values are written to `/etc/sysconfig/hardware/`.

Options for the module should be written in the format `option=value`. Each entry should be space separated, for example, `io=0x300 irq=5`. **Note:** If two cards are configured with the same module name, options will be merged while saving.

Get a list of available network cards by pressing **Select from List**.

If you have a **PCMCIA** network card, select PCMCIA. If you have a **USB** network card, select USB.

Manual Network Card Configuration

Network Configuration

Device Type: Ethernet Configuration Name: 0

Kernel Module

Hardware Configuration Name: static-0

Module Name: Options:

☐ PCMCIA ☐ USB

Select from List

Back Abort Next

設定網路位址

設定介面的設備類型和組態名稱。從提供的選項選取設備類型。根據您的需要指定一個組態名稱。預設值通常相當有用，而且是可接受的。有關組態名稱命名慣例的資訊，請參閱 `getcfg(8)` 手冊頁。

如果您選取 **無線** 做為介面的設備類型，請設定操作模式、網路名稱 (ESSID) 以及下一個對話方塊 **無線網路卡組態** 中的加密。按一下 **確定** 完成您網路卡的組態設定。請參閱 [節 22.1.3, "利用 YaST 進行組態" \[271\]](#) 中有關無線網路卡組態的詳細資訊。如果要設定其他介面類型，請接著進行網路位址設定：

自動位址設定 (透過 DHCP)

如果您的網路包含 DHCP 伺服器，您可以用它來自動設定您的網路位址。如果您用的是 DSL 連線而非 ISP 指定的靜態 IP 時，應該使用此選項。如果您決定使用 DHCP，請在選取 **DHCP 用戶端** 選項之後設定其詳細資料。指定 DHCP 伺服器是否應該執行廣播要求以及使用任何識別碼。按照預設值，DHCP 伺服器會使用網路卡的硬體位址以辨別介面。如果有虛擬主機設定，透過相同介面與不同主機通訊，會需要使用識別子來分辨他們。

設定靜態位址

如果有靜態位址，請啟動該選項。然後輸入您網路的位址和子網路遮罩。預設的子網路遮罩應該會符合一般家用網路的需求。

選取 下一步 來結束對話方塊，或接著設定主機名稱、名稱伺服器 and 路由詳細資料 (請參閱 DNS 伺服器 (↑啟動) 和 路由 (↑啟動) 章節)。

進階可讓您指定更複雜的設定。在 詳細設定 下，可以使用 使用者控制，將管理員 (root) 的網路卡控制權委託給一般使用者。就行動操作而言，可讓使用者更有彈性地調整變更的網路連線，因此可以自行控制介面的啟用和停用。可在此對話方塊中設定 MTU (傳輸單元最大值) 和 設備啟用 類型。

38.4.2 數據機

在「YaST 控制中心」，可在 網路卡設備 下存取數據機組態。如果無法自動偵測到您的數據機，請開啟手動組態對話方塊。在開啟的對話方塊中，輸入數據機在 數據機 下用來連線的介面。

圖形 38.4 數據機組態

Enter all modem configuration values.

Modem Device specifies to which port your modem is connected: ttyS0, ttyS1, etc., refer to serial ports and usually correspond to COM1, COM2, etc., in DOS/Windows. ttyACM0 and ttyACM1 refer to USB ports.

If you are on a PBX, you probably need to enter a **Dial Prefix**. Often, this is 9 or 0.

Choose **Dial Mode** according to your phone link. Most telephone companies use **Tone Dial** as the **Dial Mode**. Check the additional check boxes to turn on your modem speaker (**Speaker On**) or for your modem to wait until it detects a dial tone (**Detect Dial Tone**).

Press **Details** to configure the baud rate and the modem initialization strings.

Modem Parameters

Modem Device
/dev/modem

Dial Prefix (if needed)

Dial Mode
☒ Tone Dialing
☐ Pulse Dialing

Special Settings
☒ Speaker On
☒ Detect Dial Tone

Details

Back Abort Next

如果您使用的是專用交換機 (PBX)，您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。如果要在沒有 root 許可權的情況下將數據機控制權委託給一般使用者，請啟用 **使用者控制**。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** 一般表示式下，指定一個一般表示式。KInternet 中的 **撥號前置號碼** (一般使用者均可修改)，必須符合此一般表示式。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，選擇 **ISP** (網際網路服務提供者)。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。這包括撥接連線名稱、**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**，以提示您在每次連線時輸入密碼。

在最後一個對話方塊中，可以指定其他的連線選項：

視需要撥號

如果您啟用了視需要撥號，請至少設定一個名稱伺服器。

連線時修改 DNS

此選項是依照預設啟用的，每次您連線到網際網路時，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線後傳輸其領域名稱伺服器，則應停用此選項並手動輸入 **DNS** 資料。

簡易模式

此選項預設為啟用。使用它，就會忽略 **ISP** 伺服器送出的輸入提示，以避免干擾連線過程。

外部防火牆介面 和 重新啟動防火牆

選取這些選項即可啟用 **SUSEfirewall2**，在您連線到網際網路時提供保護，防止外來攻擊。

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線。

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 *完成* 以結束此對話方塊。

38.4.3 ISDN

使用此模組來為您的系統設定一個或多個 ISDN 卡。如果 YaST 無法自動偵測您的 ISDN 卡，請以手動選取它。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 38.5 ISDN 組態

Start Mode: With **OnBoot**, the driver is loaded during system boot. For **Manual**, the driver must be started with the **reiscan start** command. Only the user root can do this. **HotPlug** is a special case for PCMCIA and USB devices.

ISDN Protocol: In most cases, the protocol is Euro-ISDN.

Area Code: Enter your local area code for the ISDN line here, without a leading zero and without a country prefix.

Dial Prefix: If you need a prefix to get an public line, enter it here. This is only used on a internal S0 bus and the most common one is "0".

If you do not want to log all your ISDN traffic, uncheck **Start ISDN Log**.

ISDN Card Information

Vendor: Pyramid
ISDN Card: Speed Fax+ PCI

Driver: HiSax driver

ISDN Protocol

☒ Euro-ISDN (EDSS1)
☐ ITR6
☐ Leased Line
☐ NI1

Country: Germany Code: +49

Area Code: Dial Prefix:

☒ Start ISDN Log

Device Activation

At Boot Time

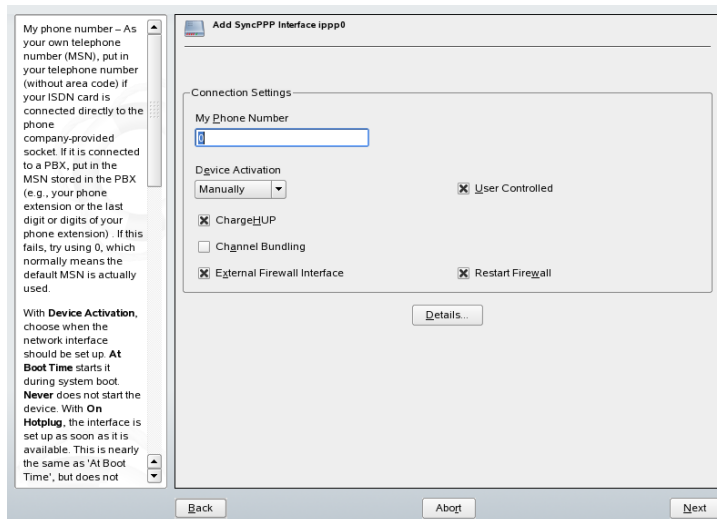
Back Abort OK

在下一個對話方塊中 (如 圖形 38.5, "ISDN 組態" [526] 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 *ITR6*。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)。

啟動模式定義應該如何啟動 ISDN 介面：開機時會使系統每次啟動時都初始化 ISDN 驅動程式。手動會要求您以 root 身分使用 `rcisdn start` 指令來載入 ISDN 驅動程式。用於 PCMCIA 或 USB 設備的熱插式設備，會在插入設備後載入驅動程式。完成上述所有設定之後，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 38.6 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 連線會提供 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 連線的第一組 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 匯流排並在連線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。如需更多資訊，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線，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* 來啟用連線的 *SUSEfirewall2*。如果要讓一般使用者不需要管理員許可即可啟用和停用介面，請選取 *使用者控制*。

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，即在您每次連線時 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，可為連線指定一個時限未使用網路的時間 (以秒為單位)，之後即會自動終止連線。以 *下一步* 確認您的設定值。YaST 顯示設定介面的摘要。如果要啟用這些設定，請選取 *完成*。

38.4.4 纜線數據機

在某些國家 (奧地利、美國)，透過電視纜線來存取網際網路是相當普遍的。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 *自動位址設定 (透過 DHCP)* 或 *靜態位址設定*。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。

38.4.5 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 模組含有多個對話方塊，可根據下列其中一種通訊協定在其中設定 DSL 連線的參數：

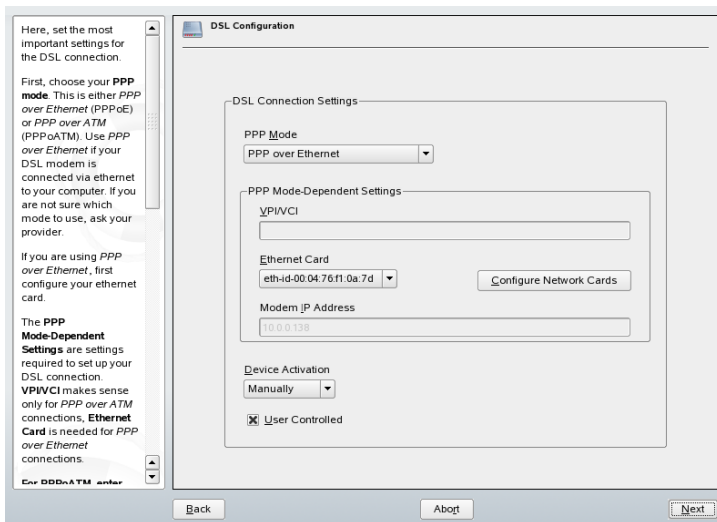
- PPP over Ethernet (PPPoE)
- PPP over ATM (PPPoATM)
- CAPI for ADSL (Fritz 網路卡)
- 點對點通道通訊協定 (PPTP) -- 奧地利

PPPoE 或 PPTP 類型的 DSL 連線組態，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 [節 38.4.1, "使用 YaST 設定網路卡" \[522\]](#))。如果是 DSL 連線，會自動指定位址但不是透過 DHCP，，因此您不可啟用 *自動位址設定 (透過 DHCP)* 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。只用來初始化網路卡，而不代表 DSL 連線等等。

圖形 38.7 DSL 組態



如果要開始設定 DSL 組態 (請參閱 [圖形 38.7, "DSL 組態" \[530\]](#))，請先選取 PPP 模式和用來連接 DSL 數據機的乙太網路卡 (大多為 eth0)。然後使用 **設備啟用** 來指定是否要在開機時建立 DSL 連線。按一下 **使用者控制** 來授權一般使用者，讓他們不需要 root 許可權即可透過 KInternet 來啟用或停用介面。您也可以使用該對話方塊來選擇您的國家和當地一些主控它的 ISP。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。

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，每次連線時，會送出名稱伺服器的 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線。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 的組態非常類似 DSL 的設定。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。

38.5 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。

所有內建的網路卡以及熱插式網路卡 (PCMCIA、USB、一些 PCI 卡) 會透過熱插方式進行偵測到並加以設定。系統是透過下列兩種不同的方法來偵測網路卡：第一種是以實體設備，第二種則是以介面。插入設備或偵測到設備時則會觸發熱插事件。此熱插事件使用程序檔 `hwup` 觸發設備的起始化。起始化網路卡作為新的網路介面時，核心會產生另一個熱插事件，使用 `/ifup` 觸發介面的設定。

核心根據介面名稱註冊的暫時順序加以編號。起始化順序是指定名稱的決定因素。如果數個網路卡中的一個失敗，所有依序起始化網路卡的編號就會改變。對於真正的可熱插式網路卡，設備連接的順序才是決定因素。

為了讓組態具有彈性，已經分開設備 (硬體) 及介面的組態，而且不再以介面名稱來管理組態與設備和介面之間的對應。設備組態位於 `/etc/sysconfig/hardware/hwcfg-*`。介面組態位於 `/etc/sysconfig/network/ifcfg-*`。組態名稱是以描述設備及其關聯介面的方式來指定的。因為之前驅動程式與介面名稱的對應需要靜態介面名稱，所以此對應不再發生於 `/etc/modprobe.conf`。在這種新的概念中，此檔案中的別名項目造成不想要的副作用。

組態名稱 (`hwcfg-` 或 `ifcfg-` 之後的任何項目) 都可以透過插槽、裝置專用 ID 或介面名稱來描述設備。例如，PCI 卡的組態名稱可以是 `bus-pci-0000:02:01.0` (PCI 插槽) 或 `vpid-0x8086-0x1014-0x0549` (廠商和產品 ID)。相關聯介面的名稱可以是 `bus-pci-0000:02:01.0` 或 `wlan-id-00:05:4e:42:31:7a` (MAC 位址)。

若要指定特定網路組態到任何特定類型的卡 (一次僅能插入一種) 而不是特定卡，請選取較不特定的組態名稱。例如，`bus-pcmcia` 可以用於所有 PCMCIA 卡。在另一方面，之前的介面類型會限制名稱。例如，`wlan-bus-usb` 可以指定給連接到 USB 埠的 WLAN 卡。

系統永遠使用最佳描述介面或提供介面之設備的組態。搜尋最適用組態是由 `getcfg` 處理的。`getcfg` 的輸出會傳達可以用來描述設備的所有資訊。有關組態名稱規格的詳細資料，請參閱 `getcfg` 的手冊頁。

透過描述的方法，即使網路設備不一定永遠以相同的順序起始化，網路介面還是可以具有正確的組態設定。不過，介面的名稱仍然取決於起始化順序。有兩種方式可以確保確實存取特定網路卡介面：

- `getcfg-interfaceconfiguration name` 傳回相關聯網路介面的名稱。因此，可以輸入一部份的組態名稱(例如，防火牆、`dhcpcd`、路由、各種虛擬網路介面 (通道))，取代介面名稱，因為這種名稱並不是永久不變的。
- 永久介面名稱可以指定給其組態不包含介面名稱的所有介面。透過介面組態(`ifcfg-*`)中的 `PERSISTENT_NAME= pname` 項目可以執行這個動作。然而，永久名稱 `pname` 不能與核心自動指定的名稱相同。所以，不允許 `eth*`、`tr*`、`wlan*` 等名稱。相反地，請使用 `net*` 或是 `external`、`internal`、`dmz` 之類的描述名稱。永久名稱僅能在註冊後立即指定給介面，這表示必須重新載入網路卡的驅動程式或執行 `hwup device description`。針對此用途，僅使用指令 `rcnetworkrestart` 是不夠的。

重要：使用永久介面名稱

永久介面名稱的使用尚未測試於所有方面。因此，一些應用程式可能無法自由地處理選取的介面名稱。

`ifup` 需要現有介面，因為它不會起始化硬體。硬體的起始化是由指令 `hwup` (由 `hotplug` 或 `coldplug` 執行) 處理的。起始化設備時，會透過 `hotplug` 自動為新介面執行 `ifup`，如果開始模式是 `onboot`、`hotplug` 或 `auto`，就會設定介面，而且會啟動 `network` 服務。之前，是使用指令 `ifup interfacename` 來觸發硬體起始化。現在已經反轉此程序。首先，會起始化硬體元件，接著是所有其他的動作。採取這種方式，則永遠可以使用現有的組態集，儘可能以最佳的方式設定不同數目的設備。

表格 38.5, "手動網路組態程序檔" [532] 總結了與網路組態相關的最重要程序檔。只要可能，會依硬體和介面區分程序檔。

表格 38.5 手動網路組態程序檔

組態階段	指令	功能
硬體	<code>hw{ up, down, status }</code>	<code>hw*</code> 程序檔由熱插式子系統執行，用來起始化設備、復原起始化、或查詢設

組態階段	指令	功能
		備狀態。詳細資訊請參閱 hwup 的手冊頁。
介面	getcfg	getcfg 可以用來查詢與組態名稱或硬體描述相關聯的介面名稱。詳細資訊請參閱 getcfg 的手冊頁。
介面	if{up, down, status}	if* 程序檔會開啟現有網路介面或傳回指定介面的狀態。詳細資訊請參閱 ifup 的手冊頁。

有關熱插拔和永久設備名稱的詳細資訊，請參閱 [章 32, 熱插拔系統\[451\]](#) 與 [章 33, 使用 udev 的動態裝置節點 \[457\]](#)。

38.5.1 組態檔

本節提供網路組態檔的概觀，並說明其用途和使用的格式。

/etc/syconfig/hardware/hwcfg-*

這些檔案包含網路卡及其他設備的硬體組態；其中包含所需的參數，如核心模組、啟動模式和程序檔關聯。詳細資訊請參閱 hwup 的手冊頁。無論現有硬體為何，啟動 coldplug 時會套用 hwcfg-static-* 組態。

/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註解，而且可以在 `ifcfg-*` 檔案中使用，以更高優先順序處理。

/etc/sysconfig/network/routes, ifroute-*

TCP/IP 封包的靜態路由在此決定。各種系統工作所需的所有靜態路由可以在 `/etc/sysconfig/network/routes` 檔案中輸入：路由至主機、透過閘道的路由至主機，及路由至網路。針對需要個別路由的每個介面，定義其他的組態檔：`/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。

第四欄只與連線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。不需要的蘭位必須包含 `-` 減號，以確保可正確解析指令。請參閱 `routes(5)` 一文取得更多資訊。

/etc/resolv.conf

主機所屬的領域指定於此檔案 (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`)。可以指定多個領域名稱。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。透過輸入數行且每行都以 `nameserver` 開頭的方法，可使用多個名稱伺服器。在註解前加上 `#` 符號。YaST 在此檔案中輸入指定名稱伺服器。範例 38.5, "`/etc/resolv.conf`" [535] 顯示出 `/etc/resolv.conf` 有可能的樣子。

範例 38.5 `/etc/resolv.conf`

```
# Our domain
search example.com
#
# We use sun (192.168.0.20) as nameserver
nameserver 192.168.0.20
```

一些如 `pppd` (`wvdial`)、`ipppd` (`isdn`)、`dhcpcd` (`dhcpcd` 和 `dhclient`)、`pcmcia` 及 `hotplug` 之類的服務會修改檔案 `/etc/resolv.conf`，方法是使用程序檔 `modify_resolvconf`。如果檔案 `/etc/resolv.conf` 已經由此程序檔暫時修改，它會包含預先定義的註解，提供的資訊包含修改它的服務、原始檔案的備份位置以及如何關閉自動修改機制。如果數次修改 `/etc/resolv.conf`，檔案會以巢狀形式包含所作的修改。即使反轉時使用與修改順序不同的順序，還是可以完全反轉此程序。需要這種彈性的服務包括 `isdn`、`pcmcia`、`hotplug`。

如果沒有以正常的方式終止服務，可以使用 `modify_resolvconf` 還原原始檔。另外，在系統開機時，會執行檢查，看看是否有沒有清理、已修改的 `resolv.conf` (例如，系統當機後)，在這種情況下，會還原原始 (未修改) 的 `resolv.conf`。

YaST 使用指令 `modify_resolvconf check` 找出是否已修改 `resolv.conf`，接著將警告使用者還原檔案會遺失所有變更。除此之外，YaST 不會依靠 `modify_resolvconf`，意即會透過 YaST 變更 `resolv.conf` 的影響與手動變更的影響是一樣的。無論是哪種情形，變更永遠是有效的。上述提及的服務所需的修改只是暫時的。

`/etc/hosts`

在此檔中 (請參閱 範例 38.6, "`/etc/hosts`" [536])，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 連線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、完整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 38.6 `/etc/hosts`

```
127.0.0.1 localhost
192.168.0.20 sun.example.com sun
192.168.0.0 earth.example.com earth
```

`/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面。請參閱 [範例 38.7, "`/etc/networks`" \[536\]](#)。

範例 38.7 `/etc/networks`

```
loopback      127.0.0.0
localnet      192.168.0.0
```

`/etc/host.conf`

名稱解析--透過解析器庫分析主機和網路的名稱--由此檔案收集。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 38.6, "`/etc/host.conf` 的參數" \[536\]](#) 顯示出可用的參數。`/etc/host.conf` 範例是顯示在 [範例 38.8, "`/etc/host.conf`" \[537\]](#)。

表格 38.6 `/etc/host.conf` 的參數

<code>order hosts, bind</code>	指定針對名稱解析存取服務的順序。可用的引數為 (以空格或是逗號分隔): <i>hosts</i> : 搜尋 <code>/etc/hosts</code> 檔案 <i>bind</i> : 存取名稱伺服器 <i>NIS</i> : 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> ，但除此之外，並不會對網路組態有任何影響。

`trim domainname` 指定的領域名稱在主機名稱解析後會與主機名稱分隔(只要主機名稱包括領域名稱)。只有在本地領域分離出來的名稱位於 `/etc/hosts` 檔案，但是仍然使用附加的領域名稱進行辨識時，這個選項才有用。

範例 38.8 `/etc/host.conf`

```
# We have named running
order hosts bind
# Allow multiple addrs
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 38.9, "`/etc/nsswitch.conf`" [537]。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) (請參閱 章 40, 領域名稱系統 [547])。

範例 38.9 `/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 38.7, "透過 `/etc/nsswitch.conf` 的可用資料庫" [538]。此外，將來應該還有 `automount`、`bootparams`、`netmasks`、和 `publickey`。NSS 資料庫的組態選項是列在 表格 38.8, "NSS「資料庫」的組態選項" [538]。

表格 38.7 透過 */etc/nsswitch.conf* 的可用資料庫

<code>aliases</code>	<code>sendmail</code> 所執行的郵件別名；請參閱 <code>man 5 aliases</code> 。
<code>ethers</code>	乙太網路位址。
群組	<code>getgrent</code> 所使用的使用者群組。請參閱 <code>group</code> 的 <code>man</code> 頁面。
主機	<code>gethostbyname</code> 及類似功能所使用的主機名稱與 IP 位址。
<code>netgroup</code>	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 <code>netgroup(5)</code> 一文。
<code>networks</code>	<code>getnetent</code> 所使用的網路名稱與位址。
<code>passwd</code>	<code>getpwent</code> 所使用的使用者密碼；請參閱 <code>passwd(5)</code> 一文。
<code>protocols</code>	<code>getprotoen</code> 所使用的網路通訊協定；請參閱 <code>protocols(5)</code> 一文。
<code>rpc</code>	<code>getrpcbyname</code> 及類似功能所使用的遠端程序呼叫名稱與位址。
<code>services</code>	<code>getservent</code> 使用的網路服務。
<code>shadow</code>	<code>getspnam</code> 所使用的使用者遮蔽密碼；請參閱 <code>shadow(5)</code> 一文。

表格 38.8 NSS「資料庫」的組態選項

<code>files</code>	直接存取檔案，例如 <code>/etc/aliases</code>
<code>db</code>	透過資料庫存取

<code>nis、nisplus</code>	NIS，請參閱 章 41, 使用 NIS [567]
<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`，以縮短這段等待時間。

/etc/HOSTNAME

這是沒有附加領域名稱的主機名稱。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。

38.5.2 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 38.9, "網路程式的一些啟動程序檔" \[539\]](#) 中提供了一些程序檔的說明。

表格 38.9 網路程式的一些啟動程序檔

<code>/etc/init.d/network</code>	此程序檔會處理網路介面的組態。硬體必須先由 <code>/etc/init.d/coldplug</code> (透過 <code>hotplug</code>) 起始化。如果沒有啟動 <code>network</code>

服務，透過熱插拔插入網路介面時，將無法執行它們。

<code>/etc/init.d/inetd</code>	啟動 <code>xinetd</code> 。 <code>xinetd</code> 可用來使伺服器服務供系統使用。例如，只要開啟 FTP 連線，它即可啟動 <code>vsftpd</code> 。
<code>/etc/init.d/portmap</code>	啟動 RPC 伺服器所需的 <code>portmapper</code> ，例如 NFS 伺服器。
<code>/etc/init.d/nfsserver</code>	啟動 NFS 伺服器。
<code>/etc/init.d/sendmail</code>	控制傳送郵件的程序。
<code>/etc/init.d/ypserv</code>	啟動 NIS 伺服器。
<code>/etc/init.d/ypbind</code>	啟動 NIS 用戶端。

38.6 smpppd 做為撥號助理

大部份的家用使用者都沒有連線到網際網路的專線，而是使用撥號連線。視撥號方法 (ISDN 或 DSL) 而定，連線是由 `ipppd` 或 `pppd` 來控制。基本上，連線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線，不會產生撥號連線的其他成本，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線。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線。

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 `ipppd` 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，並傳輸關於連線目前狀態的資訊。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線。

38.6.1 設定 smpppd

smpppd 所提供的連線會自動由 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*

如果主機具有數個 IP 位址，請使用此參數以決定 smpppd 應該接受連線的 IP 位址。

host-range = *min ip max 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)` and `smpppd.conf(5)`。

38.6.2 設定 KInternet、cinternet 與 qinternet 以供遠端使用

KInternet、cinternet 以及 qinternet 可用以控制本地或遠端 smpppd。cinternet 是圖形化 KInternet 對等的指令行程式。qinternet 基本上與 KInternet 相同，但是不使用 KDE 程式庫，因此可以在沒有 KDE 的情況下使用它，而且必須分開安裝。

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`。這個檔案只使用三個選項：

`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` 選項會命令建立連至本地 `smpppd` 的連線。`gateway` 會指向閘道的 `smpppd`。這個連線將會依 `config-file` 中 `server` 下的指定來建立。`slp` 會命令前端，連線至透過 `SLP` 所找到的 `smpppd`。

`server = server`

此處指定 `smpppd` 所執行的主機。

`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)` and `cinternet(8)`。

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，都知道某一服務的可用性。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 支援透過 SLP 提供的安裝來源，並包含 SLP 整合的系統服務。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，例如安裝伺服器、YOU 伺服器、檔案伺服器或是 SUSE Linux 上的列印伺服器。

39.1 註冊自己的服務

許多在 SUSE Linux 下的應用程式，透過 `libslp` 程式庫的使用，已經具有整合的 SLP 支援。如果尚未使用 SLP 支援來編譯服務，請使用下列其中一種方式讓 SLP 編譯服務：

透過 `/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。

註冊檔另外也包含 `watch-tcp-port` 與 `description` 兩個變數。第一個變數將 SLP 服務宣告連結至是否為作用中的相關服務，因為 `slpd` 檢查服務的狀態。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

上述程序的唯一差異是在中央檔案內群組所有的服務。

以 `slptool` 動態註冊

如果應該從專用的程序檔為 SLP 註冊服務，請使用 `slptool` 指令行前端。

39.2 在 SUSE Linux 中的 SLP 前端

SUSE Linux 包含數個前端，可透過網路的檢查及使用，來啟用 SLP 資訊：

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。也可以從處理 SLP 資訊的程序檔呼叫 `slptool`。

YaST SLP 瀏覽器

YaST 包含獨立的 SLP 瀏覽器，以樹狀圖表在 *網路服務* → *SLP 瀏覽器* 下列出區域網路中所有透過 SLP 所宣告的服務

Konqueror

當 Konqueror 做為網路瀏覽器時，可以在 `slp:` 顯示區域網路中所有可用的 SLP 服務。在主要視窗中按一下圖示，以取得更多關於相關服務的詳細資訊。如果您使用 Konqueror 加上 `service: /`，則請在瀏覽器視窗中按一次相關圖示，以設定與選取服務的連線。

39.3 啟動 SLP

如果您要提供服務，必須在系統上執行 `slpd`。如果只是要做服務查詢，並不需要啟動此精靈。就像大部份在 SUSE Linux 之下的系統服務一樣，`slpd` 精靈是利用獨立的 `init` 程序檔來控制。預設精靈為非作用中。若要為持續某个工作階段而啟動它，請以 `root` 的身份執行 `rcslpd start` 以啟動它，並執行 `rcslpd stop` 以停止它。請以 `restart` 或 `status` 執行重新啟動或狀態檢查。如果預設 `slpd` 應該啟動，請以 `root` 的身份執行 `insservslpd` 指令。這將會在系統開機時，自動在要啟動的服務集合中包括 `slpd`。

39.4 若需詳細資訊

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.com>

OpenSLP 計劃的首頁。

`file:/usr/share/doc/packages/openslp/*`

這個目錄包含所有關於 SLP 的可用文件，包括 `README`、`SuSE` (含有 SUSE Linux 詳細資訊)、上方所提及的 RFC 以及兩個介紹性的 HTML 文件。想要使用 SLP 功能的程式設計人員，應該安裝 `openslp-devel` 套件，以參閱它所提供的 *程式設計人員指南*。

領域名稱系統

必須使用 DNS (領域名稱系統) 將領域名稱和主機名稱解析為 IP 位址。例如，藉由這種方式，IP 位址 192.168.0.0 會指定給主機名稱 `earth`。在設定您自己的名稱伺服器前，請參閱 [節 38.3, "名稱解析" \[520\]](#) 中有關 DNS 的一般資訊。以下設定範例是指 BIND。

40.1 DNS 概論

40.2 使用 YaST 進行組態

您可以使用 YaST 的 DNS 模組設定區域網路的 DNS 伺服器。第一次啟動模組時，會啟動精靈以提示您決定有關伺服器管理的一些基本設定。完成此最初設定程序會產生一個非常基本的伺服器組態，應該在其基本方面都能運作。進階模式可以處理更多進階設定任務。

40.2.1 精靈組態

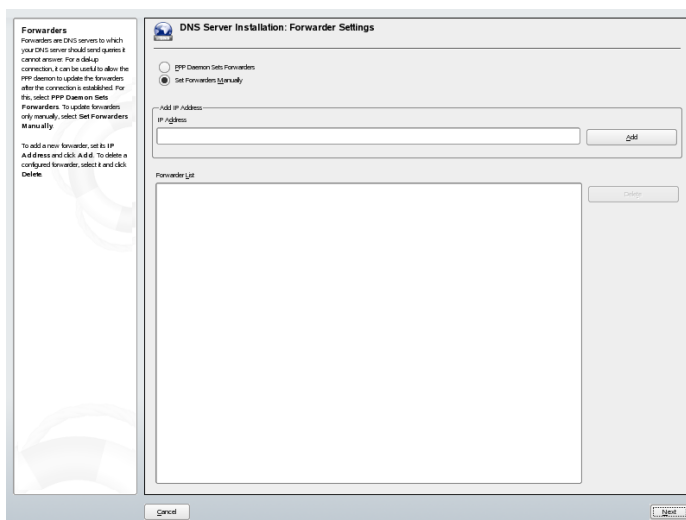
精靈包含三個步驟或對話方塊。在對話方塊中適當的地方，能夠讓您進入進階組態模式。

Forwarder 設定

第一次啟動模組時，請參閱 [圖形 40.1, "DNS 伺服器安裝：Forwarder 設定" \[548\]](#) 中所示的對話方塊。它可以決定 PPP 精靈是否應該提供透過 DSL 或

ISDN 在撥接時 forwarder 的清單 (PPP 精靈設定 *Forwarder*)，或者是否要提供您自己的清單 (手動設定 *Forwarder*)。

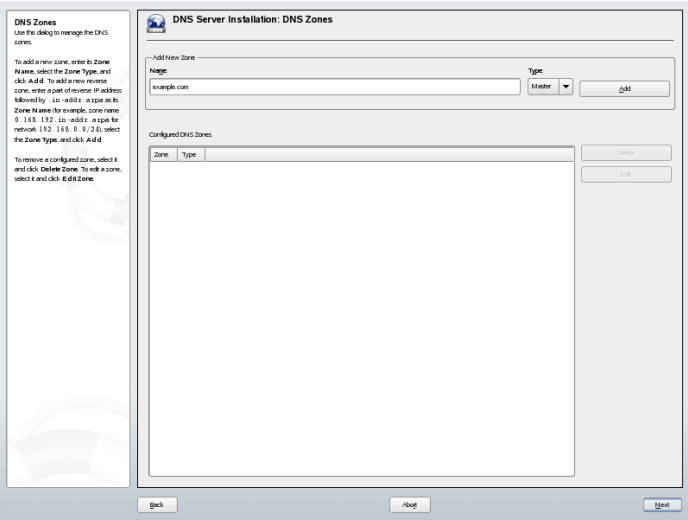
圖形 40.1 DNS 伺服器安裝：Forwarder 設定



DNS 區域

此對話方塊包含數個部分，負責管理區域檔案，在 [節 40.5, "區域檔案" \[560\]](#) 中可找到它的相關說明。對於新區域，在 *區域名稱* 中提供名稱。若要新增反向區域，名稱的結尾必須是 `.in-addr.arpa`。最後，選取 *區域類型* (主要或從屬)。請參閱 [圖形 40.2, "DNS 伺服器安裝：DNS 區域" \[549\]](#)。按一下 *編輯區域*，設定現有區域的其他設定。若要移除區域，按一下 *刪除區域*。

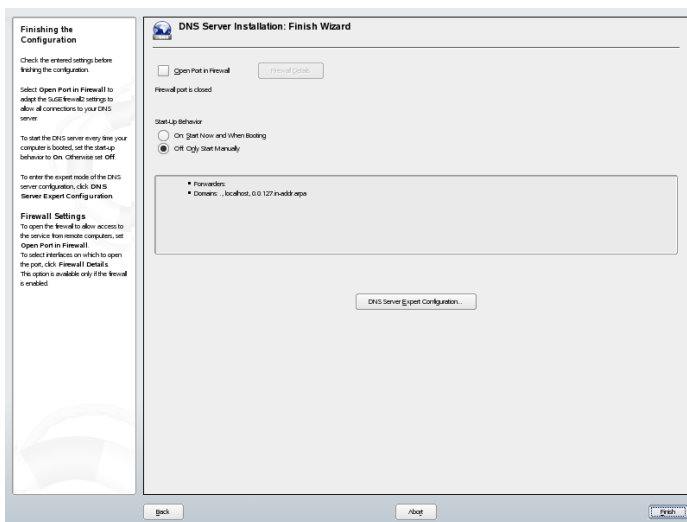
圖形 40.2 DNS 伺服器安裝：DNS 區域



完成精靈

在這個最後的對話方塊中，您可以開啟在安裝期間所啟動防火牆中的 DNS 服務埠，並決定是否要啟動 DNS。進階組態也可以從此對話方塊存取。請參閱 [圖形 40.3, "DNS 伺服器安裝：完成精靈" \[550\]](#)。

圖形 40.3 DNS 伺服器安裝：完成精靈



40.2.2 進階組態

啟動模組後，YaST 會開啟顯示數個組態選項的視窗。完成該視窗可讓 DNS 伺服器組態的基本功能就位運作：

啟動

在開機下，定義系統開機（啟動系統時）或手動時是否啟動 DNS 伺服器。若要立即啟動 DNS 伺服器，選取 **立即啟動 DNS 伺服器**。若要停止 DNS 伺服器，選取 **立即停止 DNS 伺服器**。若要儲存目前的設定，選取 **立即儲存設定並重新啟動 DNS 伺服器**。您可以使用 **在防火牆中開啟埠** 在防火牆中開啟 DNS 埠，並使用 **防火牆細節** 修改防火牆設定。

Forwarder

這個對話方塊與啟動精靈組態後所開啟的對話方塊相同 (請參閱 [Forwarder 設定 \[547\]](#))。

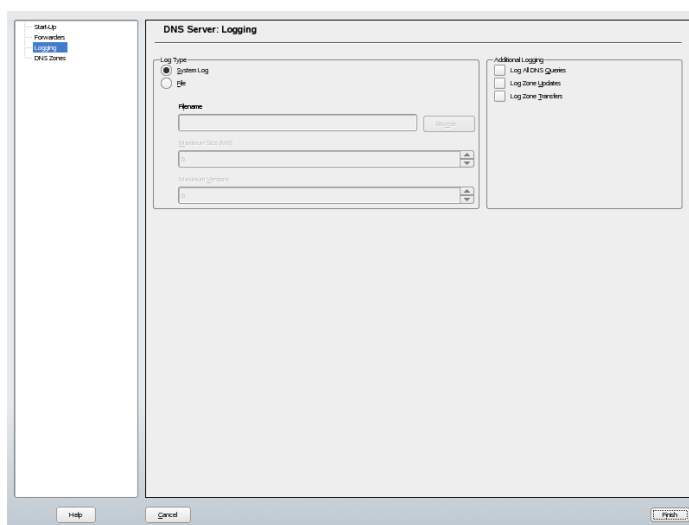
記錄

本區段允許您設定 DNS 伺服器應該記錄的內容及方法。在 **記錄類型** 下，指定 DNS 伺服器應該寫入記錄資料的位置。選取 **記錄到系統記錄** 來使用全系統的記錄檔 `/var/log/messages` 或選取 **記錄到檔案** 指定不同的檔案。在

後者情況下，另外也要指定最大檔案大小(以 MB 為單位)以及要儲存的記錄檔數量。

進一步選項可從 *其他記錄* 下存取。啟用 *記錄所有 DNS 查詢* 會記錄每個查詢，這個選項會讓記錄檔的大小變得非常大。所以，除了偵錯用途外，啟用此選項並不是理想的作法。若要記錄 DHCP 與 DNS 伺服器之間區域更新期間的資料流量，啟用 *記錄區域更新*。若要記錄從主伺服器到從屬伺服器區域傳輸期間的資料流量，啟用 *記錄區域傳輸*。請參閱 [圖形 40.4, "DNS 伺服器：記錄" \[551\]](#)。

圖形 40.4 DNS 伺服器：記錄



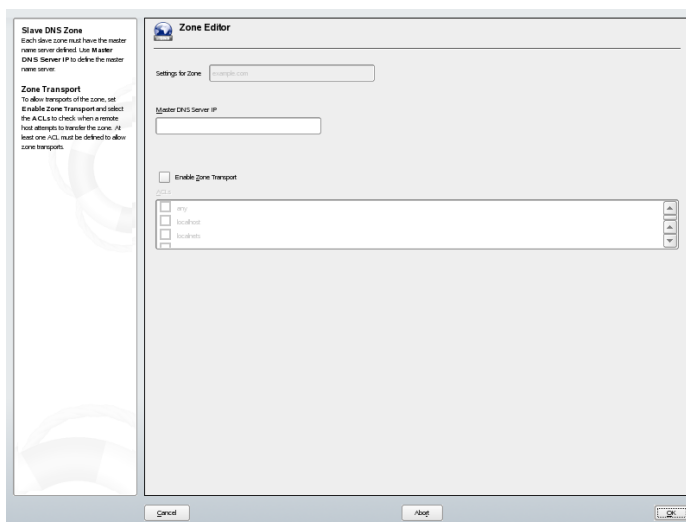
DNS 區域

這個對話方塊說明精靈的組態。請參閱 [節 40.2.1, "精靈組態" \[547\]](#)。

從屬區域編輯器

如果在 [DNS 區域 \[551\]](#) 所述的步驟中選取的區域類型為 *從屬*，就會開啟此對話方塊。在 *主要 DNS 伺服器* 下，指定從屬伺服器將取得其資料的主要伺服器。若要限制對伺服器的存取，可以從清單選取其中一個 ACL。請參閱 [圖形 40.5, "DNS 伺服器：從屬區域編輯器" \[552\]](#)。

圖形 40.5 DNS 伺服器：從屬區域編輯器



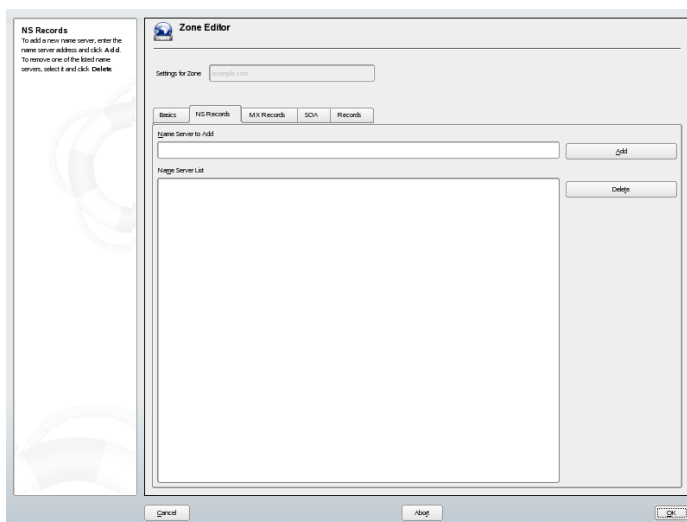
主要區域編輯器

如果在 [DNS 區域 \[551\]](#) 所述的步驟中選取的區域類型為 **主要**，就會開啟此對話方塊。對話方塊包含數頁：**基本** (先開啟的頁面)、**NS 記錄**、**MX 記錄**、**SOA** 以及 **記錄**。

區域編輯器 (NS 記錄)

此對話方塊允許您定義指定區域的替代名稱伺服器。請確定您自己的名稱伺服器包含於清單中。若要新增記錄，在 **要新增的名稱伺服器** 下輸入其名稱，然後使用 **新增** 確認動作。請參閱 [圖形 40.6, "DNS 伺服器：區域編輯器 \(NS 記錄\)" \[553\]](#)。

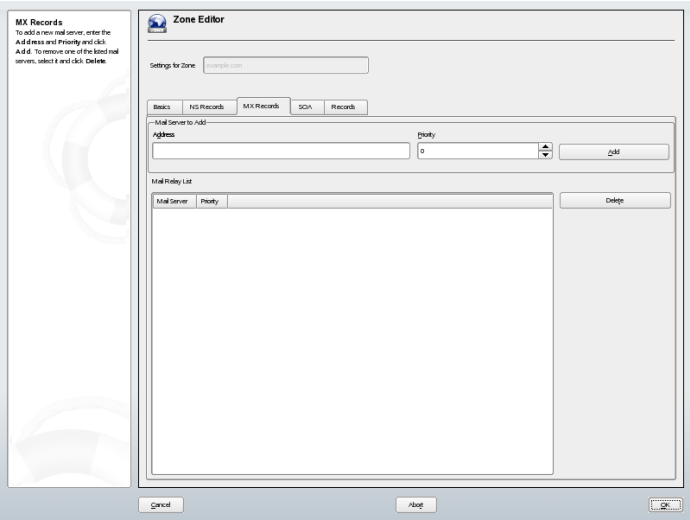
圖形 40.6 DNS 伺服器：區域編輯器 (NS 記錄)



區域編輯器 (MX 記錄)

若要新增目前區域的郵件伺服器到現有清單，請輸入對應位址及優先順序值。完成後，選取 **新增** 確認該動作。請參閱 [圖形 40.7, "DNS 伺服器：區域編輯器 \(MX 記錄\)" \[554\]](#)。

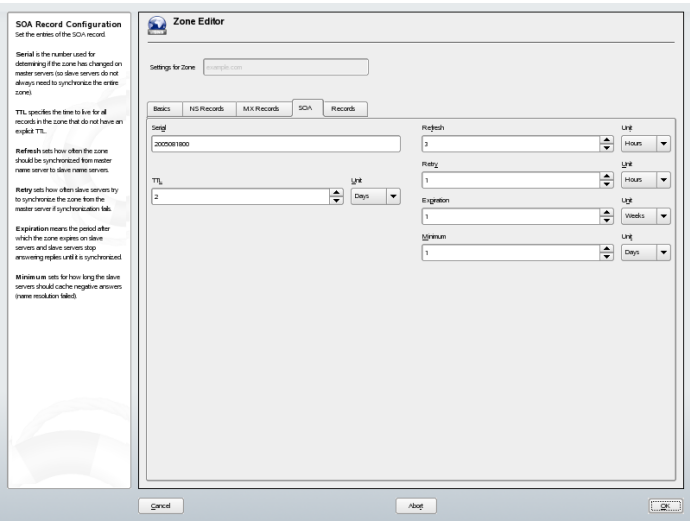
圖形 40.7 DNS 伺服器：區域編輯器 (MX 記錄)



區域編輯器 (SOA)

此頁允許您建立 SOA (授權開始) 記錄。如需個別選項的說明，請參閱 [範例 40.6](#), "檔案 /var/lib/named/world.zone" [561]。

圖形 40.8 DNS 伺服器：區域編輯器 (SOA)



區域編輯器 (記錄)

這個對話方塊是管理名稱解析。在 *記錄金鑰* 中，輸入主機名稱，然後選取其類型。*A-Record* 代表主項目。這個的值應該是 IP 位址。*CNAME* 是別名。使用 *NS* 與 *MX* 類型，以取得 *NS* 記錄與 *MX* 記錄標籤中所提供的擴充資訊的詳細或部份記錄。這三個類型可以解析成現有的 A 記錄。*PTR* 是供反向區域所使用。它是 A 記錄的相反。

40.3 啟動名稱伺服器 BIND

在 SUSE Linux 系統上，名稱伺服器 BIND (*Berkeley Internet name domain*，柏克萊網際網路名稱領域) 已經預先設定，所以安裝後即可啟動，不會有任何問題。如果您已具有可以運作的網際網路連線，而且也在 `/etc/resolv.conf` 中輸入了 `127.0.0.1` 做為 `localhost` 的名稱伺服器位址，表示您通常已經有工作名稱解析，就不需要知道提供者的 DNS。BIND 透過根名稱伺服器完成名稱解析，顯見處理程序較慢。一般而言，提供者的 DNS 應該在 `forwarders` 之下的組態檔 `/etc/named.conf` 中輸入其 IP 位址，以確保有效及安全的名稱解析。如果目前都能運作，名稱伺服器會純粹當作 *僅快取* 名稱伺服器執行。只有您設定它自己的區域時，它才會變成適當的 DNS。關於這個的簡單範例包含於 `/usr/share/doc/packages/bind/sample-config` 的文件中。

提示: 自動使用名稱伺服器資訊

根據網際網路連線或網路連線的類型，可以自動對目前的情況使用名稱伺服器資訊。若要這樣做，將檔案 `/etc/sysconfig/network/config` 中的變數 `MODIFY_NAMED_CONF_DYNAMICALY` 設定為 `yes`。

不過，在負責的機構指定正式領域前，請不要設定。即使您有自己的領域而且是由提供者管理的，最好也不要使用，否則 BIND 不會轉寄此領域的要求。例如，此領域將無法存取提供者的網頁伺服器。

若要啟動名稱伺服器，以 `root` 的身份輸入指令 `rcnamedstart`。如果右邊出現綠色的「完成」，即表示已成功啟動稱為 `named` 的名稱伺服器程序。使用 `host` 或 `dig` 程式立即測試本地系統上的名稱伺服器，應該會傳回 `localhost` 作為預設伺服器，位址為 `127.0.0.1`。如果不是這種情形，`/etc/resolv.conf` 可能包含不正確的名稱伺服器項目，或是檔案根本不存在。對於第一次測試，輸入 `host 127.0.0.1`，這通常都能成功。如果看到錯誤訊息，使用

`rcnamed status`，檢查伺服器是否確實執行。如果名稱伺服器沒有啟動或是未以預期的方式運作，通常在 `/var/log/messages` 記錄檔中可以找到原因。

若要使用提供者的名稱伺服器或將網路上已經執行的名稱伺服器作為 `forwarder`，請在 `forwarders` 下的 `options` 區段中輸入對應的 IP 位址。[範例 40.1](#)，["named.conf 中的轉寄選項" \[556\]](#) 中包含的位址只是範例。根據您自己的設定調整這些項目。

範例 40.1 *named.conf* 中的轉寄選項

```
options {
    directory "/var/lib/named";
    forwarders { 10.11.12.13; 10.11.12.14; };
    listen-on { 127.0.0.1; 192.168.0.99; };
    allow-query { 127/8; 192.168.0/24; };
    notify no;
};
```

`options` 項目後面跟著是區域的項目、`localhost` 以及 `0.0.127.in-addr.arpa`。在 `"."` 之下的 `type hint` 項目應該永遠是存在的。不需要修改對應檔案，而且應該依其原狀運作。另外也請確定每個項目的前後都有「;」，且大括號在正確的位置。變更組態檔 `/etc/named.conf` 或區域檔後，使用 `rcnamed reload` 告訴 BIND 重新讀取這些檔案。使用 `rcnamed restart` 停止和重新啟動名稱伺服器會達成相同的結果。輸入 `rcnamed stop` 隨時停止伺服器。

40.4 組態檔 */etc/named.conf*

BIND 名稱伺服器本身的所有設定儲存於檔案 `/etc/named.conf`。不過，要處理之領域的區域資料，包括主機名稱、IP 位址等，儲存於 `/var/lib/named` 目錄中的不同檔案中。詳細資訊會在稍後說明。

`/etc/named.conf` 粗略分為兩個部分。其中一個是一般設定的 `options` 區段，另一個則是由個別領域的 `zone` 項目組成。`logging` 區段和 `acl` (存取控制清單) 項目是選擇性的。註解行的開頭是 `#` 符號或 `//`。在 [範例 40.2](#)，["基本的 /etc/named.conf" \[557\]](#) 中顯示了最基本的 `/etc/named.conf`。

範例 40.2 基本的 `/etc/named.conf`

```
options {
    directory "/var/lib/named";
    forwarders { 10.0.0.1; };
    notify no;
};

zone "localhost" in {
    type master;
    file "localhost.zone";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "127.0.0.zone";
};

zone "." in {
    type hint;
    file "root.hint";
};
```

40.4.1 重要組態選項

"filename" 目錄;

指定 BIND 可以在其中找尋包含區域資料檔案的目錄。通常是 `/var/lib/named`。

forwarders { *ip-address*; };

指定無法直接解析 DNS 要求時應該轉寄的目標名稱伺服器 (大部分是提供者)。以 `10.0.0.1` 之類的 IP 位址取代 *ip-address*。

forward first;

將會在嘗試透過根名稱伺服器解析 DNS 要求前先加以轉寄。除了 `forward first`，可以寫入 `forward only` 轉寄所有要求，而不會有任何要求傳送到根名稱伺服器。這對於防火牆組態是可以理解的。

listen-on port 53 { 127.0.0.1; *ip-address*; };

告訴 BIND 哪個網路介面和哪個連接埠要接受用戶端查詢。`port 53` 不需要明確指定，因為 53 是預設連接埠。輸入 `127.0.0.1` 允許來自本地主機的要求。如果完全省略此項目，預設會使用所有介面。

listen-on-v6 port 53 {any; };

告訴 BIND 哪個連接埠應該聆聽 IPv6 用戶端要求。除了 `any` 外只能使用 `none`。就 IPv6 而言，伺服器僅接受萬用字元位址。

query-source address * port 53;

如果防火牆封鎖外送 DNS 要求，則需要這個項目。這樣會告訴 BIND 從外部的連接埠 53 張貼要求，而不是從任何高於 1024 的連接埠張貼。

query-source-v6 address * port 53;

告訴 BIND 哪個連接埠用於 IPv6 查詢。

allow-query { 127.0.0.1; net; };

定義用戶端可以張貼 DNS 要求的網路。使用如 `192.168.1/24` 的位址資訊取代 `net`。尾端的 `/24` 是網路遮罩的縮寫表示式，在此例中為 `255.255.255.0`。

allow-transfer ! *;;

控制哪個主機可以要求區域傳輸。在範例中，這類要求是使用 `! *` 完全拒絕。沒有這個項目，可以從任一處要求區域傳輸，沒有限制。

statistics-interval 0;

如果沒有這個項目，BIND 每小時在 `/var/log/messages` 中會產生數行統計資訊。指定 `0` 則完全不會顯示這些統計數字，或設定以分鐘為單位的間隔時間。

cleaning-interval 720;

這個選項定義 BIND 清除其快取的時間間隔。這樣每次清除時會觸發 `/var/log/messages` 中的項目。時間單位為分鐘。預設值是 60 分鐘。

interface-interval 0;

BIND 會定期搜尋網路介面，尋找新的或不存在的介面。如果此值設定為 `0`，就不會執行這個動作，且 BIND 僅會聆聽啟動時偵測到的介面。此外，請以分鐘為單位定義間隔時間。預設值是 60 分鐘。

notify no;

當變更區域資料或重新啟動名稱伺服器時，`no` 會防止通知其他名稱伺服器。

40.4.2 記錄

記錄的內容、方式及位置可以廣泛地在 BIND 中設定。一般而言，預設設定應該足夠。[範例 40.3, "關閉記錄的項目" \[559\]](#) 顯示出這類項目的最簡單格式，而且完全停用任何記錄。

範例 40.3 關閉記錄的項目

```
logging {  
    category default { null; };  
};
```

40.4.3 區域項目

範例 40.4 *my-domain.de* 的區域項目

```
zone "my-domain.de" in {  
    type master;  
    file "my-domain.zone";  
    notify no;  
};
```

在 `zone` 之後，指定要管理的領域名稱 (`my-domain.de`)，後面跟著 `in` 以及大括號包住的相關選項區塊，如 [範例 40.4, "my-domain.de 的區域項目" \[559\]](#) 中所示。若要定義 *slave zone*，切換 `type` 為 `slave` 並指定管理此區域的名稱伺服器為 `master` (也可能成為另一個主要的從屬)，如 [範例 40.5, "other-domain.de 的區域項目" \[559\]](#) 中所示。

範例 40.5 *other-domain.de* 的區域項目

```
zone "other-domain.de" in {  
    type slave;  
    file "slave/other-domain.zone";  
    masters { 10.0.0.1; };  
};
```

區域選項：

type master;

藉由指定 `master`，告訴 BIND 區域由本地名稱伺服器處理。這假設區域檔案已經以正確格式建立。

type slave;

此區域傳輸自另一個名稱伺服器。必須與 `masters` 一起使用。

type hint;

`hint` 類型的區域，用來設定根名稱伺服器。此區域定義可以維持原狀。

file my-domain.zone or file "slave/other-domain.zone";

此項目指定領域之區域資料所在的檔案。從屬並不需要這個檔案，因為可從另一個名稱伺服器取得此資料。若要分別主要和從屬檔案，請為從屬檔案使用目錄 `slave`。

masters { server-ip-address; };

僅從屬區域需要此項目。它指定應該傳輸區域檔案的名稱伺服器。

allow-update {! *; };

此選項控制外部寫入存取，將允許用戶端產生 DNS 項目，通常由於安全性的緣故不需要此項目。沒有此項目，則完全不允許區域更新。以下項目會產生相同的結果，因為 `!*` 有效地禁止任何這類活動。

40.5 區域檔案

需要兩種類型的區域檔案。其中一個指定 IP 位址給主機名稱，而另一個則相反：提供主機名稱給 IP 位址。

提示：在區域檔案中使用點符號

．在區域檔案中具有重要意義。如果主機名稱最後不用．結尾，會附加區域。指定完整領域名稱的主機名稱必須以．結尾，才能避免再次附加領域。缺少點符號或錯置其位置最經常造成名稱伺服器組態錯誤。

第一個要考慮的情況是負責領域 `world.cosmos` 的區域檔案 `world.zone`，如 [範例 40.6, "檔案 /var/lib/named/world.zone" \[561\]](#) 中所示。

範例 40.6 檔案 `/var/lib/named/world.zone`

```
$TTL 2D
world.cosmos. IN SOA      gateway root.world.cosmos. (
                        2003072441 ; serial
                        1D         ; refresh
                        2H         ; retry
                        1W         ; expiry
                        2D )       ; minimum

                        IN NS      gateway
                        IN MX      10 sun

gateway IN A      192.168.0.1
        IN A      192.168.1.1
sun      IN A      192.168.0.2
moon     IN A      192.168.0.3
earth    IN A      192.168.1.2
mars     IN A      192.168.1.3
www      IN CNAME  moon
```

行 1:

\$TTL 定義應該套用到此檔案中所有項目的預設有效時間。在此範例中，項目的有效時間是兩天 (2 D)。

行 2:

這是 SOA (授權開始) 控制記錄開始的地方：

- 要管理的領域名稱是第一個位置的 `world.cosmos`。這以 `.` 結尾，否則會第二次附加區域。或者，可以在此輸入 `@`，這樣會從 `/etc/named.conf` 中的對應項目擷取區域。
- 在 `IN SOA` 之後是名稱伺服器的名稱，做為此區域的主要伺服器。名稱會從 `gateway` 擴展為 `gateway.world.cosmos`，因為它沒有用 `.` 結尾。
- 後面跟著負責此名稱伺服器之人員的電子郵件地址。因為 `@` 符號已經具有特殊意義，所以在此輸入 `.` 來代替。對於 `root@world.cosmos`，項目必須讀成 `root.world.cosmos.`。必須包含在結尾，防止新增區域。
- (將所有到) 的行都包含在 SOA 記錄中。

行 3:

`serial number` 是任意號碼，每次變更此檔案時就會增加。需要通知次要名稱伺服器(從屬伺服器)這種變更。對於這種情形，十個數字的日期及執行號碼，寫法是 `YYYYMMDDNN`，已成為習慣格式。

行 4:

`refresh rate` 指定次要名稱伺服器確認區域 `serial number` 的時間間隔。在此例中，是一天。

行 5:

`retry rate` 指定次要名稱伺服器在發生錯誤時嘗試再次連絡主要伺服器的時間間隔。在此例中，是兩小時。

行 6:

`expiration time` 指定次要名稱伺服器沒有重新取得與主要伺服器的連絡時捨棄快取資料後的時間範圍。在此例中，是一週。

行 7:

`SOA` 記錄中的最後一個項目指定 `negative caching TTL`，亦即其他伺服器沒有解析 `DNS` 查詢之結果可以快取的時間。

行 9:

`IN NS` 指定負責此領域的名稱伺服器。`gateway` 會擴展為 `gateway.world.cosmos`，因為它沒有以 `.` 結尾。可能有數行會像這樣，其中一行是主要名稱伺服器，而每個次要名稱伺服器各一行。如果 `/etc/named.conf` 中的 `notify` 不是設定為 `no`，此處列出的所有名稱伺服器會收到區域資料變更的通知。

行 10:

`MX` 記錄指定接收、處理和轉寄領域 `world.cosmos` 電子郵件的郵件伺服器。在此範例中，是主機 `sun.world.cosmos`。主機名稱前的號碼是優先設定值。如果有多個 `MX` 項目，會先優先使用具有最小值的郵件伺服器，而如果郵件無法送到此伺服器，就會嘗試使用下一個較高的值。

12 至 17 行:

這些是指定給主機名稱的一或多個 `IP` 位址的實際位址記錄。列示於此的名稱沒有 `.`，因為沒有包含其領域，所以會在後面加上 `world.cosmos`。將兩個 `IP` 位址指定給主機 `gateway`，因為它有兩張網路卡。如果主機位址是傳統位址 (`IPv4`)，記錄會使用 `A` 標示。如果位址是 `IPv6` 位址，項目會使用 `A6` 標示。`IPv6` 位址之前的記錄是 `AAAA`，現在已廢除不用。

行 18:

別名 `www` 可以用來定址 `mond` (`CNAME` 表示 *canonical name* (標準名稱))。

擬領域 `in-addr.arpa` 用來反向查詢 IP 位址到主機名稱。它會以反向標記法附加到位址的網路部分。因此 `192.168.1` 會解析為 `1.168.192.in-addr.arpa`。請參閱 [範例 40.7, "反向查詢" \[563\]](#)。

範例 40.7 反向查詢

```
$TTL 2D
1.168.192.in-addr.arpa. IN SOA gateway.world.cosmos. root.world.cosmos. (
                                2003072441      ; serial
                                1D                ; refresh
                                2H                ; retry
                                1W                ; expiry
                                2D )             ; minimum

                                IN NS            gateway.world.cosmos.

1                               IN PTR          gateway.world.cosmos.
2                               IN PTR          earth.world.cosmos.
3                               IN PTR          mars.world.cosmos.
```

行 1:

\$TTL 定義套用到此處所有項目的標準 TTL。

行 2:

組態檔應該為網路 `192.168.1.0` 啟用反向查詢。假設區域稱為 `1.168.192.in-addr.arpa`，則不應該新增到主機名稱。因此，以完整格式輸入所有主機名稱，亦即包含領域以及結尾的 `.`。其餘的項目與之前 `world.cosmos` 範例中所述的項目相同。

3 至 7 行:

請參閱之前的 `world.cosmos` 範例。

行 9:

同樣地，此行指定負責此區域的名稱伺服器。不過，這一次，以完整格式輸入名稱，亦即包含領域以及結尾的 `.`。

11 至 13 行:

這些是相關主機上 IP 位址的指標記錄提示。行的開頭僅會輸入 IP 位址的最後一個部分，結尾不加上 `.`。對此附加區域 (不加上 `.in-addr.arpa`) 會造成完整 IP 位址變成反向順序。

通常，不同 BIND 版本之間的區域傳輸應該是沒有問題的。

40.6 區域資料的動態更新

動態更新 (*dynamic update*) 這個詞是指新增、變更或刪除主要伺服器的區域檔案中項目的作業。此機制於 RFC 2136 中有詳細描述。利用新增選擇性的 `allow-update` 或 `update-policy` 規則，來設定每個區域項目個別的動態更新。動態更新的區域不應該手動修改。

使用指令 `nsupdate` 將要更新的項目傳送到伺服器。對於此指令的完整語法，請查閱 `nsupdate` 的手冊頁 (`man 8 nsupdate`)。為了安全性的緣故，這類更新應該使用 TSIG 金鑰加以執行，如 節 40.7, "安全交易" [564] 所述。

40.7 安全交易

安全交易可以透過採用共用秘密金鑰 (也稱為 TSIG 金鑰) 的交易簽名 (TSIG) 來完成。本節說明如何產生及使用這類金鑰。

不同伺服器之間的通訊以及動態更新區域資料而言，是需要安全交易的。讓存取控制依靠金鑰比單純依靠 IP 位址要來得安全許多。

使用以下指令產生 TSIG 金鑰 (有關詳細資訊，請參閱 `man dnssec-keygen`):

```
dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

這樣會建立兩個檔案，名稱類似如下：

```
Khost1-host2.+157+34265.private Khost1-host2.+157+34265.key
```

金鑰本身 (如 `ejIkuCyyGJwwuN3xAteKgg==` 的字串) 在兩個檔案中都可找到。如果要用於交易，第二個檔案 (`Khost1-host2.+157+34265.key`) 必須傳輸到遠端主機，最好是以安全的方式傳輸 (例如，使用 `scp`)。在遠端伺服器上，金鑰必須包含於檔案 `/etc/named.conf` 內，才能開啟 `host1` 與 `host2` 之間的安全通訊：

```
key host1-host2. {  
    algorithm hmac-md5;  
    secret "ejIkuCyyGJwwuN3xAteKgg==";  
};
```

警告: `/etc/named.conf` 的檔案權限

確定 `/etc/named.conf` 的權限受到適當的限制。此檔案的預設值是 0640，擁有者為 `root` 及群組 `named`。另一種方法是，將金鑰移到具有特殊限定權限的額外檔案，是後續從 `/etc/named.conf` 包含進來的。

若要讓伺服器 `host1` 能夠使用 `host2` (在此範例中位址為 192.168.2.3) 的金鑰，伺服器的 `/etc/named.conf` 必須包含以下規則：

```
server 192.168.2.3 {  
    keys { host1-host2. ;};  
};
```

類比項目必須包含於 `host2` 的組態檔中。

針對 IP 位址和位址範圍定義的任何 ACL (存取控制清單，不與檔案系統 ACL 混淆) 來新增 TSIG 金鑰，以啟用異動保全性。對應項目應該看起來如下：

```
allow-update { key host1-host2. ;};
```

此主題在 `update-policy` 下的 *BIND Administrator Reference Manual* 中有詳細討論。

40.8 DNS 安全性

DNSSEC 或 DNS 安全性細述於 RFC 2535。在 BIND 手冊中的 DNSSEC 可用工具會討論到。

視為安全的區域必須具有關聯的一或多個區域金鑰。這些金鑰是使用 `dnssec-keygen` 產生，如同主機金鑰一樣。目前是使用 DSA 加密演算法產生這些金鑰。產生的公用金鑰應該包含於套用 `$INCLUDE` 規則的對應區域檔案中。

藉由指令 `dnssec-makekeyset`，產生的所有金鑰會封裝為一組，然後必須透過安全的方法傳輸到父區域。在父區域上，會使用 `dnssec-signkey` 簽署該組金鑰。接著會使用透過此指令產生的檔案，以 `dnssec-signzone` 指令簽署區域，然後會為 `/etc/named.conf` 中每個區域產生要包含在內的檔案。

40.9 若需詳細資訊

如需其他資訊，請參閱 *BIND Administrator Reference Manual*，安裝於 `/usr/share/doc/packages/bind/` 下。另外也請參閱手冊參考的 RFC 以及 BIND 隨附的手冊頁。有關 SUSE Linux 中 BIND 的最新資訊，請參閱 `/usr/share/doc/packages/bind/README.SuSE`。

使用 NIS

一旦網路中的多個 UNIX 系統想要存取常用資源，所有使用者和群組身分對於該網路中所有機器均相同就變得很重要。網路應該對使用者透明化：無論使用者是使用哪些機器，他們永遠都可以在相同環境中找到自己。這可藉由 NIS 和 NFS 服務來達成。NFS 透過網路配送檔案系統，將在 [章 42, 使用 NFS 共用檔案系統 \[575\]](#) 中討論。

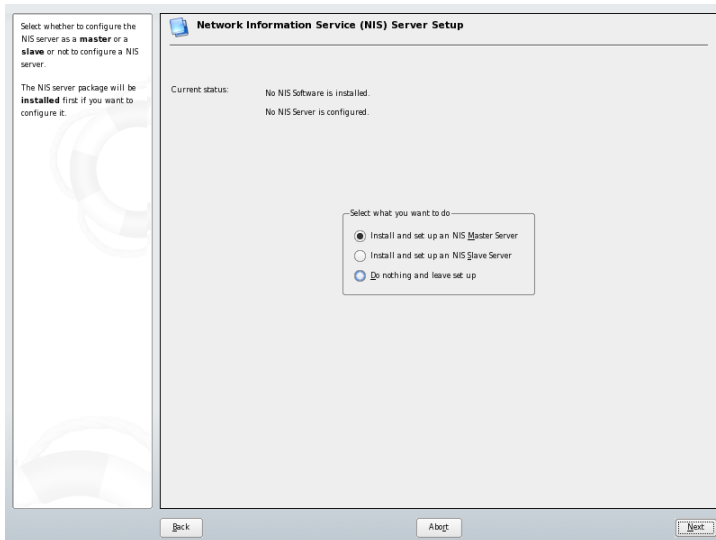
NIS (網路資訊服務) 可描述成像資料庫的服務，提供透過網路對 `/etc/passwd`、`/etc/shadow` 和 `/etc/group` 內容的存取權。NIS 也可用於其他用途 (例如，提供如 `/etc/hosts` 或 `/etc/services` 檔案的內容以供使用)，但這超過本介紹的範圍。人們通常將 NIS 視為 *YP*，因它與網路「黃頁」的概念相似。

41.1 使用 YaST 設定 NIS 伺服器

若要進行組態，從 YaST 模組 *網路服務* 選取 *NIS 伺服器*。如果目前在您的網路中沒有 NIS 伺服器存在，請在下一個畫面中啟用 *安裝與設定主要 NIS 伺服器*。YaST 會立即安裝需要的套件。

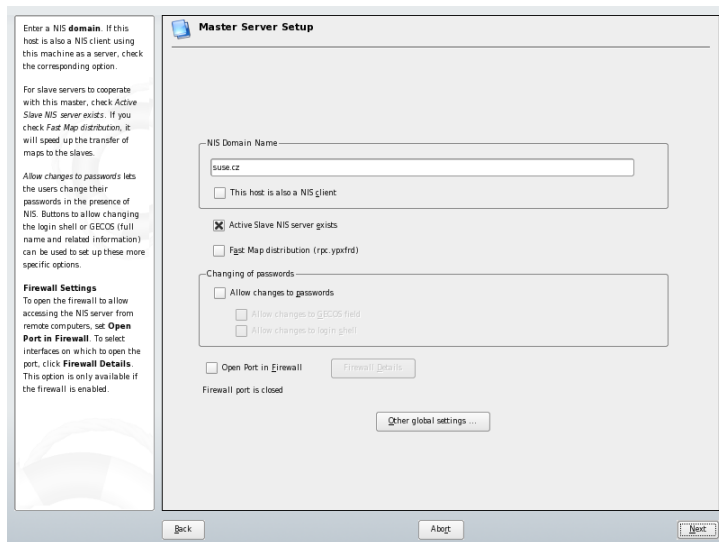
如果已安裝 NIS 軟體，按一下 *建立主要 NIS 伺服器*。如果您已經有 NIS 伺服器 (*主要伺服器*)，便可以新增 NIS 次要伺服器 (例如，如果您想要設定新的子網路)。首先，主要伺服器的組態如描述。按一下 *不要儲存離開設定* 會不儲存變更，回到「YaST 控制中心」。

圖形 41.1 NIS 伺服器設定



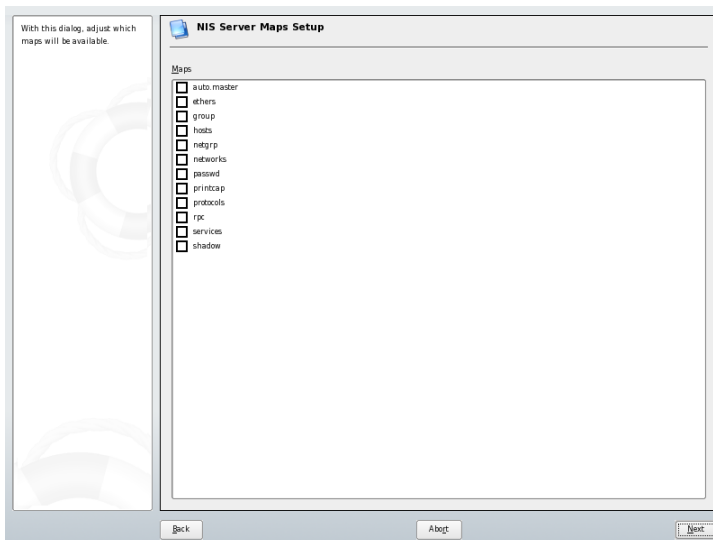
安裝所有的套件後，在設定對話方塊的上方輸入NIS領域名稱，會在 圖形 41.1, "NIS 伺服器設定" [568] 顯示。以核取方塊定義主機是否也是 NIS 用戶端，可讓使用者登入並從 NIS 伺服器存取資料。勾選想要套用的所有方塊，包括 變更密碼選項。按一下 其他全域設定設定其他的設定。現在正存取可變更來源目錄，合併密碼並設定最少使用者和群組 ID 的畫面。按一下 確定 以回到主要對話方塊。按一下 下一步 繼續設定。

圖形 41.2 主要伺服器設定



在下一個畫面，指定要變成可用的地圖。按一下 下一步帶您往下一個可決定是否允許主機查詢 NIS 伺服器的畫面。您可以新增、刪除和編輯主機。按一下 完成 儲存變更，並結束設定對話方塊。

圖形 41.3 NIS 伺服器主要設定



若要設定網路中其他 NIS (次要伺服器)，請現在啟用 安裝與設定次要 NIS 伺服器。如果已安裝 NIS 軟體，按一下 建立 NIS 次要伺服器 然後按一下 下一步 繼續。在下一個畫面，輸入 NIS 領域名稱和勾選套用的方塊。

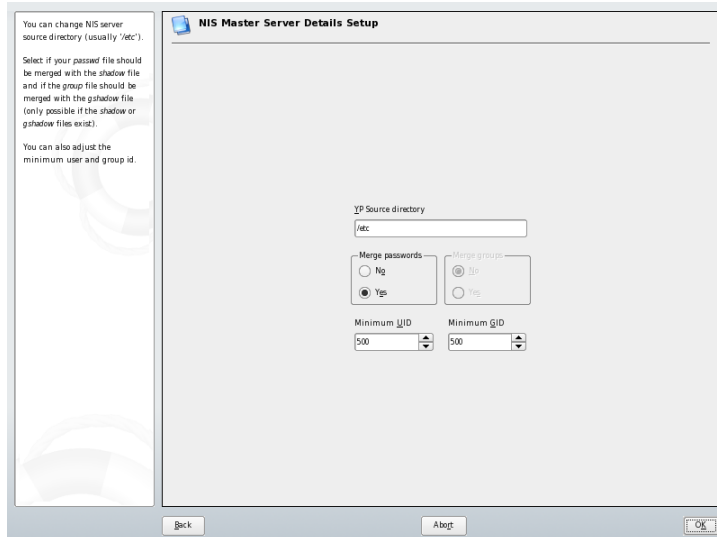
若要讓您網路中的使用者 (包括本地使用者和透過 NIS 伺服器管理的使用者) 變更 NIS 伺服器的密碼 (以指令 `yppasswd`)，請啟用對應選項。如此便可使用 允許變更 GECOS 欄位 和 允許變更登入外圍程序 選項。"GECOS" 表示使用者可使用指令 `ypchfn` 變更其名稱與位址設定。例如，"SHELL" 讓使用者以 `ypchsh` 指令變更其預設外圍程序，以便從 `bash` 切換到 `sh`。

按一下 其他全域設定 設定其他的設定。現在存取畫面，如 圖形 41.4, "變更 NIS 伺服器的目錄與同步化檔案" [571] 中所示，您可變更 NIS 伺服器的來源目錄 (預設為 `/etc`)。此外，在此處可合併密碼與群組。設定值應該為 是，如此檔案 (`/etc/passwd`、`/etc/shadow` 和 `/etc/group`) 便可同步。同時需決定最小使用者與群組 ID。按一下 確定 以確認您的設定值並回到上一個畫面。

設定完成後，按一下 下一步 到下一個畫面。在下一個對話方塊，勾選可用的地圖，然後按一下 下一步 繼續。在最後一個畫面，輸入可查詢 NIS 伺服器的主機。您可以按一下 適當按鈕 新增、編輯或刪除主機。按一下 完成 儲存變更，並結束設定。

然後按 下一步。

圖形 41.4 變更 NIS 伺服器的目錄與同步化檔案



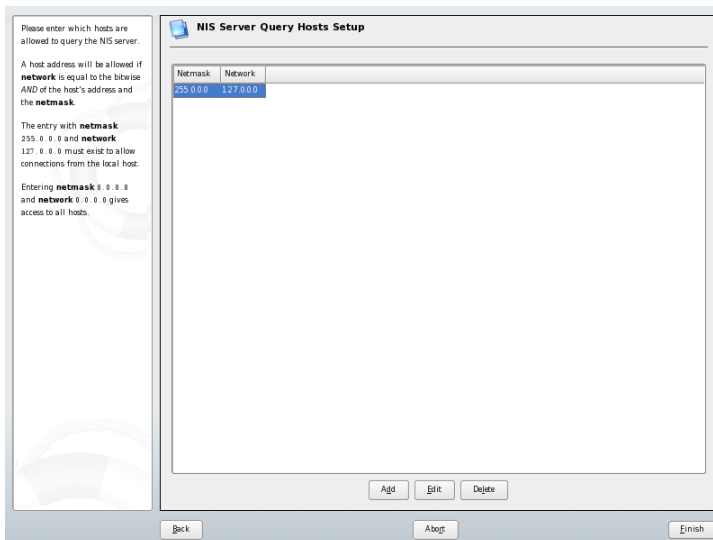
如果您先前啟用了 可用的從屬 NIS 伺服器存在，請輸入要做為次要伺服器的主機名稱然後按 下一步。如果您不使用次要伺服器，會略過次要伺服器組態，直接繼續資料庫組態的對話方塊。此處，請指定 映射，從 NIS 伺服器傳輸到用戶端的部分資料庫。預設值通常足夠使用。

下一步 會繼續最後一個對話方塊，如 圖形 41.5, "設定 NIS 伺服器的要求權限" [572] 中所示。指定可從哪些網路傳送要求至 NIS 伺服器。通常，這是您的內部網路。在此例中，應該是下列兩個項目：

```
255.0.0.0      127.0.0.0
0.0.0.0        0.0.0.0
```

第一個項目讓您可從自己的主機連線，即 NIS 伺服器。第二個讓所有具有相同網路存取權的主機，可以傳送要求至伺服器。

圖形 41.5 設定 NIS 伺服器的要求權限



重要: 自動防火牆組態

如果您的系統上有防火牆在作用中 (SUSEfirewall2), YaST 會在選取 *在防火牆中開啟埠* 時, 啟用 portmap 服務以調整其 NIS 伺服器的組態。

41.2 設定 NIS 用戶端

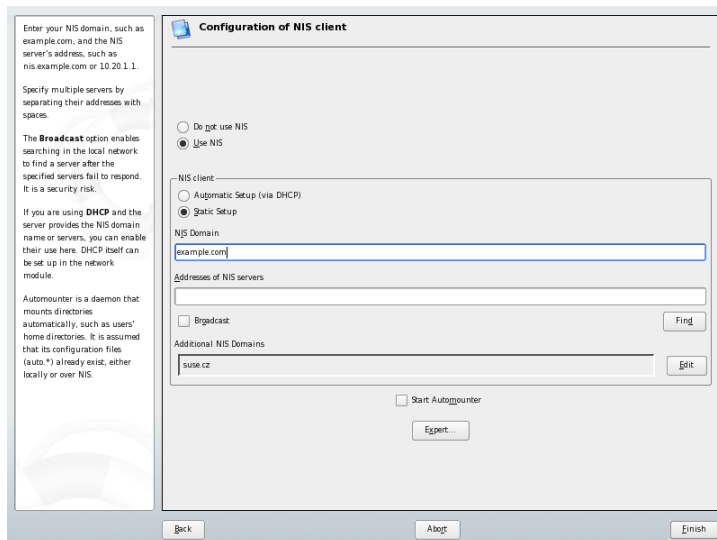
使用此模組以設定 NIS 用戶端。在選擇使用 NIS 以及自動掛載器 (視狀況而定) 之後, 會開啟此對話方塊。選擇主機是否有靜態 IP 位址, 或是要接受 DHCP 所發出的 IP 位址。DHCP 也提供 NIS 領域和 NIS 伺服器。若需有關 DHCP 的詳細資訊, 請參閱 [章 43, DHCP \[581\]](#)。如果使用靜態 IP 位址, 請手動指定 NIS 領域和 NIS 伺服器。請參閱 [圖形 41.6, "設定 NIS 伺服器的領域和位址" \[573\]](#)。尋找可讓 YaST 搜尋網路中作用中的 NIS 伺服器。*廣播* 可讓您在指定伺服器無法回應時, 搜尋內部網路, 找尋伺服器。

您也可以在 *NIS 伺服器位址* 輸入位址, 並以空格分隔, 指定多個伺服器。

在進階設定中，如果您不想要其他主機查詢您的用戶端所使用的伺服器，請關閉 **回答遠端主機**。勾選 **中斷伺服器**，則用戶端可接收透過未授權的連接埠通訊的伺服器之回覆。若需進一步資訊，請參閱 `man ypbind`。

完成設定後，按一下 **完成 變更他們**，並回到「YaST 控制中心」。

圖形 41.6 設定 NIS 伺服器的領域和位址



使用 NFS 共用檔案系統

如 [章 41, 使用 NIS \[567\]](#) 中所述，與 NIS 一起使用的 NFS 讓網路對使用者而言透明化。使用 NFS，可以透過網路配送檔案系統。使用者登入哪個終端機都沒關係。他永遠可在相同環境中找到自己。

和 NIS 相似，NFS 是非對稱服務。有 NFS 伺服器 and NFS 用戶端。一台機器可同時扮演這兩種角色--它可透過網路提供檔案系統(輸出)，也可以從其他主機裝載檔案系統(輸入)。一般而言，這些伺服器必須具有非常大的硬碟容量，其他用戶端裝載其檔案系統。

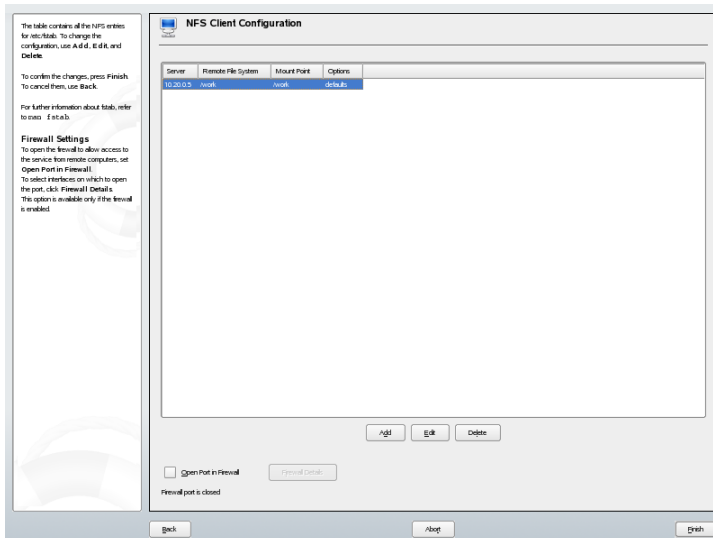
重要: DNS 所需

原則上，可以只使用 IP 位址來進行所有的輸出。不過，若要避免逾時，您應該具有運作中的 DNS 系統。這是基於記錄目的所必備的，因為裝載的精靈會進行反向查詢。

42.1 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。使用 YaST 模組 *NFS 用戶端* 是最簡單的達成方式。僅需鍵入 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裝載點。在第一個對話方塊中，按一下 *新增* 之後便可完成。按一下 *開啟防火牆* 中的 *連接埠* 開啟防火牆，可從遠端電腦存取服務。在核取方塊旁顯示防火牆的狀態。請按一下 *確定* 來儲存變更。請參閱 [圖形 42.1, "使用 YaST 的 NFS 用戶端組態" \[576\]](#)。

圖形 42.1 使用 YaST 的 NFS 用戶端組態



42.2 手動輸入檔案系統

您可輕易地從 NFS 伺服器手動輸入檔案系統。唯一的先決條件是執行中的 RPC 埠對應程式，以 root 使用者身份輸入指令 `rpcportmap start` 便可啟動。只要符合此先決條件，便可像在本地硬碟一樣，以下列語法使用指令 `mount`，將對應機器上輸出的遠端檔案系統裝載在檔案系統中：

```
mount host:remote-path local-path
```

例如，如果要輸入來自機器 Sun 的使用者目錄，請使用以下指令：

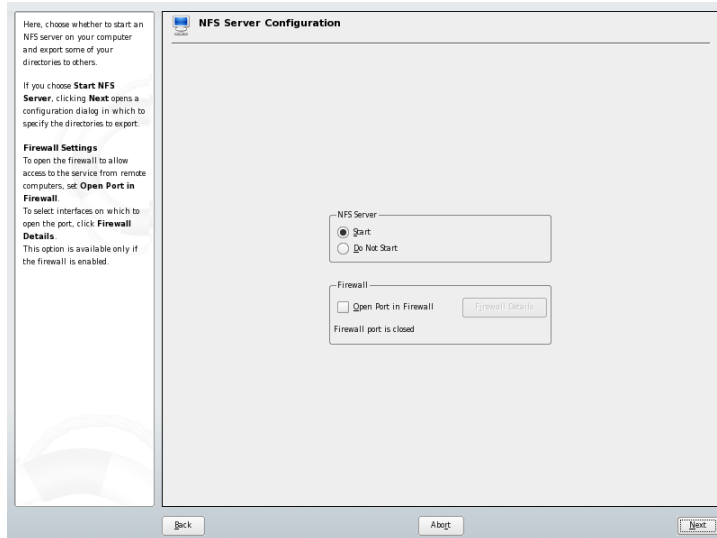
```
mount sun: /home /home
```

42.3 以 YaST 輸出檔案系統

使用 YaST，將您網路中的主機轉變為 NFS 伺服器 可將目錄和檔案輸出到所有授與存取權的主機之伺服器。可不在每一個主機上本地安裝時，提供應用程式給團隊中的所有工作夥伴。若要安裝此種伺服器，請啟動 YaST 並選取 **網路服**

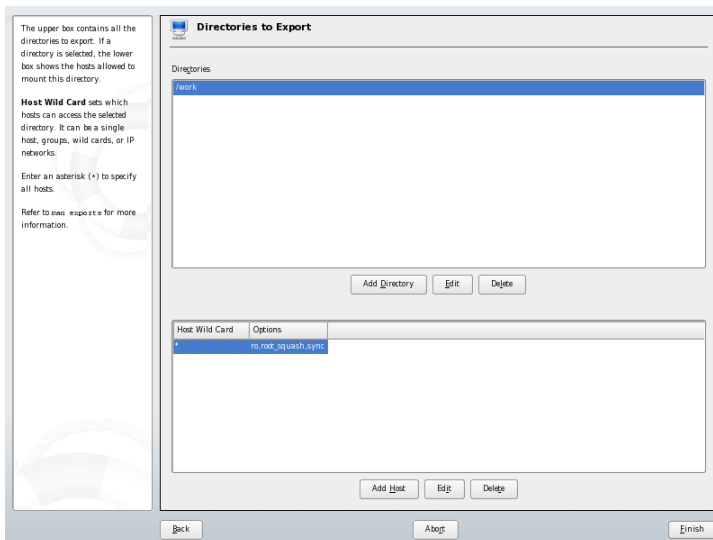
務 → *NFS 伺服器*。會開啟像 圖形 42.2, "*NFS 伺服器組態工具*" [577] 中的對話方塊。

圖形 42.2 *NFS 伺服器組態工具*



下一步，啟用 *啟動NFS 伺服器*，並按一下 *下一步*。在上方的文字欄位中，鍵入要輸入的目錄。在下方，輸入可以存取它們的主機。在 圖形 42.3, "*使用 YaST 設定 NFS 伺服器*" [578] 中顯示了此對話方塊。每一台主機有四個選項可設定：單一主機、網路群組、萬用字元和IP 網路。man exports 提供這些選項更詳盡的說明。*結束* 可完成組態。

圖形 42.3 使用 YaST 設定 NFS 伺服器



重要：自動防火牆組態

如果您的系統上有防火牆在作用中 (SUSEfirewall2)，YaST 會在選取 在防火牆中開啟埠 時，啟用 `nfs` 服務以調整其 NFS 伺服器組態。

42.4 手動輸出檔案系統

如果您不想使用 YaST，請確定 NFS 伺服器上有執行下列系統：

- RPC portmapper (`portmap`)
- RPC 裝載精靈 (`rpc.mountd`)
- RPC NFS 精靈 (`rpc.nfsd`)

若要在系統啟動時使用程序檔 `/etc/init.d/portmap` 和 `/etc/init.d/nfsserver` 來啟動這些服務，請輸入指令 `insserv /etc/init.d/nfsserver` 和 `insserv /etc/init.d/portmap`。同時在組態檔 `/etc/exports` 中定義哪些檔案系統應輸出至哪些主機。

對每一個要輸出的目錄，必須有一行來設定哪些主機可以何種權限存取該目錄。此目錄中的所有子目錄也會自動輸出。被授權的機器通常會以完整名稱來指定 (包括領域名稱)，但是也可能會使用萬用字元，如 * 或 ? (以 Bash 外圍程式相同方式擴展)。如果在此處沒有指定機器，任何機器均可以指定權限輸入此檔案系統。

在機器名稱之後以括號設定輸出檔案系統的權限。最重要的選項顯示於 [表格 42.1, "輸出檔案系統的權限" \[579\]](#) 中。

表格 42.1 輸出檔案系統的權限

選項	意義
ro	檔案系統以唯讀權限輸出 (預設值)。
rw	檔案系統以讀寫權限輸出。
root_squash	這將可確保輸入機器的 root 使用者在此檔案系統上沒有 root 權限。可藉由將使用者 ID 65534 指定給具有使用者 ID 0 (root) 的使用者來達成。此使用者 ID 應設為 nobody (預設值)。
no_root_squash	請勿將使用者 ID 0 指定為使用者 ID 65534，請維持 root 權限的有效性。
link_relative	將絕對連結 (這些是以 / 開頭) 轉換為 .. / 的順序。只有在裝載機器的整個檔案系統時才有用 (預設值)。
link_absolute	符號連結保留不動。
map_identity	使用者 ID 在用戶端與伺服器上是完全相同的 (預設值)。
map_daemon	用戶端與伺服器沒有相符的使用者 ID。這會讓 nfsd 建立使用者 ID 的轉換表。此工作必須有 ugidd 精靈。

您的 exports 檔案看起來會像 [範例 42.1, "/etc/exports" \[580\]](#)。 /etc /exports 是由 mountd 與 nfsd 讀取。如果您對此檔案做任何變更，請重新啟動 mountd 和 nfsd 讓變更生效。使用 rcnfsserver restart 可輕易完成。

範例 42.1 */etc/exports*

```
#
# /etc/exports
#
/home          sun(rw)   venus(rw)
/usr/X11       sun(ro)   venus(ro)
/usr/lib/texmf sun(ro)   venus(rw)
/              earth(ro,root_squash)
/home/ftp      (ro)
# End of exports
```


DHCP

動態主機組態通訊協定 (DHCP, dynamic host configuration protocol) 的用途是為了從伺服器集中指定網路設定，而不是在每個工作站上分別設定。設定要使用 DHCP 的主機對於自己的靜態位址並沒有控制權。它可以根據伺服器的指示完全且自動地設定自己本身。

使用 DHCP 的一個方式是識別每個使用網路卡硬體位址 (大部份的情況下是固定的) 的用戶端，以後每次用戶端連接伺服器時，將提供相同的設定值給用戶端。或者 DHCP 也可以設定成讓伺服器從專門使用的位址集區動態指定位址給每個有關的用戶端。在後者的情形中，DHCP 伺服器在每次從用戶端接受到要求時指定相同的位址給用戶端，即使經過較長的時間也是一樣。當然只有在網路的用戶端數量不超過位址數量時才能這樣運作。

由於這些可以採用的方法，因此 DHCP 透過這兩個方式讓系統管理員的工作變得更容易。任何有關位址及一般網路組態的變更 (甚至更大的變更)，只要編輯伺服器的組態檔，即可集中執行。這樣比重新設定許多工作站便利許多。另外也更易於整合機器 (特別是新機器) 到網路中，因為可以從集區對其指定 IP 位址。從 DHCP 伺服器擷取適當網路設定，對於經常使用不同網路的筆記型電腦特別有用。

DHCP 伺服器不僅提供 IP 位址和網路遮罩，也提供主機名稱、領域名稱、閘道和名稱伺服器位址供用戶端使用。除此之外，DHCP 也允許集中設定一些其他的參數，例如，用戶端可以輪詢目前時間的時間伺服器或甚至是列印伺服器。

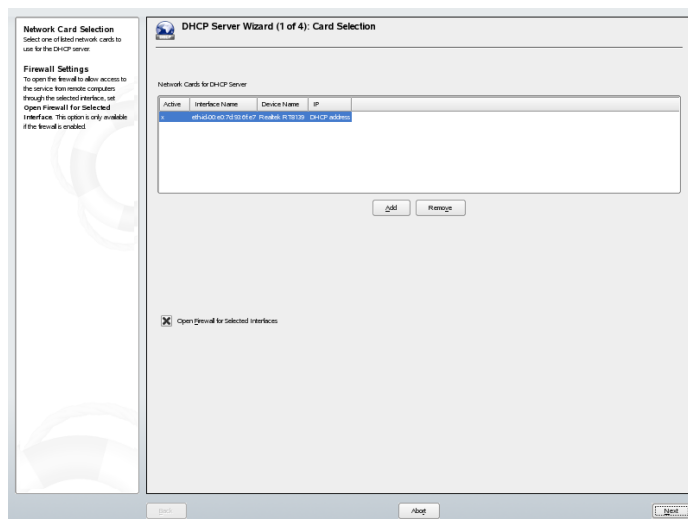
43.1 使用 YaST 設定 DHCP 伺服器

第一次啟動模組時，會啟動精靈以提示您決定有關伺服器管理的一些基本設定。完成此最初設定程序會產生一個非常基本的伺服器組態，應該在其基本方面都能運作。進階模式可以處理更多進階設定任務。

選擇介面卡

在第一步驟中，YaST 會尋找系統上可用的網路介面，然後顯示清單。從清單中，選取 DHCP 伺服器要連接的介面，並按一下 **新增** 然後選取 **開啟** 選取介面的防火牆以開啟此介面的防火牆。請參閱 [圖形 43.1, "DHCP 伺服器：選擇介面卡" \[582\]](#)。

圖形 43.1 DHCP 伺服器：選擇介面卡



全域設定

在輸入欄位，提供 DHCP 伺服器應該管理的所有用戶端的網路細節。這些細節包括領域名稱、時間伺服器的位址、主要及次要名稱伺服器的位址、列印和 WINS 伺服器的位址 (供同時具有 Windows 與 Linux 用戶端的混和網路使用)、閘道位址以及租用時間。請參閱 [圖形 43.2, "DHCP 伺服器：全域設定" \[583\]](#)。

圖形 43.2 DHCP 伺服器：全域設定

Global Settings
Here, make several DHCP settings.

Domain Name sets the domain for which the DHCP server leases IP to clients.

Primary Name Server IP and Secondary Name Server IP offer these name servers to the DHCP clients. These values must be IP addresses.

Default Gateway inserts the value as the default route in the routing table of clients.

Time Server tells clients to use this server for time synchronization.

Print Server offers this server as the default print server.

WINS Server offers this server as the WINS server (Windows Internet Naming Service).

Default Lease Time sets the time after which the leased IP expires and the client must ask for an IP again.

DHCP Server Wizard (2 of 4): Global Settings

Domain Name:

Primary Name Server IP:

Secondary Name Server IP:

Default Gateway (Router):

WINS Server:

Default Lease Time: Hours

Buttons:

動態 DHCP

在此步驟中，設定應該如何將動態 IP 位址指定給用戶端。若要這樣做，指定伺服器可以指定位址給 DHCP 用戶端的 IP 範圍。所有這些位址應該涵蓋在相同的網路遮罩下。另外也指定用戶端可以保留其 IP 位址、不用要求延續出租的時間限制。或者，指定最長出租時間，也就是伺服器保留特定用戶端之 IP 位址的時間。請參閱 [圖形 43.3, "DHCP 伺服器：動態 DHCP" \[584\]](#)。

圖形 43.3 DHCP 伺服器：動態 DHCP

The screenshot shows the 'DHCP Server Wizard (3 of 4): Dynamic DHCP' window. On the left is a sidebar with two sections: 'IP Address Range' and 'Lease Time'. The main area is divided into two sections. The top section, 'IP Address Range', contains fields for 'Current Network' (10.20.0.0) and 'Current Network' (255.255.255.0), and a 'Lease Time' section with 'Default' (4) and 'Maximum' (2) fields. The bottom section, 'Lease Time', contains a 'Lease Time' section with 'Default' (4) and 'Maximum' (2) fields. The window has 'Back', 'Next', and 'Cancel' buttons at the bottom.

IP Address Range
Here, set the First IP Address and the Last IP Address to lease to the clients. These addresses must have the same network. For instance, 192.168.1.1 and 192.168.1.64.

Lease Time
Here, set the Default Lease time for the current IP address range, which sets the optimal IP refreshing time for clients.
Maximum (optional value) sets the maximum time period for which the IP is blocked for the client on the DHCP server.

DHCP Server Wizard (3 of 4): Dynamic DHCP

IP Address Range:

Current Network: 10.20.0.0

Current Network: 255.255.255.0

First IP Address: 10.20.0.1

Last IP Address: 10.20.0.255

Lease Time:

Default: 4 Hours

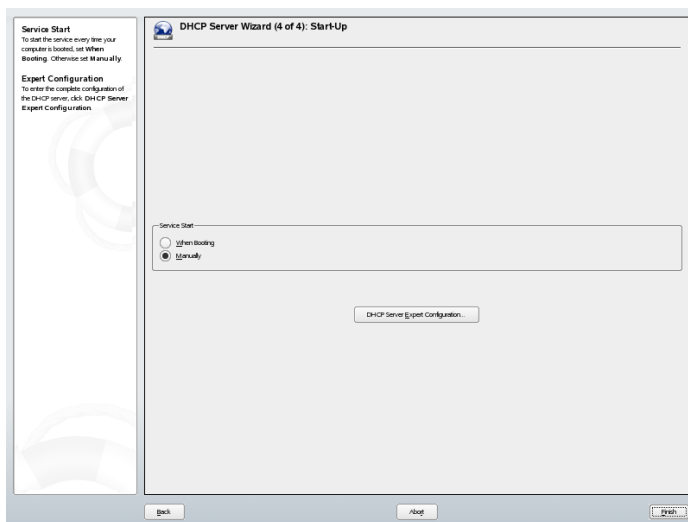
Maximum: 2 Days

Back Next Cancel

完成組態及設定啟動模式

在組態精靈的第三步驟後，會顯示最後一個對話方塊，定義DHCP伺服器的啟動方式。在此您可以指定是否在系統開機時自動啟動DHCP伺服器，或在需要時才手動啟動(例如，為了測試)。按一下 完成 完成伺服器的組態。請參閱 圖形 43.4, "DHCP 伺服器：啟動" [585]。

圖形 43.4 DHCP 伺服器：啟動



43.2 DHCP 軟體套件

DHCP 伺服器與 DHCP 用戶端皆可用於 SUSE Linux。可用的 DHCP 伺服器是 `dhcpd` (由 Internet Software Consortium 發佈)。在用戶端，選擇兩種不同的 DHCP 用戶端程式：`dhclient` (也是 ISC 發佈的) 以及 `dhcpd` 套件中的 DHCP 用戶端精靈。

SUSE Linux 預設會安裝 `dhcpd`。該程式易於使用，而且每次系統開機時就會自動啟動來等待 DHCP 伺服器。它不需要組態檔來執行其工作，而且大部份的標準設定可以直接使用。如果是更複雜的情形，請使用 ISC `dhclient`，這是透過組態檔 `/etc/dhclient.conf` 來控制的。

43.3 DHCP 伺服器 `dhcpd`

任何 DHCP 系統的核心都是動態主機組態通訊協定精靈。這個伺服器會依照組態檔 `/etc/dhcpd.conf` 中定義的設定，出租位址並監看其使用情形。藉由變更此檔案中的參數及值，系統管理員可以透過數種方式影響程式的行為。請參

閱 範例 43.1, "組態檔 /etc/dhcpd.conf" [586] 中的 /etc/dhcpd.conf 基本範例檔案。

範例 43.1 組態檔 /etc/dhcpd.conf

```
default-lease-time 600;          # 10 minutes
max-lease-time 7200;             # 2 hours

option domain-name "cosmos.all";
option domain-name-servers 192.168.1.1, 192.168.1.2;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option
subnet-mask 255.255.255.0;

subnet 192.168.1.0 netmask 255.255.255.0
{
    range 192.168.1.10 192.168.1.20;
    range 192.168.1.100 192.168.1.200;
}
```

這個簡單的組態檔應足以讓 DHCP 伺服器指定網路中的 IP 位址。請確定每行結尾都插入分號，否則將不會啟動 dhcpd。

上述範例檔可以分為三個部分。第一個部份定義要求用戶端申請續約前 IP 位址預設的租用秒數 (default-lease-time)。其中也包含機器保留 DHCP 伺服器指定之 IP 位址、不用申請續約的最長時間的陳述式 (max-lease-time)。

在第二部份中，一些基本網路參數定義於全域層級：

- 行 option domain-name 定義了您網路的預設領域。
- 利用 option domain-name-servers 項目，最多可指定三個 DNS 伺服器值，用來解析 IP 位址為主機名稱，反之亦然。理想上來說，設定 DHCP 前請先設定您機器上或網路上其他地方的名稱伺服器。該名稱伺服器也可以為每個動態位址定義主機名稱，反之亦然。若要瞭解如何設定您自己的名稱伺服器，請參閱 章 40, 領域名稱系統 [547]。
- option broadcast-address 這一行定義了要求用戶端所使用的廣播位址。
- 利用 option routers，告訴伺服器要將無法送至區域網路上主機的資料封包傳送到什麼地方(依據提供的來源和目的主機位址及子網路遮罩)。在大多數的情況中，特別是較小的網路，此路由與網際網路閘道是完全一樣的。

- 利用 `option subnet-mask`，指定指定給用戶端的網路遮罩。

檔案的最後一個部分是為了定義網路，包括子網路遮罩。為了完成，指定 DHCP 精靈用來指定 IP 位址給有關用戶端的位址範圍。在此範例中，用戶端的任何指定位址可能介於 192.168.1.10 與 192.168.1.20 以及 192.168.1.100 與 192.168.1.200 之間。

編輯這幾行後，應該能夠使用指令 `rcdhcpd start` 啟動 DHCP 精靈。應該是立即就能使用。使用指令 `rcdhcpd check-syntax` 執行簡短的語法檢查。如果在組態上遇到沒有預期的問題，例如何伺服器出現錯誤而中止或在啟動時沒有傳回 `done`，則查詢主系統記錄 `/var/log/messages` 或主控台 10 (`Ctrl` + `Alt` + `F10`) 上的資訊，應該就可找出問題所在。

在預設 SUSE Linux 系統上，為了安全的緣故，會在 `chroot` 環境中啟動 DHCP 精靈。組態檔必須複製到 `chroot` 環境，如此精靈才可以找到這些檔案。通常是不需要擔心發生這種情形，因為指令 `rcdhcpd start` 會自動複製檔案。

43.3.1 使用固定 IP 位址的用戶端

如上面所說，DHCP 也可以在每個要求時，指定預先定義的靜態位址給特定用戶端。明確指定的位址永遠比集區的動態位址優先採用。不僅如此，靜態位址永遠不會過期，與動態位址不同，例如，可用的位址不夠時，伺服器則需要在用戶端間重新散佈位址。

若要識別設定為靜態位址的用戶端，`dhcpd` 會使用硬體位址，這是全球唯一、固定的數字代碼，由六對八位元組組成，用來識別所有網路設備 (例如，00:00:45:12:EE:F4)。如果相對的行 (如 [範例 43.2, "組態檔的增加部分" \[587\]](#) 中所示) 新增到 [範例 43.1, "組態檔 /etc/dhcpd.conf" \[586\]](#) 的組態檔，則 DHCP 精靈在任何情況下，永遠會指定相同的資料集給對應的用戶端。

範例 43.2 組態檔的增加部分

```
host earth {
hardware ethernet 00:00:45:12:EE:F4;
fixed-address 192.168.1.21;
}
```

對應用戶端的名稱 (`hosthostname`, `earth`) 會輸入到第一行，而在第二行則是 MAC 位址。在 Linux 主機上，使用指令 `ifstatus` 後面接著網路設備 (例

如，eth0)，可以判定此位址。如有必要，首先使用 `ifup eth0` 啟用網路卡。輸出應該包含如下內容

```
link/ether 00:00:45:12:EE:F4
```

在上述範例中，具有網路卡、且 MAC 位址為 00:00:45:12:EE:F4 的用戶端會自動指定 IP 位址 192.168.1.21 及主機名稱 **earth**。幾乎所有情況中輸入的硬體類型會是 ethernet，儘管通常在 IBM 系統上找到的是 token-ring，也是可以支援的。

43.3.2 SUSE Linux 系統

為了改善安全性，ISC 之 DHCP 伺服器的 SUSE 版本隨附應用 Ari Edelkind 的 non-root/chroot 修補程式。這樣可讓 dhcpd 利用使用者 ID nobody 執行，而且也能執行於 chroot 環境下 (/var/lib/dhcp)。如要這樣做，組態檔 dhcpd.conf 必須位於 /var/lib/dhcp/etc。init 程序檔啟動時會自動複製檔案到此目錄。

伺服器與此功能相關的行為，是透過檔案 /etc/sysconfig/dhcpd 中的項目來控制的。若是不要在 chroot 環境下執行 dhcpd，請將 /etc/sysconfig/dhcpd 中的 DHCPD_RUN_CHROOTED 變數設定為 "no"。

若要讓 dhcpd 從 chroot 環境內解析主機名稱，必須也要複製以下其他組態檔案：

- /etc/localtime
- /etc/host.conf
- /etc/hosts
- /etc/resolv.conf

啟動 init 程序檔時，這些檔案會複製到 /var/lib/dhcp/etc/。如果 /etc/ppp/ip-up 之類的程序檔動態修改這些檔案時，則進行所需變更時，也要考慮到這些複本。不過，如果組態檔僅指定 IP 位址 (而不是指定主機名稱) 時，則不需要擔心。

如果您的組態包括應該要複製到 chroot 環境的其他檔案，請在檔案 /etc/sysconfig/dhcpd 中的變數 DHCPD_CONF_INCLUDE_FILES 下指定這些檔

案。若是要確定重新啟動系統記錄精靈時，仍保持 DHCP 記錄設備持續運作，必須要在檔案 `/etc/sysconfig/syslog` 中變數 `SYSLOGD_PARAMS` 下新增選項 `"-a /var/lib/dhcp/dev/log"`。

43.4 若需詳細資訊

有關 DHCP 的詳細資訊請參閱 *Internet Software Consortium* (<http://www.isc.org/products/DHCP/>) 的網站。在 `dhcpd`、`dhcpd.conf`、`dhcpd.leases` 以及 `dhcp-options` 中的手冊頁也可以找到資訊。

使用 `xntp` 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，通常有必要同步化所有機器的系統時間，但手動調整時間並非理想的方法。`xntp` 提供解決這些問題的機制。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機參考的時鐘，例如收音機控制的時鐘。

44.1 使用 YaST 設定 NTP 用戶端

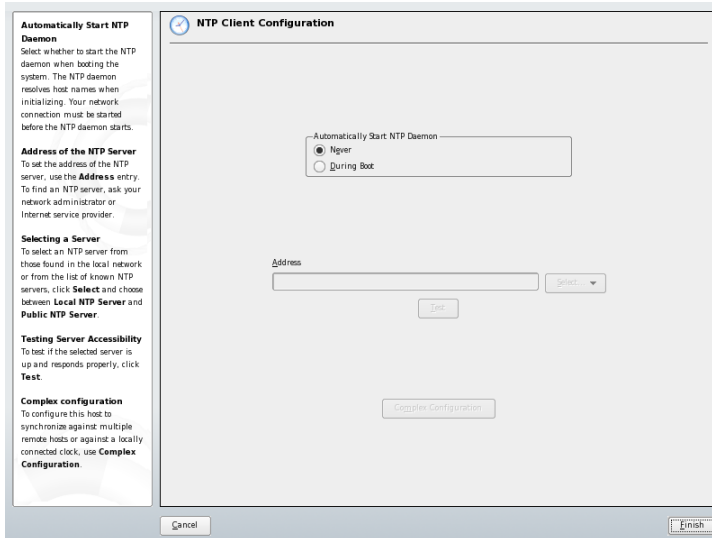
`xntp` 會預先設定為使用本機電腦時鐘做為時間參考。不過，使用 (BIOS) 時鐘，僅供萬一沒有更精確的時間來源時備用。SUSE Linux 利於使用 YaST 設定 NTP 用戶端。不執行 `SUSEfirewall` 的用戶端，可使用快速或複雜的設定，因為他們受到內部網路的保護。接下來將描述這兩種組態。

44.1.1 快速 NTP 用戶端組態

簡單的 NTP 用戶端組態 (網路服務 → NTP 用戶端) 是由兩個對話方塊所組成。請在第一個對話方塊中設定 `xntpd` 的開始模式以及要查詢的伺服器。當系統開機

時，若要自動啟動 `xntpd`，請按一下 **開機時**。然後按一下 **選取** 存取第二個對話方塊，並在其中為網路選取適當的時間伺服器。

圖形 44.1 *YaST: 設定 NTP 用戶端*

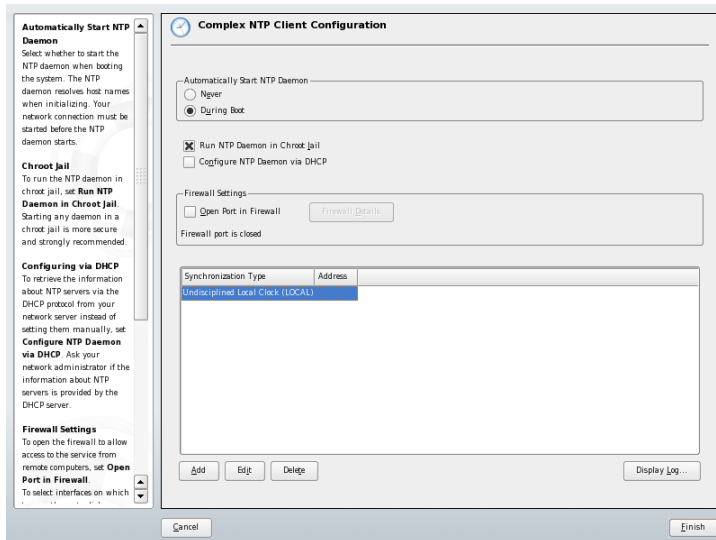


在詳細的伺服器選取對話方塊中，決定執行時間同步化時，要使用區域網路中的時間伺服器 (**本地 NTP 伺服器**)，還是使用可以處理時區的網際網路時間伺服器 (**公用 NTP 伺服器**)。若需本機時間伺服器，請按一下 **查詢**，開始 SLP 查詢，在網路中尋找可用的時間伺服器。從搜尋結果清單中選取最合適的時間伺服器，並按一下 **確定**，結束對話方塊。若需公用時間伺服器，請選取您的國家 (時區) 並從 **公用 NTP 伺服器** 下的清單選取適當的伺服器，然後按一下 **確定**，結束對話方塊。在主對話方塊中，使用 **測試** 來測試所選取伺服器的可用性，並按一下 **完成**，結束對話方塊。

44.1.2 複雜的 NTP 用戶端組態

依照快速組態所述選取了啟動模式後，即可從 *NTP 用戶端* 模組主對話方塊中的複雜組態下，存取 NTP 用戶端的複雜組態，如 [圖形 44.1, "YaST: 設定 NTP 用戶端" \[592\]](#) 所顯示。

圖形 44.2 YaST: 複雜的 NTP 用戶端組態



在複雜 NTP 用戶端組態中，決定是否要在 chroot jail 中啟動 xntpd。因為它可以防止攻擊者損毀整個系統，因此在 xntpd 遭受攻擊時，會有較高的安全性。透過 DHCP 設定 NTP 精靈可以設定 NTP 用戶端，而能透過 DHCP 取得網路中可用的 NTP 伺服器清單。

用戶端要查詢的伺服器以及其他時間來源會列在清單下半部。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。下列是可用的選項：

伺服器

另一個對話方塊可讓您選取 NTP 伺服器 (如 節 44.1.1, "快速 NTP 用戶端組態" [591] 中所述)。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。您可以在輸入欄位指定 xntpd 的其他選項。如需詳細資訊，請參閱 /usr/share/doc/packages/xntp-doc (xntp-doc 套件的一部份)。

點

「點」是指建立對稱關係的機器：它可同時做為時間伺服器及用戶端。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、裝置名稱以及其他選項。按一下 **裝置校正**，即可微調裝置。如需關於本機收音機時鐘作業的詳細資訊，可在 `/usr/share/doc/packages/xntp-doc/html/refclock.htm` 中找到。

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，否則請勿啟動廣播。

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。

44.2 設定網路中的 xntp

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，如果網路中有某個名稱為 `ntp.example.com` 的可用時間伺服器，請將它的名稱，即 `server ntp.example.com` 加入 `/etc/ntp.conf` 檔案。若要新增更多時間伺服器，請以關鍵字伺服器插入其他的行。在以 `rcxntpd start` 指令啟動 `xntpd` 後，將需要一個小時才能使時間穩定下來，而且會建立累積記錄檔案以校正本機電腦時鐘。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。

有兩種方式可將 NTP 機制做為用戶端。首先，用戶端可以定期查詢已知伺服器的時間。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。

44.3 設定本機參考時鐘

軟體套件 `xntp` 包含與本機參考時鐘連線的驅動程式。支援的時鐘清單可以在 `/usr/share/doc/packages/xntp-doc/html/refclock.htm` 檔案的 `xntp-doc` 套件中找到。每個驅動程式都與數字關聯。在 `xntp` 中，實際組態工作是利用虛擬 IP 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，會以 `127.127.t.u` 的格式指定特殊的 IP 位址給它們。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，個別裝置都具有描述組態細節的特殊參數。`/usr/share/doc/packages/xntp-doc/html/driverNN.htm` 檔案 (`NN` 是驅動程式的號碼) 提供關於時鐘特定類型的資訊。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 `xntp-doc` 套件後，就可以在 `/usr/share/doc/packages/xntp-doc/html` 目錄中找到 `xntp` 的文件。`/usr/share/doc/packages/xntp-doc/html/refclock.htm` 檔案提供描述驅動程式參數的驅動程式頁面連結。

LDAP — 一種目錄服務

「輕量型目錄存取協定」(Lightweight Directory Access Protocol, LDAP) 是通訊協定集合用以存取和維護資訊目錄。LDAP 可做為許多用途，像是使用者與群組管理、系統組態管理或是位址管理。本章針對 OpenLDAP 如何運作以及如何使用 YaST 來管理 LDAP 資料提供基本的概念。雖然 LDAP 協定有數種執行方式，但是本章只著重於 OpenLDAP 執行方式。

在網路環境中保持重要資訊的結構性和容易存取是相當重要的。要達到此一目的，可使用黃頁之類的目錄服務，它會以結構良好並且快速搜尋的形式來保存資訊以供存取。

在理想的狀況下，中央伺服器會將資料保存在目錄中，並使用特定的協定將資料分送給所有用戶端。資料的結構允許各種不同的應用程式進行存取。如此，個別的行事曆工具和電子郵件用戶端就不需要維護自己的資料庫，而是存取一個中央儲存區。此一特性明顯降低管理資訊的需要。使用 LDAP 等開放且標準化的協定，以盡量確保所有不同的用戶端應用程式都能存取這類的資訊。

此處所指的目錄是一種最佳化的資料庫，可供快速而有效的讀取和搜尋：

- 為了能夠同時大量讀取，寫入權限僅限管理員所執行的少量更新作業。舊式的資料庫會進行最佳化，以容許在最短時間內接受最多的資料量。
- 由於寫入的存取限制極大，因此採用目錄服務來管理大多數未變更的靜態資料。舊式資料庫中的資料經常變更(動態資料)。舉例來說，企業目錄中的電話號碼變更頻率，會低於會計部門所管理數字的變更頻率。
- 管理靜態資料時，很少會更新現存的資料集。管理動態資料時，特別是銀行帳戶或會計相關的資料集時，最重要的是保持這些資料的一致性。如果要從

一筆資料扣除一個數目，然後將它加到另一筆資料，則必須在同一筆交易內同時進行這兩個動作，以確保整個資料集的餘額正確無誤。資料庫可支援此種交易。目錄則否。目錄可接受短暫性的資料不一致。

LDAP 這類的目錄服務並不是設計來支援複雜的更新或查詢。所有的應用程式在存取這種服務時，都應該以快速簡便的方式進行。

許多目錄先前即已存在，而且現在仍存在於 Unix 內和 Unix 外。Novell NDS、Microsoft ADS、Banyan 的 Street Talk 和 OSI 標準的 X.500 是其中的幾個例子。LDAP 原先的設計只是 DAP 的簡易版。DAP 是一個目錄存取協定，可用來存取 X.500。X.500 標準負責管理目錄項目的階層組織。

LDAP 是 DAP 的精簡版。在不失去 X.500 項目階層的前提下，擁有 LDAP 跨平台功能並節省資源。由於使用 TCP/IP，使得在擴充應用程式和 LDAP 服務之間建立界面的工作變得更加容易。

然而 LDAP 的功能已經提昇。在愈來愈多的使用中，LDAP 已不需要 X.500 的支援而可作為獨立的解決方案使用。LDAP 支援使用 LDAPv3 (openldap2 套件中的協定版本) 的轉介，此一特性使它能够具備實作分散式資料庫。SASL (簡單驗證及安全性階層) 的使用也是一項新嘗試。

LDAP 超越其原始設計，其用途不限於查詢 X.500 伺服器的資料。slapd 開放來源伺服器可將物件資訊儲存在本地資料庫。另外有一個稱為 slurpd 的延伸套件，負責複製多個 LDAP 伺服器。

openldap2 套件包含：

slapd

是一個獨立的 LDAPv3 伺服器，用來管理 BerkeleyDB 架構資料庫中的物件資訊。

slurpd

此程式可將本地 LDAP 伺服器中的資料變更複製到其它安裝在網路中的 LDAP 伺服器。

其它系統維護工具

slapcat、slapadd、slapindex

45.1 LDAP 與 NIS 的比較

Unix 系統管理員習慣使用 NIS 伺服器進行名稱解析和網路資料散佈。/etc 中的檔案所包含的組態資料，以及 group、hosts、mail、netgroup、networks、passwd、printcap、protocols、rpc 和 services 等目錄中所包含的組態資料，都是由網路中的用戶端散佈。因為這些檔案是簡單的文字檔，因此相當容易維護。不過，因為缺乏結構，較大量的資料處理變得更加困難。NIS 的設計只適用於 Unix 平台，因此不可能作為異質性網路的中央資料管理員。

和 NIS 不同，LDAP 伺服器不限於純 Unix 網路。Windows 伺服器 (2000 或更新) 支援 LDAP 的目錄服務功能。Novell 也提供 LDAP 服務。其它非 Unix 系統也可為前述應用程式工作提供更多支援。

LDAP 規則可套用於任何必須進行中央管理的資料結構。以下為幾個應用程式範例：

- 作為 NIS 服務的替代品使用
- 郵件路由 (postfix、sendmail)
- 郵件用戶端的通訊錄，例如 Mozilla、Evolution 和 Outlook。
- BIND9 名稱伺服器的區域描述管理
- 在異質網路中，利用 Samba 進行使用者驗證

這個清單還可加長，因為 LDAP 是可以延伸的，和 NIS 有所不同。由於清楚定義的資料階層結構可方便搜尋，使得管理大量資料的工作更為容易。

45.2 LDAP 目錄樹的結構

LDAP 的目錄具有樹狀結構。階層中的所有目錄項目 (稱為物件) 都有一個規定的位置。此階層稱為 *目錄資訊樹* (DIT)。通往所需項目的完整路徑稱為 *可辨識名稱* 或 DN，可用來清楚辨識路徑。此項目路徑中的單一節點稱為 *相關可辨識名稱* 或 RDN。可將物件指定為下列兩種類型：

容器

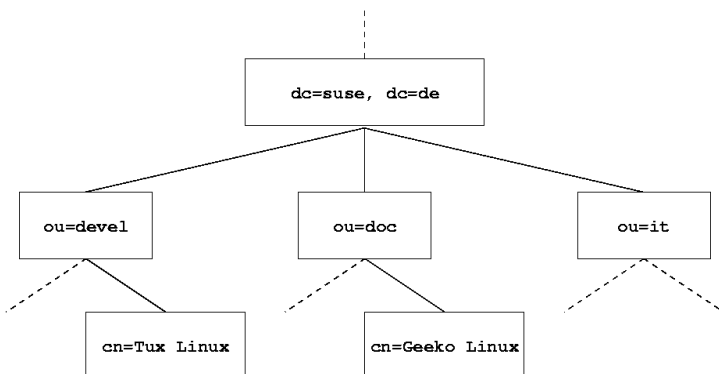
這些物件可包含其它物件。這些物件的類別為 `root` (目錄樹的根元素，此元素其實並不存在)、`c` (國家)、`ou` (組織單位) 和 `dc` (領域元件)。此模型可媲美檔案系統中的目錄 (資料夾)。

分葉

這些物件位於分支末端，而且沒有附屬物件。例如 `person`、`InetOrgPerson` 或 `groupofNames`。

目錄階層的頂端有一個根元素 `root`。其中可包含 `c` (國家)、`dc` (領域元件) 或 `o` (組織) 作為其附屬元件。從下列範例 (顯示於 [圖形 45.1, "LDAP 目錄結構" \[600\]](#)) 可清楚看出 LDAP 目錄樹中的關係。

圖形 45.1 LDAP 目錄結構



這個完整的結構圖是由一棵虛擬的目錄資訊樹所組成。其中包含三個階層的項目。每個項目分別對應到圖上的一個方塊。在此範例中，虛擬的 SUSE 員工 `Geeko Linux` 有一個完整有效的可辨識名稱，即 `cn=Geeko Linux, ou=doc, dc=suse, dc=de`。此名稱的形成是將 RDN `cn=Geeko Linux` 加到前一個項目 `ou=doc, dc=suse, dc=de` 的 DN。

哪一種類型的物件應儲存在 DIT 的全域決定，是根據綱要而定的。物件類型則是由物件類別來決定。物件類別決定相關物件必須或可以被指定哪一種屬性。因此，綱要的內容必須包括所有物件類別的定義，以及所需應用程式案例中使用的屬性。有幾個通用綱要 (請參閱 RFC 2252 和 2256)。不過，仍可建立自訂的綱要。如果用來操作 LDAP 的伺服器環境需要，也可使用多個綱要來互相輔助。

表格 45.1, "常用物件類別和屬性" [601] 提供一個小型綜覽，介紹範例中所使用的 core.schema 和 core.schema 物件類別，包括所需的屬性和有效的屬性值。

表格 45.1 常用物件類別和屬性

物件類別	意義	範例項目	強制屬性
dcObject	<i>domainComponent</i> (領域的名稱元件)	suse	dc
organizationalUnit	<i>organizationalUnit</i> (組織單位)	doc	ou
inetOrgPerson	<i>inetOrgPerson</i> (內部網路或國際網路的相關人員資料)	Geeko Linux	sn 與 cn

範例 45.1, "摘自 schema.core" [601] 中摘述一段綱要指示詞及其說明 (將各行編號以便說明)。

範例 45.1 摘自 schema.core

```
#1 attributetype ( 2.5.4.11 NAME ( 'ou' 'organizationalUnitName' )
#2         DESC 'RFC2256: organizational unit this object belongs to'
#3         SUP name )

...
#4 objectclass ( 2.5.6.5 NAME 'organizationalUnit'
#5         DESC 'RFC2256: an organizational unit'
#6         SUP top STRUCTURAL
#7         MUST ou
#8 MAY (userPassword $ searchGuide $ seeAlso $ businessCategory
        $ x121Address $ registeredAddress $ destinationIndicator
        $ preferredDeliveryMethod $ telexNumber
        $ teletexTerminalIdentifier $ telephoneNumber
        $ internationaliSDNNumber $ facsimileTelephoneNumber
        $ street $ postOfficeBox $ postalCode $ postalAddress
        $ physicalDeliveryOfficeName
        $ st $ l $ description) )
...
```

organizationalUnitName 屬性類型和對應的 organizationalUnit 物件類別在此作為範例之用。第 1 行的重點為屬性名稱、其唯一的 OID (物件識別碼) (數值) 和屬性的縮寫。

第 2 行提供屬性說明和 DESC。另外還包括對應的 RFC，可據此列出相關定義。第 3 行的 SUP 表示此屬性所隸屬的上級屬性類型。

第 4 行開始為 `organizationalUnit` 物件類型的定義，內容包括 `OID` 和物件類別名稱，如同屬性定義一樣。第 5 行為物件類別的簡短說明。第 6 行的 `sup top` 項目表示此物件類別不隸屬於其它物件類別。第 7 行始於 `MUST`，列出所有必須和 `organizationalUnit` 類型物件配合使用的屬性類型。第 8 行始於 `MAY`，列出所有允許和此物件類別配合使用的屬性類型。

有關綱要的使用，可在 `OpenLDAP` 文件中找到一篇很好的介紹。安裝好之後，請至 `/usr/share/doc/packages/openldap2/admin-guide/index.html` 尋找這篇文章。

45.3 使用 `slapd.conf` 來設定伺服器

在您安裝好的系統的 `/etc/openldap/slapd.conf` 中有一個完整的組態檔，可供您的 `LDAP` 伺服器使用。此處有這個單一項目的簡介，對於必要的調整也有說明。開頭為井字符號 (`#`) 的項目為非作用中的項目。必須移除這個備註字元，才能啟用這些項目。

45.3.1 `slapd.conf` 中的全域指示詞

範例 45.2 *`slapd.conf`: 包括綱要指示詞*

```
include      /etc/openldap/schema/core.schema
include      /etc/openldap/schema/cosine.schema
include      /etc/openldap/schema/inetorgperson.schema
include      /etc/openldap/schema/rfc2307bis.schema
include      /etc/openldap/schema/yast.schema
```

在 [範例 45.2, "slapd.conf: 包括綱要指示詞" \[602\]](#) 中，`slapd.conf` 的第一個指示詞指定一個綱要，`LDAP` 目錄會根據此綱要來組織其架構。`core.schema` 項目為強制性。這個指示詞後面會附加其他必要綱要。可在包含的 `OpenLDAP` 文件中找到更多資訊。

範例 45.3 *`slapd.conf`: `pidfile` 和 `argsfile`*

```
pidfile      /var/run/slapd/slapd.pid
argsfile     /var/run/slapd/slapd.args
```

這兩個檔案包含 `PID` (處理程序 ID) 和一些用來啟動 `slapd` 程序的引數。此處不需要修改。

範例 45.4 *slapd.conf*:存取控制

```
# Sample Access Control
#       Allow read access of root DSE
# Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
# access to dn="" by * read
#       access to * by self write
#           by users read
#           by anonymous auth
#
# if no access controls are present, the default is:
#       Allow read by all
#
# rootdn can always write!
```

範例 45.4, "[slapd.conf:存取控制](#)" [603] 摘自 `slapd.conf`，此檔案在伺服器上負責管制 LDAP 目錄的存取權。只要特定資料庫區段中沒有宣告任何自訂的規則，在 `slapd.conf` 全域區段中的設定均為有效。這些自訂規則可覆寫全域宣告。正如此處所提到的，所有使用者都有讀取目錄的權利，但只有管理員 (`rootdn`) 才可在目錄中寫入資料。LDAP 中的存取控制規定是一個非常複雜的程序。下列秘訣可提供協助：

- 每一項存取規則都具有下列結構：

```
access to <what> by <who> <access>
```

- *what* 是一個替代符號，代表有存取權的物件或屬性。個別的目錄分支可由個別規則來明確地保護。也可使用一般表示式來處理目錄樹的區域。`slapd` 會按照組態檔中所列的順序來評估所有規則。較一般性的規則應列在較特定的規則之後——會評估 `slapd` 視為有效的第一條規則，並忽視其後的所有項目。
- *who* 會決定誰應獲得存取權，以進入由 *what* 所決定的區域。可使用一般表示式。第一次的配對完成後 `slapd` 會再次中止 *who* 的評估，因此必須在較一般性的規則之後列出較特定的規則。在 [表格 45.2, "使用者群組及其存取權"](#) [604] 中所顯示的項目是有可能的。

表格 45.2 使用者群組及其存取權

標籤	範圍
*	沒有例外的所有使用者
anonymous	未驗證的 (「匿名」) 使用者
使用者	驗證的使用者
self	以目標物件連線的使用者
dn.regex=<regex>	符合一般表示式的所有使用者

- `access` 會指定存取類型。使用 [表格 45.3, "存取類型" \[604\]](#) 中所列的選項。

表格 45.3 存取類型

標籤	存取範圍
none	沒有存取權
auth	連絡伺服器
compare	比較物件的存取
search	搜尋過濾器的使用
讀取	讀取權
寫入	寫入權

`slapd` 會比較用戶端所要求的存取權與在 `slapd.conf` 中授予的存取權。如果用戶端所要求的權利比規則中的等級更低或同等，則可獲得存取權。如果用戶端要求的權利比規則中的宣告更高，則會被拒絕。

範例 45.5, "slapd.conf:存取控制範例" [605] 顯示使用一般表示式即可任意開發簡單存取控制權的範例。

範例 45.5 *slapd.conf*:存取控制範例

```
access to dn.regex="ou=([ ^, ]+), dc=suse, dc=de "  
by dn.regex="cn=administrator, ou=$1, dc=suse, dc=de" write  
by user read  
by * none
```

此規則宣告只有個別的管理員有寫入個別 `ou` 項目的權利。其它所有通過驗證的使用者都有讀取權，其餘的人則沒有存取權。

提示: 建立存取規則

如果沒有 `access to` 的規則或者沒有符合 `by` 的指示詞，則無法取得存取權。只會授予明確宣告的存取權。如果沒有任何宣告的規則，則管理員依照預設原則擁有寫入權，其他人則有讀取權。

請至已安裝的 `openldap2` 套件的線上文件尋找詳細資訊及 LDAP 存取權的組態範例。

除了使用 (`slapd.conf`) 中央組態檔來管理存取許可權之外，也可使用存取控制資訊 (ACI)。ACI 允許在 LDAP 樹中儲存個別物件的存取資訊。此種類型的存取控制尚未普遍，開發者目前仍將它視為一項實驗。請參閱 <http://www.openldap.org/faq/data/cache/758.html> 以取得更多資訊。

45.3.2 slapd.conf 中的資料庫特定指示詞

範例 45.6 *slapd.conf*: 資料庫特定指示詞

```
database bdb
checkpoint      1024      5
cachesize       10000
suffix "dc=suse,dc=de"
rootdn "cn=admin,dc=suse,dc=de"
# Cleartext passwords, especially for the rootdn, should
# be avoided. See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw secret
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd/tools. Mode 700 recommended.
directory /var/lib/ldap
# Indices to maintain
index objectClass eq
```

資料庫類型 (在此例中是 Berkeley 資料庫) 是在這段文字的第一行決定 (請參閱 [範例 45.6, "slapd.conf: 資料庫特定指示詞" \[606\]](#))。checkpoint 可決定異動記錄要保留多少資料量 (單位為 kb) 之後才會寫入實際資料庫, 以及決定兩次寫入動作之間的相隔時間 (單位為分鐘)。cachesize 可設定資料庫快取中保留的物件數。suffix 可決定這部伺服器應該負責 LDAP 樹的哪個部份。接下來的 rootdn 決定擁有本伺服器管理權的人。此處宣告的使用者不需要有 LDAP 項目或以一般使用者的身份出現。管理員密碼是使用 rootpw 來設定。在這裡可以不使用 secret 而是以雜湊值來輸入 slappasswd 所建立的管理員密碼。directory 指示詞表示資料庫目錄所在的目錄 (在檔案系統中) 儲存在伺服器中。最後一個指示詞 index objectClass eq 會啟動所有物件類別索引的維護作業。在此處可根據經驗, 加入一個使用者最常搜尋的屬性。此處使用的是資料庫自訂的 Access 規則, 而非全域的 Access 規則。

45.3.3 啟動及停止伺服器

一旦 LDAP 伺服器完全設定好, 且所有需要的項目已依照 [節 45.4, "LDAP 目錄中的資料處理" \[607\]](#) 中描述的樣式建立之後, 輸入 rclldap start 以 root 的身份來啟動 LDAP 伺服器。輸入 rclldap stop 指令可手動停止伺服器。使用 rclldap status 來請求執行中 LDAP 伺服器的狀態。

[節 28.2.3, "使用 YaST 設定系統服務 \(Runlevel\)" \[394\]](#) 中所描述的 YaST runlevel 編輯器, 可讓系統在啟動和暫停時自動啟動及停止伺服器。也可以透過 [節 28.2.2,](#)

"[Init 程序檔](#)" [391] 中所述的指令提示，利用 `inserv` 指令來建立對應的連結至啟動和停止程序檔。

45.4 LDAP 目錄中的資料處理

OpenLDAP 提供一系列的工​​具，用來管理 LDAP 目錄中的資料。以下說明四種最重要的工具，分別用來新增、刪除、搜尋和修改資料集。

45.4.1 將資料加入 LDAP 目錄

一旦您 LDAP 伺服器 `/etc/openldap/slapd.conf` 中的組態正確並且就緒 (即擁有正確的 `suffix`、`directory`、`rootdn`、`rootpw` 和 `index`) 項目，請接著輸入記錄。OpenLDAP 提供這項工作所需的 `ldapadd` 指令。可能的話，將多個物件一次加入資料庫是較實際的作法。LDAP 可處理此處的 LDIF 格式 (LDAP 資料交換格式)。LDIF 是一個簡單的文字檔，可包含任意的屬性及數值組合。有關可用的物件類別和屬性的資料，請參閱 `slapd.conf` 中宣告的綱要檔。在 [圖形 45.1, "LDAP 目錄結構" \[600\]](#) 中用來建立概略架構的 LDIF 檔會和 [範例 45.7, "LDIF 檔案範例" \[607\]](#) 中的檔案相似。

範例 45.7 LDIF 檔案範例

```
# The SUSE Organization
dn: dc=suse,dc=de
objectClass: dcObject
objectClass: organization
o: SUSE AG dc: suse

# The organizational unit development (devel)
dn: ou=devel,dc=suse,dc=de
objectClass: organizationalUnit
ou: devel

# The organizational unit documentation (doc)
dn: ou=doc,dc=suse,dc=de
objectClass: organizationalUnit
ou: doc

# The organizational unit internal IT (it)
dn: ou=it,dc=suse,dc=de
objectClass: organizationalUnit
ou: it
```

重要: LDIF 檔的編碼

LDAP 可使用 UTF-8 (Unicode)。母音字元的編碼必須正確。使用支援 UTF-8 的編輯器，例如 **Kate** 或最新版的 **Emacs**。否則，應避免母音字元和其它特殊字元，或使用 `recode` 將輸入重新編碼為 UTF-8。

儲存字尾為 `.ldif` 的檔案，然後用下列指令傳送給伺服器。

```
ldapadd -x -D <dn of the administrator> -W -f <file>.ldif
```

`-x` 在此例中會關閉 SASL 的驗證。`-D` 會宣告呼叫操作的使用者。此處輸入管理員的有效 DN，如同 `slapd.conf` 中的設定一樣。在此範例中，有效 DN 為 `cn=admin,dc=suse,dc=de`。`-W` 可避免將密碼輸入指令行 (以非加密文字) 並啟動一個密碼提示。此密碼是先前在 `slapd.conf` 中使用 `rootpw` 決定的。`-f` 傳送檔案名稱。請參閱 [範例 45.8, "ldapadd 和 example.ldif" \[608\]](#) 中執行 `ldapadd` 的詳細內容。

範例 45.8 *ldapadd 和 example.ldif*

```
ldapadd -x -D cn=admin,dc=suse,dc=de -W -f example.ldif
```

```
Enter LDAP password
adding new entry "dc=suse,dc=de"
adding new entry "ou=devel,dc=suse,dc=de"
adding new entry "ou=doc,dc=suse,dc=de"
adding new entry "ou=it,dc=suse,dc=de"
```

個別的使用者資料可建立在不同的 LDIF 檔。[範例 45.9, "Tux 的 LDIF 資料" \[608\]](#) 會新增 Tux 至新的 LDAP 目錄。

範例 45.9 *Tux 的 LDIF 資料*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
objectClass: inetOrgPerson
cn: Tux Linux
givenName: Tux
sn: Linux
mail: tux@suse.de
uid: tux
telephoneNumber: +49 1234 567-8
```

LDIF 檔可包含任意數量的物件。可以將整個目錄分支一起傳給伺服器，也可只傳送一部份，如個別物件範例所示。如果需要經常修改某些資料，建議使用單一物件的細分。

45.4.2 修改 LDAP 目錄中的資料

工具 `ldapmodify` 可用來修改資料集。最簡單的作法是先修改對應的 LDIF 檔，然後將修改過的檔案傳送給 LDAP 伺服器。如果要將同事 Tux 的電話號碼從 +49 1234 567-8 改為 +49 1234 567-10，則必須依照 [範例 45.10, "修改過的 LDIF 檔 tux.ldif" \[609\]](#) 中的方式來編輯 LDIF 檔案。

範例 45.10 修改過的 LDIF 檔 *tux.ldif*

```
# coworker Tux
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

使用下列指令將修改過的檔案輸入 LDAP 目錄：

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W -f tux.ldif
```

或者，也可以直接將要修改的屬性傳送給 `ldapmodify`。此項作業程序描述如下：

1. 啟動 `ldapmodify` 並輸入您的密碼：

```
ldapmodify -x -D cn=admin,dc=suse,dc=de -W
Enter LDAP password:
```

2. 輸入變更並注意是否符合下列語法順序：

```
dn: cn=Tux Linux,ou=devel,dc=suse,dc=de
changetype: modify
replace: telephoneNumber
telephoneNumber: +49 1234 567-10
```

請在 `ldapmodify(1)` man 頁面中，尋找有關 `ldapmodify` 及其語法的詳細資訊。

45.4.3 搜尋或讀取 LDAP 目錄中的資料

OpenLDAP 透過 `ldapsearch` 提供一個指令行工具，用來搜尋和讀取 LDAP 目錄中的資料。下列為簡易查詢的語法：

```
ldapsearch -x -b dc=suse,dc=de "(objectClass=*)"
```

選項 `-b` 決定搜尋基礎——用來執行搜尋的目錄樹區段。在目前這個範例中，此搜尋基礎為 `dc=suse,dc=de`。如果要在 **LDAP** 目錄的次區段中執行更詳細的搜尋 (例如，只在 `devel` 部門搜尋) 可使用 `-b` 將此區段傳送至 `ldapsearch`。`-x` 會要求啟動簡單的驗證。`(objectClass=*)` 會宣告目錄中的所有物件都應該讀取。在建立一個新的目錄樹後，可使用本指令來確認所有項目都已正確記錄，而且伺服器的回覆也符合需要。更多有關使用 `ldapsearch` 的資訊，可在對應的 `man` 頁面 (`ldapsearch(1)`) 找到。

45.4.4 刪除 LDAP 目錄中的資料

使用 `ldapdelete` 來刪除不要的資料。其語法和前述指令相似。例如，要刪除 `Tux Linux` 的整個項目：可發出下列指令：

```
ldapdelete -x -D cn=admin,dc=suse,dc=de -W cn=Tux \
Linux,ou=devel,dc=suse,dc=de
```

45.5 YaST LDAP 用戶端

YaST 包含一個模組，可用來設定 **LDAP** 式的使用者管理。如果您在安裝時未啟用此功能，則可選取 **網路服務** → **LDAP 用戶端** 來啟用模組。YaST 會自動啟用 **LDAP** 所需的任何 **PAM** 和 **NSS** 相關變更 (如下述)，並安裝必要檔案。

45.5.1 標準程序

具備作用於用戶端機器背景的處理程序基本知識，可協助您瞭解 YaST **LDAP** 用戶端模組的運作方式。如果已啟動 **LDAP** 以進行網路驗證或者已呼叫 YaST 模組，則表示已安裝 `pam_ldap` 和 `nss_ldap` 套件，而且兩個對應的組態檔也已經調整。`pam_ldap` 為 **PAM** 模組，負責登入程序和 **LDAP** 目錄之間的協調，也是驗證資料的來源。已安裝 `pam_ldap.so` 專用模組並調整 **PAM** 組態 (請參閱 [範例 45.11](#), "[適用於 LDAP 的 pam_unix2.conf](#)" [610])。

範例 45.11 適用於 **LDAP** 的 `pam_unix2.conf`

```
auth:      use_ldap
account:   use_ldap
password:  use_ldap
session:   none
```

在手動設定額外服務以使用 LDAP 時，應納入 PAM 組態檔中的 PAM LDAP 模組，該組態檔與 `/etc/pam.d` 中的服務相對應。可在 `/usr/share/doc/packages/pam_ldap/pam.d/` 中找到適用於個別服務的組態檔。將適當的檔案複製到 `/etc/pam.d`。

透過 `nsswitch` 機制進行的 `glibc` 名稱解析適用於含有 `nss_ldap` 的 LDAP。安裝此套件時，已在 `/etc/` 中建立一個新的調整過的 `nsswitch.conf` 檔案。更多有關 `nsswitch.conf` 的作業可在 [節 38.5.1, "組態檔" \[533\]](#) 中找到。`nsswitch.conf` 必須包含下列幾行，以供 LDAP 進行使用者管理及驗證。請參閱 [範例 45.12, "nsswitch.conf 中的調整" \[611\]](#)。

範例 45.12 `nsswitch.conf` 中的調整

```
passwd: compat
group: compat

passwd_compat: ldap
group_compat: ldap
```

這幾行會先命令 `glibc` 的查詢程式庫評估 `/etc` 中的對應檔案，然後存取 LDAP 伺服器，以它作為驗證及使用者資料的來源。可對此機制進行測試，例如使用 `getent passwd` 指令來讀取使用者資料庫的內容。所傳回的資料集應包含一份有關您系統上的本地使用者和 LDAP 伺服器儲存的所有使用者的調查。

如果要避免一般使用者透過 LDAP 以 `ssh` 或 `login` 登入伺服器，則 `/etc/passwd` 和 `/etc/group` 檔案必須各自增加一行。該行的內容為 `+::: /sbin/nologin in /etc/passwd and +::: in /etc/group`。

45.5.2 LDAP 用戶端的組態

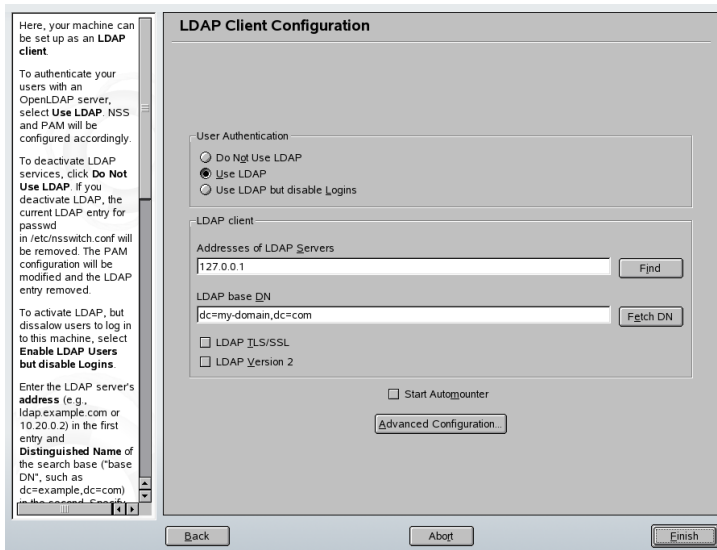
起始調整 `nss_ldap` 之後，YaST 就會開始處理 `pam_ldap`、`/etc/passwd` 和 `/etc/group`；您只需將用戶端連接到伺服器，讓 YaST 透過 LDAP 進行使用者管理。["基本組態" \[612\]](#) 中有說明這個基本設定。

使用 YaST LDAP 用戶端進一步設定 YaST 群組和使用者組態模組。這包括操作新使用者和群組的預設設定，以及指定給使用者或群組的屬性數目和性質。與傳統的使用者或群組管理解決方案相比，LDAP 使用者管理可讓您將更多不同屬性指定給使用者和群組。這會在 ["設定 YaST 群組和使用者管理模組" \[614\]](#) 中說明。

基本組態

如果您選擇 LDAP 使用者管理，或是在已安裝系統的「YaST 控制中心」選取網路服務 → LDAP 用戶端，就會在安裝期間開啟基本 LDAP 用戶端組態對話方塊 (圖形 45.2, "YaST: LDAP 用戶端的組態" [612])。

圖形 45.2 YaST: LDAP 用戶端的組態

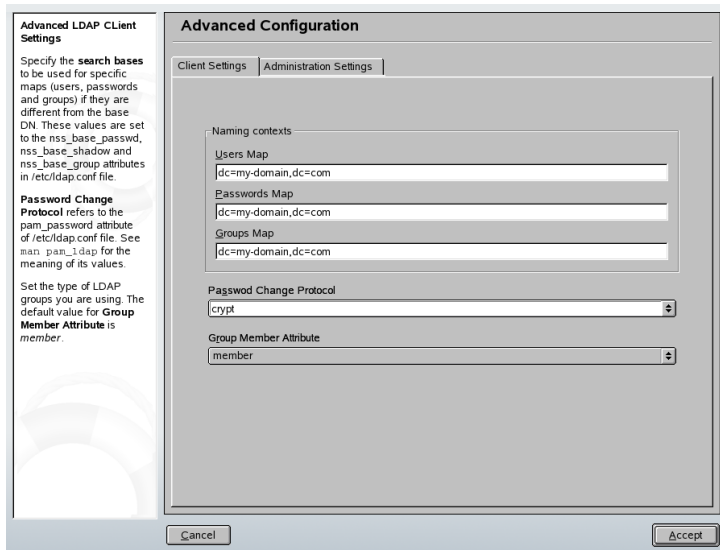


若要對 OpenLDAP 伺服器驗證機器使用者，並透過 OpenLDAP 啟用使用者管理，請按照下列步驟進行：

- 1 按一下使用 *LDAP* 以使用 LDAP。如果您要使用 LDAP 進行驗證，但是不想讓其他使用者登入這個用戶端，請改為選取使用 *LDAP* 但停用登入。
- 2 輸入要使用的 LDAP 伺服器的 IP 位址。
- 3 輸入 *LDAP* 基礎 DN 以選取 LDAP 伺服器上的搜尋基礎。
- 4 如果需要伺服器的 TLS 或 SSL 受保護通訊，請選取 *LDAP TLS/SSL*。
- 5 如果 LDAP 伺服器仍然使用 LDAPv2，請選取 *LDAP* 版本 2 以明確地使用此協定。

- 6 選取啟動自動裝載器以將遠端目錄 (例如遠端管理的 /home) 裝載至用戶端。
- 7 按一下完成以套用設定。

圖形 45.3 YaST: 進階組態



如果要以管理者身份修改伺服器上的資料，按一下 **進階組態**。下列對話方塊分為兩個標籤。請參閱 [圖形 45.3, "YaST: 進階組態" \[613\]](#):

- 1 在用戶端設定標籤中，依照您的需要來調整下列設定：
 - a 如果使用者、密碼和群組的搜尋基礎與 *LDAP* 基礎 *DN* 指定的全域搜尋基礎不同，請在使用者對應、密碼對應和群組對應中，輸入這些不同的命名內容。
 - b 指定密碼變更協定。變更密碼時使用的標準方法是 `crypt`，表示會使用由 `crypt` 產生的密碼雜湊。如需這個選項和其他選項的詳細資料，請參閱 `pam_ldap` man 頁面。
 - c 指定要與群組成員屬性搭配使用的 *LDAP* 群組。它的預設值是 `member`。

2 在管理設定中，調整下列設定：

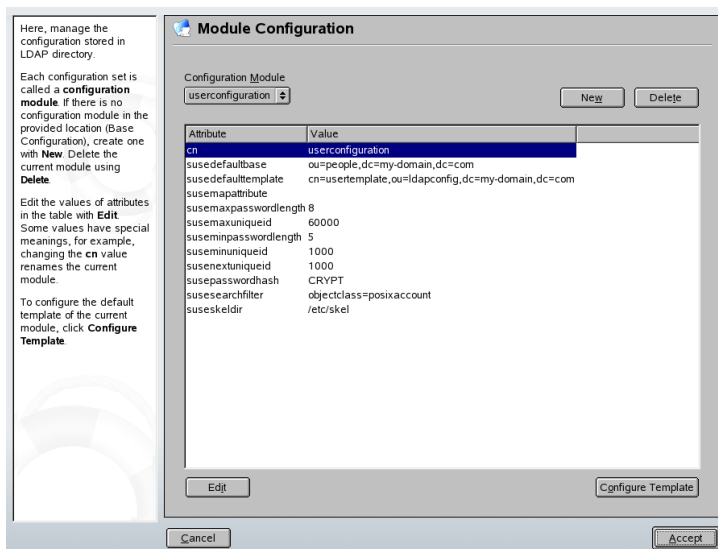
- a 透過組態基礎 *DN* 來設定儲存使用者管理資料的基礎。
- b 在管理員 *DN* 輸入適當的值。這個 *DN* 必須與 `/etc/openldap/slapd.conf` 所指定的 `rootdn` 值相同，才能讓這位特定使用者操作 LDAP 伺服器中儲存的資料。
- c 勾選建立預設組態物件，在伺服器上建立基本組態物件，以透過 LDAP 啟用使用者管理。
- d 如果用戶端機器會成為網路中的主目錄的檔案伺服器，請勾選此機器上的主目錄。
- e 按一下接受以離開進階組態，再按完成以套用設定。

按一下設定使用者管理設定以編輯 LDAP 伺服器上的項目。接著會根據伺服器中儲存的 ACL 和 ACI 來授予伺服器組態模組的權限。遵循 ["設定 YaST 群組和使用者管理模組" \[614\]](#) 中概述的程序。

設定 YaST 群組和使用者管理模組

使用 YaST LDAP 用戶端來調整 YaST 模組，以進行使用者和群組管理，並在需要時加以延伸。利用個別屬性的預設值來定義樣板，以簡化資料註冊。此處所建立的預設會以 LDAP 物件的形式儲存在 LDAP 目錄中。使用者資料註冊仍以使用者和群組管理的一般 YaST 模組來完成。所註冊的資料會在伺服器中儲存為 LDAP 物件。

圖形 45.4 YaST: 模組組態



模組組態對話方塊 (圖形 45.4, "YaST: 模組組態" [615]) 允許建立新模組、選取和修改現有的模組組態，以及設計和修改這些模組的樣板。

若要建立新的組態模組，請按照下列步驟進行：

- 1 按一下 **新增** 並選取要建立的模組類型。若為使用者組態模組，請選取 `suseuserconfiguration`；若為群組組態，請選擇 `susegroupconfiguration`。

- 2 選擇新樣板的名稱。

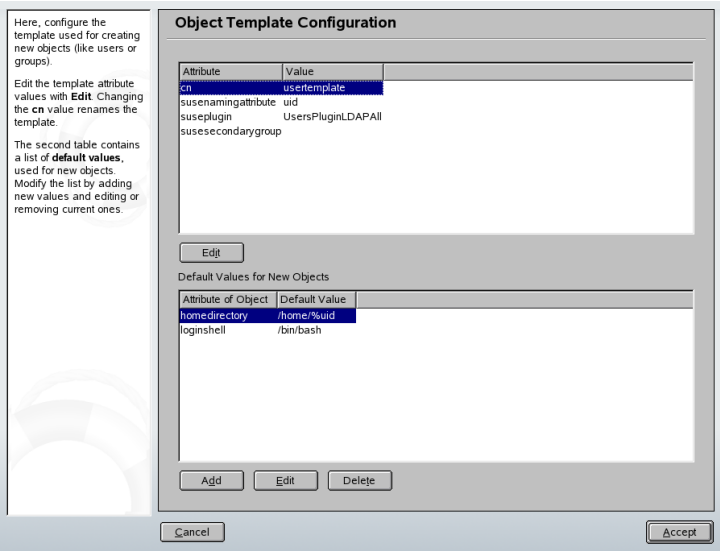
接著該內容會顯示一個表格，列出所有可用於此模組的屬性和其指定值。除了所有設定的屬性之外，表格中也會列出現用綱要允許但未使用的其它屬性。

- 3 選取個別屬性，按下 **編輯**，然後輸入新值，即可接受或調整要用於群組和使用使用者組態的預設值。變更模組的 `cn` 屬性，即可重新命名模組。按一下 **刪除** 可刪除目前選取的模組。
- 4 按一下 **確定** 之後，新模組就會加入選項功能表。

群組和使用者管理的 YaST 模組會將合理的標準值內嵌至樣板中。若要編輯與組態模組相關聯的樣板，請按照下列步驟進行：

- 1 在模組組態對話方塊中，按一下設定樣板。
- 2 根據您的需求來決定要指定給此樣板的一般屬性值，或是保留空白。LDAP 伺服器上的空白屬性會被刪除。
- 3 修改、刪除或新增新物件的預設值 (LDAP 樹中的使用者或群組組態)。

圖形 45.5 YaST: 物件樣板組態



將模組的 `suseDefaultTemplate` 屬性值設定為調整過的樣板 DN，以連接樣板與其模組。

提示

使用變數樣式來取代絕對值，即可透過其它屬性建立一個屬性的預設值。例如，建立新使用者時，會自動以 `sn` 和 `givenName` 的屬性值建立 `cn=%sn %givenName`。

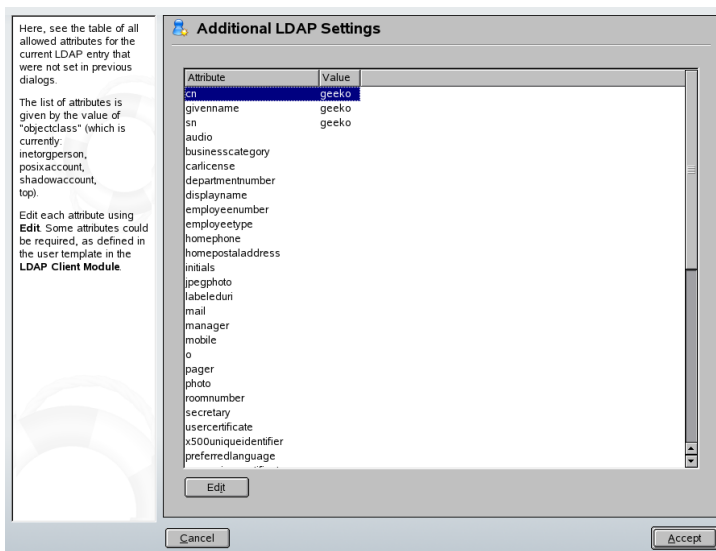
一旦所有模組和樣板都已正確設定並可執行時，即可使用 YaST 以一般方式來註冊新群組和使用者。

45.6 在 YaST 中設定 LDAP 使用者和群組

實際的使用者和群組資料註冊與不使用 LDAP 時的程序只有些微不同。以下是使用者管理簡介。群組管理程序與此類似。

- 1 經由 *安全性與使用者* → *使用者管理* 進入 YaST 使用者管理。
- 2 使用 *設定過濾器*，將使用者的檢視畫面限制為 LDAP 使用者，並輸入根 DN 的密碼。
- 3 按一下 *新增* 並輸入新使用者的組態。就會開啟有四個索引標籤的對話方塊：
 - a *使用者資料* 標籤中的指定使用者名稱、登入及密碼。
 - b 勾選新使用者的群組成員、登入外圍程序及主目錄的 *詳細資料* 標籤。若有需要，可將預設值變更為更符合您需求的值。利用 "[設定 YaST 群組和使用者管理模組](#)" [614] 中說明的程序，即可定義預設值以及密碼設定的預設值。
 - c *修改或接受預設的密碼設定*。
 - d 進入 *插件* 標籤，選取 LDAP 插件，然後按一下 *啟動*，即可設定指定給新使用者的其他 LDAP 屬性 (請參閱 [圖形 45.6, "YaST: 其它 LDAP 設定"](#) [618])。
- 4 按一下 *接受* 以套用設定，並離開使用者組態。

圖形 45.6 YaST: 其它 LDAP 設定



最初的使用者管理輸入表中包含 *LDAP* 選項。此功能可用來將 LDAP 搜尋過濾器套用到可用的使用者，或者也可選取 *LDAP 使用者及群組組態* 來連接設定 LDAP 使用者和群組的模組。

45.7 若需詳細資訊

本章節刻意將一些較為複雜的主題排除在外，例如 SASL 組態或分散工作量至多個從屬伺服器的 LDAP 伺服器的複製。有關這兩個主題的詳細資訊，可在 *OpenLDAP 2.2 管理員指南* 中找到 (請參閱下列資訊)。

OpenLDAP 專案的網站提供 LDAP 初學者及進階使用者詳盡的說明。

OpenLDAP Faq-O-Matic

是一個豐富的問答集，內容包括安裝、組態設定及 OpenLDAP 的使用方法。請參閱 <http://www.openldap.org/faq/data/cache/1.html>。

快速入門指南

簡潔的逐步示範，教您如何安裝您的第一部 LDAP 伺服器。請參閱 <http://www.openldap.org/doc/admin22/quickstart.html>，或是參閱已

安裝系統的 `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`。

OpenLDAP 2.2 管理員指南

詳細介紹所有重要的 LDAP 組態，包括存取控制及加密。請參閱 <http://www.openldap.org/doc/admin22/>，或是參閱已安裝系統的 `/usr/share/doc/packages/openldap2/admin-guide/quickstart.html`。

了解 LDAP

有關 LDAP 基本原則的一般性介紹：<http://www.redbooks.ibm.com/redbooks/pdfs/sg244986.pdf>。

有關 LDAP 的出版品：

- *LDAP System Administration (LDAP 系統管理)*，作者 Gerald Carter (ISBN 1-56592-491-6)
- *Understanding and Deploying LDAP Directory Services (瞭解和部署 LDAP 目錄服務)*，作者 Howes, Smith, and Good (ISBN 0-672-32316-8)

有關 LDAP 的最終參考資料為對應的 RFC (要求建議)，2251 至 2256。

Apache 網頁伺服器

根據 <http://www.netcraft.com> 指出，Apache 擁有超過 60% 的市場佔有率，是世界上使用最普遍的網頁伺服器。針對網頁應用程式設計的 Apache，通常是用於 Linux 系統，並且可配合 MySQL 資料庫以及 PHP 和 Perl 程式語言使用。此結合一般稱為 LAMP。

本章將介紹版本 2.x 的網頁和應用程式伺服器軟體 Apache。本文將說明 Apache 的安裝和設定，以及部分可用模組的使用方式。

46.1 序言和詞彙

本節將提供與網頁相關和 Apache 特定的常用詞彙定義。

重要：詞彙

本文提到的 *Apache* 都是指版本 2.x 的 Apache。如需有關 Apache 1.x 的詳細資訊，請參閱 [Apache 網站內容](#)。

46.1.1 網頁伺服器

網頁伺服器會傳送用戶端要求的網頁。這時用戶端可能是類似 Konqueror 的網頁瀏覽器，或是任何可以連接到全球資訊網 (World Wide Web, WWW) 的其他設備。這些網頁可以完整地儲存在磁碟 (靜態網頁) 中，或是為處理外部實體查詢而產生 (動態網頁)，例如資料庫或網頁服務。

46.1.2 HTTP

用戶端和網頁伺服器之間的通訊，會使用超文字傳輸協定 (HTTP) 進行。目前的 HTTP 1.1 版本已記載於 RFC 2068 和更新的 RFC 2616。這些 RFC 可以從 <http://www.w3.org> 取得。

46.1.3 URL

URL 是指「一致資源定址器 (Uniform Resource Locator)」。用戶端會透過例如 <http://www.example.com/index.html> 的 URL，來向伺服器要求網頁。URL 包含：

協定

常用的協定：

http://

HTTP 協定

https://

HTTP 安全加密版

ftp://

下載及上傳檔案的傳輸協定

領域

在這個範例中的領域就是 `www.example.com`。領域就是指對應到 IP 位址的名稱。因此，`www.example.com` 只會唯一對應到某 IP 位址 (例如 123.456.789.1)。而這組位址數字，也是識別執行網頁伺服器電腦時的唯一專屬號碼。將領域名稱對應到 IP 位址的動作，通常稱為*名稱解析*。領域可以細分成幾個部份，也就是此範例中的 `www`、`example` 和 `com`。最後一個領域名稱部分就是最上層領域 (Top Level Domain, TLD)。在這個範例中的 TLD 就是 `com`。TLD 代表名稱解析程序的最上層。TLD 可分為通用 (gTLD，例如 `com`、`org` 和 `net`)，或是國家專用 (ccTLD，例如 `de` 即代表德國)。所有的領域部分集合起來，就可產生完整領域名稱 (Fully Qualified Domain Name, FQDN)。

資源

在這個範例中的資源是指 `index.html`。此部份指定連結到資源的完整路徑。資源可以是檔案，和此範例一樣。然而，它可能也是 CGI 程序檔、JavaServer 頁面，或是一些其他資源。

負責任的網際網路機制，例如領域名稱系統 (DNS)，可以將針對領域 `www.example.com` 的查詢，轉遞到一部或多部擁有資源的電腦。接著 Apache 會將實際的資源傳送給用戶端，在此範例中是指 `index.html` 網頁。這時該檔案是位於最上層目錄。不過，資源也可以位於子目錄中，例如 <http://www.example.com/linux/novell/suse>。

46.1.4 指示詞

在設定 Apache 時，通常要用指示詞來作為"組態選項"的同義字。指示詞是和 Apache 網頁伺服器軟體相關的技術詞彙。

46.2 安裝

SUSE Linux 上的 Apache，會以標準的預先定義組態來執行原始設定。您可以依照本章中的說明，在短時間之內完成設定並執行 Apache 網頁伺服器。您必須登入為 `root` 身分，才能安裝和設定 Apache。

46.2.1 使用 YaST 安裝 Apache

SUSE Linux Apache2 套件與 Apache 網站 (<http://httpd.apache.org>) 上提供的標準軟體套件，在檔案系統和應用程式配置上有些許不同。以下章節將敘述安裝 SUSE Linux Apache2 套件的詳細說明，並解釋適用的變化。

若要安裝簡易的 Web 伺服器，請執行下列步驟：

過程 46.1 快速安裝

- 1 以 GUI (圖形使用者介面) 或指令行模式啟動 YaST。
- 2 選取網路服務 → HTTP 伺服器。

- 3 按一下繼續以確認要安裝套件 Apache2 和 Apache2-prefork。
- 4 完成安裝後，會出現 *Apache 組態精靈*，接著您可以開始設定 Web 伺服器。

如上述說明中指出的缺點，您所安裝的套件不包含 PHP 且沒有資料庫支援功能。若要安裝包含 PHP 和資料庫支援功能的 Web 伺服器，請執行下列步驟：

過程 46.2 簡易安裝 Web 伺服器

- 1 以 GUI (圖形使用者介面) 或指令行模式啟動 YaST。
- 2 選取軟體 → 安裝和移除軟體。
- 3 選取過濾器中的選擇，然後核取包含 Apache2 的簡易 Web 伺服器。
- 4 按確定。
- 5 確認安裝個別套件後，即可完成 SUSE Linux Apache2 安裝程序。

若是進階使用者，SUSE Linux 會提供自訂套件選擇。若要自訂安裝 Web 伺服器，請執行下列步驟：

過程 46.3 使用 YaST 安裝預設 Apache RPM

- 1 以 GUI (圖形使用者介面) 或指令行模式啟動 YaST。選取軟體 → 安裝和移除軟體。
- 2 選取過濾器中的搜尋，接著在搜尋欄位中輸入 apache2。
- 3 選取要安裝的 apache2。
- 4 接著使用步驟 2 和步驟 3 來選擇模組。請參閱 [節 46.5, "Apache 模組" \[649\]](#)。
- 5 選擇模組後，請按接受。
- 6 接著，系統會提示您選擇一個用於所需 apache2-MPM 套件的相依性：apache2-prefork 或 apache2-worker。如需兩者差異處的說明，請參閱 [節 46.2.2, "多重處理模組" \[625\]](#)。如果您無法確定要選擇哪一個選項，

請選取 `apache2-prefork` 套件 (此選項為以 Unix 為基礎的作業系統所使用的預設值)，接著按確定。

7 確認安裝個別套件後，即可完成 SUSE Linux Apache2 安裝程序。

注：啟動 Web 伺服器

安裝 Apache 不會自動啟動 Web 伺服器。如需控制 Apache 啟動和關閉的資訊，請參閱 [節 46.3.3, "啟動、開始和停止 Apache" \[643\]](#)。

46.2.2 多重處理模組

如 [使用 YaST 安裝預設 Apache RPM \[624\]](#) 中所提及，SUSE Linux 會提供兩種不同的多重處理模組 (MPM) 搭配 Apache 使用。MPM 會負責接收和處理網頁伺服器所收到的要求，因此屬於網頁伺服器軟體的核心部分。

Prefork MPM

Prefork MPM 會實作未產生執行緒、正在進行 Prefork 的網頁伺服器。這個模組會使得網頁伺服器產生類似 Apache 版本 1.x 的行為，也就是將每個要求獨立分開，並為個別子程序進行 fork 來加以處理。這樣發生問題的要求就不會影響其他要求，進而避免網頁伺服器出現鎖定現象。

雖然透過這種以程序為主的方法可以提供穩定性，但是比起 Worker MPM 方法，Prefork MPM 會耗用較多的系統資源。Unix 作業系統會將 Prefork MPM 當作預設 MPM。

重要：本文件的 MPM

本文件會假設 Apache 是使用 Prefork MPM。

Worker MPM

Worker MPM 會提供多執行緒網頁伺服器。執行緒是"輕量級"的程序。執行緒和程序相比的優點是，它消耗的資源較少。Worker MPM 不只會 Fork 子程序，它還可使用執行緒和伺服器處理程序，來為要求提供服務。完成 Prefork 的子程序屬於多執行緒。

這種方法因為耗用比 Prefork MPM 更少的系統資源，因此可以提高 Apache 的執行效能。主要缺點之一就是 worker MPM 的穩定性：當某執行緒毀損時，程序的所有執行緒都會受到影響。最嚴重的情況下，甚至還會造成伺服器當機。尤其是在負載量高的 Apache 上使用 CGI (已於 ["通用開道介面：mod_cgi"](#) [651] 說明) 時，可能就會因執行緒無法與系統資源進行通訊而產生內部伺服器錯誤。

在 Apache 上使用 worker MPM 的另外一點爭議，就是並非所有可用的 Apache 模組 (see [節 46.5, "Apache 模組"](#) [649]) 都能安全地使用執行緒，這樣就無法配合 worker MPM 使用。

警告: PHP 屬於 Apache 模組 (mod_php)

並非所有可用的 PHP 模組都是安全執行緒。因此最好不要搭配 worker MPM 來使用 mod_php。

46.2.3 預設的檔案系統與應用程式配置

SUSE Linux 會在預設位置中放入 Apache 套件檔案。這裡將列出最重要檔案的位置。

二進位檔

大部分在 SUSE Linux Apache 中的執行檔都會附加 2。這樣就可以在同時安裝 Apache 1.x 和 Apache 2.x 時，方便識別不同版本的二進位檔案。

/usr/sbin/httpd2

指向 [節 46.2.2, "多重處理模組"](#) [625] 所述選擇多重處理模組的符號連結。預設值是 httpd2-prefork。啟動程序檔會根據 MPM 的系統組態設定，來維護此 symlink。

/usr/sbin/httpd2-prefork

實際的 Apache2 執行檔。

/usr/sbin/apache2ctl

由 Apache HTTPD 專案提供，可用來啟動和停止網頁伺服器的控制程序檔。如需更多詳細資訊，請參閱 [節 46.3.3, "啟動、開始和停止 Apache"](#) [643] 或執行 `/usr/sbin/apache2ctl help`。

/etc/init.d/apache2

啟動和停止程序檔，可以提供 SUSE Linux 安裝的完整功能，並在開機階段啟動 Apache。這個程序檔會在啟動和停止伺服器之前先檢查有效的組態，以及覆寫組態的位置。它還可以讓使用者更方便地包含其他組態檔、載入模組，甚至在不用修改程序檔情況下啟動不同的伺服器例項。

/usr/sbin/rcapache2

方便使用的 `/etc/init.d/apache2 symlink`，因為其路徑預設不會出現 `/etc/init.d/`。只要使用 `rcapache2 start` 就可以啟動 Apache。

/usr/sbin/htpasswd2

可用來產生 `.htaccess` 驗證加密密碼的公用程式。如需如何使用這項工具的詳細資訊，請參閱 `htpasswd2(1) Man` 頁面。

組態檔案

大部分的組態檔案會出現在 `/etc/apache2` 底下。如需如何變更組態設定的詳細資訊，請參閱 [節 46.3, "組態" \[629\]](#)。

/etc/apache2/httpd.conf

上層組態檔案。儘可能不要變更這個檔案。這個檔案主要是用來包含其他組態檔案，以及宣告全域設定。

/etc/apache2/*.conf

有些外部 Apache 模組會將組態檔案放在目錄 `/etc/apache2/` 中，而且通常會用模組本身名稱 (`mod_*.conf`) 作為字首。

/etc/apache2/conf.d/*

其中會包含來自特定套件之其他不同組態檔案的目錄。如需取得範例說明，請參閱 ["服務 PHP: mod_php4、mod_php5" \[657\]](#)。

/etc/apache2/vhosts.d/*

其中包含虛擬主機之選擇性組態檔案的目錄。如需詳細資料，請參閱 [節 46.4, "虛擬主機" \[644\]](#)。

/etc/sysconfig/apache2

相關於 Apache2 的 SUSE Linux 組態檔案。其中包含控制 Apache 網頁伺服器的所有相關組態參數。YaST 會使用 `/etc/sysconfig/apache2` 來設定

Apache，請參閱 節 46.3.1, "使用 YaST 設定 Apache" [630]。它也可以手動進行編輯，請參閱 節 46.3.2, "手動設定 Apache" [637]。

日誌檔

依預設，Apache 會在下列檔案中提供有關其 Runtime 狀態的各種資訊：

`/var/log/apache2/error_log`

Apache 會在這個檔案中，記錄啟動、關閉通知與所有 Runtime 錯誤。

`/var/log/apache2/access_log`

所有送到網頁伺服器的要求都會記錄在這個檔案。預設的項目格式會使用組合格式，以便顯示有關主機、傳送要求的使用者代理程式與參照 URI 的資訊。

Document Root

實體目錄 `/srv/www/htdocs` 是 Apache 從其中提供網頁的預設位置。它扮演了用戶端要求的"根目錄"。如果要使用 Apache 發佈網頁，請將檔案結構儲存在此目錄或其下方。

例如 `http://www.example.com/index.html` 的 URL，將會參照到名為 `example.com` 領域之 SUSE Linux Apache 預設組態中的 `/srv/www/htdocs/index.html`。

46.2.4 手動建立模組

Apache 是以模組化方式建立，這表示這些模組可以提供網頁伺服器軟體本身的功能。這樣一來，進階使用者就可以撰寫自訂模組，來延伸 Apache 功能。如需詳細資訊，請參閱下面介紹的 Man 頁面。

apache2-devel

若要開發 Apache 模組或是編譯協力廠商模組，除了適當的開發工具外，還需使用 `apache2-devel` 套件。`apache2-devel` 也包含了 `apxs2` 工具，這是在編譯 Apache 其他模組時，需要用到的工具。

apxs2

apxs2 二進位檔案位在 `/usr/sbin` 下方：

- `/usr/sbin/apxs2`--適合用來建立可配合任何 MPM 使用的延伸模組。安裝位置是 `/usr/lib/apache2`。
- `/usr/sbin/apxs2-prefork`--適合用於 `prefork` MPM 模組。安裝位置是 `/usr/lib/apache2-prefork`。
- `/usr/sbin/apxs2-worker` -- 適合用於 `worker` MPM 模組。

apxs2 會安裝可供所有 MPM 使用的模組。另外兩個程式，則只能安裝供個別 MPM 使用的模組。apxs2 會將模組安裝在 `/usr/lib/apache2`，而 `apxs2-prefork` 則會將模組安裝在 `/usr/lib/apache2-prefork`。

apxs2 可以從原始程式碼進行模組編譯和安裝(其中包括必要的組態檔案變更)，並建立可於 Runtime 載入 Apache 的動態共用物件(DSO)。使用 `cd /path/to/module/source; apxs2 -c -i mod_foo.c` 指令，從原始程式碼安裝模組。有關 apxs2 的其他選項，將於 `apxs2(1)` man 頁面說明。接著根據 [節 46.3.2, "手動設定 Apache" \[637\]](#) 中的描述，使用 `APACHE_MODULES` 項目，在 `/etc/sysconfig/apache2` 中啟動這些模組。

46.3 組態

在 SUSE Linux 中執行 Apache 時，可以使用下面兩種方法進行設定：透過 YaST 或是手動設定。手動設定組態可以提供較詳細的設定，但是缺乏 YaST GUI 提供的方便性。

重要：組態變更

有些 Apache 的組態值變更，必須等到 Apache 重新啟動之後，才能生效。如果是透過 YaST 完成組態，並且「**HTTP 服務**」已經核取「**啟用**」，就會自動重新啟動。如需有關手動重新啟動的詳細資訊，請參閱 [節 46.3.3, "啟動、開始和停止 Apache" \[643\]](#) 內容。大多數的組態變更只需要透過 `rcapache2 reload` 進行重新載入。

46.3.1 使用 YaST 設定 Apache

您可以使用 YaST，將您網路上的主機變成 Web 伺服器。若要設定這類的伺服器，請啟動 YaST 並選取**網路服務** → **HTTP 伺服器**。第一次啟動模組時，會啟動「HTTP 伺服器精靈」以提示您如何決定有關伺服器管理的一些基本設定。

HTTP 伺服器精靈

「HTTP 伺服器精靈」包含五個步驟(或對話方塊)。在最後一個步驟的對話方塊中，您可以進入進階組態模式以進行更多特定的設定。

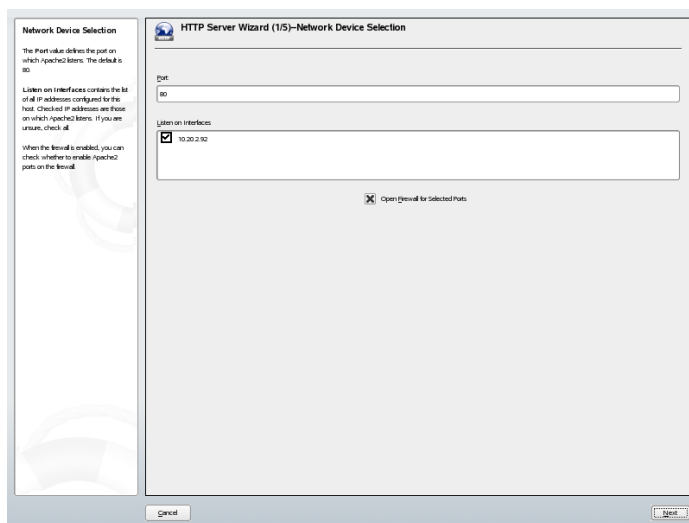
網路設備選擇

在此，您可以指定 Apache 用來監聽內送要求的網路介面和連接埠。您可以選取任何現有網路介面及其 IP 位址的組合。若連接埠(連接埠隸屬以下三種範圍：熟知的連接埠、註冊的連接埠和動態或私人的連接埠)不供其他服務使用，則皆可供您使用。

另外，預設設定為監聽所有在連接埠 80 上的網路介面(IP 位址)。當防火牆啟動時，您可以檢查是否已啟動防火牆上的 Apache 連接埠。

核取**開啟選取埠的防火牆**選項，可開啟防火牆中 Web 伺服器監聽的連接埠。若要使 Web 伺服器在網路(包括 LAN、WAN 或公用網際網路)上為可用狀態，請核取此選項。但是，若是處於測試階段且不須由外部網路存取 Web 伺服器，則可以關閉監聽連接埠。當您接受使用預設設定或已變更的任何設定後，請按一下**下一步**以繼續進行設定。

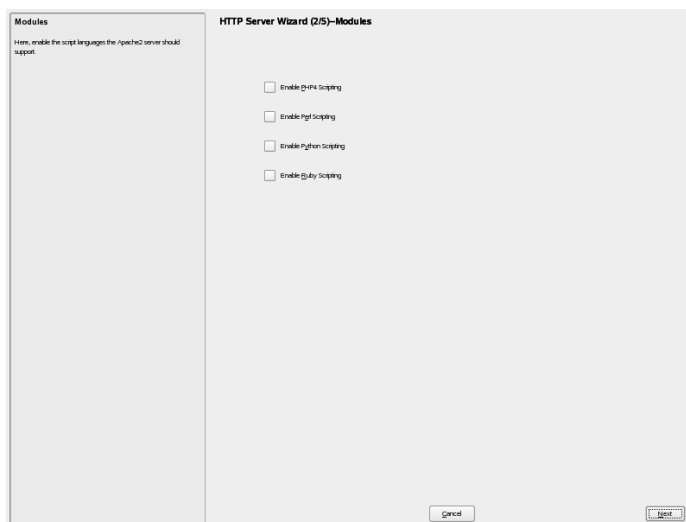
圖形 46.1 HTTP 伺服器精靈：網路設備選擇



模組

SUSE Linux Apache 套件隨附多種不同的 Apache 模組。這些模組可以擴充 Apache 的功能並適用於許多工作。您可以在 *模組組態* 選項中載入和卸載啟動伺服器時，使用的各種 Apache 模組。如需有關模組的詳細說明，請參閱 [節 46.5, "Apache 模組" \[649\]](#)。按一下 *下一步* 繼續。

圖形 46.2 HTTP 伺服器精靈：模組

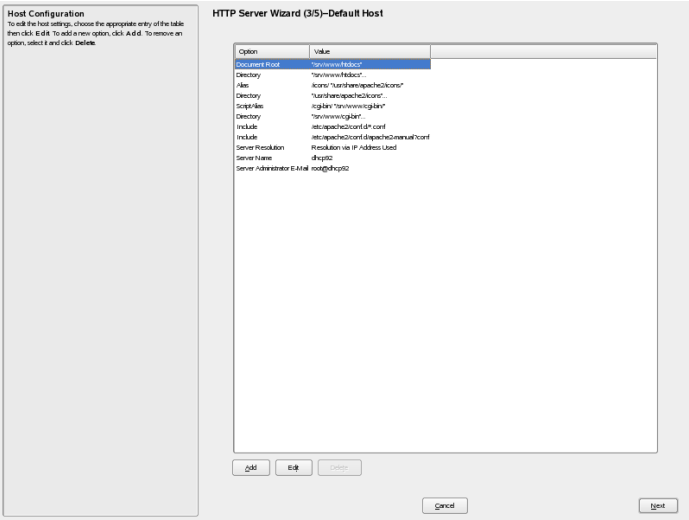


預設主機

此選項與預設 Web 伺服器相關。如 [節 46.4, "虛擬主機" \[644\]](#) 中所述，Apache 可以在一部實體電腦上提供多個網域。組態檔案中第一個宣告的網域 (或虛擬主機) 通常會被視為預設主機。若要編輯主機設定，請在表格中選擇要適當的項目，然後按一下 **編輯**。若要新增主機，請按一下 **新增**。若要刪除主機，請選取主機，然後按一下 **刪除**。

您可以在這個步驟中決定是否要將 SSL (Secure Sockets Layer，安全通訊端層) 選項和值加入主機設定中。請閱讀 ["新增 SSL 支援" \[637\]](#) 中的說明，以取得更多相關資訊。

圖形 46.3 HTTP 伺服器精靈：預設主機



此處為伺服器預設值的清單：

Document Root

如 "Document Root" [628] 中所述， /srv/www/htdocs 是 Apache 從其中提供網頁的預設位置。

Directory

/srv/www/htdocs 是網頁的位置。

Alias

藉由 Alias 指令的幫助， URL 可以映射到實體檔案系統位置。這表示即使某個路徑不在檔案系統的 Document Root 中，仍可藉由該路徑別名的 URL 進行存取。

預設的 SUSE Linux Alias /icons 會指向 /usr/share/apache2/icons，即目錄索引檢視模式中顯示的 Apache 圖示。

Directory

/usr/shareapache2/icons 是 Alias 目錄的位置。

Script Alias

類似 `Alias` 指令。`ScriptAlias` 指令可以映射檔案系統的 URL 位置。兩者差別在於 `ScriptAlias` 可以將目標目錄指定作為一個 CGI 位置，這表示 CGI 程序檔會在該位置中執行。

Directory

`/srv/www/cgi-bin` 是 `ScriptAlias` 目錄的位置。

Include

`/etc/apache2/conf.d/*.conf` 是一個包含隨附於某些套件的組態檔案的目錄。`/etc/apache2/conf.d/apache2-manual.conf` 是一個包含所有 `apache2-manual` 組態檔案的目錄。

Server Resolution

此選項是指 [節 46.4, "虛擬主機" \[644\]](#)。

按 *HTTP* 標題決定要求伺服器選項可以讓一個 `VirtualHost` (虛擬主機) 對該伺服器名稱的要求做出回應 (請參閱 [節 46.4.1, "以名稱為基礎的虛擬主機" \[645\]](#))。

按伺服器 IP 位址決定要求伺服器選項可讓 Apache 按用戶端傳送的 HTTP 標題資訊選取要求的主機。如需有關以 IP 為基礎的虛擬主機的更多資訊，請參閱 [節 46.4.2, "以 IP 為基礎的虛擬主機" \[647\]](#)。

Server Name

這可以指定用戶端用來存取 Web 伺服器的預設 URL。使用 FQDN (請參閱 [領域 \[622\]](#)) 即可在 `http://FQDN` 或其 IP 位址存取 Web 伺服器。

Server Administrator E-Mail

伺服器管理員電子郵件選項可提供 Web 伺服器管理員的電子郵件位址。

完成設定預設主機步驟後，請按一下下一步繼續下一個組態對話方塊。

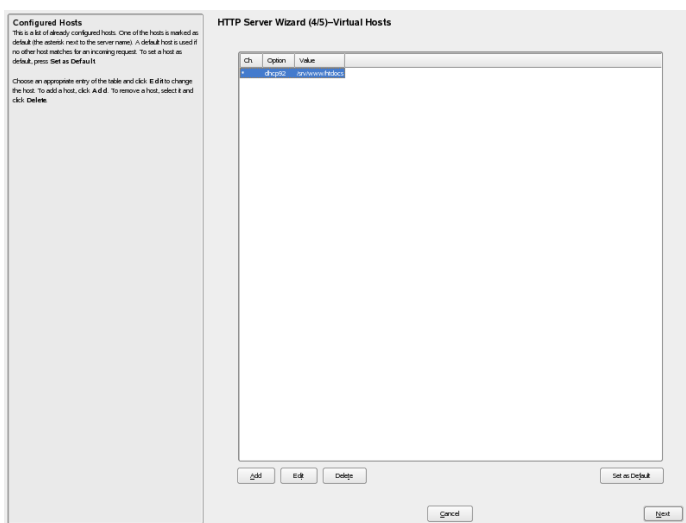
虛擬主機

在此步驟中，精靈會顯示已設定的虛擬主機清單 (請參閱 [節 46.4, "虛擬主機" \[644\]](#))。且其中一個主機會被標示為預設主機 (伺服器名稱旁邊會有一個星號)。若要設定預設主機，請選取該伺服器，接著按一下設定為預設。

若要新增主機，請按一下新增，接著會出現一個對話方塊，請在對話方塊中輸入該主機的基本資訊。*伺服器識別*中包括伺服器名稱、伺服器來源根目錄

和管理員電子郵件。視窗左側框架中的說明文字則會對每個項目做出詳細說明。*伺服器解析*是用來決定辨別主機的方法。您可以指定要從 HTTP 標題識別要求伺服器，或是按伺服器 IP 位址識別要求伺服器。另一個識別主機的方法，是按用戶端連線到伺服器時使用的 IP 位址來決定虛擬主機。您也可以核取該選項來啟用 SSL 支援。另外，您還可以指定憑證檔案的路徑。只要按一下瀏覽選擇路徑即可，預設目錄為 `/etc/apache2/ssl.crt`。輸入所有資訊後，請按一下下一步繼續組態設定的最後一個步驟。

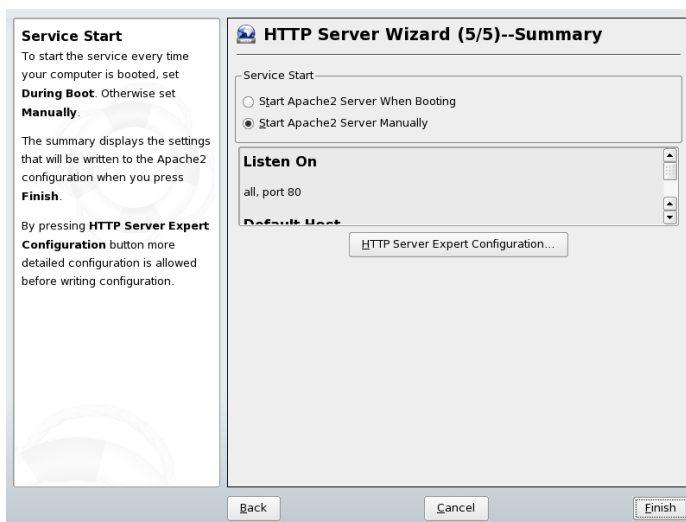
圖形 46.4 HTTP 伺服器精靈：虛擬主機



摘要

這是精靈的最後一個步驟。您可以在這裡決定 Apache 伺服器啟動的方式和時間：啟動時間或手動啟動。之前選取的連接埠也會與預設和虛擬主機一同顯示。如果您接受目前設定，請按一下完成以完成組態設定。

圖形 46.5 HTTP 伺服器精靈：摘要



HTTP 伺服器進階組態

HTTP 伺服器模組還可讓您對目前組態進行更多調整。按一下 *HTTP 伺服器進階組態* 可查看更多組態選項。以下為您可以進行變更的組態：

監聽

選取 *監聽* 設定，接著按一下 *編輯* 會開啟新的視窗，請在該視窗新增、刪除或編輯項目。

模組

選取 *模組* 設定並按一下 *編輯* 後，您可以按一下 *切換狀態* 以變更 Apache2 模組的狀態。按一下 *新增模組* 以新增模組。

預設主機

選取 *預設主機* 並按一下 *編輯*，接著可開始編輯主機設定。您也可以新增、編輯或刪除選項。

主機

選取 *主機* 並按一下 *編輯*，接著您可以新增、刪除、編輯主機設定，或選取要做為預設主機的主機。

您可以在以上所有對話方塊中按一下**紀錄檔案**，以查看錯誤錯誤紀錄和存取紀錄。按一下**確定**完成組態設定並返回「YaST 控制中心」。

新增 SSL 支援

若要新增 SSL 選項至主機，請在「HTTP 伺服器精靈」的步驟 3 (預設主機) 中按一下**新增**。如果您的伺服器已完成設定且您將不再使用精靈，您可以選取「HTTP 伺服器組態」對話方塊中的**預設主機**或按一下**編輯**和**新增**來安裝 SSL 選項。這兩種方式中皆會出現一個快顯視窗，您可以將視窗捲動到 SSL 選項，接著按**確定**確認動作。接著系統會要求您為選取的選項輸入一個值。設定值僅為**開啟**或**關閉**兩種值，該對話方塊會要求您輸入適當的值。如果您無法確定要輸入哪個值，請參考設定 SSL 時有關參數值的文件。按一下**確定**後，該選項和值會出現在主機組態清單中。按一下**下一步**可進入組態對話方塊中的下一個步驟。

如果 SSL 出現在主機組態清單中，請按一下**編輯**開啟 SSL 組態對話方塊。如果清單中沒有顯示 SSL 選項，請按一下**新增**，選取 SSL 並按一下**確定**，接著會自動開啟對話方塊。您可以在此新增、刪除或編輯 SSL 選項。按一下**確定**返回「HTTP 伺服器精靈」。

46.3.2 手動設定 Apache

手動設定 Apache 是指透過 root 使用者身份來編輯純文字組態檔案。

重要: 沒有用於 Apache2 的 SUSEconfig 模組

SUSE Linux 已經移除用於 Apache2 的 SUSEconfig 模組。變更 `/etc/sysconfig/apache2` 之後，不再需要執行 `SuSEconfig`。

`/etc/sysconfig/apache2`

`/etc/sysconfig/apache2` 可控制部分的 Apache 全域設定，例如要載入的模組、要包含的其他組態檔案、伺服器應該啟動的旗標，以及應該加入到指令行的旗標。在這個檔案中的每個組態選項都會詳細加以說明，因此本文將不予以介紹。針對一般用途的網頁伺服器，`/etc/sysconfig/apache2` 應該可以符合任何組態需求。如果需要特定的組態，請參閱 ["在 /etc/apache2/httpd.conf: Global Environment 區段中的 Apache 指示詞" \[638\]](#) 的相關內容。

重要：自動在伺服器啟動時產生的檔案

`/etc/sysconfig/apache2` 會在網頁伺服器啟動或重新啟動時，自動建立或編輯下列檔案。

- `/etc/apache2/sysconfig.d/loadmodule.conf` -- 在 Runtime 載入的模組
- `/etc/apache2/sysconfig.d/global.conf` -- 整個伺服器的一般設定
- `/etc/apache2/sysconfig.d/include.conf` -- 已包含組態檔案的清單

請不要手動編輯這些檔案。而是編輯 `/etc/sysconfig/apache2` 中的相關設定。

如需精確的組態調整內容，請查閱 `/etc/apache2/*` 中的檔案，尤其是針對虛擬主機、全域環境或主要伺服器的手動組態變更。

在 `/etc/apache2/httpd.conf`: Global Environment 區段中的 Apache 指示詞

SUSE Linux 會以 `/etc/apache2/httpd.conf` 作為其他組態檔案的參考重點。僅針對啟用 `/etc/sysconfig/apache2` 沒有提供的功能，進行檔案編輯。位在 `httpd.conf` *Global Environment* (全域環境) 區段中的指示詞，會影響 Apache 的所有操作。

下列段落內容將介紹一些 YaST 沒有提供的指示詞。Global Environment 和 VirtualHost，都會用到類似 Document Root ([Document Root \[633\]](#)) 等基本和必要的核心指示詞。

下列參數和指示詞將依邏輯連結和組態範圍進行排列。這些內容都會設定在 `/etc/apache2/httpd.conf`。

LoadModule *module_identifier* /path/to/module

LoadModule 指示詞可以指定在 Runtime 載入 Apache 模組。
module_identifier 是根據使用說明所指定的模組名稱。/path/to/module 可以是指向檔案的絕對或相對路徑。

範例 46.1 *LoadModule 指示詞*

```
LoadModule rewrite_module /usr/lib/apache2-prefork/mod_rewrite.so
```

在 SUSE Linux 中，並不需要直接使用 LoadModule 陳述式。而是使用 /etc/sysconfig/apache2 中的 APACHE_MODULE。

MaxClients 數目

Apache 可以同時處理的用戶端數目上限。MaxClients 必須大到足夠處理網站預計同時收到的要求數目，而且必須小到確保實體 RAM 足以處理所有處理程序的大小。

Timeout 秒數

指定 Apache 在報告要求逾時之前的等候時間。

在 /etc/apache2/httpd.conf: Main Server 區段中的 Apache 指示詞

當所有 VirtualHost 都沒有處理用戶端要求，而必須轉由預設或主要伺服器來處理這些要求時，就會套用 Main Server（主要伺服器）區段中的指示詞。此外，這個區段所定義的參數就是所有已設定虛擬主機的預設值。因此，所有在 Main Server 的指示詞也可以從 VirtualHost 內容進行設定，以便覆寫這些預設值。

DirectoryIndex 檔案名稱

設定 Apache 要搜尋的檔案，以完成未指定檔案的 URL。預設值是 index.html。例如，當用戶端要求 URL `http://www.example.com/foo/`，而目錄 foo 包含檔名為 index.html 的檔案時，Apache 就會向用戶端傳送這個網頁。請使用空格分隔來宣告多個檔案。

範例 46.2 *DirectoryIndex* 指示詞

```
DirectoryIndex index.html index.shtml start.php begin.pl
```

AllowOverride All | None | 選項

這個指示詞只能用於 `<Directory></Directory>` 宣告。請參閱 [Directory \[633\]](#)。

`AllowOverride` 可指定 `.htaccess` 檔案 (或由 `AccessFileName` 指定的其他檔案, 請參閱 "[AccessFileName 檔案名稱](#)" [641]) 可以覆寫哪些存取和顯示選項。

下面是可能的設定值:

全部

`.htaccess` 檔案可以覆寫全部選項。

無

`.htaccess` 檔案不可以覆寫任何選項。

AuthConfig

可以使用 `.htaccess` 檔案來為目錄進行密碼保護。

FileInfo

允許在 `.htaccess` 檔案中使用控制文件類型的指示詞。典型的範例就是使用 `ErrorDocument` 來設定自訂的錯誤頁面 (請參閱 <http://httpd.apache.org/docs-2.0/mod/core.html#errordocument>)。

索引

如果找不到任何 `DirectoryIndex` 文件, Apache 就會使用這個參數來控制目錄內容的顯示畫面。

限制

可以控制目錄或特定用戶端檔案的存取權限。在 `.htaccess` 檔案中使用 `Allow`、`Deny` 和 `Order` 等指示詞, 便可達到控制目的。如需有關使用這些指示詞的詳細資訊, 請參閱存取模組說明文件 (http://httpd.apache.org/docs-2.0/mod/mod_access.html)。

選項

允許在 `.htaccess` 檔案中使用 `Options` 和 `XBitHack` 指示詞。`Options` 指示詞 (<http://httpd.apache.org/docs-2.0/mod/core.html#options>) 可用來控制特定目錄中將提供哪些伺服器功能。`XBitHack` 指示詞 (http://httpd.apache.org/docs-2.0/mod/mod_include.html#xbithack) 允許將包含執行位元集的檔案分析成 SSI (請參閱 "伺服器端包含 `mod_include`" [650])。

重要

這些設定會循環套用到目前的目錄及其子目錄。除了「全部」和「無」，其餘選項都可以用空格分隔來加以組合。

範例 46.3 `AllowOverride` 指示詞

```
<Directory /srv/www/htdocs>
    AllowOverride None
</Directory>
<Directory /srv/www/htdocs/project>
    AllowOverride All
</Directory>
<Directory /srv/www/htdocs/project/webapp>
    AllowOverride Indexes Limit AuthConfig
</Directory>
```

`AccessFileName` 檔案名稱

`AccessFileName` 可設定能夠覆寫全域存取許可權、和其他目錄設定的檔案名稱 (請參閱 [Directory \[633\]](#))。

預設設定是 `.htaccess`。請使用空格分隔來宣告多個檔案。

範例 46.4 `AccessFileName` 指示詞

```
AccessFileName .htaccess .acl permission.txt
```

`ErrorLog` 檔案 / " / 指令"

指定 Apache 在其中記錄錯誤訊息的檔案名稱。不然，Apache 也可以記錄至指令或是程序檔。預設設定是 `/var/log/apache2/error_log`。

範例 46.5 *ErrorLog* 指示詞

```
ErrorLog /var/log/apache2/error_log  
ErrorLog "| /path/to/script"
```

LogLevel 層級

這個指示詞可以設定日誌訊息要記錄的詳細程度。下列是根據遞增的詳細程度層級 (遞減的訊息嚴重性) 所排列出的層級

- emerg
- alert
- crit
- error
- warn
- notice
- info
- debug

預設設定是 warn，這是每日操作建議使用層級。使用 info 和 debug 層級，可以提供有效資訊來達到除錯目的。

範例 46.6 *LogLevel* 指示詞

```
LogLevel debug
```

在 `/etc/apache2/httpd.conf`: **Virtual Hosts** 區段中的 **Apache** 指示詞

如果要在實體電腦上提供多個領域或是主機名稱，就必須使用 `VirtualHost` 容器。這些容器會宣告在組態的 `Virtual Hosts` (虛擬主機) 區段。如需有關虛擬主機使用語法和功能的詳細資訊，請參閱 [節 46.4, "虛擬主機" \[644\]](#)。

46.3.3 啟動、開始和停止 Apache

如果要在開機時間啟動 Apache 網頁伺服器，請使用 YaST 的 Runlevel 編輯器。如果要啟動它，請選取 YaST 中的「系統」→「系統服務 (Runlevel)」。接著，再瀏覽到「*apache2*」項目。選擇「啟用」，就可讓 Apache 在電腦開機時自動啟動。有經驗的使用者可能會使用 `chkconfig` 工具，透過下面指令行來達到相同效果：`/sbin/chkconfig -a apache2`。

如果要啟動或停止 Apache，請以 `root` 使用者身分來使用 `/usr/sbin/rcapache2` 程序檔。這樣 `/usr/sbin/rcapache2` 就會採用下列參數來啟動和停止 Apache 網頁伺服器：

start

啟動 Apache 網頁伺服器。

startssl

啟動提供 SSL 支援的 Apache 網頁伺服器。如需有關設定提供 SSL 之 Apache 的詳細資訊，請參閱 ["新增 SSL 支援" \[637\]](#) 和 ["Secure Sockets Layer 和 Apache: mod_ssl" \[654\]](#) 的內容。

停止

停止 Apache 網頁伺服器。

configtest

可在不需要實際停止、啟動或重新啟動網頁伺服器情況下，測試 Apache 組態。因為每次伺服器在啟動、重新載入或重新啟動時都會強制進行這項測試，所以通常並不需要明確地執行這項測試。

重新啟動

首先停止網頁伺服器，接著再重新啟動。

try-restart

重新啟動執行中的網頁伺服器。

restart-hup

向 Apache 網頁伺服器傳送 `SIGHUP` 訊號來重新啟動。這是很少使用的做法。

graceful 和 reload

在關機之前，通知所有已進入 fork 狀態的 Apache 處理程序先完成它們的要求，接著再停止網頁伺服器。當每個處理程序都結束之後，就會取代成新開始的處理程序，最後完全重新啟動 Apache。

提示

`rcapache2 reload` 是生產環境中較受歡迎的 Apache 重新啟動方法，因為這種方法可以讓所有用戶端在不會導致連線中斷情況下取得服務。

狀態

檢查 Apache 網頁伺服器的 Runtime 狀態。

範例 46.7 下面是啟動和停止 Apache 時的範例輸出

```
tux@sun # rcapache2 status
Checking for httpd2:                                unused

tux@sun # rcapache2 configtest
Syntax OK

tux@sun # rcapache2 start
Starting httpd2 (prefork)                            done

tux@sun # rcapache2 status
Checking for httpd2:                                running

tux@sun # rcapache2 graceful
Reload httpd2 (graceful restart)                    done

tux@sun # rcapache2 status
Checking for httpd2:                                running
```

錯誤組態檔案可能會導致 Apache 無法正確啟動，或是完全無法啟動。如果完全不能啟動，可能連任何訊息都無法顯示。請務必在每次啟動和重新啟動時檢查主要錯誤日誌。

46.4 虛擬主機

虛擬主機一詞，是指 Apache 從相同一部實體電腦提供多個 URI (資源識別字串，Universal Resource Identifier)。這是指同時由一部實體電腦的單一網頁伺服器來執行多個領域 (例如，`www.example.com` 和 `www.example.net`)。

使用虛擬主機的目的，經常是為了節省管理工作 (只需要維護一部網頁伺服器) 和硬體成本花費 (不需要將各個領域安裝在特定伺服器)。虛擬主機可以使用名稱、IP 或是連接埠作為基礎。

虛擬主機可以透過 YaST 進行設定 (請參閱 [預設主機 \[632\]](#))，或是手動編輯 `httpd.conf` 的 `Virtual Host` 區段 (請參閱 [節 46.3.2, "手動設定 Apache" \[637\]](#))。

依預設，在 SUSE Linux 中執行的 Apache，已經在 `/etc/apache2/vhosts.d/` 準備好虛擬主機的組態檔案。這個目錄 (`vhost.template`) 也會提供虛擬主機的基本樣板。虛擬主機組態也可以加入到其他位置，例如加入到一個檔案，再接著將其包含到組態中。

重要

使用 `httpd2 -s` 指令，便可方便地檢查虛擬主機的設定。這個指令會輸出 Apache 可理解的虛擬主機設定，並協助您確認有得到預期結果。如果在執行 Apache 時有使用例如 `-DSSL` 的旗標，您就必須在測試時使用相同的旗標，例如 `httpd2 -S -DSSL`。

46.4.1 以名稱為基礎的虛擬主機

使用以名稱為基礎的虛擬主機時，每個 IP 位址可以為超過一個網站提供服務。Apache 會使用用戶端所傳送 HTTP 標題中的主機欄位，將其要求傳送連線到符合其中一個虛擬主機宣告的 `ServerName` 項目。如果沒有找到相符的 `ServerName`，就會預設使用第一個指定的 `VirtualHost`。

`NameVirtualHost` 會啟動 Apache 組態的 `Virtual Host` (虛擬主機) 區段。

NameVirtualHost

`NameVirtualHost` 會判斷 Apache 網頁伺服器的所在 IP 位址，以及 (選擇性) 要用哪個連接埠來傾聽已在 HTTP 標題中包含領域名稱的用戶端要求。

第一個引數可以是完整領域名稱 (Fully Qualified Domain Name)，但是建議最好使用 IP 位址。第二個引數是連接埠 (選擇性)。根據預設會使用連接埠 80，而且可以透過 `Listen` 指示詞 ([網路設備選擇 \[630\]](#)) 來設定。

IP 位址和連接埠號碼都可以使用萬用字元 `*`，來接收所有介面上的要求。IPv6 位址必須包在方括號中。

範例 46.8 以名稱為基礎的 *VirtualHost* 項目變化

```
# NameVirtualHost IP-address[:Port]
NameVirtualHost 192.168.1.100:80
NameVirtualHost 192.168.1.100
NameVirtualHost *:80
NameVirtualHost *
NameVirtualHost [2002:c0a8:164::]:80
```

<VirtualHost></VirtualHost> 以名稱為基礎的內容

<VirtualHost></VirtualHost> 區塊包含了特定領域所套用的資訊。Apache 會使用這個區段所包含的指示詞，來接收已定義 *VirtualHost* 的用戶端要求。這裡也可以使用 *VirtualHost* 內容所允許的任何 Apache 指示詞。開啟 *VirtualHost* 標籤，就可採用以名稱為基礎之虛擬主機組態中的下列引數：

- 先前使用 *NameVirtualHost* 指示詞所宣告的 IP 位址 (或是完整領域名稱)。
- 先前使用 *NameVirtualHost* 指示詞所宣告的選擇性連接埠號碼。

允許使用萬用字元 `*` 作為 IP 位址的替代符號。這個語法只適用於在 *NameVirtualHost ** 結合使用萬用字元。如果是使用 IPv6 位址，該位址就必須用方括號包住。

範例 46.9 以名稱為基礎的 VirtualHost 指示詞

```
<VirtualHost 192.168.1.100:80>
    ServerName www.example.com
    DocumentRoot /srv/www/htdocs/example.com
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com-error_log
    CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>

<VirtualHost 192.168.1.100:80>
    ServerName www.example.net
    DocumentRoot /srv/www/htdocs/example.net
    ServerAdmin webmaster@example.net
    ErrorLog /var/log/apache2/www.example.net-error_log
    CustomLog /var/log/apache2/www.example.net-access_log common
</VirtualHost>

<VirtualHost [2002:c0a8:164::]>
    # 2002:c0a8:164:: is the IPv6 equivalent to 192.168.1.100
    ServerName www.example.org
    DocumentRoot /srv/www/htdocs/example.org
    ServerAdmin webmaster@example.org
    ErrorLog /var/log/apache2/www.example.org-error_log
    CustomLog /var/log/apache2/www.example.org-access_log common
</VirtualHost>
```

在這個範例中，`www.example.com` 和 `www.example.net` 等領域，都是裝載在 IP 位址 `192.168.1.100` 的電腦上。所有傳入到網頁伺服器的要求，都會以第一個 `VirtualHost` 作為預設虛擬主機。

而 `ErrorLog` (已說明於 "[ErrorLog 檔案 / 指令](#)" [641]) 與 `CustomLog` (請參閱 http://httpd.apache.org/docs-2.0/mod/mod_log_config.html#customlog) 等指示詞則不需要包含領域名稱。在此，使用您選擇的名稱。

46.4.2 以 IP 為基礎的虛擬主機

這項替代虛擬主機組態方法需要為一台機器設定多個 IP。一個 Apache 例項可裝載多個領域，每個領域都會指定不同的 IP。

重要: IP 位址與以 IP 為基礎的虛擬主機

實體伺服器必須為每部以 IP 為基礎的虛擬主機設定一個 IP 位址。當該電腦沒有安裝多張網路卡時，也可以使用虛擬網路介面 (IP 別名)。

設定 IP 別名

如果要讓 Apache 裝載多個 IP，該實體電腦必須可以接受多個 IP 的要求。此稱為多重 IP 裝載。此外還必須啟動核心中的 IP 別名。這是 SUSE Linux 中的預設值。

當核心的 IP 別名設定好之後，就可以使用指令 `ifconfig` 和 `route` 來設定主機上的其他 IP。這些指令必須以 `root` 身份執行。

在下列範例中，我們假設已經主機已經將 IP 192.168.0.10 指定給網路設備 `eth0`。輸入指令 `ifconfig`，檢視主機 IP。使用下列的指令，可以新增其他 IP 位址：

```
ip addr add 192.168.0.20/24 dev eth0
ip addr add 192.168.0.30/24 dev eth0
```

所有 IP 位址都已指定給相同的實體網路設備 (`eth0`)。

<VirtualHost></VirtualHost> 以 IP 為基礎的內容

當系統設定好 IP 別名 (或者主機已經裝置多張網路卡) 之後，就可以設定 Apache。每部虛擬伺服器都必須具備個別的 `VirtualHost` 區塊。

下列範例將示範，Apache 正執行於 IP 192.168.1.10 的電腦上，其中還另外裝載了 IP 192.168.0.20 與 192.168.0.30 的兩個領域。這個特殊範例只能應用於內部網路，因為從 192.168.0.0 到 192.168.0.255 之間的 IP，不會傳送經過公用網際網路。

範例 46.10 以 IP 為基礎的 VirtualHost 指示詞

```
<VirtualHost 192.168.0.20>
    ServerName www.example.com
    DocumentRoot /srv/www/htdocs/example.com
    ServerAdmin webmaster@example.com
    ErrorLog /var/log/apache2/www.example.com-error_log
    CustomLog /var/log/apache2/www.example.com-access_log common
</VirtualHost>

<VirtualHost 192.168.0.30>
    ServerName www.example.net
    DocumentRoot /srv/www/htdocs/example.net
    ServerAdmin tux@example.net
    ErrorLog /var/log/apache2/www.example.net-error_log
    CustomLog /var/log/apache2/www.example.net-access_log common
</VirtualHost>
```

這裡出現的 VirtualHost 指示詞，只能指定給 192.168.0.10 以外的其他介面。如果 192.168.0.10 也設定 Listen 指示詞 (已說明於 [網路設備選擇 \[630\]](#))，這時就必須建立以 IP 為基礎的個別虛擬伺服器，為該介面或是已套用在 /etc/apache2/httpd.conf 的 Main Server (主要伺服器) 區段所找到的指示詞 (請參閱 "[在 /etc/apache2/httpd.conf: Main Server 區段中的 Apache 指示詞](#)" [\[639\]](#)) 回應 HTTP 要求。

46.5 Apache 模組

Apache 軟體屬於模組設計：除了部分核心工作，其餘所有功能皆由模組處理。這種進步幅度很大，甚至連 HTTP 都是由模組 (http_core) 處理。

Apache 模組可以在 Buildtime 編譯成 Apache 二進位檔案，或是在 Runtime 動態載入。如需有關 Runtime 載入的詳細資訊，請參閱 "[LoadModule module_identifier/path/to/module](#)" [\[639\]](#) 了解手動載入模組，以及參閱 [模組 \[631\]](#) 了解使用 YaST 載入模組。

SUSE Linux 中的 Apache 已在 apache2 RPM (這裡省略字首「mod_」) 中，隨附提供下列可用模組：access、actions、alias、asis、auth、auth_anon、auth_dbm、auth_digest、auth_ldap、autoindex、cache、case_filter、case_filter_in、cern_meta、cgi、charset_lite、dav、dav_fs、deflate、dir、disk_cache、dumpio、echo、env、expires、ext_filter、file_cache、headers、imap、include、info、ldap、log_config、log_forensic、logio、mem_cache、mime、mime_magic、negotiation、proxy、proxy_connect、proxy_ftp、proxy_http、rewrite、setenvif、speling、ssl、status、

suexec、unique_id、userdir、usertrack 與 vhost_alias。此外，SUSE Linux 還提供下列 Apache 模組，作為需要個別安裝的 RPM 套件：

apache2-mod_auth_mysql、apache2-mod_fastcgi、
apache2-mod_macro、apache2-mod_murka、apache2-mod_perl、
apache2-mod_php4、apache2-mod_php5、apache2-mod_python 與
apache2-mod_ruby。

本節將針對這些模組中的部分模組進一步討論說明。如需基本版本中的其他模組說明，請造訪 Apache 模組網站，位址是 <http://httpd.apache.org/docs-2.0/mod/>。如需取得協力廠商模組，請造訪 <http://modules.apache.org/>。

Apache 模組可以細分為三種不同類別：基本模組、延伸模組和外部模組。

46.5.1 基本模組

基本模組會依預設編譯到 Apache。這些模組可以提供使用，除非已在 Buildtime 明確省略。SUSE Linux Apache 只有編譯最少的基本模組，但是所有這些模組都可以用共用物件方式取得：這些模組並非包含在 /usr/sbin/httpd2 二進位檔案本身，而是透過設定 APACHE_MODULES，在 Runtime 包含到 /etc/sysconfig/apache2。

伺服器端包含 mod_include

mod_include 提供了在將資料傳送給用戶端之前的檔案處理方法。一般說來，mod_include 是用來為文件包含在送達用戶端之前會依序分析成 HTML 的檔案。這就是其稱為 Server-Side Includes (SSI) 的原因。

使用 SSI 時，伺服器端就會在受到格式化 SGML 註解觸發時，執行特殊指令。這些 SGML 指令會提供下列語法：

```
<!--#element attribute=value -->
```

如需元件和屬性值的清單，請參閱 http://httpd.apache.org/docs-2.0/mod/mod_include.html 所提供的 mod_include 說明文件。

如果要在 SUSE Linux 使用 `mod_include`，請在 `/etc/sysconfig/apache2` 的 `APACHE_MODULES` 加入 `include`，或是使用 [模組 \[631\]](#) 所介紹的 YaST。

提示

使用 `XBitHack` 指示詞 (http://httpd.apache.org/docs-2.0/mod/mod_include.html#xbithack)，指示 Apache 透過 SSI 指示詞的 `execute` 位元集來分析檔案。

這樣表示，您不需要變更檔案副檔名來標示其中包含 SSI 元件 (如上面範例中的 `.shtml`)，而是使用一般的 `.html` 檔案並執行 `chmod +x myfile.html` 就可進行分析。

通用閘道介面：mod_cgi

`mod_cgi` 可讓 Apache 傳送由外部 CGI (Common Gateway Interface) 程式或程序檔所建立的內容。它會扮演介於實體電腦之可用程式設計語言和 Apache 網頁伺服器之間的例項。理論上，CGI 程序檔可以用任何程式設計語言來編寫。通常會使用類似 Perl 或 C 等程式設計語言。`mod_cgi` 是在網站中包含動態內容的常用方法。

CGI 程式設計不同於「一般」程式設計；因為 CGI 程式和程序檔必須能夠產生 `Content-type: text/html` MIME 類型來建立 HTML 輸出。

範例 46.11 下面是使用 Perl 的簡單 CGI 程序檔

```
# /path/to/perl
print "Content-type: text/html \n \n";
print "Hello, World. ";
```

繫結至程式設計語言 (例如 `mod_php5`) 和 `mod_cgi` 等模組之間的差異，在於是否可以結合 `mod_cgi` 和 `mod_suexec` (請參閱 ["使用 mod_suexec，依據不同使用者身份來執行 CGI" \[653\]](#))。這個結合結果，可讓 CGI 程序檔透過指定使用者 ID 來執行。通常，單獨使用 `mod_cgi` 或 `mod_php5` 的程序檔，會以 Apache 使用者的使用者 ID (SUSE Linux 的預設設定：`wwwrun`) 來執行。針對程式設計語言所設計的模組 (例如 `mod_php5` 或 `mod_ruby`)，會在 Apache 中內嵌常駐解譯器，以便在該 Apache 使用者 ID 之下執执行程序檔。

這樣一來，透過 `mod_suexec` 的 CGI 就可提高系統管理明確性，因為 CGI 處理程序會指派給個別使用者，而不是指派給網頁伺服器本身。同時，檔案系統

也會因為採用下列組合而提高安全性：程序檔只會繼承使用者的檔案系統權限。如果是使用模組，程序檔會獲得網頁伺服器的使用者檔案許可權，這可能會造成檔案系統資料不慎外洩。

當用戶端向網頁伺服器發出的要求停止時，CGI 就會終止。這樣表示 CGI 不是常駐程式，而且會在終止之後釋放所有佔用的資源。這項優點在程式設計不正確時更能顯出其妙用。而使用模組時，因為解譯器屬常駐型，所以程式設計的錯誤效應會一直累積。這樣可能會導致無法釋出資源，例如資料庫連線，最後必須重新啟動 Apache。

如果要在 SUSE Linux 使用 `mod_cgi`，請在 `/etc/sysconfig/apache2` 的 `APACHE_MODULES` 加入 `cgi`，或是使用 [模組 \[631\]](#) 所介紹的 YaST。在 SUSE Linux 中的預設 CGI 目錄是 `/srv/www/cgi-bin/`。

如果要手動編輯 Apache 組態檔案，請依照下面範例作為設定 `mod_cgi` 的指示方針。

範例 46.12 手動啟動 `mod_cgi`

```
# Global Environment
LoadModule cgi_module /path/to/mod_cgi.so

# Main Server and/or Virtual Host and/or
# Directory and/or .htaccess context
AddHandler cgi-script .cgi .pl

# Main Server and/or Virtual Host context
ScriptAlias /cgi-bin/ /srv/www/cgi-bin/

# Alternatively, explicitly allow CGI scripts in a directory
# Main Server and/or Virtual Host context
<Directory /srv/www/some/dir>
    Options +ExecCGI
</Directory>
```

46.5.2 延伸模組

一般說來，Apache 軟體套件會包含標示為延伸的模組，但是通常不會使用靜態方式將這些模組編譯到伺服器中。在 SUSE Linux 中，這類模組可以依照共用物件方式，在 Runtime 時載入到 Apache 提供使用。

使用 `mod_suexec`，依據不同使用者身份來執行 CGI

結合 `mod_cgi` ("通用閘道介面: `mod_cgi` " [651]) 和 `mod_suexec`，便可依照特定使用者和群組來執行 CGI 程序檔。您必須使用 `/usr/sbin/suexec2` 中的 `suEXEC` 程式，來達到這個目的。Apache 會在每次執行 CGI 程序檔或程式時，呼叫這個包裝函式。接著，包裝函式和程式都會取得所指派的設定使用者和群組 ID。這樣就可依據設定的使用者或群組來執行。

雖然這項方法可以大幅降低使用者產生 CGI 程序檔時的相關安全性風險，但是在使用時仍要考量下列重要事項：

使用 *suEXEC* 的考量

- `suEXEC docroot` -- 所有程序檔都會限制在這個基本目錄中執行。這樣表示搭配 `suexec` 的程序檔一定要在 `docroot` 內執行，否則將會產生錯誤。`docroot` 會在 `suEXEC` 編譯期間進行設定，而且不能在 `Runtime` 進行變更。SUSE Linux 的預設設定是 `/srv/www`。
- `uidmin` -- 這是指使用者在透過 `suEXEC` 執行程序檔時，必須具備的最小 ID。這樣可以避免程序檔由系統使用者 (例如，`root`) 執行。請不要建立包含 ID 小於 `uidmin` 的使用者 (當操作 `mod_suexec` 時一定會使用該使用者時)。在 SUSE Linux 的預設 `uidmin` 是 96。
- `gidmin` -- 這個設定概念和 `uidmin` 相同，但是是針對群組 ID。在 SUSE Linux 的預設 `gidmin` 是 96。
- 目錄和檔案許可權 -- 所執行的程序檔應該屬於相同於已指定的 `suEXEC` 使用者和群組。此外，非此擁有者的其他使用者絕對不能寫入這個檔案。同時，只有該擁有者具有程序檔所在目錄的寫入權限。
- `suEXEC safepath` -- 所有在程序檔 (例如 Perl) 中所使用的程式必須出現在已標示為可安全執行 `suexec` 的路徑。`safepath` 會在 `suEXEC` 編譯階段完成設定，而且不可以在 `Runtime` 時進行變更。SUSE Linux 中的預設 `safepath` 是 `/usr/local/bin:/usr/bin:/bin`。

如果發生 `mod_suexec` 所導致的錯誤，請查閱 `/var/log/apache2/suexec.log` 中的 `suexec` 日誌檔案。

如果要在 SUSE Linux 使用 `mod_suexec`，請在 `/etc/sysconfig/apache2` 的 `APACHE_MODULES` 加入 `suexec`，或是使用 [模組 \[631\]](#) 所介紹的 YaST。請務必記住，執行 `suexec` 時需要用到 `mod_cgi`。

`mod_suexec` 最能在虛擬主機環境中表現其功能，請參閱 [節 46.4, "虛擬主機" \[644\]](#)。如果要指定執行 CGI 程序檔時所依據的特定使用者和群組，請在提供虛擬主機宣告的檔案中使用下列語法 (SUSE Linux 的預設檔案是 `/etc/apache2/vhosts.d/*`)：

範例 46.13 `mod_suexec` 組態

```
<VirtualHost 192.168.0>
# ...
ScriptAlias /cgi-bin/ /srv/www/vhosts/www.example.com/cgi-bin/
SuexecUserGroup tux users
# ..
</VirtualHost>
```

在這個範例中，`SuexecUserGroup` 使用者名稱 群組語法會為所有位於 `/srv/www/vhosts/www.example.com/cgi-bin/` 的程序檔，指派 `tux` 的使用者 ID，以及使用者的群組 ID。

Secure Sockets Layer 和 Apache: `mod_ssl`

`mod_ssl` 會為用戶端和網頁伺服器之間的 HTTP 通訊，提供使用安全通訊端層 (Secure Sockets Layer, SSL)、以及傳輸層安全性 (Transport Layer Security, TLS) 通訊協定的強式加密。為了完成這個目的，伺服器會在回覆任何 URL 要求之前，先傳送提供伺服器有效身分相關資訊的 SSL 憑證。這樣一來，就可以保證該伺服器是此通訊的唯一專屬正確端點。此外，該憑證會在用戶端和伺服器端建立加密連線，以便在沒有洩漏敏感、純文字內容的風險情況下傳輸資訊。使用 `mod_ssl` 搭配 Apache 的最明顯特徵，就是 URL 的字首都加上 `https://`，而不是 `http://`。

在網頁伺服器端上，SSL 和 TLS 要求的預設連接埠是 443。這樣和"一般" Apache 傾聽連接埠 80、SSL/TLS 啟用 Apache 傾聽連接埠 443 之間，就不會產生衝突。事實上，HTTP 和 HTTPS 可以執行相同的 Apache 例項。通常這時會使用一部虛擬主機 (請參閱 [節 46.4, "虛擬主機" \[644\]](#))，將連接埠 80 和連接埠 443 的要求分派到不同的虛擬伺服器。

重要: 以名稱為基礎的虛擬主機和 SSL

我們不可能在只有一個 IP 位址的伺服器上，執行多部啟用 SSL 的虛擬主機。連線到這類設定的使用者，將會在每次造訪該 URL 時收到警告訊息，表示憑證與伺服器名稱不符。每個啟用 SSL 的領域都需要使用不同的 IP 位址或連接埠，才能根據有效的 SSL 憑證來完成通訊。

不管是否有收到該警告訊息，您還是會獲得相同於任何有效 SSL 網站提供的加密層級。這樣表示只要是可接受的警告訊息，網頁伺服器和用戶端之間仍屬安全的通訊。這時唯一包含伺服器身分的內容 (經過有效 SSL 憑證保證) 將會遭到捨棄。

如果要在 SUSE Linux 使用 `mod_ssl`，請在 `/etc/sysconfig/apache2` 的 `APACHE_MODULES` 加入 `ssl`，或是使用 [模組 \[631\]](#) 所介紹的 YaST。此外，網頁伺服器必須設定成傾聽標準的 HTTPS 連接埠 443。這項設定可以在 `/etc/apache2/listen.conf` 手動完成，或是使用 YaST 的「傾聽」功能表項目 (請參閱 [網路設備選擇 \[630\]](#)) 來完成設定。

以 root 身分輸入 `cd /usr/share/doc/packages/apache2;`
`./certificate.sh`，就可以建立測試用 SSL 憑證。請遵循畫面指示來建立此 SSL 憑證。最後產生的憑證檔案會出現在目錄 `/etc/apache2/ssl*` 中。

包括全球認證的"實際"憑證，是由類似 Thawte (<http://www.thawte.com/>) 或 Verisign (www.verisign.com) 等供應商發出。

如果要手動編輯 Apache 組態檔案，請依照下面範例作為設定 `mod_ssl` 的指示方針。

範例 46.14 手動設定 *mod_ssl*

```
# Global Environment
# listen on the standard SSL port
Listen 443
# load module only if rcapache2 start-ssl was issued
<IfDefine SSL>
LoadModule ssl_module /path/to/mod_ssl.so
</IfDefine>

# Main Server context
# include global (server-wide) SSL configuration
# that is not specific to any virtual host
# only if ssl_module was loaded
<IfModule mod_ssl.c>
Include /etc/apache2/ssl-global.conf
</IfModule>
```

提示

請不要忘記為連接埠 443 上已啟用 SSL 的 Apache 開啟防火牆。移至「安全性和使用者」→「防火牆」→「允許服務」，就可以透過 YaST 來進行開啟。接著，將「HTTPS 伺服器」加入到「允許服務」清單中。

46.5.3 外部模組

正式說來，標示為外部的模組不會包含於 Apache 發行版本。然而，SUSE Linux 會提供幾種已經可以使用的外部模組。本章將大概介紹一些外部模組及其功能。

使用 Perl 來管理 Apache: *mod_perl*

mod_perl 會在 Apache 中內嵌常駐 Perl 解譯器。這樣就可以避免 *mod_cgi* 每次收到 CGI 要求時就呼叫外部執行檔，而降低可能的負荷。*mod_perl* 則可另外允許配合 Perl 程式設計語言，來控制多種 Apache 功能。

如果要在 SUSE Linux 使用 *mod_perl*，請安裝 *apache2-mod_perl* RPM，並透過 YaST (模組 [631]) 或是手動方式，在 */etc/sysconfig/apache2* 啟用模組。完成安裝和啟動之後，*/etc/apache2/conf.d/* 之下就會出現另一個組態檔案 *mod_perl.conf*。此外，*mod_perl* 啟動程序檔也會依照 *mod_perl-startup.pl* 進行安裝。如需有關如何使用這個模組的詳細資訊，請參閱 *mod_perl* 網站 (<http://perl.apache.org/>) 所提供的說明文件。

服務 PHP: mod_php4、mod_php5

PHP 是一種受歡迎的程式設計語言，其原本是針對在網路上的用途而設計。目前有兩種版本，PHP4 和 PHP5。其中 PHP4 是使用傳統的 PHP 概念和方法，而 PHP5 則引進新的物件導向程式設計可能性，以及其他多種進階功能。SUSE Linux 會同時提供 mod_php4 和 mod_php5。這兩個程式可以將 PHP 解譯器內嵌到 Apache，成為其中的常駐模組。

如果要在 SUSE Linux 使用 mod_php4 或 mod_php5，請分別安裝 RPM (apache2-mod_php4、apache2-mod_php5)，並透過 YaST (模組 [631]) 或是手動方式，在 /etc/sysconfig/apache2 啟用模組。

完成安裝和啟動之後，/etc/apache2/conf.d/ 下面就會出現針對個別模組所產生的不同組態檔案 (php4.conf 或 php5.conf)。PHP 網站 (<http://www.php.net>) 已針對結合 Apache 使用 PHP，提供了相當豐富的說明資源。

Python 和 Apache: mod_python

mod_python 會在 Apache 中內嵌 Python 解譯器。Python 是一種物件導向的程式設計語言，語法相當簡潔易懂。其中較少見但卻十分方便的特色，就是程式結構是由原始程式碼縮排來決定，而不是由類似 begin 和 end 等常見分界元件來決定。

如果要在 SUSE Linux 使用 mod_python，請安裝 apache2-mod_python RPM，並透過 YaST (模組 [631]) 或是手動方式，在 /etc/sysconfig/apache2 啟用模組。如需有關如何使用這個模組的詳細資訊，請參閱 mod_python 網站 (<http://www.modpython.org/>) 所提供的說明文件。

Apache 的 Ruby 解譯器: mod_ruby

mod_ruby 會在 Apache 網頁伺服器中內嵌 Ruby 解譯器，允許執行 Ruby CGI 程序檔。Ruby 是一種相當新的物件導向、高層次的程式設計語言，在某些功能方面很類似 Perl 和 Python。和 Python 一樣，它有一個清楚、有效的語法。另一方面，Ruby 採用縮寫 (例如以 \$.r 來表示輸入檔中的最後一行數字)，有些程式設計師很喜歡這種特點，但也有些人不喜歡這個特點。Ruby 的基本概念很像 Smalltalk。

如果要在 SUSE Linux 使用 `mod_ruby`，請安裝 `apache2-mod_ruby` RPM，並透過 YaST (模組 [631]) 或是手動方式，在 `/etc/sysconfig/apache2` 啟用模組。如需有關如何使用這個模組的詳細資訊，請參閱 `mod_ruby` 網站 (<http://www.modruby.net/en/index.rbx>) 所提供的說明文件。

原始檔案系統存取：`mod_dav`

`mod_dav` 可提供 Apache 的 WebDAV (以 Web 為基礎的分散式製作與版本設定) 功能。WebDAV 是 HTTP 通訊協定的延伸功能，可以讓使用者共同編輯並管理遠端伺服器的檔案。WebDAV 具有類似 FTP 的功能，主要差異是它會採用 HTTP 通訊協定來存取伺服器。實際上，`mod_dav` 可以讓 Apache 網頁伺服器提升成為進階的遠端檔案系統。

如果不需要，最好限制透過 WebDAV 存取目錄的權限。至少要實行的預防措施，就是在 Location 指示詞中設定 WebDAV 資源的 HTTP 基本驗證，以及「限制」條件。

如果要存取 WebDAV 資源，用戶端必須安裝可提供 WebDAV 功能的軟體。SUSE Linux 已經隨附下列 WebDAV 功能：字首為 `webdav:` // 或 `webdavs:` // (適用於透過 SSL 連線執行 WebDAV 情況) 的 Konqueror，可用來連接至 Apache WebDAV 檔案系統。

`mod_dav` 需要使用模組 `mod_dav_fs`，以便提供 WebDAV 的實際檔案系統存取權限。如果要在 SUSE Linux 使用 `mod_dav`，請透過 YaST (模組 [631]) 或是手動方式，在 `/etc/sysconfig/apache2` 中啟用模組。為 `mod_dav_fs` 執行相同步驟。如需有關如何使用這個模組的詳細資訊，請參閱 `mod_dav` 網站 (http://httpd.apache.org/docs-2.0/mod/mod_dav.html) 所提供的說明文件。

提供使用者首頁：`mod_userdir`

SUSE Linux 中的 `mod_userdir`，已經預設將每位使用者在 `~/public_html` 資料夾的內容提供作為公開網頁。這些網頁的存取 URL 將為 `http://www.example.com/~使用者名稱/`。

提示

基於安全性考量，SUSE Linux 中的 `mod_userdir` 會禁止使用者存取在 `root` 使用者主目錄中的任何目錄。此外，您可以使用下列指令行，特別指定只有特定使用者可以擁有公開首頁：

```
# Main server context
UserDir disabled
UserDir enabled tux wilber
```

如果要在 SUSE Linux 使用 `mod_userdir`，請透過 YaST (模組 [631]) 或是手動方式，在 `/etc/sysconfig/apache2` 中啟用模組。如需有關如何使用這個模組的詳細資訊，請參閱 `mod_userdir` 網站 (http://httpd.apache.org/docs-2.0/mod/mod_userdir.html) 所提供的說明文件。

變更 URL 結構：mod_rewrite

`mod_rewrite` 通常被認為是"操縱 URL 的瑞士刀。"這個模組可根據指定規則集，來重寫即時收到的要求 URL。從

`http://www.example.com/display.php?cat=2&article=1&lang=de`
的 `http://www.example.com/2/1/de`，就可看出典型的執行結果。

「[URL 重寫指南](#)」會介紹這種功能強大、但是屬於複雜模組的優點和缺點：

"您可能在初次用過 `mod_rewrite` 之後就像抱到燙手山芋，從此不願意再使用；或是發現它的強大功用，從此愛不釋手。"

下列組態內容中都可以設定 `RewriteRule` 集：主機伺服器、虛擬主機、目錄和 `.htaccess` 檔案。<http://httpd.apache.org/docs-2.0/misc/rewriteguide.html> 所提供的「URL 重寫指南」，可以作為了解使用 `mod_rewrite` 進行 URL 重寫的絕佳起點。

如果要在 SUSE Linux 使用 `mod_rewrite`，請透過 YaST (模組 [631]) 或是手動方式，在 `/etc/sysconfig/apache2` 中啟用模組。

46.6 安全性

向公用網際網路公開的網頁伺服器，必須持續進行系統管理。這時當然會無法避免地出現與軟體、意外錯誤設定相關的安全性問題。下面是可用來處理這些問題的幾項秘訣。

保持更新

SUSE會在發現Apache軟體弱點時，發出安全性建議事項。其中會包含應該要儘速套用的弱點修正指示。如需取得SUSE安全性公告郵件清單，請造訪http://www.suse.com/us/private/support/online_help/maillinglists/。如需SUSE Linux套件安全性問題的線上最新資訊，請造訪<http://www.novell.com/linux/security/securitysupport.html>。

此外，您應該要訂閱Apache公告郵件清單(<http://httpd.apache.org/lists.html#http-announce>)，其中將會張貼最新發行與錯誤修正程式。

DocumentRoot 許可權

根據預設，DocumentRoot目錄/srv/www/htdocs與CGI目錄/srv/www/cgi-bin屬於root使用者。您不應該變更這些許可權。如果所有人都可以寫入目錄，任何使用者都可以將檔案放到目錄中。這些檔案接著可能會由Apache透過wwwrun許可權來執行，而這種情況可能會造成使用者取得不預期的檔案系統資源存取權限。請使用子目錄/srv/www/htdocs與/srv/www/cgi-bin，以便結合Directory指示詞(請參閱[Directory \[633\]](#))來組織使用者或特定領域資料。

CGI 與 SSI 目錄

使用Perl、PHP、SSI或是任何其他程式設計語言的互動式程序檔，基本上都可以執行任意指令。限制CGI與SSI(請參閱"[通用闡道介面：mod_cgi](#)" [\[651\]](#)、[Script Alias](#) [\[634\]](#)與"[伺服器端包含 mod_include](#)" [\[650\]](#))不要全域執行，而是執行到特定目錄中，就可以提供降低風險的選項。

另一種可能做法，是搭配mod_suexec(請參閱"[使用 mod_suexec，依據不同使用者身份來執行 CGI](#)" [\[653\]](#))來執行常見CGI。針對Apache模組，具有例如"[服務 PHP：mod_php4、mod_php5](#)" [\[657\]](#)所述安全性敏感組態的解譯器，可以協助維護安全的網路環境。

存取許可權

通常在測試環境中，網頁伺服器的存取許可權會因為測試設定緣故，而較為輕忽處理。這種情形可能會導致意外洩漏敏感資訊，甚至讓不當使用者存取整部伺服器。請使用 `Order` 指示詞 (http://httpd.apache.org/docs-2.0/mod/mod_access.html#order) 並結合 `.htaccess` 檔案 (請參閱 "[AccessFileName 檔案名稱](#)" [641])，設定特定網站僅供指定使用者或用戶端存取。

此外，您還可以使用「混淆安全性防禦」方法：這種做法的典型範例，就是在非標準的连接埠上執行 Apache (請參閱 [網路設備選擇](#) [630])。這樣就會產生附加连接埠的 URL，例如 `http://www.example.com:8765`；測試環境可以接受這種 URL。

46.7 疑難排解

如果 Apache 未啟動，網頁就無法存取，或是使用者無法連線到網頁伺服器，找出問題的發生原因是很重要的工作。下面是可在其中尋找錯誤原因、以及檢查重點項目的幾個常見位置。

首先是提供詳細錯誤原因的 `rcapache2` (請參閱 [節 46.3.3, "啟動、開始和停止 Apache"](#) [643])，因此在實際操作 Apache 時如果有使用這個指令，將對您非常有幫助。有時這個指令會引導您使用二進位檔案 `/usr/sbin/httpd2`，來啟動或停止網頁伺服器。請避免這樣做，並改而使用 `rcapache2` 程序檔。`rcapache2` 甚至還會提供解決組態錯誤的秘訣和提示。

第二個不可輕忽的地方，就是日誌檔案的重要性 (請參閱 "[日誌檔](#)" [628])。針對嚴重和不嚴重的錯誤，都可以從 Apache 日誌檔中找出錯誤發生原因。此外，如果日誌檔案需要更詳細的資訊，還可以透過 `LogLevel` 指示詞來控制日誌訊息的詳細程度 (請參閱 "[LogLevel 層級](#)" [642])。

提示

使用 `tail -F /var/log/apache2/*_log &` 來檢視 Apache 日誌訊息。接著再執行 `rcapache2 restart`。現在，請嘗試連接到瀏覽器，並檢查輸出結果。

這時經常發生的錯誤，就是沒有在伺服器防火牆組態中開啟 Apache 連接埠。如果是使用 YaST 來設定 Apache，就要透過其他選項來注意這個特定問題。

如果無法透過這些功能任何一項來追查出錯誤原因，則請查閱 http://httpd.apache.org/bug_report.html 的線上 Apache 問題資料庫。此外，也可以從 <http://httpd.apache.org/userslist.html> 取得可用的郵件清單，聯絡 Apache 使用者社群。推薦的新聞群組是 comp.infosystems.www.servers.unix。

46.8 若需詳細資訊

Apache 是廣泛使用的網頁伺服器。得以想見，目前已有許多網站從各種品質角度，提供有關 Apache 的支援和協助。在任何情況下，如果要研究 Apache 和相關可能性，都應該先以 <http://httpd.apache.org/docs-2.0/> 作為起點。

此外，RPM 套件 `apache2-doc` 還包含適用於本機安裝的 Apache 手冊和參考文件。如需取得一些 SUSE 特定組態提示，請參閱包含快速參考的 `/usr/share/doc/packages/apache2` 檔案。

RPM 套件 `apache2-example-pages`，還包含顯示網頁伺服器相關資訊的 Apache 範例網頁。

46.8.1 Apache 模組

以下位址提供有關外部 [節 46.5.3, "外部模組"](#) [656] Apache 模組來源的詳細資訊：

- <http://httpd.apache.org/docs-2.0/mod/>
- <http://www.php.net/manual/en/install.unix.apache2.php>
- <http://www.modpython.org/>
- <http://www.modruby.net/>
- <http://perl.apache.org/>

46.8.2 CGI

如需有關 `mod_cgi` (請參閱 "通用闡道介面: `mod_cgi`" [651]) 和 CGI 程式設計的詳細資訊，請參閱下面位址提供的說明內容：

- <http://www.modperl.com/>
- <http://www.modperlcookbook.org/>
- <http://www.fastcgi.com/>
- <http://www.boutell.com/cgiic/>

46.8.3 其他資源

如果您在 SUSE Linux 遇到與 Apache 有關的問題，請查閱 SUSE 支援資料庫，位址為：<http://portal.suse.com/sdb/en/index.html>。

Apache 的歷程網址是 http://httpd.apache.org/ABOUT_APACHE.html。此頁面也說明稱伺服器為 Apache 的原因。

關於從 1.3 升級至 2.0 版的資訊，可由下列網址取得：<http://httpd.apache.org/docs-2.0/en/upgrading.html>。

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膝上型電腦或 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。

47.1 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。當您出門在外使用膝上型電腦時，所有需要的檔案複本都必須在本機硬碟上。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工作，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。下列摘要讓您概略了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。

47.1.1 Unison

Unison 不是網路檔案系統。實際上，只是在本機儲存和編輯檔案。您可以手動執行 Unison 程式，將檔案同步化。第一次執行同步化時，會在兩個主機上建立資料庫，包含總和檢查碼、時戳以及選取檔案的權限。下次執行時，Unison 可以識別哪些檔案已變更，並建議要從其他主機傳輸或傳輸至其他主機。通常可以接受所有的建議。

47.1.2 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區，檔案及檔案的變更都儲存在此。在本機上執行的變更會交付至儲存區，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。

47.1.3 subversion

subversion 與不斷演變的 CVS 大不相同，subversion 是一脈相承的開發計劃。subversion 是 CVS 的技術改善後繼加強版。

與舊一代的產品相比，subversion 在許多方面都已大幅改善。由於其傳統包袱之故，CVS 只能維護檔案但卻無法維護目錄。目錄在 subversion 中也有版本歷史，而且可以像檔案一樣進行複製和重新命名。它也可以將中繼資料新增至每個檔案以及每個目錄。使用版本設定就可以充份維護此中繼資料。與 CVS 相反的是，subversion 支援透過專用通訊協定進行通透網路存取，例如 WebDAV (以 Web 為基礎的分散式製作與版本設定)。WebDAV 擴充了 HTTP 通訊協定的功能，而能對遠端網頁伺服器上的檔案進行協同寫入存取。

subversion 大部份是由已經存在的軟體套件組合而成。因此，Apache 網頁伺服器與 WebDAV 延伸程式總是與 subversion 一起執行。

47.1.4 mailsync

`mailsync` 與前面小節所述的同步化工具不同，它只能同步化信箱之間的電子郵件。這個程序可以套用至本機信箱檔案以及 IMAP 伺服器上的信箱。

根據電子郵件標頭所包含的訊息 ID，將個別訊息同步化或刪除。在個別信箱之間與信箱階層之間都可以同步化。

47.1.5 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。

47.2 選取程式時所要考慮的決定性因素

47.2.1 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。`subversion`、`CVS` 以及 `WebDAV` 都採用這種模式。

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。這是 `unison` 所遵循的概念，`rsync` 實際上使用的是用戶端模式，但是任何用戶端都可以當成伺服器使用。

47.2.2 可攜式

在許多其他的作業系統上也可以使用 subversion、CVS 以及 unison，包含各種 Unix 與 Windows 系統。

47.2.3 互動式與自動化

在 subversion、CVS、WebDAV 以及 Unison 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。

47.2.4 衝突：事件與解決方案

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 subversion 或 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。subversion 或 CVS 中的衝突通常都可以輕易解決。

Unison 會報告衝突，並允許將受影響的檔案排除在同步化之外。然而，變更的合併不像在 subversion 或 CVS 中那樣容易。

兩相比較下，subversion 或 CVS 在發生衝突時可以部份接受變更，但 WebDAV 只會在成功地進行完整修改時執行登記。

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。

47.2.5 選取和新增檔案

在其標準組態中，Unison 會同步化整個目錄樹狀結構。出現在樹狀結構中的新檔案會自動包括在同步化中。

在 subversion 或 CVS 中，必須分別使用 `svn add` 或 `cvs add` 指令，以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 `svn update` 以及 `svn status` 或 `cvs update` 輸出中的問號時。

47.2.6 歷程記錄

subversion 或 CVS 還有另一項功能，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。

47.2.7 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。subversion 與 CVS 在伺服器上還需要額外的空間，供儲存區資料庫使用。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。

47.2.8 GUI

Unison 提供圖形使用者介面，可顯示 Unison 所要執行的同步化程序。接受提議或排除同步化中的個別檔案。在文字模式中，以互動式的方式確認個別程序。

有經驗的使用者通常會從指令行執行 subversion 或 CVS。然而，在 Linux 系統中也有 cervisia 之類的圖形使用者介面，其他作業系統中也有 wincvs 之類的圖形使用者介面。許多開發工具 (像是 kdevelop) 以及文字編輯器 (像是 emacs) 都支援 CVS 或 subversion。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。

47.2.9 使用者親切性

Unison 與 rsync 使用簡便，而且也適於新手使用。CVS 與 subversion 則比較難操作。使用者必須瞭解儲存區與本機資料之間的互動。對資料的變更應該先在本機上與儲存區合併。這是使用 `cvs update` 或 `svn update` 指令來執行。接著必須使用 `cvs commit` 或 `svn commit` 指令將資料傳送回儲存區。只要瞭解此程序，即使是新手也能輕鬆使用 CVS 或 subversion。

47.2.10 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。Unison、CVS、rsync 以及 subversion 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 或 Unison。同樣地不建議在不安全的網路中使用 *pserver* 機制存取 CVS。subversion 與 Apache 一起執行，便提供了必要的安全性措施。

47.2.11 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。儘管 subversion 不像 CVS 一樣普及，它已運用於像是 subversion 專案本身的生產環境中。

Unison 仍算是相當新的產品，但是宣稱擁有高水準的穩定性。然而，它對於使用者錯誤更為敏感。一旦確認要將檔案刪除同步化，就不能再還原檔案。

表格 47.1 檔案同步化工具的功能：-- = 非常差、- = 差或無法使用、o = 中等、+ = 好、++ = 佳、x = 可用

	unison	CVS/subv.	rsync	mailsync
用戶端/伺服器	相等	C-S/C-S	C-S	相等
可攜式	Lin、 Un*x、Win	Lin、Un*x、 Win	Lin、 Un*x、Win	Lin、Un*x
互動	x	x/x	x	-
速度	-	o/+	+	+
衝突	o	++/++	o	+
檔案 Sel。	目錄	選項/檔案、目 錄	目錄	信箱
歷程記錄	-	x/x	-	-

	unison	CVS/subv.	rsync	mailsync
硬碟空間	0	--	0	+
GUI	+	o/o	-	-
困難度	+	o/o	+	0
攻擊	+(ssh)	+/(ssh)	+(ssh)	+(SSL)
資料損失	+	++/++	+	+

47.3 Unison 的介紹

Unison 是同步化和傳輸整個目錄樹狀結構的最佳解決方案。同步化是以雙向執行，而且可以透過直覺式圖形前端程式來控制。也可以使用主控台版本。同步化也可以自動化，因此不需要與使用者互動，但是經驗是必要的。

47.3.1 需求

Unison 必須安裝在用戶端以及伺服器上。在此環境中，*伺服器*一詞是指第二部遠端主機 (與 CVS 不同，詳述於 [節 47.1.2, "CVS" \[666\]](#))。

在下一小節中，Unison 是與 ssh 一起使用。在此例中，必須在用戶端上安裝 SSH 用戶端，並在伺服器上安裝 SSH 伺服器。

47.3.2 使用 Unison

Unison 所使用的方法是兩個目錄 (*roots*) 彼此互相關聯。此關聯是象徵性的 它不是線上的連線。以此為例，目錄配置如下列所示：

用戶端：	/home/tux/dir1
伺服器：	/home/geeko/dir2

您想要同步化這兩個目錄。使用者在用戶端上的身份為 **tux**，在伺服器上的身份為 **geeko**。第一件要做的事就是測試用戶端與伺服器之間的通訊是否可運作：

```
unison -testserver /home/tux/dir1 ssh: //geeko@server //homes/geeko/dir2
```

最常遭遇的問題為：

- 在用戶端與伺服器上使用的 **Unison** 版本是不相容的。
- 伺服器並不允許 **SSH** 連線。
- 兩個指定的路徑都不存在。

如果每個項目都可以運作，請省略 `-testserver` 選項。在第一次同步化期間，**Unison** 並不知道兩個目錄之間的關係，並且會針對個別檔案與目錄的傳輸方向送出建議。在 *動作* 資料欄中的箭頭會指出傳輸的方向。問號表示 **Unison** 無法建議傳輸方向，因為兩個版本都已經變更或者是新的版本。

方向鍵可以用來設定個別項目的傳輸方向。如果所有顯示項目的傳輸方向都正確，則只要按一下 *到*。

若要控制 **Unison** 的特性 (例如，對於沒有衝突錯誤的檔案是否要自動執行同步化)，可以在啟動該程式時，利用指定的指令行參數來控制。使用 `unison --help` 可檢視所有參數的完整清單。

範例 47.1 `~/.unison/example.prefs` 檔案

```
root=/home/tux/dir1
root=ssh: //wilber@server //homes/wilber/dir2
batch=true
```

使用者目錄 `~/.unison` 會為每一組同步化資料保留一個同步化日誌。組態集合，像是 `~/.unison/example.prefs`，也可以儲存於此目錄中。若要啟動同步化，請依照 `unison example.prefs` 中所示，將此檔案指定為指令行參數。

47.3.3 若需詳細資訊

Unison 的官方文件特別有用。因此，本小節只提供簡短的介紹。完整的手冊可在 <http://www.cis.upenn.edu/~bcpierce/unison/> 中以及在 **SUSE** 套件 `unison` 中找到。

47.4 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。

47.4.1 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 儲存區的資料應該包含在定期備份中。

當設定 CVS 伺服器時，透過 SSH 授與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 CVS_ROOT=tux@server: /serverdir
```

`cvs init` 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 指令都需要註解。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
```

47.4.2 使用 CVS

現在可以使用 `cvs co synchome` 從所有主機取出同步化儲存區。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 `cvs commit`。

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 `cvsv commit file1 directory1` 中指定它們。在將它們交付至伺服器前，必須使用像是 `cvsv add file1 directory1` 之類的指令，將新檔案與目錄新增至儲存區。接著，使用 `cvsv commit 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區，請現在取出 (參閱上方)。

使用 `cvsv update` 啟動與伺服器的同步化。依照 `cvsv update 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有版本的差異，請使用 `cvsv diff` 或 `cvsv diff file1 directory1` 指令。使用 `cvsv -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

已更新本機版本。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

已修改本機版本。如果在伺服器上有一些變更，可以在本機複本中將差異合併。

P

本機版本已使用伺服器版本修補。

C

本機檔案與儲存區中目前的版本衝突。

?

這個檔案並不存在於 CVS。

M 狀態是指在本機修改過的檔案。或是將本機複本交付至伺服器，或是移除本機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。如果您交付本機已修改的檔案，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，請查看檔案中的衝突記號(>與<)，並在兩個版本之間做一選擇。這有可能是相當麻煩的工作，您可能會決定捨棄變更、刪除本機檔案，然後輸入 `cvs up`，以便從伺服器擷取目前的版本。

47.4.3 若需詳細資訊

本小節僅針對 CVS 的許多可能性提供簡短的介紹。下列 URL 提供許多相關文件：

```
http://www.cvshome.org/  
http://www.gnu.org/manual/
```

47.5 Subversion 簡介

Subversion 是免費的開放式來源版本設定控制系統，而且普遍視之為 CVS 的後繼加強版，意即在 CVS 中已推出的功能，通常在 subversion 中也有。另外特別推薦的原因是，它兼具了 CVS 的優點，卻不必忍受其缺點。許多功能已經在 [節 47.1.3, "subversion" \[666\]](#) 中都已簡要說明過。

47.5.1 安裝 Subversion 伺服器

在伺服器上安裝儲存區資料庫是相單簡單的程序。Subversion 為此提供了專用的管理工具。建立新儲存區所要輸入的指令是：

```
svnadmin create /path/to/repository
```

其他的選項使用 `svnadmin help` 就可列出。與 CVS 大不相同的是，subversion 使用的不是 RCS 為基礎，而是 Berkeley Database。請確定不要在遠端檔案系統上安裝儲存區，例如 NFS、AFS 或 Windows SMB。該資料庫需要 POSIX 鎖定機制，但是這些檔案系統不支援。

`svnlook` 指令會提供現有儲存區的資訊。

```
svnlook info /path/to/repository
```

伺服器必須經過設定才能讓不同的使用者存取儲存區。您可以使用 Apache 網頁伺服器以及 WebDAV 來執行此動作，或使用 `svnserve` 以 subversion 來封裝伺服器。只要一啟動和執行 `svnserve`，就可以使用 `svn://` 或 `svn+ssh://URL` 來

存取儲存區。呼叫 `svn` 時應該要自我驗證的使用者，可以設定在 `/etc/svnserve.conf`。

要選用 `Apache` 或 `svnserve` 的決定取決於許多因素。建議您瀏覽 `subversion` 的書籍。在 [節 47.5.3, "若需詳細資訊" \[678\]](#) 中可以找到更多它的詳細資訊。

47.5.2 使用與作業

使用 `svn` 指令 (與 `cv`s 相似) 來存取 `subversion` 儲存區。符合對應儲存區且由設定正確之伺服器所提供的內容，可供任何用戶端以下列指令之一存取：

```
svn list http://svn.example.com/path/to/project
```

或

```
svn list svn://svn.example.com/path/to/project
```

使用 `svn checkout` 指令，將現有的專案儲存在目前的目錄中 (取出)：

```
svn checkout http://svn.example.com/path/to/project nameofproject
```

「取出」會在用戶端上建立新的子目錄 `nameofproject`。然後就可以執行各項作業 (新增、複製、重新命名、刪除)：

```
svn add file
svn copy oldfile newfile
svn move oldfile newfile
svn delete file
```

這些指令也可以用在目錄上。`subversion` 可以另外記錄檔案或目錄的屬性：

```
svn propset license GPL foo.txt
```

先前的範例會為 `license` 屬性設定 `GPL` 值。使用 `svn proplist` 顯示屬性：

```
svn proplist --verbose foo.txt
Properties on 'foo.txt':
license : GPL
```

使用 `svn commit` 將變更儲存至伺服器。另一個使用者可以使用 `svn update` 與伺服器同步化，在其工作目錄中併入您的變更。

與 `CVS` 不同的是，`subversion` 不需要使用 `svn status` 存取儲存區，就可以顯示它的工作目錄狀態。本機變更會顯示在五個資料欄中，而第一個資料欄是最重要的：

"

沒有變更。

'A'

物件已標示為新增。

'D'

物件已標示為刪除。

'M'

物件已修改。

'C'

物件發生衝突。

'I'

物件被忽略。

'?'

物件不是由版本設定控制所維護。

!!

物件已報告為遺失。當未使用 `svn` 指令而刪除或移動物件時，就會出現旗標。

'~'

物件是被當做檔案來維護，但是物件曾被目錄所取代，或是目錄被物件所取代。

第二個資料欄會顯示屬性的狀態。在 `subversion` 書籍中可以參閱所有其他欄的意義。

使用指令 `svn help` 以取得某指令的參數描述：

```
svn help proplist
proplist (plist, pl): 列出檔案、目錄或版本的所有屬性
用法: 1. proplist [PATH...]
      2. proplist --revprop -r REV [URL]
```

1. 列出工作複本中的版本屬性。
2. 列出儲存區版本上的尚未設定版本之遠端屬性。

...

47.5.3 若需詳細資訊

最主要的參考資料為<http://subversion.tigris.org/>的 subversion 專案首頁。可以在您安裝套件 subversion-doc 後，於 <file:///usr/share/doc/packages/subversion/html/book.html> 目錄中找到推薦的書籍，您也可以在线上找到，網址為<http://svnbook.red-bean.com/svnbook/index.html>。

47.6 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。

47.6.1 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 `rcrsyncd start` 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 `insserv rsyncd` 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線的日誌檔。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` 日誌檔中。若要啟動實際的傳輸，請提供目標目錄。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。

47.6.2 若需詳細資訊

關於 `rsync` 的重要資訊，請參閱 `man` 頁面。指令為 `man rsync` 和 `man rsyncd.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。

47.7 mailsync 簡介

mailsync 主要適用於下列三項工作：

- 將本機儲存的電子郵件以及儲存在伺服器上的郵件同步化。
- 將信箱移轉至另一種格式或另一個伺服器
- 信箱的完整性檢查或搜尋重複資料

47.7.1 組態與使用

mailsync 可以區分信箱本身 (*store*) 以及兩個信箱之間的連線 (*channel*)。「儲存區」與「通道」的定義是儲存在 `~/. mailsync` 中。下列段落說明一些儲存區的範例。

簡單的定義可能會如下所示：

```
store saved-messages {  
    pat Mail/saved-messages  
    prefix Mail/  
}
```

`Mail/` 是使用者主目錄下之子目錄，它包含電子郵件資料夾，其中包括 `saved-messages` 資料夾。如果 `mailsync` 是以 `mailsync -m saved-messages` 啟動，它會列出 `saved-messages` 中所有訊息的索引。如果建立了下列定義

```
store localdir {  
    pat Mail/*  
    prefix Mail/  
}
```

`mailsync -m localdir` 指令會列出所有儲存在 `Mail/` 之下的訊息。相對的，`mailsync localdir` 指令會列出資料夾名稱。IMAP 伺服器上某儲存區的設定將如下所示：

```
store imapinbox {  
    server {mail.edu.harvard.com/user=gulliver}  
    ref {mail.edu.harvard.com}  
    pat INBOX  
}
```

上述範例只能處理 IMAP 伺服器的主要資料夾。子資料夾的儲存區將如下所示：

```
store imapdir {  
server {mail.edu.harvard.com/user=gulliver}  
ref {mail.edu.harvard.com}  
pat INBOX.*  
prefix INBOX.  
}
```

如果 IMAP 伺服器支援加密的連線，則伺服器設定應該變更為

```
server {mail.edu.harvard.com/ssl/user=gulliver}
```

或者，如果不知道伺服器的憑證，則變更為

```
server {mail.edu.harvard.com/ssl/novalidate-cert/user=gulliver}
```

稍後會說明該字首。

目前在 Mail / 下的資料夾應該連線至 IMAP 伺服器的子目錄：

```
channel folder localdir imapdir {  
msinfo .mailsync.info  
}
```

mailsync 會使用 msinfo 檔案來追蹤已同步化的訊息。

mailsync folder 指令會執行下列動作：

- 展開兩邊的信箱圖樣。
- 從產生的資料夾名稱移除字首。
- 同步化成對的資料夾 (如果它們不存在就建立它們)。

因此在 IMAP 伺服器上的 INBOX.sent-mail 資料夾會與 Mail/sent-mail 本機資料夾同步化 (如果有前述的定義)。在個別資料夾之間的同步化將會依照下列方式執行：

- 如果在兩邊都已經有某一訊息，則不會執行任何動作。
- 如果在某一邊缺少某訊息而且為新訊息 (未列在 msinfo 檔案)，就會傳輸至該處。
- 如果訊息僅存在於某一邊而且為舊的 (已列於 msinfo 檔案)，就會刪除該處的訊息 (因為另一邊原先所存在的該訊息已遭刪除)。

若想事先知道同步化期間要傳送哪些訊息，以及要刪除哪些訊息，請使用 `mailsync older localdir` 來啟動具有通道與儲存區的 `mailsync`。此指令會產生所有在本機主機上列為新訊息的清單，以及同步化期間在 IMAP 那一邊將刪除的所有訊息清單。同樣地，`mailsync folder imapdir` 指令會產生所有在 IMAP 那一邊為新訊息的清單，以及同步化期間在本機主機上將刪除的所有訊息清單。

47.7.2 可能的問題

在資料遺失的事件中，最安全的方法就是刪除相關的通道日誌檔 `msinfo`。因此，僅存在於某一邊的所有訊息會被視為新訊息，所以會在下一次同步化期間予以傳輸。

只有具有訊息 ID 的訊息會包括在同步化中。缺乏訊息 ID 的訊息則會被忽略，亦即不會傳輸或刪除它們。缺乏訊息 ID 通常是有問題的程式在傳送或撰寫訊息時所造成。

在某些 IMAP 伺服器上，會使用 `INBOX` 來處理主要資料夾，而子資料夾則會以隨機選取的名稱來處理 (與 `INBOX` 及 `INBOX.name` 相對)。因此，像這樣的 IMAP 伺服器無法為子資料夾指定專用的圖樣。

成功地將訊息傳輸至 IMAP 伺服器後，`mailsync` 所使用的信箱驅動程式 (`c-client`) 會設定特殊的狀態旗標。因此，`mutt` 之類的電子郵件程式就無法將這些訊息識別成新訊息。請使用 `-n` 的選項以停用特殊狀態旗標的設定值。

47.7.3 若需詳細資訊

在 `/usr/share/doc/packages/mailsync/` 中的 `README`，是包括在 `mailsync` 中，它提供更多資訊。在此連線中，RFC 2076「一般網際網路訊息標頭」特別有趣。

Samba

使用 Samba，就可以將 Unix 機器設為 DOS、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。除了說明基本功能，本章節還會介紹 Samba 組態的基本概念，並說明可在網路中設定 Samba 的 YaST 模組。

可在數位文件中取得更多有關 Samba 的資訊。如果已安裝 Samba，若要取得更多的線上文件與範例，請在指令列輸入 `apropos samba` 以顯示部分手冊頁面或直接瀏覽 `/usr/share/doc/packages/samba` 目錄。在 `examples` 子目錄中可以找到加備註的範例組態 (`smb.conf.SuSE`)。

隨附 `samba` 套件版本 3 的一些重要新功能包含：

- 支援 Active Directory
- 加強 Unicode 支援
- 已全面修訂內部驗證機制
- 改善 Windows 200x 與 XP 列印系統的支援
- 在 Active Directory 領域中可以將伺服器設為成員伺服器
- 採用 NT4 領域，允許將 NT4 領域移轉至 Samba 領域

提示: 移轉至 Samba3

將 Samba 2.x 移至 Samba 3 時，有一些特殊的項目需要考慮。此主題的討論包含在「Samba HOWTO 文件集」中，裡面有整個章節專門討論它。在安裝 samba-doc 套件後，可以在 /usr/share/doc/packages/samba/Samba-HOWTO-Collection.pdf 中找到 HOWTO。

Samba 使用依據 NetBIOS 服務的 SMB 通訊協定 (伺服器訊息區塊)。由於來自 IBM、Microsoft 發行了此通訊協定的壓力，因此其他的軟體製造商可以建立對 Microsoft 領域網路的連線。使用 Samba 時，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。

NetBIOS 是用來進行機器之間通訊的軟體介面 (API)。在此提供了名稱服務。它允許連線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。現在可以針對不同的網路架構實行 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。Samba 預設是使用此對應名稱。

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 系統層次上整合 SMB 資源。

SMB 伺服器透過共用提供用戶端硬碟空間。共用包含伺服器上的目錄以及其子目錄。它是利用名稱來輸出，並且可藉由其名稱來存取。共用名稱可以設成任何名稱 它並不需要是輸出目錄的名稱。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。

48.1 設定伺服器

如果您想要使用 Samba 做為伺服器，請安裝 `samba`。以 `rcnmb start && rcsmmb start` 啟動 Samba 所需的服務，並以 `rcsmmb stop && rcnmb stop` 停止它們。

Samba 的主要組態檔為 `/etc/samba/smb.conf`。這個檔案可以分成兩個邏輯部份。`[global]` 區段包含中央與全域設定值。`[share]` 區段包含個別檔案與印表機共用。利用此方法，就可以在 `[global]` 區段中以不同的方式或以全域方式設定關於共用的細節，它可以加強組態檔的結構透明化。

48.1.1 全域區段

下列 `[global]` 區段的參數需要做一些調整，以符合網路設定的需求，讓其他機器可以透過 Windows 環境中的 SMB 存取 Samba 伺服器。

workgroup = TUX-NET

這一行是將 Samba 伺服器指定給工作群組。以網路環境中適當的工作群組取代 TUX-NET。除非您已將這個名稱指定給網路中的其他機器，否則 Samba 伺服器將會以其 DNS 名稱來顯示。如果沒有 DNS 名稱，請使用

`netbiosname=MYNAME` 設定伺服器名稱。如需關於此參數的詳細資訊，請參閱 `mansmb.conf`。

os level = 2

此參數會觸發 Samba 伺服器是否嘗試變成其工作群組的 LMB (本地主要的瀏覽器)。請選擇一個非常低的值，以使現有的 Windows 網路能避免設定不當的 Samba 伺服器所造成的任何問題。如需關於此重要主題的詳細資訊，請參閱套件文件中 `textdocs` 子目錄下的 `BROWSING.txt` 與 `BROWSING-Config.txt` 檔案。

如果沒有其他的 SMB 伺服器在網路中 (例如，Windows NT 或 2000 伺服器)，而且您想要 Samba 伺服器保留在本地環境中所存在的所有系統清單，請將 `os level` 設成更高的值 (例如，65)。接著就會將 Samba 伺服器選擇成本地網路的 LMB。

當變更此設定值時，請小心地考慮這個值將會如何影響現有的 Windows 網路環境。首先請在獨立的網路中或在一天中非重要的時間測試變更。

wins support 與 wins server

若要將 Samba 伺服器整合至含有主動 WINS 伺服器的現有 Windows 網路中，請啟用 `wins server` 選項，並將其值設為該 WINS 伺服器的 IP 位址。

如果您的 Windows 機器已連線至獨立的子網路中，而且應該要能夠看到彼此，則需要設定 WINS 伺服器。若要將 Samba 伺服器變成像這樣的 WINS 伺服器，請設定 `wins support = Yes` 選項。請確定網路中只有一個 Samba 伺服器啟用了這個設定值。`wins server` 與 `wins support` 選項絕不能在 `smb.conf` 檔案中同時啟用。

48.1.2 共用

下列範例說明如何將 CD-ROM 光碟機與使用者目錄 (`homes`) 開放給 SMB 用戶端使用。

[cdrom]

若要避免不小心將 CD-ROM 光碟機開放成共用，請以備註符號停用這些行 (在此例中為分號)。請在第一個資料欄中移除分號，以便和 Samba 共用 CD-ROM 光碟機。

範例 48.1 CD-ROM 共用

```
;[cdrom]
;comment = Linux CD-ROM
;path = /media/cdrom
;locking = No
```

[cdrom] and comment

`[cdrom]` 這個項目是在網路上所有的 SMB 用戶端都可以看到的共用名稱。可以另外再加入一個 `comment`，以進一步描述共用。

`path = /media/cdrom`

`path` 會輸出 `/media/cdrom` 目錄。

利用限制非常嚴格的預設組態，就可以將這種共用只開放給出現在此系統上的使用者共用。如果這個共用應該開放每個人使用，請將 `guest ok = yes` 加入組態。這個設定值可以將讀取權限開放給網路上的每個人使用。建議您處理此參數時必須極為小心。這將會在 `[global]` 區段中套用更多此參數的使用。

[homes]

[home] 共用在這裏特別重要。如果使用者擁有 Linux 檔案伺服器以及其自己主目錄的有效帳戶與密碼，就可以連線到主目錄。

範例 48.2 主目錄共用

```
[homes]
comment = Home Directories
valid users = %S
browseable = No
read only = No
create mask = 0640
directory mask = 0750
```

[homes]

只要沒有其他的共用，使用共用的使用者名稱連線至 SMB 伺服器，就會使用 [homes] 共用指示來動態產生共用。共用的產生名稱是使用者名稱。

valid users = %S

只要成功地建立連線，就會以共用的具體名稱取代 %S。[homes] 共用，永遠為使用者名稱。因此，使用者共用的存取權限僅限於使用者。

browseable = No

這個設定值讓共用在網路環境變成無形的。

read only = No

根據預設，Samba 會利用 read only = Yes 參數，以禁止寫入任何輸出共用的權限。若要開放共用為可寫入的，請設定 read only = No 的值，這與 writeable = Yes 同義。

create mask = 0640

以 MS Windows NT 為基礎的系統將無法理解 UNIX 權限的概念，因此在建立檔案時，它們將無法指定權限。create mask 參數可以定義指定給新建立檔案的存取權限。這只會套用至可寫入的共用。實際上，這個設定值表示擁有者具有讀取與寫入權限，而擁有者的主要群組成員則具有讀取權限。valid users = %S 可以在即使群組具有讀取權限時禁止讀取權限。若想要使群組具有讀取或寫入權限，請停用 valid users = %S 一行。

48.1.3 安全性層級

SMB 通訊協定是源自於 DOS 與 Windows 並直接考慮安全性的問題。每個共用存取權都以密碼保護。SMB 具有三個檢查權限的可能方法：

共用層級安全性 (安全性 = 共用):

每個共用都必須指定密碼。每個知道此密碼的人員都具有該共用的存取權。

使用者層級安全性 (安全性 = 使用者):

這個變化引用了使用者對 SMB 的概念。每個使用者都必須以自己的密碼註冊伺服器。在註冊後，伺服器可以視使用者名稱將存取權授與個別輸出的共用。

伺服器層級安全性 (安全性 = 伺服器):

對其用戶端而言，Samba 會模擬在使用者層級模式中工作。不過，它會將所有的密碼查詢傳遞給另一個將會處理驗證的使用者層級模式伺服器。這個設定值將需要另一個參數 (password server =)。

共用、使用者以及伺服器層級安全性之間的差異會套用至整個伺服器。因為無法針對伺服器組態的個別共用提供共用層級的安全性，並針對其他的共用提供使用者層級的安全性。然而，您可以針對系統上每個設定的 IP 位址執行個別的 Samba 伺服器。

在「Samba HOWTO 文件集」中可以找到關於此主題的詳細資訊。至於在一個系統上的多個伺服器，請注意 interfaces 與 bind interfaces only 選項。

提示

如需進行 Samba 伺服器的簡單管理工作，也可以使用 **swat** 程式。它提供簡單的網頁介面，以便於設定 Samba 伺服器。在網頁瀏覽器中，開啟 <http://localhost:901> 並以 root 的使用者身份登入。然而，必須同時在 `/etc/xinetd.d/samba` 與 `/etc/services` 檔案中啟動 **swat**。若要這麼做，請在 `/etc/xinetd.d/samba` 中，編輯 `disable` 一行，將它設為 `disable = no`。在 **man** 頁面中提供了關於 **swat** 的詳細資訊。

48.2 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。這可藉由 Samba 伺服器來完成。在以 Windows 為基礎的網路中，這個工作是由設定成主要領域控制器 (PDC) 的 Windows NT 伺服器來處理。在 `smb.conf` 的 `[global]` 區段中必須編輯的項目如 [範例 48.3, "在 smb.conf 中的全域區段"](#) [689] 所示。

範例 48.3 在 `smb.conf` 中的全域區段

```
[global]
    workgroup = TUX-NET
    domain logons = Yes
    domain master = Yes
```

如果使用加密的密碼來驗證 (這是維護良好的 MS Windows 9x 安裝、MS Windows NT 4.0 加上 Service Pack 3 以及所有之後產品的預設值)，Samba 伺服器必須能夠處理它們。在 `[global]` 區段中的 `encrypt passwords = yes` 項目可以啟用此功能 (在 Samba 版本 3，這已是預設值)。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領域帳戶 (Windows NT 領域概念所需)：

範例 48.4 設定機器帳戶

```
useradd hostname \
smbpasswd -a -m hostname
```

使用 `useradd` 命令，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`) 包含自動化此工作的設定值。

範例 48.5 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：

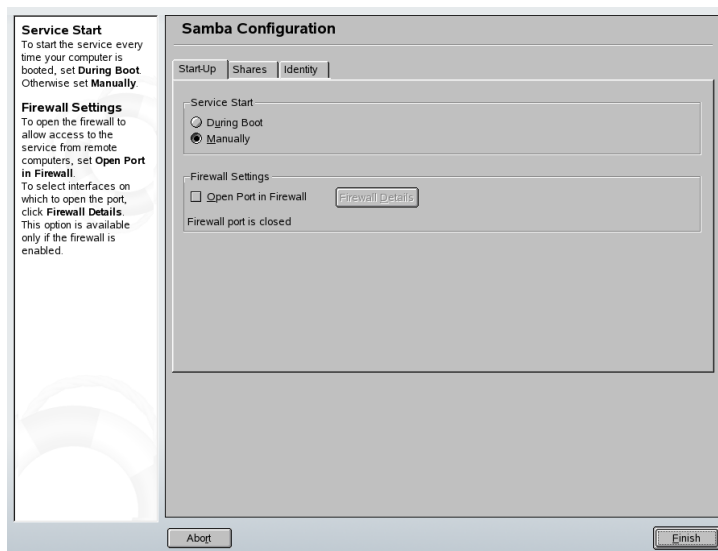
```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

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」。

48.3 使用 YaST 設定 Samba 伺服器

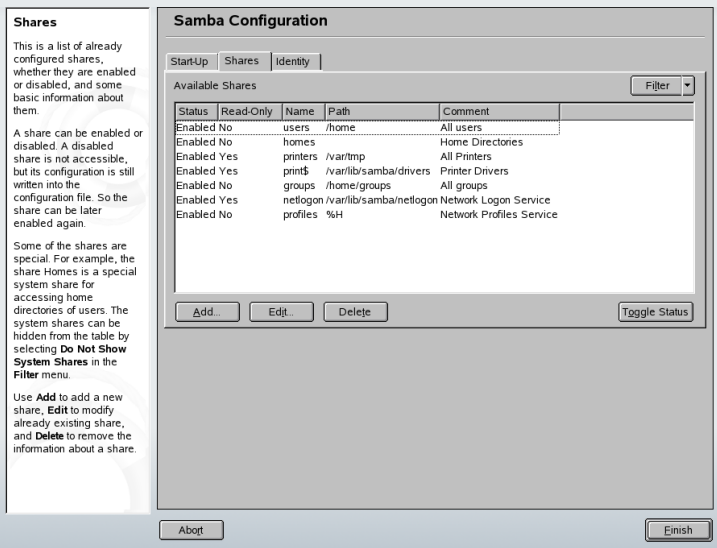
選取 Samba 伺服器要控制的工作群組或領域以啟動伺服器設定。從工作群組或領域名稱選取現有的，或輸入新的名稱。在下一個步驟中，指定伺服器應該做為 PDC (主要領域控制器) 或做為 BDC (備份領域控制器)。

圖形 48.1 Samba 設定--啟動



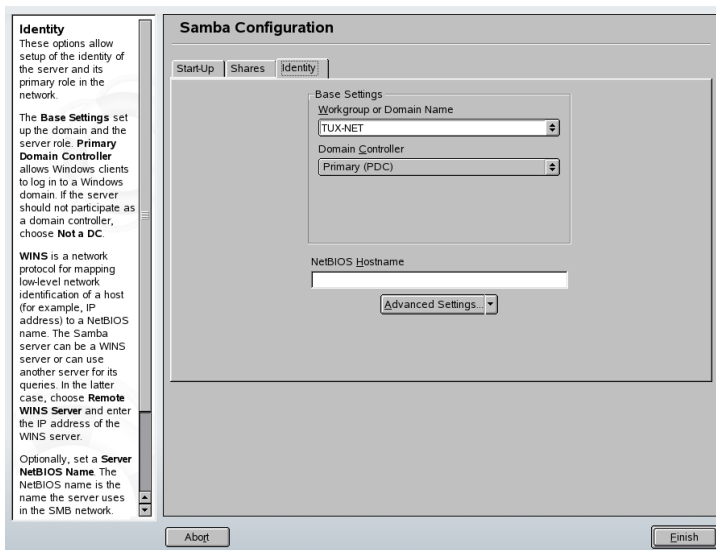
在 啟動啟動 Samba，如 圖形 48.1, "Samba 設定--啟動" [690] 所示。使用 在防火牆中開啟埠 以及 防火牆細節，以調整伺服器上的防火牆，使 `netbios-ns`、`netbios-dgm`、`netbios-ssn` 以及 `microsoft-ds` 服務的埠對所有外部與內部的介面開放，因而可確保 Samba 伺服器的作業能順利進行。

圖形 48.2 Samba 設定--共有



在 共用中 (圖形 48.2, "Samba 設定--共有" [691]), 決定要啟動的 Samba 共用。使用 切換狀態 以切換 作用中 與 非作用中。按一下 新增 以新增新的共用。

圖形 48.3 Samba 設定--識別



在 識別 中 (如 圖形 48.3, "Samba 設定--識別" [692] 所示), 決定主機要關聯的領域 (基礎設定) 以及是否要在網路中使用替代的主機 (*NetBIOS* 主機名稱)。

48.4 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。

48.4.1 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 *Samba* 工作群組對話方塊中輸入領域或工作群組。按一下 瀏覽 以顯示可以使用滑鼠選取的所有可用的群組與領域。如果您啟動 *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。

48.4.2 Windows 9x 與 ME

Windows 9x 與 ME 已經有 TCP/IP 的內建支援。不過，預設並不會安裝它。如果要新增 TCP/IP，請進入 *控制台* → *系統* 中，然後選擇 *新增* → *通訊協定* → *Microsoft 的 TCP/IP*。在重新啟動 Windows 機器後，按兩下網路環境中的桌面圖示以尋找 Samba 伺服器。

提示

如果要使用 Samba 伺服器上的印表機，請從對應的 Windows 版本安裝標準或 Apple-PostScript 印表機驅動程式。最好是將此與 Linux 印表機佇列連結，它可以接受 Postscript 做為輸入格式。

48.5 最佳化

`socket options` 是 Samba 版本隨附的範例組態中，所提供的其中一個可能的最佳化組態。其預設的組態會參考本地的乙太網路。如需關於 `socket options` 的其他資訊，請參閱 `smb.conf` 手冊頁的相關小節，以及 `socket(7)` 的手冊頁。在「Samba HOWTO 文件集」的 Samba 效能調整一章中，提供了進一步的資訊。

在 `/etc/samba/smb.conf` 中的標準組態，是用來提供以 Samba 團隊預設值為基礎的有用設定。不過您無法直接使用該組態，特別是在網路組態與工作群組名稱。加備註的範例組態 `examples/smb.conf`。SuSE 包含對本地需求調整相當有用的資訊。

提示

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。
