管理指南

SUSE Linux Enterprise Server 12 SP5

管理指南

SUSE Linux Enterprise Server 12 SP5

描述系統管理任務，如維護、監控及自訂初始安裝的系統。

出版日期： 2019 年 11 月 04 日

SUSE LLC
10 Canal Park Drive
Suite 200
Cambridge MA 02141
USA
https://www.suse.com/documentation

目錄

關於本指南

本指南的目標使用者為專業網路管理員和系統管理員，供其在操作 SUSE® Linux Enterprise 期間使用。因此，本指南的重點只在於確定 SUSE Linux Enterprise 的設定正確，而且網路所需服務都已備妥，讓它在初始安裝後即可正常運作。至於如何確定 SUSE Linux Enterprise 能夠與您企業的應用程式軟體正確相容，或其核心功能能否符合您的要求，則不在本指南的討論範圍。本指南假設已經進行了全面的要求稽核，並且已要求進行安裝，或者已要求進行此類稽核的測試安裝。

本指南包含下列內容：

支援與一般任務
: SUSE Linux Enterprise 提供眾多工具，可針對系統的各個方面進行自訂。這個部分介紹其中幾項。為了讓管理員對系統有初步的瞭解，本指南分析了可用的裝置技術、高度可用性的組態和進階的管理可能性。

系統
: 學習這部份，進一步瞭解作業系統基礎。SUSE Linux Enterprise 支援多種硬體架構，您可以藉此自行打造於 SUSE Linux Enterprise 上執行的應用程式。開機載入程式和開機程序資訊會協助您瞭解 Linux 系統的運作方式，以及您個人的自訂程序檔和應用程式如何與作業系統融合。

服務
: SUSE Linux Enterprise 的設計目的是要做為網路作業系統。它提供各種各樣的網路服務，例如 DNS、DHCP、Web、代理和驗證服務。它還可完全整合到異質環境中，其中包括 MS Windows 用戶端和伺服器。

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

疑難排解
: 概述了當您需要更多資訊或要執行特定任務時，如何尋找#明和其他文件。還提供了最常見的問題及其解決方法的清單。

1 可用文件

注意：線上文件和最新更新

我們的產品文件可從 http://www.suse.com/documentation/ 獲取，您也可以在此處找到最新更新，以及瀏覽或下載各種格式的文件。

此外，您安裝的系統的 `/usr/share/doc/manual` 下通常會提供產品文件。

針對本產品提供的文件如下：

《安裝快速入門》文章

列出系統要求，並提供透過 DVD 或 ISO 影像安裝 SUSE Linux Enterprise Server 的分步指南。

《部署指南》

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

管理指南

描述系統管理任務，如維護、監控及自訂初始安裝的系統。

《Virtualization Guide》

概述虛擬化技術，並介紹虛擬化的整合式介面 libvirt，以及關於特定監管程式的詳細資訊。

《儲存管理指南》

提供關於如何管理 SUSE Linux Enterprise Server 上儲存裝置的資訊。

《AutoYaST》

AutoYaST 系統會使用包含安裝和組態資料的 AutoYaST 設定檔，讓您以無人管理的方式大量部署 SUSE Linux Enterprise Server 系統。該手冊會指引您完成自動安裝的基本步驟：準備、安裝及設定。

《Security Guide》

介紹系統安全性的基本概念，涵蓋了本地安全性與網路安全性方面。說明如何使用產品固有的安全軟體（例如 AppArmor），或者能夠可靠收集關於任何安全相關事件之資訊的稽核系統。

《Security and Hardening Guide》

說明安裝和設定安全 SUSE Linux Enterprise Server 的詳情，以及進一步保護和強化該安裝所需的其他安裝後程序。為管理員提供安全性相關的選項和決策。

《System Analysis and Tuning Guide》

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

《Subscription Management Tool for SLES 12 SP5》

訂閱管理工具的管理員指南 - SUSE Customer Center 的代理系統，其中包含儲存庫與註冊目標。瞭解如何安裝與設定本地 SMT 伺服器、鏡像複製與管理儲存庫、管理用戶端機器，以及設定用戶端以使用 SMT。

《GNOME 使用者指南》

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

2 意見反應

以下為可供使用的數種意見回應管道：

錯誤與增強功能要求

有關適用於產品的服務與支援選項，請參閱 http://www.suse.com/support/。openSUSE 的說明由社群提供。如需相關資訊，請參閱 https://en.opensuse.org/Portal:Support。

若要報告產品元件的錯誤，請造訪 https://scc.suse.com/support/requests 並登入，然後按一下新建。

使用者意見

我們希望得到您對本手冊以及本產品隨附之其他文件的意見和建議。請使用線上文件每頁底部的「使用者備註」功能，或造訪 http://www.suse.com/documentation/feedback.html 在其中輸入您的意見。

郵件

如需本產品文件的回饋，您還可以傳送郵件至 `doc-team@suse.com`。請務必包含文件標題、產品版本以及文件發行日期。若要報告錯誤或對增強功能提出建議，請提供問題的簡短描述，並指出對應的章節編號及頁碼（或 URL）。

3 文件慣例

本文件中使用以下注意事項與排版慣例：

- `/etc/passwd`：目錄名稱與檔案名稱
- 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、選項和參數
- `user`：使用者或群組
- **套件名稱** ：套件的名稱
- `Alt`、`Alt`—`F1`：供人按下的按鍵或案件組合；顯示的按鍵與鍵盤上一樣為大寫
- 檔案、 檔案 › 另存新檔 ：功能表項目、按鈕
- `x86_64` 本段內容僅與 AMD64/Intel 64 架構相關。箭頭標示了文字區塊的開頭與結尾。
  `System z, POWER` 本段內容僅與 `z Systems` 和 `POWER` 架構相關。箭頭標示了文字區塊的開頭與結尾。
- Dancing Penguins（「Penguins」一章，↑另一本手冊）：這是對另一本手冊中某一章的參考。
- 必須具有**根**權限才能執行的指令。通常，您也可以在這些指令前加上 `sudo` 指令，以非特權使用者身分來執行它們。

```
root # command
tux > sudo command
```

- 沒有權限的使用者也可以執行的指令。

```
tux > command
```

- 注意事項

警告：警告

繼續操作之前必須瞭解的重要資訊。提醒您注意安全問題、可能的資料遺失、硬體損毀或者實際危險。

重要：重要說明

繼續操作之前應該瞭解的重要資訊。

注意：備註

其他資訊，例如各軟體版本之間的區別。

提示：提示

有用的資訊，例如一條準則或實用的建議。

4 關於本文件的編寫

本文件是採用 SUSEDoc（DocBook 5（http://www.docbook.org）的子集合）所編寫。其中 XML 來源檔案已由 `jing`（參閱 https://code.google.com/p/jing-trang/）驗證，經由 `xsltproc` 處理，而且採用 Norman Walsh 樣式表的自訂版本轉換成 XSL-FO。完稿的 PDF 是使用 Apache Software Foundation（https://xmlgraphics.apache.org/fop）的 FOP 進行格式化所產生。用於製作本文件的開放原始碼工具和環境由 DocBook Authoring and Publishing Suite（DAPS）提供。專案的首頁可以在 https://github.com/openSUSE/daps 中找到。

本文件的 XML 原始碼可以在 https://github.com/SUSE/doc-sle 中找到。

I 一般工作

1 Bash 和 Bash 程序檔

當今時代，許多人都在使用裝有 GNOME 之類圖形使用者介面（GUI）的電腦。儘管這些介面提供了很多功能，但使用它們執行自動化任務時，還是會有限制。外圍程序是 GUI 的有效補充，本章概述了外圍程序（以 Bash 為例）的一些方面。

1.1 什麼是「外圍程序」？

一般而言，外圍程序就是 Bash（Bourne again Shell）。本章中提及的「外圍程序」指的是 Bash。實際上，可用的外圍程序不止 Bash（還有 ash、csh、ksh、zsh 等等），每個外圍程序都具有不同的功能和特性。如需有關其他外圍程序的詳細資訊，請在 YaST 中搜尋外圍程序。

1.1.1 瞭解 Bash 組態檔案

外圍程序可啓用為：

1. 互動式登入外圍程序：　使用 `--login` 選項啓用 Bash 以登入機器，或使用 SSH 登入遠端機器時會採用這種方式。
2. 「一般」互動式外圍程序：　啓動 xterm、konsole、gnome-terminal 或類似工具時通常會採用這種方式。
3. 非互動式外圍程序：　在指令行中呼叫外圍程序程序檔時會採用這種方式。

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

表格 1.1 登入外圍程序的 BASH 組態檔案

檔案	描述
`/etc/profile`	請勿修改此檔案，否則您的修改在下次更新時可能會被破坏！

檔案	描述
`/etc/profile.local`	擴充 `/etc/profile` 時使用此檔案
`/etc/profile.d/`	包含特定程式的系統級組態檔案
`~/.profile`	在此處插入登入外圍程序的使用者特定組態

請注意，登入外圍程序還會獲取表格 1.2 「非登入外圍程序的 Bash 組態檔案」中所列的組態檔案。

表格 1.2 非登入外圍程序的 BASH 組態檔案

`/etc/bash.bashrc`	請勿修改此檔案，否則您的修改在下次更新時可能會被破坏！
`/etc/bash.bashrc.local`	使用此檔案只能插入 Bash 的系統級修改
`~/.bashrc`	在此處插入使用者特定的組態

此外，Bash 還使用其他檔案：

表格 1.3 BASH 的特殊檔案

檔案	描述
`~/.bash_history`	包含您鍵入的所有指令清單
`~/.bash_logout`	登出時執行
`~/.alias`	使用者為常用指令定義的別名。如需如何定義別名的詳細資料，請參閱 `man 1 alias`。

1.1.2 目錄結構

下表概述了 Linux 系統中最重要的較高層級目錄。下列清單中提供了關於目錄與重要子目錄的更多詳細資訊。

表格 1.4 標準目錄網路樹的綜覽

目錄	內容
`/`	根目錄 — 目錄樹狀結構的起點。
`/bin`	基本的二進位檔案，例如系統管理員與一般使用者都需要使用的指令。通常還包含 Bash 等外圍程序。
`/boot`	開機載入程式的靜態檔案。
`/dev`	存取主機特定裝置所需的檔案。
`/etc`	主機特定系統的組態檔案。
`/home`	存放系統中所有擁有帳戶之使用者的主目錄。但是，`root` 的主目錄不在 `/home` 中，而是位於 `/root` 內。
`/lib`	基本的共享程式庫與核心模組。
`/media`	抽取式媒體的定點。
`/mnt`	用於暫時掛接檔案系統的定點。
`/opt`	附加應用程式軟體套件。
`/root`	超級使用者 `root` 的主目錄。
`/sbin`	基本的系統二進位檔案。
`/srv`	系統所提供之服務的資料。
`/tmp`	暫存檔案。
`/usr`	包含唯讀資料的次要階層。
`/var`	可變資料，例如記錄檔案。
`/windows`	僅當系統中同時安裝了 Microsoft Windows* 與 Linux 才可以使用。包含 Windows 資料。

以下清單提供了更多詳細資訊，以及目錄中包含的檔案與子目錄的一些範例：

`/bin`

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

`/boot`

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

`/dev`

存放代表硬體元件的裝置檔案。

`/etc`

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

`/home/USERNAME`

存放系統中每個擁有帳戶之使用者的個人資料。只有檔案擁有者或系統管理員才能修改位於此處的檔案。依預設，您的電子郵件目錄與個人桌面組態以隱藏檔案與目錄的形式存放於此，例如 `.gconf/` 和 `.config`。

注意：網路環境中的主目錄

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

`/lib`

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

`/media`

包含 CD-ROM、隨身碟及數位相機（若使用 USB）等抽取式媒體的掛接點。`/media` 通常存放系統硬碟之外的任何類型磁碟機。抽取式媒體插入或連接到系統並進行掛接後，您就可以從此處存取該媒體。

`/mnt`

此目錄提供了暫時掛接之檔案系統的定點。`root` 可在此處掛接檔案系統。

`/opt`

為安裝協力廠商軟體而保留。這裡有選擇性軟體與大型附加程式套件。

`/root`

`root` 使用者的主目錄。此處存放 `root` 的個人資料。

`/run`

`systemd` 和各個元件使用的 tmpfs 目錄。`/var/run` 是 `/run` 的符號連結。

`/sbin`

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

`/srv`

存放系統所提供之服務的資料，例如 FTP 與 HTTP。

`/tmp`

需要檔案暫時儲存區的程式會使用此目錄。

重要：開機時清理 /tmp

無法保證在系統重新開機後，先前儲存於 `/tmp` 中的資料仍然存在。這視情況而定，例如，取決於 `/etc/tmpfiles.d/tmp.conf` 中的設定。

`/usr`

`/usr` 與使用者無關，是 UNIX 系統資源（UNIX system resource）的縮寫。`/usr` 中的資料是靜態的唯讀資料，可依照**檔案系統階層標準**（Filesystem Hierarchy Standard，FHS）在不同的主機之間共用。此目錄包含所有應用程式（包括 GNOME 之類的圖形桌面），並且會在檔案系統中建立次要階層。`/usr` 存放了多個子目錄，例如 `/usr/bin`、`/usr/sbin`、`/usr/local` 以及 `/usr/share/doc`。

`/usr/bin`

包含一般情況下可存取的程式。

`/usr/sbin`

包含為系統管理員保留的程式，例如修復功能。

`/usr/local`

在此目錄中，系統管理員可安裝獨立於套裝作業系統的本地延伸。

`/usr/share/doc`

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

在 `packages` 內，可找到系統上安裝之軟體套件所包含的文件。對於每個套件，都會建立一個子目錄 `/usr/share/doc/packages/PACKAGENAME`，通常用其儲存該套件的讀我檔案，有時儲存範例、組態檔案或其他程序檔。

如果系統中安裝了 HOWTO，`/usr/share/doc` 還會包含 `howto` 子目錄，其中有許多與 Linux 軟體設定及操作相關之任務的其他文件。

`/var`

`/usr` 存放的是靜態唯讀資料，而 `/var` 存放的是系統操作時寫入的資料，因此為可變資料，例如記錄檔案或多工緩衝處理資料。如需 `/var/log/` 中包含之最重要記錄檔的綜覽，請參閱表格 40.1 「記錄檔案」。

1.2 寫入外圍程序程序檔

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

範例 1.1 列印文字的外圍程序程序檔

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

❶ 第一行以 Shebang 字元（`#!`）開頭，指出此檔案為程序檔。程序檔透過 Shebang 後面指定的解譯器執行，在此例中為 `/bin/sh`。

❷ 第二行為備註，以 # 開頭。建議將較為複雜的行設為備註，以便記住其作用。

❸ 第三行使用內建指令 `echo` 列印相應的文字。

您需要符合一些先決條件才能執行此程序檔：

1. 每個程序檔都應包含 Shebang 行（如上面的範例所示）。如果缺少該行，您需要手動呼叫直譯器。
2. 您可以將程序檔儲存於任何位置。但是，最好將其儲存於外圍程序可以找到的目錄中。外圍程序中的搜尋路徑由環境變數 `PATH` 決定。一般使用者通常沒有寫入 `/usr/bin` 的權限。因此，建議將程序檔儲存在使用者目錄 `~/bin/` 中。以上範例名為 `hello.sh`。
3. 程序檔需要執行權限。使用下列指令設定權限：

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

如果滿足了上述所有先決條件，便可以按以下方式執行程序檔：

1. 絕對路徑：　執行程序檔時可以使用絕對路徑。在此例中為 `~/bin/hello.sh`。
2. 任何位置：　如果 `PATH` 環境變數包含程序檔所在的目錄，您可以使用 `hello.sh` 來執行程序檔。

1.3 重新指向指令事件

每條指令可以使用三個通道用於輸入或輸出：

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

要重新指向這些通道，可以使用以下幾種方式：

指令 > 檔案

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

```
ls > listing.txt
```

指令 >> 檔案

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

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

指令 < 檔案

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

```
read a < foo
```

指令1 | 指令2

將左邊指令的輸出重新指向為右邊指令的輸入。例如，`cat` 指令會輸出 `/proc/cpuinfo` 檔案的內容。再由 `grep` 使用此輸出內容單獨過濾出包含 `cpu` 的行：

```
cat /proc/cpuinfo | grep cpu
```

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

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

1.4 使用別名

別名為一或多條指令的簡短定義。別名的語法為：

```
alias NAME=DEFINITION
```

例如，下行定義了一個別名 `lt`，它會輸出一份較長的清單（選項 `-l`），將其依修改時間排序（`-t`），並依排好序的倒序列印（`-r`）：

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

要檢視所有的別名定義，請使用 `alias`。若要移除別名，請使用 `unalias` 和對應的別名名稱。

1.5 使用 Bash 中的變數

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

要檢視所有環境變數，請使用 `printenv` 指令。如需瞭解變數的值，則將變數名稱做為引數插入：

```
printenv PATH
```

無論是全域變數還是本地變數，都可以使用 `echo` 進行檢視：

```
echo $PATH
```

要設定本地變數，請使用變數名稱，後面跟上等號，再跟上值：

```
PROJECT="SLED"
```

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

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

若要移除變數，請使用 `unset`：

```
unset NAME
```

下表包含了部分可在外圍程序程序檔中使用的常用環境變數：

表格 1.5 有用的環境變數

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

1.5.1 使用引數變數

例如，如果您有程序檔 `foo.sh`，可以按以下格式執行該程序檔：

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

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

下面的程序檔 `foo.sh` 可列印從 1 到 4 的所有引數：

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

如果您使用以上引數執行此程序檔，所得結果為：

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

1.5.2 使用變數替代項

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

`${VAR#pattern}`

從左側移除最短的相符項：

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

`${VAR##pattern}`

從左側移除最長的相符項：

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

`${VAR%pattern}`

從右側移除最短的相符項：

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

```
/home/tux/book/book.tar
```

`${VAR%%pattern}`

從右側移除最長的相符項：

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

`${VAR/pattern_1/pattern_2}`

以 `PATTERN_2` 取代 `PATTERN_1` 中 `VAR` 的內容：

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

1.6 分組與結合指令

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

以下清單顯示了可對指令分組的方式：

指令1 ; 指令2

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

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

指令1 && 指令2

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

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

指令1 || 指令2

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

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

`funcname(){ ... }`

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

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

可以按以下格式呼叫此函數：

```
hello Tux
```

將會列印：

```
Hello Tux
```

1.7 使用通用流程建構元

為了控制程序檔的流程，外圍程序包含 `while`、`if`、`for` 及 `case` 建構元。

1.7.1 If 控制指令

`If` 指令用於檢查運算式。例如，以下程式碼將測試目前的使用者是否為 Tux：

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

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

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

測試運算式也可以縮寫到角括弧中：

```
if [ -e /tmp/foo.txt ] ; then
```

```
  echo "Found foo.txt"
fi
```

如需更多有用的運算式，請參閱 http://www.cyberciti.biz/nixcraft/linux/docs/uniqlinuxfeatures/lsst/ch03sec02.html 。

1.7.2 使用 `for` 指令建立迴路

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

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

1.8 更多資訊

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

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

2 sudo

許多指令與系統公用程式都需要以 `root` 身分執行，才能修改檔案並/或執行只有進階使用者才能執行的任務。出於安全考量並避免在無意間執行可能造成嚴重後果的指令，一般情況下建議不要以 `root` 身分登入，而是以無特權的一般使用者身分進行操作，並使用 `sudo` 指令以執行需要更高權限的指令。

在 SUSE Linux Enterprise Server 上，`sudo` 預設設定為與 su 的運作方式類似。不過，`sudo` 允許使用者以可靈活設定的方式使用其他任何使用者擁有的權限執行指令。此舉可以將具有特定權限的角色指定給某些使用者和群組。舉例而言，可以允許 `users` 群組的成員以 `wilber` 擁有的權限執行指令。而對指令的存取還可以進一步予以限制，例如禁止指定任何指令選項。su 始終要求必須提供 `root` 密碼才能使用 PAM 進行驗證，但 `sudo` 可以設定為使用您自己的身分證明進行驗證。此舉無需透露 `root` 密碼，因此更加安全。例如，您可以允許 `users` 群組的成員以 `wilber` 身分執行 `frobnicate` 指令，但禁止其指定任何引數。這樣可以將具有特定能力的角色指定給某些使用者和群組。

2.1 `sudo` 的基本使用方法

`sudo` 簡單易用，但能力不凡。

2.1.1 執行單一指令

以普通使用者身分登入後，您可以在指令前加上 `sudo` 以 `root` 身分執行任何指令。系統會提示使用者輸入 root 密碼，一旦驗證成功，便會以 `root` 身分執行指令：

```
tux > id -un❶
tux
tux > sudo id -un
root's password:❷
root
tux > id -un
tux❸
tux > sudo id -un
❹
root
```

1. `id -un` 指令會列印目前使用者的登入名稱。
2. 輸入期間，密碼不會顯示（既不顯示為純文字，也不顯示為項目符號）。
3. 只有以 `sudo` 開頭的指令會以更高權限執行。如果在不使用字首 `sudo` 的情況下執行同樣的指令，系統會繼續以目前使用者的權限執行。
4. 在限定時間內，您無需再次輸入 `root` 密碼。

提示：I/O 重新導向

I/O 重新導向的工作方式與您的預期可能會有所不同：

```
tux >  sudo echo s > /proc/sysrq-trigger
bash: /proc/sysrq-trigger: Permission denied
tux >  sudo cat < /proc/1/maps
bash: /proc/1/maps: Permission denied
```

只有 `echo`/`cat` 二進位值會以更高權限執行，重新導向會由使用者的外圍程序以使用者自己的權限來執行。若要啟動外圍程序，可依第 2.1.2 節 「啟動外圍程序」中所述的方法，或使用 `dd` 公用程式：

```
echo s | sudo dd of=/proc/sysrq-trigger
sudo dd if=/proc/1/maps | cat
```

2.1.2 啟動外圍程序

必須在每個指令前加上 `sudo` 可能很繁瑣。雖然您可以指定一個外圍程序做為 `sudo bash` 指令，但還是建議使用下列其中一種內建機制來啟動外圍程序：

`sudo -s (<指令>)`

啟動由 `SHELL` 環境變數所指定的外圍程序或目標使用者的預設外圍程序。如果提供了指令，此指令便會傳遞給該外圍程序（使用 `-c` 選項）；否則，外圍程序會以互動模式執行。

```
tux:~ > sudo -i
root's password:
root:/home/tux # exit
tux:~ >
```

`sudo -i (<指令>)`

與 `-s` 類似，但會將該外圍程序啓動為登入外圍程序。這表示系統會處理該外圍程式的啓動檔案（`.profile` 等），並將目前的工作目錄設定為目標使用者的主目錄。

```
tux:~ > sudo -i
root's password:
root:~ # exit
tux:~ >
```

2.1.3 環境變數

`sudo` 依預設不會傳播環境變數：

```
tux > ENVVAR=test env | grep ENVVAR
ENVVAR=test
tux > ENVVAR=test sudo env | grep ENVVAR
root's password:
❶
tux >
```

❶ 輸出為空即說明在使用 `sudo` 執行的指令的網路位置中不存在環境變數 `ENVVAR`。使用 `env_reset` 選項會導致行為出現變更，請參閱表格 2.1 「有用的旗標和選項」。

2.2 設定 sudo

`sudo` 是一項非常靈活的工具，提供大量組態。

注意：因鎖定而無法使用 sudo

如果您不小心將自己鎖定在 `sudo` 之外，則可以使用 `su -` 及 `root` 密碼來獲得 root 外圍程序。若要修復該錯誤，請執行 `visudo`。

2.2.1 編輯組態檔案

`sudo` 的主要規則組態檔案為 `/etc/sudoers`。如果此檔案中存在錯誤，您可能便無法進入系統，因此強烈建議您使用 `visudo` 來編輯設定檔。此舉可防止同時變更所開啓的檔案，並會在儲存修改之前檢查語法錯誤。

您還可以透過設定 `EDITOR` 環境變數以使用除 `vi` 以外的編輯器（無論名稱為何），例如：

```
sudo EDITOR=/usr/bin/nano visudo
```

不過，`/etc/sudoers` 檔案本身由系統套件所提供，所做的修改可能會在更新時遭到破壞。因此，建議將自訂組態存入 `/etc/sudoers.d/` 目錄下的檔案中。該目錄下的所有檔案都會自動納入。若要在該子目錄下建立或編輯檔案，請執行：

```
sudo visudo -f /etc/sudoers.d/NAME
```

或使用其他編輯器（如 `nano`）：

```
sudo EDITOR=/usr/bin/nano visudo -f /etc/sudoers.d/NAME
```

注意：/etc/sudoers.d 中忽略的檔案

`/etc/sudoers` 中的 `#includedir` 指令（用於 `/etc/sudoers.d`）會忽略以 `~`（波狀符號）結尾或包含 `.`（點）.

如需 `visudo` 指令的詳細資訊，請執行 `man 8 visudo`。

2.2.2 sudoers 組態的基本語法

sudoers 組態檔案中有字串和旗標兩種選項。其中，字串可以包含任何值，旗標可以開啓或關閉。sudoers 組態檔案最重要的語法建構為：

```
# Everything on a line after a # gets ignored ❶
Defaults !insults # Disable the insults flag ❷
Defaults env_keep += "DISPLAY HOME" # Add DISPLAY and HOME to env_keep
tux ALL = NOPASSWD: /usr/bin/frobnicate, PASSWD: /usr/bin/journalctl ❸
```

❶ `#include` 和 `#includedir` 這兩個一般指令例外。其後是數字，用於指定 UID。

❷ 移除 `!` 可將指定的旗標設定為開啓。

❸ 請參閱第 2.2.3 節 「sudoers 中的規則」。

表格 2.1 有用的旗標和選項

選項名稱	描述	範例
`targetpw`	此旗標控制呼叫使用者是否需要輸入密碼。如果要求輸入目標使用者（例如 `root`）的密碼則為開啓，如果要求輸入其自己的密碼則為關閉。	`Defaults targetpw # Turn targetpw flag ON`
`rootpw`	如果設定了此選項，`sudo` 會提示使用者輸入 `root` 密碼，而非目標使用者或呼叫使用者的密碼。預設為關閉。	`Defaults !rootpw # Turn rootpw flag OFF`
`env_reset`	如果設定了此選項，`sudo` 會建構一個僅含 `TERM`、`PATH`、`HOME`、`MAIL`、`SHELL`、`LOGNAME`、`USER`、`USERNAME` 和 `SUDO_*` 集的精簡環境。此外，`env_keep` 中所列的變數會從呼叫環境輸入。預設為開啓。	`Defaults env_reset # Turn env_reset flag ON`

`env_keep`	當 `env_reset` 旗標為開啓時要保留的一系列環境變數。	`# Set env_keep to contain EDITOR and PROMPT`
		`Defaults env_keep = "EDITOR PROMPT"`
		`Defaults env_keep += "JRE_HOME" # Add JRE_HOME`
		`Defaults env_keep -= "JRE_HOME" # Remove JRE_HOME`

選項名稱	描述	範例

`env_delete`	當 `env_reset` 旗標為關閉時要移除的一系列環境變數。	`# Set env_delete to contain EDITOR and PROMPT`
		`Defaults env_delete = "EDITOR PROMPT"`
		`Defaults env_delete += "JRE_HOME" # Add JRE_HOME`
		`Defaults env_delete -= "JRE_HOME" # Remove JRE_HOME`

`Defaults` 記號還可用於建立使用者、主機和指令三者之集合的別名。除此之外，還可以將某選項僅套用到一組特定的使用者。

如需 `/etc/sudoers` 組態檔案的詳細資訊，請諮詢 `man 5 sudoers`。

2.2.3 sudoers 中的規則

sudoers 組態中的規則可能會相當複雜，本節僅介紹基本內容。每個規則後面都跟著基本規劃（`[]` 表示可選部份）：

```
#Who      Where         As whom      Tag               What
User_List Host_List = [(User_List)] [NOPASSWD:|PASSWD:] Cmnd_List
```

SUDOERS 規則的語法

`User_List`

一或多個識別碼（以 `,` 分隔）：使用者名稱、以 `%GROUPNAME` 格式表示的群組，或是以 `#UID` 格式表示的使用者 ID。否定運算可以使用 `!` 字首。

`Host_List`

一或多個識別碼（以 `,` 分隔）：完全合格的主機名稱或 IP 位址。否定運算可以使用 `!` 字首。`ALL` 是 `Host_List` 的一般選項。

`NOPASSWD:|PASSWD:`

如果使用者在 `NOPASSWD:` 之後執行的指令與 `CMDSPEC` 相符，系統不會提示使用者輸入密碼。

`PASSWD` 為預設值，只有當其與 NOPASSWD 出現在同一行時才需要指定：

```
tux ALL = PASSWD: /usr/bin/foo, NOPASSWD: /usr/bin/bar
```

`Cmnd_List`

一或多個規範（以 `,` 分隔）：指向可執行程式的路徑，後跟允許的引數或不跟任何內容。

```
/usr/bin/foo     # Anything allowed
/usr/bin/foo bar # Only "/usr/bin/foo bar" allowed
/usr/bin/foo ""  # No arguments allowed
```

`ALL` 可以用作 `User_List`、`Host_List` 和 `Cmnd_List`。

允許 `tux` 在無需輸入密碼的情況下以 `root` 身分執行所有指令的規則：

```
tux ALL = NOPASSWD: ALL
```

允許 `tux` 執行 `systemctl restart apache2` 的規則：

```
tux ALL = /usr/bin/systemctl restart apache2
```

允許 `tux` 在不帶任何引數的情況下以 `admin` 身分執行 `wall` 的規則：

```
tux ALL = (admin) /usr/bin/wall ""
```

警告：可造成嚴重後果的建構

下列類型的建構

```
ALL ALL = ALL
```

在沒有 `Defaults targetpw` 的情況下切勿使用，否則任何人都能以 `root` 身分執行指令。

2.3 常見使用案例

雖然預設組態對於簡單設定和桌面環境往往已足夠，但自訂組態也非常有用。

2.3.1 在無需提供 root 密碼的情況下使用 sudo

在具有特殊限制（「使用者 X 只能以 root」 身分執行指令 Y）的情況下，這點無法實現。在其他情況下，還是建議進行某類分隔。依慣例，wheel 群組的成員可以以 root 身分執行所有帶有 sudo 的指令。

1. 將自己新增至 wheel 群組
   如果您的使用者帳戶尚不是 wheel 群組的成員，可執行 sudo usermod -a -G wheel 使用者名稱，然後登出並再次登入，將帳戶加入其中。執行 groups 使用者名稱驗證變更是否成功。

2. 將使用呼叫使用者的密碼進行驗證的選項設定為預設設定。
   使用 visudo 建立檔案 /etc/sudoers.d/userpw（請參閱第 2.2.1 節 「編輯組態檔案」）並新增：

   ```
   Defaults !targetpw
   ```

3. 選取新的預設規則。
   視您是否希望使用者重新輸入其密碼，在 /etc/sudoers 中取消備註特定的行，並將預設規則設定為備註。

   ```
   ## Uncomment to allow members of group wheel to execute any command
   # %wheel ALL=(ALL) ALL

   ## Same thing without a password
   # %wheel ALL=(ALL) NOPASSWD: ALL
   ```

4. 對預設規則設定更多限制
   移除 /etc/sudoers 中的 allow-everything 規則或設定為備註：

   ```
   ALL     ALL=(ALL) ALL   # WARNING! Only use this together with 'Defaults targetpw'!
   ```

   警告：sudoers 中可造成嚴重後果的規則

   切勿遺忘此步驟，否則任何使用者都能以 root 身分執行任何指令！

5. 測試組態
   嘗試以 wheel 成員身分和非成員身分執行 sudo。

   ```
   tux:~ > groups
   ```

```
users wheel
tux:~ > sudo id -un
tux's password:
root
wilber:~ > groups
users
wilber:~ > sudo id -un
wilber is not in the sudoers file.  This incident will be reported.
```

2.3.2 對 X.Org 應用程式使用 sudo

使用 `sudo` 啓動圖形應用程式時，會遇到以下錯誤：

```
tux > sudo xterm
xterm: Xt error: Can't open display: %s
xterm: DISPLAY is not set
```

YaST 將選擇 ncurses 介面而非圖形介面。

若要在 `sudo` 啓動的應用程式中使用 X.Org，需要傳播環境變數 `DISPLAY` 和 `XAUTHORITY`。若要對此進行設定，請建立檔案 `/etc/sudoers.d/xorg`（請參閱第 2.2.1 節 「編輯組態檔案」）並新增下列一行：

```
Defaults env_keep += "DISPLAY XAUTHORITY"
```

若尚未設定 `XAUTHORITY` 變數，請依如下方式設定：

```
export XAUTHORITY=~/.Xauthority
```

現在，X.Org 應用程式便可正常執行：

```
sudo yast2
```

2.4 更多資訊

使用 `sudo --help` 可取得有關可用的指令行參數的簡要綜覽。說明和其他重要資訊可參閱 man 頁面 `man 8 sudo`，組態則記錄在 `man 5 sudoers` 中。

3 YaST 線上更新

SUSE 為您的產品提供持續的軟體安全性更新。依預設，更新 Applet 可以讓您的系統保持最新狀態。如需有關更新 Applet 的詳細資訊，請參閱《部署指南》，第 13 章「安裝或移除軟體」，第 13.5 節「使系統保持最新」。本章介紹了用於更新軟體套件的替代工具：YaST 線上更新。

可從更新軟體儲存庫中取得適用於 SUSE® Linux Enterprise Server 的最新修補程式。如果在安裝期間已註冊了產品，則表明已設定更新儲存庫。如果您尚未註冊 SUSE Linux Enterprise Server，可在 YaST 中啓動產品註冊。或者，您可以從信任的來源手動新增更新儲存庫。若要新增或移除儲存庫，請透過 YaST 中的軟體 › 軟體儲存庫啓動儲存庫管理員。《部署指南》，第 13 章「安裝或移除軟體」，第 13.4 節「管理軟體儲存庫與服務」 中提供了有關儲存庫管理員的詳細資訊。

注意：存取更新目錄時發生錯誤

如果您無法存取更新目錄，原因可能是訂閱已過期。SUSE Linux Enterprise Server 通常提供一年或三年的訂閱，您可以在這個時間段內存取更新目錄。訂閱期結束後，對更新目錄的存取將被拒絕。

如果存取更新目錄時遭到拒絕，系統將顯示一則警告訊息，提示您造訪 SUSE Customer Center 以查看您的訂閱。SUSE Customer Center 的網址為 https://scc.suse.com//。

SUSE 提供了具有不同關聯層級的更新：

安全性更新
: 修復嚴重的安全性問題，必須予以安裝。

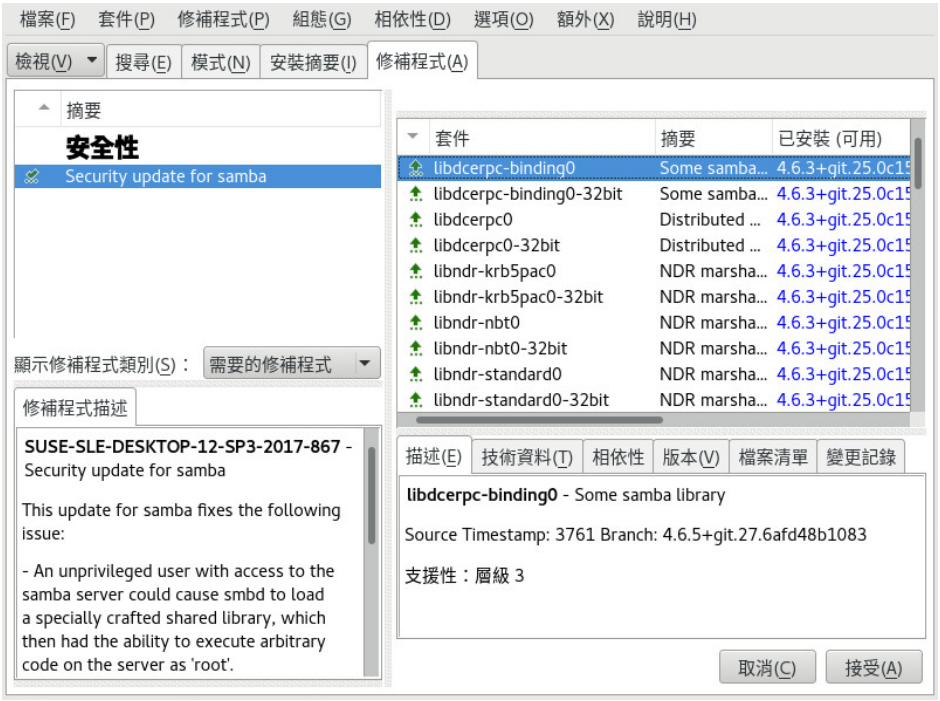
推薦性更新
: 修復可能會破壞電腦的問題。

選擇性更新
: 修復與安全性無關的問題或提供增強功能。

3.1 線上更新對話方塊

若要開啓 YaST 線上更新對話方塊，請啓動 YaST 並選取軟體 › 線上更新。或者，也可以在指令行中使用 `yast2 online_update` 指令進行啓動。

線上更新視窗包含四個區段。



圖形 3.1 YAST 線上更新

左側的摘要區段列出了 SUSE Linux Enterprise Server 可用的修補程式。修補程式根據安全性關聯程度（**安全性**、**建議**和**選擇性**）進行排序。您可以從顯示修補程式類別中選取以下其中一個選項，以變更摘要區段的檢視窗：

需要的修補程式（預設檢視窗）
: 適用於系統上安裝的套件，但尚未安裝的修補程式。

不需要的修補程式
: 適用於系統上未安裝之套件的修補程式，或已滿足其執行要求的修補程式（相關套件已從其他來源進行更新）。

所有修補程式
: SUSE Linux Enterprise Server 的所有可用修補程式。

摘要區段中的每個清單項目包含一個符號和修補程式名稱。有關可用符號及其含義的綜覽，請按 Shift—F1 。「**安全性**」和「**建議**」狀態的修補程式所要求的動作已自動預設。這些動作包括自動安裝、自動更新和自動刪除。

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

在摘要區段中選取一個項目，可在對話方塊的左下角看到簡短的修補程式描述。右上方的區段列出了所選修補程式中包含的套件（一個修補程式可以包含多個套件）。按一下右上方區段中的項目，可檢視修補程式中包含的各套件的詳細資料。

3.2 安裝修補程式

在 YaST 的「連接更新」對話方塊中，您可以一次性安裝所有可用的修補程式，也可以手動選取所需的修補程式。您還可以回復已套用至系統的修補程式。

依預設，目前您系統可用的所有新修補程式（**選擇性**修補程式除外）均已標示為可供安裝。按一下接受或套用後，這些修補程式將自動套用。如果一或多個修補程式需要將系統重新開機，則在開始安裝修補程式之前，系統會發出相關通知。此時，您可以選擇繼續安裝所選修補程式、跳過需要重新開機之所有修補程式的安裝並安裝剩餘的修補程式，或者返回修補程式手動選擇畫面。

程序 3.1 使用 YAST 線上更新套用修補程式

1. 啓動 YaST，然後選取軟體 › 線上更新。
2. 若要自動套用您系統目前可用的所有新修補程式（**選擇性**修補程式除外），請按套用或接受。
3. 首先修改要套用的修補程式選擇：
   a. 使用介面提供的相應過濾器和檢視窗。如需詳細資訊，請參閱第 3.1 節「線上更新對話方塊」。
   b. 根據您的需求和意願選取或取消選取修補程式，方法是在修補程式上按一下滑鼠右鍵，然後從內容功能表中選擇相應的動作。

重要：始終套用安全性更新

除非很有必要，否則請不要取消選取任何**安全性**相關的修補程式。因為這些修補程式負責修復嚴重的安全性問題，可防止您的系統被入侵。

c. 大部分的修補程式都會包含多套件的更新。若要變更對單一套件所執行的動作，請在套件檢視窗中的套件上按一下滑鼠右鍵，然後選擇一個動作.

d. 若要確認您的選擇並套用所選修補程式，請按一下套用或接受繼續。

4. 安裝完成後，請按一下完成離開 YaST線上更新。您的系統現在已是最新狀態。

3.3 自動線上更新

YaST 還提供有設定每日、每週或每月執行自動更新的選項。若要使用相應的模組，您需要先安裝 `yast2-online-update-configuration` 套件。

依預設更新將下載為增量 RPM。由於透過增量 RPM 重建 RPM 套件需要佔用大量記憶體和處理器資源，出於效能考量，某些設定或硬體組態可能要求您停用增量 RPM。

某些修補程式（例如核心更新或需要授權合約的套件）需要使用者互動，而這可能會讓自動更新程序停止下來。您可以設定為跳過需要使用者互動的修補程式。

程序 3.2 設定自動線上更新

1. 安裝後，啓動 YaST 並選取軟體 › 線上更新組態。
   您也可以在指令行中使用 `yast2 online_update_configuration` 指令啓動模組。

2. 啓用自動線上更新。

3. 選擇更新間隔：每日、每週或每月。

4. 若要自動接受所有授權合約，請啓用同意授權。

5. 如果您要更新程序繼續以完全自動化的方式執行，可以選取跳過互動式修補程式。

重要：跳過修補程式

如果您選擇跳過需要互動的所有套件，請經常執行手動線上更新，以便這些修補程式也能予以安裝。否則，您可能會錯過部分重要的修補程式。

6. 若要自動安裝已更新套件推薦的所有套件，請啓用包含推薦的套件。
7. 若要停用增量 RPM（出於效能方面的考量），請停用使用增量 RPM。
8. 若要根據類別（例如安全性或推薦）過濾修補程式，請啓用根據類別過濾並新增清單中的相應修補程式類別。系統將會只安裝所選類別的修補程式，跳過其他類別。
9. 按一下確定確認您的組態。

自動線上更新隨後不會自動重新啓動系統。如果套件更新需要重新開機系統才能生效，則您需要手動重新開機。

4 YaST

YaST 是 SUSE Linux Enterprise Server 的安裝和組態工具。它具有圖形介面，並且能夠在安裝期間和安裝之後快速自訂您的系統。它可用於設定硬體、設定網路、系統服務和調整安全性設定。

4.1 進階按鍵組合

YaST 有一套進階按鍵組合。

Print Screen
: 擷取並儲存螢幕畫面。當 YaST 在某些桌面環境下執行時可能不可用。

Shift—F4
: 啓用/停用為視障使用者最佳化的配色方式。

Shift—F7
: 啓用/停用除記錄錯訊息功能。

Shift—F8
: 開啓檔案對話方塊以將記錄檔案儲存至某個非標準位置。

Ctrl—Shift—Alt—D
: 傳送一個除錯事件。YaST 模組可透過執行特殊除錯動作來做出回應。結果取決於具體的 YaST 模組。

Ctrl—Shift—Alt—M
: 啓動/停止巨集記錄程式。

Ctrl—Shift—Alt—P
: 重播巨集。

Ctrl—Shift—Alt—S
: 顯示樣式表編輯器。

Ctrl—Shift—Alt—T
: 將工具集樹傾印至記錄檔案。

Ctrl—Shift—Alt—X

開啓終端機視窗（xterm）。當透過 VNC 安裝時很實用。

Ctrl—Shift—Alt—Y

顯示工具集樹瀏覽器。

5 文字模式的 YaST

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

文字模式下的 YaST 使用 ncurses 程式庫來提供簡單的虛擬圖形使用者介面。依預設，ncurses 程式庫已安裝。若要執行 YaST，終端模擬器的大小不能小於 80x25 個字元。

圖形 5.1 文字模式中的 YAST 主視窗

在文字模式中啓動 YaST 時，會顯示 YaST 控制中心（請參閱圖形 5.1）。主要視窗包含 3 個區域。左框架內包含各種模組所屬的類別。此框架在 YaST 啓動時處於使用中狀態，因此會以白色的粗框線標示。使用中的類別處於選取狀態。右框架內包含作用中類別可以使用的各個模組的綜覽。下方框架中有說明按鈕與結束按鈕。

啓動 YaST 控制中心時，會自動選取軟體類別。您可以使用 ↓ 與 ↑ 來變更類別。若要從類別中選取模組，可以使用 → 啓動右框架，然後使用 ↓ 和 ↑ 選取模組。您可以按住方向鍵不放來捲動可用模組清單。選取的模組處於選定狀態。按 Enter 可以啓動作用中的模組。

模組中的各個按鈕或選項欄位中，都有一個反白顯示的字母（預設為黃色）。您可以使用 Alt —反白的字母直接選取按鈕，而不必再使用 →| 導覽至該處。按 Alt —Q 或選取結束並按 Enter 可離開 YaST 控制中心。

提示：重新整理 YaST 對話方塊

如果 YaST 對話方塊遭毀損或破壞（例如在調整視窗大小時），請按 Ctrl—L 重新整理並還原其內容。

5.1 在模組中瀏覽

以下對 YaST 模組中控制元素的描述假設所有的功能鍵及 Alt 按鍵組合都起作用，且未被指定不同的全域功能。如需有關可能的例外狀況的資訊，請參閱第 5.3 節 「組合鍵的限制」。

在按鈕與選項清單中瀏覽

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

在選項清單中瀏覽

在作用中且包含選項清單的框架中，您可以使用方向鍵（↑ 和 ↓），以便在其中的個別元素之間進行瀏覽。如果框架中個別項目超出其寬度，您可以使用 Shift—→ 或 Shift—← 以水平方式向右捲動或向左捲動。或者使用 Ctrl—E 或 Ctrl—A。如果使用 → 或 ← 導致作用中的框架或目前的選項清單發生變更（如同在控制中心內），您也可以使用此按鍵組合。

按鈕、圓形按鈕以及核取方塊

如果要選取有空白方括號（核取方塊）的按鈕，或是有空白括號（選項圓鈕）的按鈕，請按 Space 或 Enter。或者，也可以直接使用 Alt—反白的字母 選取選項圓鈕與核取方塊。在此狀況中，您不需按 Enter 來做確認。如果您使用 →| 瀏覽至某個項目，按 Enter 即可執行所選取的動作或啓用個別的功能表項目。

功能鍵

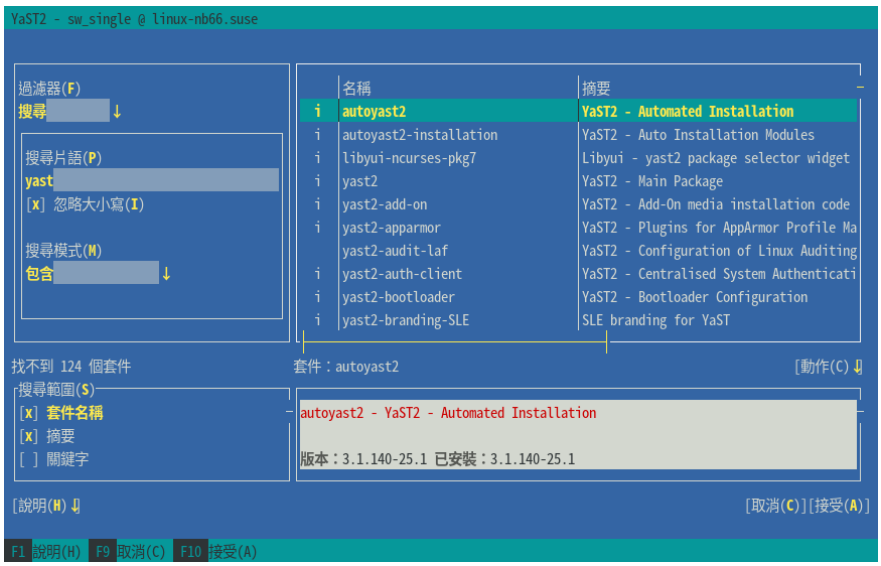
功能鍵（F1 ... F12）可讓您快速存取各種按鈕。YaST 螢幕底部的行中顯示了可用的功能鍵組合（FX）。因為不同的模組提供不同的按鈕（詳細資料、資訊、新增、刪除等等），所以各個功能鍵實際對應的按鈕視作用中的 YaST 模組而有所不同。您可以將 F10 當作接受、確定、下一步以及完成來使用。按 F1 可存取 YaST 說明。

在 ncurses 模式中使用導覽樹狀結構

某些 YaST 模組使用視窗左側的導覽樹狀結構來選取組態對話方塊。使用方向鍵（↑ 和 ↓）可以在樹狀結構中進行導覽。使用 Space 可以開啓或關閉樹狀結構中的項目。在 ncurses 模式中，於導覽樹狀結構中選取後必須按 Enter 才能顯示選取的對話方塊。這樣做的目的是為了在導覽樹狀結構時省去費時的重新描繪程序。

在軟體安裝模組中選取軟體

使用左側的過濾器可以限制顯示的套件數量。已安裝的套件標有字母 i。若要變更套件的狀態，請按 Space 或 Enter。或者，也可以使用動作功能表選取所需的狀態變更（安裝、刪除、更新、禁用或鎖定）。



圖形 5.2 軟體安裝模組

5.2 進階按鍵組合

文字模式的 YaST 具有一套進階按鍵組合。

Shift—F1

顯示進階快速鍵清單。

Shift—F4

變更色彩綱要。

Ctrl—\

結束應用程式。

Ctrl—L

重新整理螢幕。

Ctrl—D F1

顯示進階快速鍵清單。

Ctrl—D Shift—D

以螢幕擷取畫面的形式將對話方塊傾印到記錄檔案。

Ctrl—D Shift—Y

開啓 YDialogSpy 以查看工具樹階層。

5.3 組合鍵的限制

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

使用 Esc 取代 Alt

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

您可以使用 Ctrl—F 與 Ctrl—B 來往前瀏覽和往後瀏覽

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

功能鍵的限制

功能鍵（F1 ... F12）也用於執行多種功能。有些特定的功能鍵可能已由終端機所佔用，無法供 YaST 使用。不過，在純文字主控台中，應該都可以使用各種的 Alt 組合鍵與功能鍵。

5.4 YaST 指令行選項

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

```
yast -h
```

5.4.1 啓動個別模組

如果要節省時間，您可以直接啓動個別的 YaST 模組。若要啓動模組，請輸入：

```
yast <module_name>
```

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

5.4.2 透過指令行安裝套件

如果您知道套件名稱，並且此套件是由某個使用中的安裝儲存庫所提供，則可以使用指令行選項 `-i` 來安裝套件：

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

或

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

`PACKAGE_NAME` 可以是透過相依項檢查安裝的單個簡短套件名稱（例如 `gvim`），也可以是並非透過相依項檢查安裝的 RPM 套件的完整路徑。

如果您需要包含 YaST 無法提供的功能，以指令行為基礎的軟體管理公用程式，可以考慮使用 Zypper。此公用程式使用相同的軟體管理程式庫，此程式庫也是 YaST 套件管理員的基礎。第 6.1 節 「使用 Zypper」中介紹了 Zypper 的基本用法。

5.4.3 YaST 模組的指令行參數

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

```
yast <module_name> help
```

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

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

6 使用指令行工具管理軟體

本章介紹了兩個用於管理軟體的指令行工具：Zypper 與 RPM。有關在此內容中所使用辭彙的定義（例如，**儲存庫**、**修補程序**或**更新**），請參閱《部署指南》，第 13 章「安裝或移除軟體」，第 13.1 節「術語定義」。

6.1 使用 Zypper

Zypper 是用於安裝、更新和移除套件以及管理儲存庫的指令行套件管理員。這部分內容對於執行遠端軟體管理任務或透過外圍程序程序檔管理軟體非常有用。

6.1.1 一般使用情形

Zypper 的一般語法為：

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

括號中的部分為非必需。如需一般選項和所有指令的清單，請參閱 `zypper help`。若要取得特定指令的說明，請輸入 `zypper help` *指令*。

Zypper 指令

執行 Zypper 最簡單的方法就是輸入其名稱，後面跟著指令。例如，若要將所有需要的修補程式套用至系統，請使用：

```
tux > sudo zypper patch
```

全域選項

此外，您還可以選擇使用一或多個全域選項，只需在指令前面輸入它們即可：

```
tux > sudo zypper --non-interactive patch
```

在上述範例中，選項 `--non-interactive` 表示在不要求輸入任何內容的情況下執行指令（自動套用預設回答）。

特定於指令的選項

若要使用特定於特殊指令的選項，請緊接在指令後面輸入這些選項：

```
tux > sudo zypper patch --auto-agree-with-licenses
```

在上述範例中，`--auto-agree-with-licenses` 用於將所有需要的修補程式套用至系統，您不需要確認任何授權條款，而會自動接受授權條款。

引數

某些指令需要一或多個引數：例如，使用 `install` 指令時，需要指定您想要安裝的一或多個套件：

```
tux > sudo zypper install mplayer
```

某些選項還需要單個引數。以下指令會列出所有已知模式：

```
tux > zypper search -t pattern
```

您可以將上述所有指令組合使用。例如，以下指令將以詳細模式安裝 `aspell-de` 和 `aspell-fr` 套件（來自 `factory` 儲存庫）：

```
tux > sudo zypper -v install --from factory aspell-de aspell-fr
```

`--from` 選項可確保所有儲存庫均處於啓用狀態（以解決相依性問題），並且始終從指定的儲存庫要求套件。

大部分 Zypper 指令都有一個 `dry-run` 選項，對指定的指令進行模擬。它可以用於測試。

```
tux > sudo zypper remove --dry-run MozillaFirefox
```

Zypper 支援 `--userdata 字串` 全域選項。您可以使用此選項指定一個將會寫入 Zypper 的記錄檔案和外掛程式（例如 Btrfs 外掛程式）的字串。它可以用來標示和識別記錄檔案中的異動。

```
tux > sudo zypper --userdata STRING patch
```

6.1.2 使用 Zypper 安裝和移除軟體

若要安裝或移除套件，請使用以下指令：

```
tux > sudo zypper install PACKAGE_NAME
```

```
sudo zypper remove PACKAGE_NAME
```

警告：不要移除必要的系統套件

不要移除必要的系統套件，例如 `glibc` 、 `zypper` 、 `kernel` 。如果移除這些套件，系統可能會變得不穩定，甚至完全停止運作。

6.1.2.1 選取要安裝或移除的套件

可以使用 `zypper install` 和 `zypper remove` 指令，以多種方法來找到套件。

依確切的套件名稱

```
tux > sudo zypper install MozillaFirefox
```

依確切的套件名稱和版本號碼

```
tux > sudo zypper install MozillaFirefox-52.2
```

依儲存庫別名和套件名稱

```
tux > sudo zypper install mozilla:MozillaFirefox
```

`mozilla` 是要安裝套件所在之儲存庫的別名。

依含萬用字元的套件名稱

您可以選取名稱以特定字串開頭或結尾的所有套件。使用萬用字元時請務必小心，特別是在移除套件時。以下指令將安裝名稱以「Moz」開頭的所有套件：

```
tux > sudo zypper install 'Moz*'
```

提示：移除所有 -debuginfo 套件

在給問題除錯時，您有時需要暫時安裝大量 `-debuginfo` 套件，以取得關於執行中程序的詳細資訊。在完成除錯工作階段後，如果您要清理環境，請執行以下指令：

```
tux > sudo zypper remove '*-debuginfo'
```

依功能

例如，您要安裝一個 Perl 模組，但不知道套件的名稱，用功能指定便會幫上忙：

```
tux > sudo zypper install firefox
```

依功能、硬體架構或版本

可以結合功能指定硬體架構和版本：

- 所需硬體架構的名稱需附加在功能的後面，兩者以句點分隔。例如，若要指定 AMD64/Intel 64 架構（在 Zypper 中命名為 `x86_64`），請使用：

  ```
  tux > sudo zypper install 'firefox.x86_64'
  ```

- 版本必須附加在字串的末尾，並且前面必須使用運算子：`<`（小於）、`<=`（小於等於）、`=`（等於）、`>=`（大於等於）、`>`（大於）。

  ```
  tux > sudo zypper install 'firefox>=52.2'
  ```

- 也可以同時指定硬體架構與版本：

  ```
  tux > sudo zypper install 'firefox.x86_64>=52.2'
  ```

依 RPM 檔案的路徑

您也可以指定套件的本地或遠端路徑：

```
tux > sudo zypper install /tmp/install/MozillaFirefox.rpm
tux > sudo zypper install http://download.example.com/MozillaFirefox.rpm
```

6.1.2.2 同時安裝並移除套件

若要同時安裝並移除套件，請使用 `+/-` 輔助按鍵。若要安裝 `emacs` 同時移除 `vim`，請使用：

```
tux > sudo zypper install emacs -vim
```

若要移除 `emacs` 同時安裝 `vim`，請使用：

```
tux > sudo zypper remove emacs +vim
```

為了防止將名稱以 `-` 開頭的套件解譯為指令選項，請一律將此類名稱用做第二個引數。如果行不通，則在前面加上 `--`：

```
tux > sudo zypper install -emacs +vim       # Wrong
tux > sudo zypper install vim -emacs        # Correct
tux > sudo zypper install -- -emacs +vim    # Correct
tux > sudo zypper remove emacs +vim         # Correct
```

6.1.2.3 清理已移除套件的相依性

如果想讓系統在移除某套件後自動移除由於此操作而導致不再需要的所有套件，請使用 `--clean-deps` 選項：

```
tux > sudo zypper rm PACKAGE_NAME --clean-deps
```

6.1.2.4 在程序檔中使用 Zypper

依預設，Zypper 會在安裝或移除所選套件前或出現問題時要求您確認。使用 `--non-interactive` 選項可覆寫此行為。此選項必須放在實際指令（`install`、`remove` 和 `patch`）的前面，如下所示：

```
tux > sudo zypper --non-interactive install PACKAGE_NAME
```

此選項允許在程序檔與 cron 工作中使用 Zypper。

6.1.2.5 安裝或下載來源套件

若要安裝某個套件對應的來源套件，請使用：

```
tux > zypper source-install PACKAGE_NAME
```

以 `root` 身分執行時，來源套件的預設安裝位置為 `/usr/src/packages/`；以使用者身分執行時，預設安裝位置為 `~/rpmbuild`。可以在本地 `rpm` 組態中變更這些值。

使用此指令還會安裝指定套件的建構相依性。如果不想執行此操作，請新增參數 `-D`：

```
tux > sudo zypper source-install -D PACKAGE_NAME
```

若想單獨安裝建構相依套件，請使用 `-d`。

```
tux > sudo zypper source-install -d PACKAGE_NAME
```

當然，您必須在儲存庫清單中啓用含來源套件的儲存庫（預設會新增，但不會啓用），此指令才有效。如需有關儲存庫管理的詳細資料，請參閱第 6.1.5 節 「使用 Zypper 管理儲存庫」。

儲存庫中所有可用來源套件的清單可透過以下指令獲得：

```
tux > zypper search -t srcpackage
```

您還可以將所有已安裝套件的來源套件下載到本地目錄。若要下載來源套件，請使用：

```
tux > zypper source-download
```

預設下載目錄為 `/var/cache/zypper/source-download`。您可以使用 `--directory` 選項變更該目錄。若只想顯示遺失或多餘的套件而不進行任何下載或刪除操作，請使用 `--status` 選項。若要刪除多餘的來源套件，請使用 `--delete` 選項。若要停用刪除操作，請使用 `--no-delete` 選項。

6.1.2.6 從已停用的儲存庫安裝套件

通常，您只能安裝或重新整理來自啓用的儲存庫的套件。`--plus-content` 標記選項可協助您指定要重新整理的、要在目前 Zypper 工作階段期間暫時啓用的，以及要在工作階段完成後停用的儲存庫。

例如，若要啓用可以提供其他 `-debuginfo` 或 `-debugsource` 套件的儲存庫，請使用 `--plus-content debug`。可以多次指定此選項。

若要暫時啓用此類「除錯」儲存庫以安裝特定的 `-debuginfo` 套件，請依如下所示使用該選項：

```
tux > sudo zypper --plus-content debug \
   install "debuginfo(build-id)=eb844a5c20c70a59fc693cd1061f851fb7d046f4"
```

對於缺失的 debuginfo 套件，`gdb` 會報告 `build-id` 字串。

6.1.2.7 公用程式

若要驗證是否仍滿足所有相依條件並執行未滿足的相依條件，請使用：

```
tux > zypper verify
```

除了必須滿足的相依條件之外，一些套件還會「推薦」其他套件。這些推薦的套件只有在確實可用並可安裝時才會進行安裝。如果所推薦的套件在推薦方套件安裝之後（透過新增其他套件或硬體）才可用，請使用以下指令：

```
tux > sudo zypper install-new-recommends
```

此指令在接入網路攝影機或 Wi-Fi 裝置後非常有用。它將安裝裝置的驅動程式及相關軟體（如果有）。驅動程式及相關軟體只有在符合了某些硬體相依性條件後才可安裝。

6.1.3 使用 Zypper 更新軟體

Zypper 有三種方法更新軟體：安裝修補程式、安裝新版本的套件或更新整個套裝作業系統。最後一種方法可藉由 `zypper dist-upgrade` 來實現。中介紹了如何升級 SUSE Linux Enterprise Server《部署指南》，第 19 章「升級 SUSE Linux Enterprise」。

6.1.3.1 安裝全部所需的修補程式

若要安裝所有正式發佈且適用於您系統的修補程式，請執行：

```
tux > sudo zypper patch
```

系統會檢查您電腦上設定的，來自儲存庫的所有修補程式是否與您的安裝相關。如果相關（未分類為 **選擇性** 或 **功能**），則立即安裝這些修補程式。請注意，正式的更新儲存庫僅在註冊 SUSE Linux Enterprise Server 安裝之後才可用。

如果要安裝的修補程式所包含的變更需要將系統重新開機才能生效，您事先會收到警告。

單純使用 `zypper patch` 指令不會套用來自協力廠商儲存庫的套件。若要同時更新協力廠商儲存庫，請使用 `with-update` 指令選項，如下所示：

```
tux > sudo zypper patch --with update
```

若要額外安裝選擇性修補程式，請使用：

```
tux > sudo zypper patch --with-optional
```

若要安裝與特定 Bugzilla 問題相關的所有修補程式，請使用：

```
tux > sudo zypper patch --bugzilla=NUMBER
```

若要安裝與特定 CVE 資料庫項目相關的所有修補程式，請使用：

```
tux > sudo zypper patch --cve=NUMBER
```

例外，若要安裝 CVE 編號為 `CVE-2010-2713` 的安全性修補程式，請執行：

```
tux > sudo zypper patch --cve=CVE-2010-2713
```

若您只想安裝影響 Zypper 和套件管理本身的修補程式，請使用：

```
tux > sudo zypper patch --updatestack-only
```

請記住，如果您使用 `updatestack-only` 指令選項，將會丟棄原本還會更新其他儲存庫的其他指令選項。

6.1.3.2 列出修補程式

為了讓您確定修補程式是否可用，Zypper 允許您檢視以下資訊：

所需修補程式的數量
: 若要列出所需修補程式的數量（適用於您系統但尚未安裝的修補程式），請使用 `patch-check`：

  ```
  tux > zypper patch-check
  Loading repository data...
  Reading installed packages...
  5 patches needed (1 security patch)
  ```

  可以結合 `--updatestack-only` 選項使用此指令，以便僅列出影響 Zypper 和套件管理本身的修補程式。

所需修補程式的清單
: 若要列出全部所需的修補程式（適用於您系統但尚未安裝的修補程式），請使用 `list-patches`：

  ```
  tux > zypper list-patches
  ```

```
Loading repository data...
Reading installed packages...

Repository     | Name        | Version | Category | Status  | Summary
---------------+-------------+---------+----------+---------+---------
SLES12-Updates | SUSE-2014-8 | 1       | security | needed  | openssl: Update for
 OpenSSL
```

所有修補程式的清單

若要列出可用於 SUSE Linux Enterprise Server 的所有修補程式，而不論它們是否已安裝或是否適用於您的安裝，請使用 `zypper patches`。

還會列出與特定問題相關的修補程式並加以安裝。若要列出特定的修補程式，請使用 `zypper list-patches` 指令及以下選項：

依 Bugzilla 問題

若要列出與 Bugzilla 問題相關的全部所需修補程式，請使用 `--bugzilla` 選項。

若要列出針對特定錯誤的修補程式，您也可以指定錯誤編號：`--bugzilla=編號`。

若要搜尋與多個 Bugzilla 問題相關的修補程式，請在錯誤號碼之間插入逗號，例如：

```
tux > zypper list-patches --bugzilla=972197,956917
```

依 CVE 號碼

若要列出與 CVE（通用弱點與揭露）資料庫中某個項目相關的全部所需修補程式，請使用 `--cve` 選項。

若要列出針對特定 CVE 資料庫項目的修補程式，您也可以指定 CVE 編號：`--cve=編號`。若要搜尋與多個 CVE 資料庫項目相關的修補程式，請在 CVE 號碼之間插入逗號，例如：

```
tux > zypper list-patches --bugzilla=CVE-2016-2315,CVE-2016-2324
```

若要列出所有修補程式而不論是否需要安裝它們，請額外使用 `--all` 選項。例如，若要列出指定有 CVE 號碼的所有修補程式，請使用：

```
tux > zypper list-patches --all --cve
Issue | No.           | Patch             | Category    | Severity  | Status
------+---------------+-------------------+-------------+-----------+----------
cve   | CVE-2015-0287 | SUSE-SLE-Module.. | recommended | moderate  | needed
cve   | CVE-2014-3566 | SUSE-SLE-SERVER.. | recommended | moderate  | not needed
```

```
[...]
```

6.1.3.3 安裝新的套件版本

如果儲存庫中只含有新套件，但未提供修補程式，則 `zypper patch` 不起任何作用。若要使用可用的較新版本更新所有安裝的套件（同時維持系統完整性），請使用：

```
tux > sudo zypper update
```

若要更新個別套件，請在更新或安裝指令中指定套件：

```
tux > sudo zypper update PACKAGE_NAME
sudo zypper install PACKAGE_NAME
```

所有可安裝的新套件的清單可透過以下指令獲得：

```
tux > zypper list-updates
```

請注意，此指令只會列出符合以下準則的套件：

- 與已安裝套件具有相同的廠商，
- 提供套件的儲存庫相較於已安裝套件，擁有更高或相同的優先程度，
- 可安裝（符合所有相依性條件）。

所有新可用套件（無論是否可安裝）的清單可透過以下指令獲得：

```
tux > sudo zypper list-updates --all
```

若要找出新套件無法安裝的原因，請依上文所述使用 `zypper install` 或 `zypper update` 指令。

6.1.3.4 識別孤立的套件

每當您從 Zypper 中移除某個儲存庫或者升級系統時，某些套件可能會進入「孤立」狀態。這些孤立的套件不再屬於任何使用中的儲存庫。以下指令可以列出這些套件：

```
tux > sudo zypper packages --orphaned
```

有了這份清單，您可以確定是否仍然需要某個套件，或者是否可以安全移除某個套件。

6.1.4 識別使用已刪除檔案的程序和服務

在修補、更新或移除套件時，系統上可能有一些執行中的程序仍在使用更新或移除過後已刪除的檔案。執行 `zypper ps` 可以列出使用已刪除檔案的程序。如果此類程序屬於某個已知的服務，則會列出服務名稱，方便您重新啓動該服務。`zypper ps` 預設會顯示一個表格：

```
tux > zypper ps
PID   | PPID | UID | User   | Command      | Service      | Files
------+------+-----+--------+--------------+--------------+-------------------
814   | 1    | 481 | avahi  | avahi-daemon | avahi-daemon | /lib64/ld-2.19.s->
      |      |     |        |              |              | /lib64/libdl-2.1->
      |      |     |        |              |              | /lib64/libpthrea->
      |      |     |        |              |              | /lib64/libc-2.19->
[...]
```

`PID`：程序的 ID
`PPID`：父程序的 ID
`UID`：執行程序之使用者的 ID
`Login`：執行程序之使用者的登入名稱
`Command`：用於執行程序的指令
`Service`：服務名稱（僅當指令與系統服務關聯時才顯示）
`Files`：已刪除的檔案清單

可依如下所示控制 `zypper ps` 的輸出格式：

`zypper ps -s`

建立一份簡短表格，其中不會顯示已刪除的檔案。

```
tux > zypper ps -s
PID   | PPID | UID  | User    | Command      | Service
------+------+------+---------+--------------+--------------
814   | 1    | 481  | avahi   | avahi-daemon | avahi-daemon
817   | 1    | 0    | root    | irqbalance   | irqbalance
1567  | 1    | 0    | root    | sshd         | sshd
1761  | 1    | 0    | root    | master       | postfix
1764  | 1761 | 51   | postfix | pickup       | postfix
1765  | 1761 | 51   | postfix | qmgr         | postfix
2031  | 2027 | 1000 | tux     | bash         |
```

`zypper ps -ss`

僅顯示與系統服務關聯的程序。

```
PID   | PPID | UID  | User    | Command      | Service
------+------+------+---------+--------------+--------------
814   | 1    | 481  | avahi   | avahi-daemon | avahi-daemon
817   | 1    | 0    | root    | irqbalance   | irqbalance
1567  | 1    | 0    | root    | sshd         | sshd
1761  | 1    | 0    | root    | master       | postfix
1764  | 1761 | 51   | postfix | pickup       | postfix
1765  | 1761 | 51   | postfix | qmgr         | postfix
```

`zypper ps -sss`

僅顯示使用已刪除檔案的系統服務。

```
avahi-daemon
irqbalance
postfix
sshd
```

`zypper ps --print "systemctl status %s"`

顯示用於取回可能需要重新啓動的服務之狀態的指令。

```
systemctl status avahi-daemon
systemctl status irqbalance
systemctl status postfix
systemctl status sshd
```

如需服務處理的詳細資訊，請參閱第 13 章 「systemd 精靈」。

6.1.5 使用 Zypper 管理儲存庫

Zypper 的所有安裝或修補指令均依賴於一系列已知的儲存庫。若要列出系統可識別的所有儲存庫，請使用指令：

```
tux > zypper repos
```

結果類似於以下輸出：

範例 6.1 ZYPPER — 已知儲存庫的清單

```
tux > zypper repos
# | Alias        | Name          | Enabled | Refresh
--+--------------+---------------+---------+--------
1 | SLEHA-12-GEO | SLEHA-12-GEO  | Yes     | No
```

```
2 | SLEHA-12      | SLEHA-12      | Yes     | No
3 | SLES12        | SLES12        | Yes     | No
```

在各種指令中指定儲存庫時，可以使用 `zypper repos` 指令輸出的別名、URI 或儲存庫編號。儲存庫別名是儲存庫名稱的縮寫形式，用於儲存庫處理指令。請注意，修改儲存庫清單後，儲存庫的編號可能會發生變更。但別名永遠不會自行變更。

依預設，不會顯示儲存庫的詳細資料（如 URI 或優先程度）。若要列出所有詳細資料，可以使用以下指令：

```
tux > zypper repos -d
```

6.1.5.1 新增儲存庫

若要新增儲存庫，請執行

```
tux > sudo zypper addrepo URI ALIAS
```

`URI` 可以是網際網路儲存庫、網路資源、目錄，也可以是 CD 或 DVD（如需詳細資料，請造訪 http://en.opensuse.org/openSUSE:Libzypp_URIs）。`ALIAS` 是儲存庫的唯一縮寫識別碼。您可以隨意選擇，前提是它必須是唯一的。如果指定了已使用的別名，Zypper 會發出警告。

6.1.5.2 重新整理儲存庫

`zypper` 可讓您從設定的儲存庫中擷取套件中的變更。若要擷取變更，請執行：

```
tux > sudo zypper refresh
```

注意：`zypper` 的預設行為

有些指令預設會自動執行 `refresh`，因此您不需要明確執行該指令。

使用 `refresh` 指令時搭配 `--plus-content` 選項還可檢視已停用儲存庫中的變更：

```
tux > sudo zypper --plus-content refresh
```

該選項雖然會擷取儲存庫中的變更，但會使已停用儲存庫的狀態保持不變，即仍為停用。

6.1.5.3 移除儲存庫

若要從清單中移除某個儲存庫，請將指令 `zypper removerepo` 與要刪除的儲存庫的別名或編號結合使用。例如，若要從範例 6.1 「Zypper — 已知儲存庫的清單」中移除儲存庫 `SLEHA-12-GEO`，請使用下列其中一個指令：

```
tux > sudo zypper removerepo 1
tux > sudo zypper removerepo "SLEHA-12-GEO"
```

6.1.5.4 修改儲存庫

`zypper modifyrepo` 可以啓用或停用儲存庫。您也可以使用此指令變更儲存庫的內容（如重新整理行為、名稱或優先程度）。以下指令將啓用名為 `updates` 的儲存庫，開啓自動重新整理功能，並將優先程度設為 20：

```
tux > sudo zypper modifyrepo -er -p 20 'updates'
```

修改儲存庫並不局限於單個儲存庫，您也可以對群組執行該操作：

`-a`：所有儲存庫
`-l`：本地儲存庫
`-t`：遠端儲存庫
`-m 類型`：某種類型的儲存庫（其中 `類型` 可以是以下其中一種：`http`、`https`、`ftp`、`cd`、`dvd`、`dir`、`file`、`cifs`、`smb`、`nfs`、`hd`、`iso`）

若要重新命名儲存庫別名，請使用 `renamerepo` 指令。以下範例會將別名 `Mozilla Firefox` 變更為 `firefox`：

```
tux > sudo zypper renamerepo 'Mozilla Firefox' firefox
```

6.1.6 使用 Zypper 查詢儲存庫和套件

Zypper 提供多種方法來查詢儲存庫或套件。若要獲得所有可用產品、模式、套件或修補程式的清單，請使用以下指令：

```
tux > zypper products
tux > zypper patterns
tux > zypper packages
```

```
tux > zypper patches
```

若要在所有儲存庫中查詢某些套件，請使用 `search`。若要獲得有關特定套件的資訊，請使用 `info` 指令。

6.1.6.1 `zypper search` 用法

`zypper search` 指令可以對套件名稱或（視情況）對套件摘要及描述進行搜尋。以 `/` 括住的字串會解譯為正規表示式。依預設，搜尋不區分大小寫。

執行簡單搜尋以尋找包含 `fire` 的套件名稱

```
tux > zypper search "fire"
```

執行簡單搜尋以尋找確切的套件 `MozillaFirefox`

```
tux > zypper search --match-exact "MozillaFirefox"
```

同時在套件描述和摘要中搜尋

```
tux > zypper search -d fire
```

僅顯示尚未安裝的套件

```
tux > zypper search -u fire
```

顯示包含字串 `fir` 且該字串後面不是 `e` 的套件

```
tux > zypper se "/fir[^e]/"
```

6.1.6.2 `zypper what-provides` 用法

若要搜尋提供特殊功能的套件，請使用指令 `what-provides`。例如，如果您想瞭解哪個套件提供 Perl 模組 `SVN::Core`，請使用以下指令：

```
tux > zypper what-provides 'perl(SVN::Core)'
```

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

6.1.6.3 `zypper info` 用法

若要查詢單個套件，請在 `info` 指令中使用準確的套件名稱做為引數。這會顯示有關某個套件的詳細資訊。如果套件名稱與儲存庫中的所有套件名稱都不相符，該指令會輸出非套件相符項目的詳細資訊。如果您（透過使用 `-t` 選項）要求特定類型，但該類型不存在，該指令會輸出其他可用的相符項，但不提供詳細資訊。

如果您指定來源套件，該指令會顯示基於該來源套件建立的二進位套件。如果指定二進位套件，該指令會輸出用來建立該二進位套件的來源套件。

如果還想顯示套件所需/推薦的項目，請使用選項 `--requires` 和 `--recommends`：

```
tux > zypper info --requires MozillaFirefox
```

6.1.7 設定 Zypper

Zypper 現隨附一個組態檔案，透過該檔案可永久變更 Zypper 的行為（整個系統範圍或僅針對特定使用者）。對於系統範圍的變更，請編輯 `/etc/zypp/zypper.conf`。對於特定使用者的變更，請編輯 `~/.zypper.conf`。如果 `~/.zypper.conf` 尚不存在，您可以使用 `/etc/zypp/zypper.conf` 做為範本：將其複製到 `~/.zypper.conf` 並根據喜好進行調整。如需有關可用選項的說明，請參閱檔案中的備註。

6.1.8 疑難排解

如果您在存取所設定儲存庫中的套件時遇到問題（例如，儘管您知道某個套件在某個儲存庫中，但 Zypper 找不到該套件），重新整理儲存庫或許可以解決問題：

```
tux > sudo zypper refresh
```

如果不起作用，請嘗試

```
tux > sudo zypper refresh -fdb
```

此指令會強制執行全面的重新整理和資料庫重建，包括強制下載原始中繼資料。

6.1.9 Btrfs 檔案系統上的 Zypper 復原功能

如果根分割區上使用的是 Btrfs 檔案系統，且系統中安裝了 `snapper`，當 Zypper 提交對檔案系統所做的變更以建立相應的檔案系統快照時，會自動呼叫 `snapper`。這些快照可用於回復 Zypper 進行的任何變更。如需相關資訊，請參閱第 7 章 「使用 Snapper 進行系統復原和快照管理」。

6.1.10 更多資訊

如需透過指令行管理軟體的詳細資訊，請輸入 `zypper help`、`zypper help 指令`，或參閱 `zypper(8)` 手冊頁。如需詳盡的指令參考、最重要指令的**彙總表**，以及有關如何在程序檔和應用程式中使用 Zypper 的資訊，請參閱 http://en.opensuse.org/SDB:Zypper_usage。最新 SUSE Linux Enterprise Server 版本的軟體變更清單可在 http://en.opensuse.org/openSUSE:Zypper versions 中找到。

6.2 RPM — 套件管理員

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

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

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

提示：軟體開發套件

對於多個套件，軟體開發所需的元件（程式庫、標題、包含檔案等）已置於獨立的套件中。只有在您想要自行編譯軟體（例如，最新的 GNOME 套件）時，才需要這些開發套件。根據副檔名 `-devel` 即可識別出這些套件，例如 `alsa-devel` 和 `gimp-devel` 套件。

6.2.1 確認套件驗證性

RPM 套件具有 GPG 簽名。若要驗證 RPM 套件的簽名，請使用 `rpm --checksig` PACKAGE-1.2.3.rpm 指令來確定該套件是來自 SUSE 還是另一個可信機構。特別建議在從網際網路更新套件時使用此指令。

修復作業系統中的問題時，您可能需要將問題暫時修復 (PTF) 安裝到線上系統中。SUSE 提供的套件已使用特殊 PTF 金鑰簽署。但是，與 SUSE Linux Enterprise 11 相比，在 SUSE Linux Enterprise 12 系統上，預設不會輸入此金鑰。若要手動輸入該金鑰，請使用以下指令：

```
tux > sudo rpm --import \
/usr/share/doc/packages/suse-build-key/suse_ptf_key.asc
```

輸入該金鑰後，您可以在系統上安裝 PTF 套件。

6.2.2 管理套件：安裝、更新和解除安裝

一般而言，安裝 RPM 歸檔很簡單，只需執行：`rpm -i` PACKAGE.rpm。使用此指令可安裝套件，但是必須滿足其相依性條件，而且不能與其他套件衝突。如果 `rpm` 要求要安裝的套件必須符合相依性要求，會顯示錯誤訊息。RPM 資料庫會在背景確定未產生衝

突，亦即特定檔案僅可屬於一個套件。藉由選擇不同選項，您可以强迫 `rpm` 忽略這些預設，但只有進階使用者才可以這樣做。否則，會危及系統完整性，還可能危害更新系統的能力。

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

- 如果系統管理員未變更組態檔，`rpm` 會安裝新版本的相應檔案。系統管理員不需要做任何動作。
- 如果更新前系統管理員曾變更組態檔案，則 `rpm` 會以副檔名 `.rpmorig` 或 `.rpmsave`（備份檔案）儲存變更的檔案，並安裝新套件中的版本。僅當原先安裝的檔案和較新的版本不同時，才執行此操作。在這種情況下，請比較備份檔案（`.rpmorig` 或 `.rpmsave`）與新安裝的檔案，然後再對新檔案做一次變更。之後，請刪除所有 `.rpmorig` 和 `.rpmsave` 檔案，以免日後的更新出現問題。
- 如果組態檔已存在，且如果在 `.spec` 檔案中指定了 `noreplace` 標籤，便會出現 `.rpmnew` 檔案。

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

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

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

若要移除套件，請輸入 `rpm -e PACKAGE`。此指令只在不存在未解決的相依性問題時才會刪除套件。只要其他應用程式還需要它，理論上無法刪除 Tcl/Tk。即使是這種情況下，RPM 還是可從資料庫呼叫以得到協助。如果由於某種原因無法進行這樣的刪除操作（即使不存在其他相依性問題），或許可以使用選項 `--rebuilddb` 來重建 RPM 資料庫。

6.2.3 Delta RPM 套件

增量 RPM 套件包含舊版與新版 RPM 套件之間的差異。將增量 RPM 套用到舊版 RPM 上會產生一個全新的 RPM。但是您不需要取得舊版的 RPM，因為增量 RPM 也可以和安裝的 RPM 配合使用。增量 RPM 套件的大小比修補程式 RPM 還小，這一特點有利於透過網際網路傳送更新套件。缺點是使用增量 RPM 的更新作業會比一般或修補程式 RPM 消耗更多的 CPU 週期。

`makedeltarpm` 和 `applydelta` 二進位檔案屬於增量 RPM 套裝軟體（`deltarpm` 套件）的一部分，可協助您建立並套用增量 RPM 套件。您可以使用下列指令建立名稱為 `new.delta.rpm` 的增量 RPM。下列指令假設 `old.rpm` 和 `new.rpm` 都已存在：

```
tux > sudo makedeltarpm old.rpm new.rpm new.delta.rpm
```

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

```
tux > sudo applydeltarpm new.delta.rpm new.rpm
```

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

```
tux > sudo applydeltarpm -r old.rpm new.delta.rpm new.rpm
```

如需技術詳細資訊，請參閱 `/usr/share/doc/packages/deltarpm/README`。

6.2.4 RPM 查詢

`rpm` 指令在使用 `-q` 選項時會啓動查詢，以便檢查 RPM 歸檔（藉由新增選項 `-p`），並查詢所安裝套件的 RPM 資料庫。有多個切換參數可用於指定所需的資訊類型。請參閱表格 6.1 「最重要的 RPM 查詢選項」。

表格 6.1 最重要的 RPM 查詢選項

`-i`	套件資訊

`-l`	檔案清單
`-f FILE`	查詢包含 `FILE` 檔案的套件（完整的路徑必須以 `FILE` 指定）
`-s`	含有狀態資訊的檔案清單（隱含 `-l`）
`-d`	只列出文件檔案（隱含 `-l`）
`-c`	只列出組態檔案（隱含 `-l`）
`--dump`	含有完整詳細資訊的檔案清單（與 `-l`、`-c` 或 `-d` 配合使用）
`--provides`	列出另一個套件可以使用 `--requires` 要求的套件功能
`--requires`、`-R`	套件所需的功能
`--scripts`	安裝程序檔（預先安裝、後續安裝、解除安裝）

例如，`rpm -q -i wget` 指令可顯示如 範例 6.2 「`rpm -q -i wget`」 中所示的資訊。

範例 6.2 `rpm -q -i wget`

```
Name        : wget
Version     : 1.14
Release     : 17.1
Architecture: x86_64
Install Date: Mon 30 Jan 2017 14:01:29 CET
Group       : Productivity/Networking/Web/Utilities
Size        : 2046483
License     : GPL-3.0+
Signature   : RSA/SHA256, Thu 08 Dec 2016 07:48:44 CET, Key ID 70af9e8139db7c82
Source RPM  : wget-1.14-17.1.src.rpm
Build Date  : Thu 08 Dec 2016 07:48:34 CET
Build Host  : sheep09
Relocations : (not relocatable)
Packager    : https://www.suse.com/
Vendor      : SUSE LLC <https://www.suse.com/>
URL         : http://www.gnu.org/software/wget/
Summary     : A Tool for Mirroring FTP and HTTP Servers
```

```
Description :
Wget enables you to retrieve WWW documents or FTP files from a server.
This can be done in script files or via the command line.
Distribution: SUSE Linux Enterprise 12
```

只有在您指令完整檔案名稱及完整路徑時，選項 `-f` 才會有作用。盡可能提供很多檔案名稱。例如：

```
tux > rpm -q -f /bin/rpm /usr/bin/wget
rpm-4.11.2-15.1.x86_64
wget-1.14-17.1.x86_64
```

如果只知道檔案名稱的一部分，可使用範例 6.3 「搜尋套件的程序檔」中所示的外圍程序檔。執行時，可將部份檔案名稱當作參數傳給程序檔。

範例 6.3 搜尋套件的程序檔

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

`rpm -q --changelog` PACKAGE 指令會顯示有關特定套件的詳細變更資訊清單，並依日期排序。

藉由安裝的 RPM 資料庫，可執行驗證檢查。這些檢查可以使用 `-V` 或 `--verify` 來啓動。使用此選項，`rpm` 可顯示從安裝開始，套件中所有變更過的檔案。`rpm` 使用八個字元的符號來提供下列變更的提示：

表格 6.2 RPM 驗證選項

`5`	MD5 檢查總數
`S`	檔案大小
`L`	符號連結
`T`	修改時間
`D`	主要和次要的裝置編號

U	擁有者
G	群組
M	模式（許可權和檔案類型）

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

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

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

6.2.5 安裝與編譯來源套件

所有來源套件均帶有副檔名 .src.rpm（來源 RPM）。

注意：安裝的來源套件

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

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

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

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

`BUILD`
: 所有來源均在此目錄中解壓縮、修補和編譯

`RPMS`
: 儲存完整二進位套件的地方

`SRPMS`
: 此處為來源 RPM

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

警告：系統完整性

請不要以系統元件（`glibc`、`rpm` 等）做試驗，因為這樣會危害系統的穩定性。

以下範例使用 `wget.src.rpm` 套件。安裝來源套件之後，會獲得類似下列清單中所列的檔案：

```
/usr/src/packages/SOURCES/wget-1.11.4.tar.bz2
/usr/src/packages/SOURCES/wgetrc.patch
/usr/src/packages/SPECS/wget.spec
```

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

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

`-bc`
: 執行與 `-bp` 相同動作，但是會額外編譯。

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

`-bb`

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

`-ba`

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

`--short-circuit`

略過部分步驟。

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

請記住，從 SUSE Linux Enterprise Server 12 開始，已棄用規格檔案中的 `BuildRoot` 指令。如果您仍然需要此功能，請使用 `--buildroot` 選項做為因應措施。如需更詳細的背景，請造訪 https://www.suse.com/support/kb/doc?id=7017104 並檢視其中的支援資料庫。

6.2.6 以 build 編譯 RPM 套件

許多套件中都包含不想要的檔案，它們會在 build 程序中增到執行系統中，因為導致危險產生。若要避免此狀況，可以使用 `build`，它會建立要在其中建置套件的定義環境。若要建立此 chroot 環境，必須提供 `build` 程序檔與完整的套件樹狀結構。此樹狀結構可在硬碟上、透過 NFS 或從 DVD 取得。使用 `build --rpms DIRECTORY` 設定位置。與 `rpm` 不同，`build` 指令在來源目錄中尋找 `.spec` 檔。若要以掛接在系統的 `/media/dvd` 之下的 DVD 建立 `wget`（如上面的範例），請以 `root` 身分執行下列指令：

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

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

`build` 程序檔提供其他多個選項。例如，讓程序檔偏好使用您自己的 RPM、省略建置環境的啓始化，或將 `rpm` 指令限制在上述某個階段。可使用 `build --help` 以及參閱 `build` man 頁面來存取其他資訊。

6.2.7 RPM 歸檔和 RPM 資料庫工具

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

具有完整功能的套件管理員是以 YaST 模組的方式提供。如需詳細資料，請參閱《部署指南》，第 13 章「安裝或移除軟體」。

7 使用 Snapper 進行系統復原和快照管理

在 Linux 上建立檔案系統快照以執行復原的功能是過去使用者常常要求的一項功能。現在，Snapper 與 Btrfs 檔案系統或簡易佈建的 LVM 磁碟區合用，填補了這個空白。

Btrfs 是一個適用於 Linux 的全新寫時複製檔案系統，它支援子磁碟區（每個實體分割區中的一或多個可獨立掛接的檔案系統）的檔案系統快照（子磁碟區於特定時間點之狀態的副本）。快照在使用 XFS、Ext4 或 Ext3 格式化之簡易佈建的 LVM 磁碟區上同樣受支援。Snapper 可讓您建立和管理這些快照。它包含一個指令行和一個 YaST 介面。從 SUSE Linux Enterprise Server 12 開始，還可以從 Btrfs 快照開機。如需詳細資訊，請參閱第 7.3 節 「透過從快照開機來執行系統復原」。

您可以使用 Snapper 執行以下任務：

- 復原 zypper 和 YaST 所做的系統變更。如需詳細資料，請參閱第 7.2 節 「使用 Snapper 復原變更」。
- 從之前的快照還原檔案。如需詳細資料，請參閱第 7.2.2 節 「使用 Snapper 還原檔案」。
- 透過從快照開機來復原系統。如需詳細資料，請參閱第 7.3 節 「透過從快照開機來執行系統復原」。
- 手動建立即時快照並管理現有的快照。如需詳細資料，請參閱第 7.5 節 「手動建立和管理快照」。

7.1 預設設定

SUSE Linux Enterprise Server 上的 Snapper 設定為充當系統變更的「復原工具」。依預設，SUSE Linux Enterprise Server 的根分割區（/）使用 Btrfs 格式化。如果根分割區（/）足夠大（大約超過 16GB），則會自動啓用快照建立功能。預設不會在 / 外的分割區上建立快照。

提示：在已安裝系統中啓用 Snapper

如果您在安裝期間停用了 Snapper，以後隨時都可啓用它。若要進行此操作，請執行以下指令以建立根檔案系統的預設 Snapper 組態：

```
tux > sudo snapper -c root create-config /
```

之後，依第 7.1.3.1 節 「停用/啓用快照」所述啓用不同的快照類型。

請記住，若要使用快照，需依照安裝程式的建議設定一個包含子磁碟區的 Btrfs 根檔案系統，並且需有一個大小至少為 16 GB 的分割區。

建立快照時，快照和原件都會指向檔案系統中的同一區塊。因此，快照最初並不佔用額外的磁碟空間。如果原始檔案系統中的資料經過修改，則會複製變更後的資料區塊，同時保留快照的舊資料區塊。因此，快照便會佔用與已修改資料相同的空間。這樣，經過一段時間之後，快照配置的空間不斷增大。因而，從包含快照的 `Btrfs` 檔案系統刪除檔案可能無法釋放磁碟空間！

注意：快照位置

快照始終位於建立快照所在的同一分割區或子磁碟區上。而無法儲存到其他分割區或子磁碟區上。

因此，包含快照的分割區必須比「一般」分割區更大。確切大小很大程度上取決於保留的快照數和資料修改量。一般來說，您應該考慮使用比平常大兩倍的大小。為了防止磁碟上的空間耗盡，系統會自動清理舊快照。如需詳細資訊，請參閱第 7.1.3.4 節 「控制快照歸檔」。

7.1.1 快照類型

儘管快照本身在技術方面並無區別，但我們根據觸發它們的事件將其分成三類：

時間軸快照

每小時建立一個快照。系統會自動刪除舊快照。依預設，系統會保留過去十天、十個月或十年的第一個快照。時間軸快照預設已停用。

安裝快照

每當使用 YaST 或 Zypper 安裝一個或多個套件時，均會建立一對快照：安裝開始前建立一個（「前」），安裝結束後建立另一個（「後」）。如果重要系統元件（如核心）已經安裝，則快照對會標示為重要（`important=yes`）。系統會自動刪除舊快照。依預設，系統會保留最近十個重要快照以及最近十個「一般」快照（包括管理快照）。預設系統會啓用安裝快照。

管理快照

每當您使用 YaST 管理系統時，均會建立一對快照：啓動 YaST 模組時建立一個（「前」），關閉模組時建立另一個（「後」）。系統會自動刪除舊快照。依預設，系統會保留最近十個重要快照以及最近十個「一般」快照（包括安裝快照）。預設系統會啓用管理快照。

7.1.2 從快照中排除的目錄

出於不同原因，需要將一些目錄從快照中排除。下列清單顯示排除的所有目錄：

`/boot/grub2/i386-pc`、`/boot/grub2/x86_64-efi`、`/boot/grub2/powerpc-ieee1275`、`/boot/grub2/s390x-emu`

不支援對開機載入程式組態進行復原。上面列出的目錄是架構專屬目錄。前兩個目錄位於 AMD64/Intel 64 機器上，後兩個目錄分別位於 IBM POWER 和 IBM z Systems 上。

`/home`

如果 `/home` 不在獨立的分割區上，系統會將其排除以避免在復原時發生資料遺失。

`/opt`、`/var/opt`

協力廠商產品通常會安裝到 `/opt`。系統會將該目錄排除以避免在復原時解除安裝這些應用程式。

`/srv`

包含 Web 和 FTP 伺服器的資料。系統會將該目錄排除以避免在復原時發生資料遺失。

`/tmp`、`/var/tmp`、`/var/cache`、`/var/crash`

包含暫存檔案和快取的所有目錄均會從快照中排除。

`/usr/local`

在手動安裝軟體時會用到此目錄。系統會將該目錄排除，以免在復原時解除安裝這些安裝的軟體。

`/var/lib/libvirt/images`

使用 libvirt 管理之虛擬機器影像的預設位置。已排除，以確定復原期間虛擬機器影像不會取代為舊版本。依預設，此子磁碟區是使用**寫入時不複製**選項建立的。

`/var/lib/mailman` 、 `/var/spool`

系統會排除包含郵件或郵件佇列的目錄以避免復原之後遺失郵件。

`/var/lib/named`

包含 DNS 伺服器的區域資料。從快照中排除該目錄是為了確保名稱伺服器在復原之後可以運作。

`/var/lib/mariadb` 、 `/var/lib/mysql` 、 `/var/lib/pgqsl`

這些目錄包含資料庫資料。依預設，這些子磁碟區是使用**寫入時不複製**選項建立的。

`/var/log`

記錄檔案位置。從快照中排除該目錄是為了在復原損毀系統之後能夠對記錄檔案進行分析。

7.1.3 自訂設定

SUSE Linux Enterprise Server 隨附一個合理的預設設定，該設定適合大多數使用案例。不過，您可以根據自己的需求對建立自動快照以及快照保留的方方面面進行設定。

7.1.3.1 停用/啓用快照

三種快照類型（時間軸、安裝、管理）均可獨立啓用或停用。

停用/啓用時間軸快照

啓用： `snapper-c root set-config "TIMELINE_CREATE=yes"`

停用： `snapper -c root set-config "TIMELINE_CREATE=no"`

預設會啓用時間軸快照，根分割區除外。

停用/啓用安裝快照

啓用：　安裝套件 `snapper-zypp-plugin`

停用：　解除安裝套件 `snapper-zypp-plugin`

預設系統會啓用安裝快照。

停用/啓用管理快照

啓用：　在 `/etc/sysconfig/yast2` 中將 `USE_SNAPPER` 設定為 `yes`。

停用：　在 `/etc/sysconfig/yast2` 中將 `USE_SNAPPER` 設定為 `no`。

預設系統會啓用管理快照。

7.1.3.2 控制安裝快照

使用 YaST 或 Zypper 安裝套件時所建立的快照對由 `snapper-zypp-plugin` 處理。何時建立快照由 XML 組態檔案 `/etc/snapper/zypp-plugin.conf` 定義。依預設，該檔案如下所示：

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <snapper-zypp-plugin-conf>
 3  <solvables>
 4   <solvable match="w"❶ important="true"❷>kernel-*❸</solvable>
 5   <solvable match="w" important="true">dracut</solvable>
 6   <solvable match="w" important="true">glibc</solvable>
 7   <solvable match="w" important="true">systemd*</solvable>
 8   <solvable match="w" important="true">udev</solvable>
 9   <solvable match="w">*</solvable>❹
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

❶ match 屬性定義模式是 Unix 外圍程序樣式的萬用字元（`w`）還是 Python正規表示式（`re`）。

❷ 如果符合指定模式且對應的套件標示為 important（例如核心套件），則快照也會標示為 important。

❸ 用於比對套件名稱的模式。根據 `match` 屬性的設定，特殊字元也可能會被解譯為外圍程序萬用字元或正規表示式。此模式符合所有以 `kernel-` 開頭的套件名稱。

❹ 此行無條件符合所有套件。

使用此組態時，只要安裝套件即會建立快照對（第 9 行）。如果標示為 important 的核心、dracut、glibc、systemd 或 udev 套件已安裝，快照對也會標示為 important（第 4 行至第 8 行）。系統會評估所有規則。

要停用規則，請使用 XML 備註將其刪除或停用。若想避免系統在每次安裝套件時都建立快照對，可將第 9 行設為備註：

```
 1 <?xml version="1.0" encoding="utf-8"?>
 2 <snapper-zypp-plugin-conf>
 3  <solvables>
 4   <solvable match="w" important="true">kernel-*</solvable>
 5   <solvable match="w" important="true">dracut</solvable>
 6   <solvable match="w" important="true">glibc</solvable>
 7   <solvable match="w" important="true">systemd*</solvable>
 8   <solvable match="w" important="true">udev</solvable>
 9   <!-- <solvable match="w">*</solvable> -->
10  </solvables>
11 </snapper-zypp-plugin-conf>
```

7.1.3.3 建立和掛接新子磁碟區

系統支援在 `/` 階層下建立新的子磁碟區，並永久掛接該磁碟區。此類子磁碟區將從快照中排除。切勿在現有快照中建立此類子磁碟區，因為在復原之後，您將無法再刪除快照。

SUSE Linux Enterprise Server 上設定了 `/@/` 子磁碟區，該子磁碟區用做永久子磁碟區（例如 `/opt`、`/srv`、`/home` 等）的獨立根分割區。您建立和永久掛接的任何新子磁碟區都需要在這個初始根檔案系統中建立。

若要這樣做，請執行以下指令。在此範例中，從 `/dev/sda2` 建立了一個新子磁碟區 `/usr/important`。

```
tux > sudo mount /dev/sda2 -o subvol=@ /mnt
tux > sudo btrfs subvolume create /mnt/usr/important
tux > sudo umount /mnt
```

`/etc/fstab` 中的相應項目需類似於：

```
/dev/sda2 /usr/important btrfs subvol=@/usr/important 0 0
```

提示：停用寫入時複製（COW）

子磁碟區可能包含經常變更的檔案，例如虛擬化磁碟影像、資料庫檔案或記錄檔案。如果是這樣，可考慮對此磁碟區停用寫入時複製功能，以免複製磁碟區塊。在 `/etc/fstab` 中使用 `nodatacow` 掛接選項可實現此目的：

```
/dev/sda2 /usr/important btrfs nodatacow,subvol=@/usr/important 0 0
```

或者，若要為單個檔案或目錄停用寫入時複製功能，請使用指令 `chattr +C 路徑`。

7.1.3.4 控制快照歸檔

快照會佔用磁碟空間。為了防止磁碟用盡而導致系統中斷，會自動刪除舊快照。預設會保留最多 10 個重要的安裝快照與管理快照，以及最多 10 個普通的安裝快照與管理快照。如果這些快照佔用的空間超過根檔案系統大小的 50%，則會刪除其他快照。永遠會至少保留 4 個重要快照和 2 個普通快照。

如需如何變更這些值的指示，請參閱第 7.4.1 節 「管理現有的組態」。

7.1.3.5 對簡易佈建的 LVM 磁碟區使用 Snapper

除了針對 `Btrfs` 檔案系統建立快照之外，Snapper 還支援針對使用 XFS、Ext4 或 Ext3 格式化之簡易佈建的 LVM 磁碟區建立快照（但不支援對一般 LVM 磁碟區建立快照）。如需 LVM 磁碟區的詳細資訊以及設定指示，請參閱《部署指南》，第 12 章「進階磁碟設定」，第 12.2 節「LVM 組態」。

若要對簡易佈建的 LVM 磁碟區使用 Snapper，您需要為它建立一個 Snapper 組態。在 LVM 上，需要使用 `--fstype=lvm(檔案系統)` 指定檔案系統。`檔案系統` 的有效值有 `ext3`、`etx4` 或 `xfs`。範例：

```
tux > sudo snapper -c lvm create-config --fstype="lvm(xfs)" /thin_lvm
```

您可以依第 7.4.1 節 「管理現有的組態」 中所述根據需求調整此組態。

7.2 使用 Snapper 復原變更

SUSE Linux Enterprise Server 上的 Snapper 已預先設定為充當復原 `zypper` 和 YaST 所做變更的工具。要充當復原工具，Snapper 設定為在每次執行 `zypper` 和 YaST 的前後建立一對快照。此外，Snapper 還可讓您還原遭意外刪除或修改的系統檔案。出於此目的需要啓用根分割區的時間軸快照 — 如需詳細資訊，請參閱第 7.1.3.1 節 「停用/啓用快照」。

預設會為根分割區及其子磁碟區設定上述的自動快照。為了讓這些快照可供其他分割區（例如 `/home`）使用，您可以建立自訂組態。

重要：復原變更與復原的比較

使用快照還原資料時，您必須知道 Snapper 可以處理兩種完全不同的案例。

復原變更
: 當如下文所述復原變更時，系統會比較兩個快照，並復原這兩個快照之間的變更。使用此方法還允許明確選取要還原的檔案。

復原
: 當如第 7.3 節 「透過從快照開機來執行系統復原」中所述執行復原時，系統會重設回建立快照當時的狀態。

復原變更時，還可以將快照與目前系統進行比較。根據此類比較還原全部檔案時，其效果等同於執行復原。但是，還是建議使用第 7.3 節 「透過從快照開機來執行系統復原」中所述的方法復原，因為它的速度更快並且可讓您在執行復原之前複查系統。

警告：資料一致性

在建立快照時，沒有任何一種機制可確保資料的一致性。如果在建立快照的同時寫入某個檔案（如資料庫），將導致檔案損毀或寫入不完整。還原此類檔案將會導致問題。此外，有些系統檔案（例如 `/etc/mtab`）是絕對不能還原的。因此，強烈建議您始終仔細檢閱已變更檔案及其差異的清單。請只還原真正屬於您要執行回復的檔案。

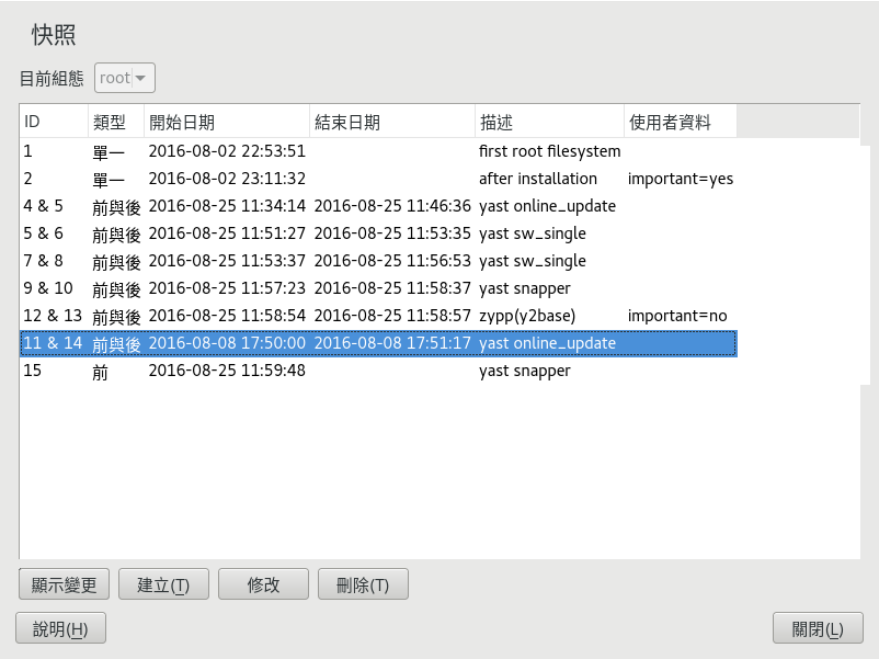
7.2.1 復原 YaST 和 Zypper 變更

如果您在安裝期間使用 `Btrfs` 設定根分割區，系統將會自動安裝 Snapper（已預先設定為用於復原 YaST 或 Zypper 所做的變更）。每當您啓動 YaST 模組或 Zypper 交易時，都會建立兩個快照：擷取啓動模組之前檔案系統狀態的「前快照」，以及完成模組之後的「後快照」。

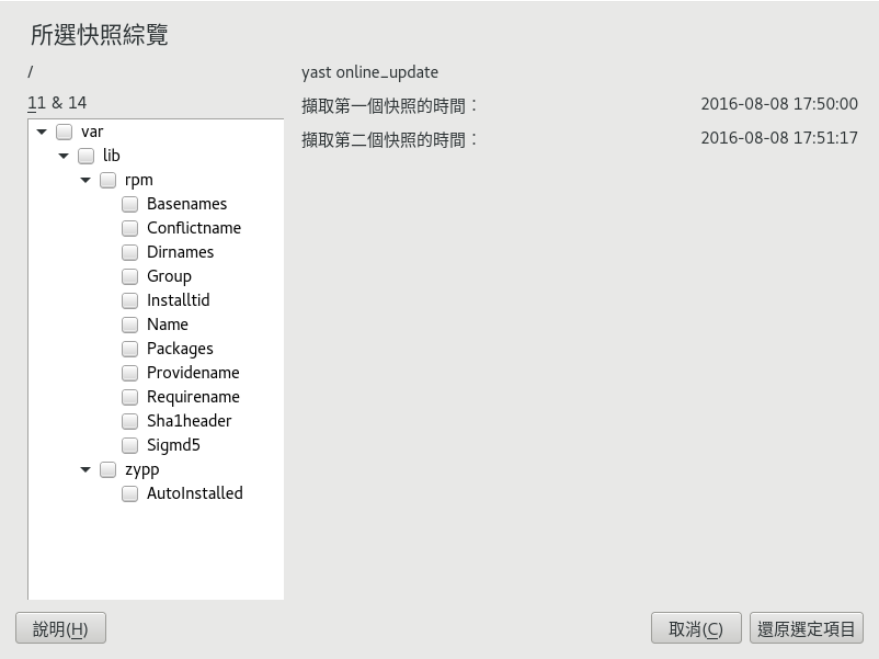
您可以使用 YaST Snapper 模組或 `snapper` 指令行工具，透過從「前快照」還原檔案來復原 YaST/Zypper 所做的變更。比較兩個快照時，這些工具還可讓您查看哪些檔案已經過變更。此外，您還可以顯示某檔案的兩個版本之間的差異（`diff`）。

程序 7.1 使用 YAST SNAPPER 模組復原變更

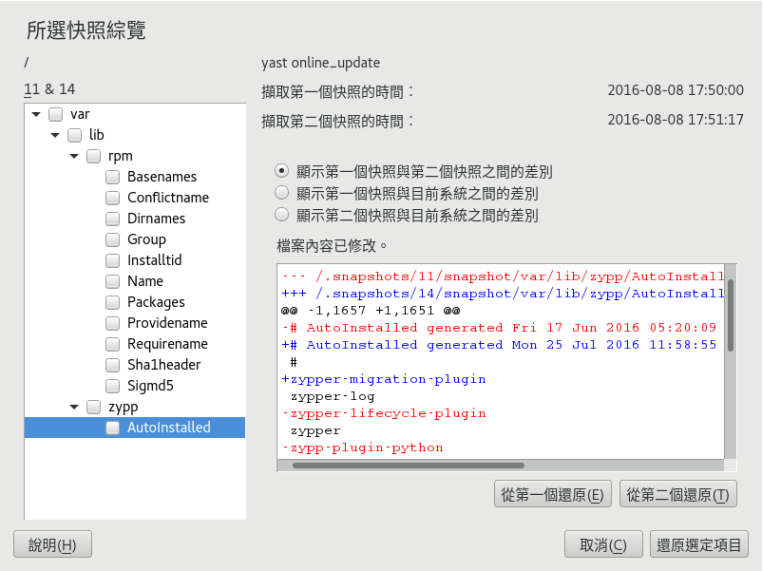
1. 從 YaST 的其他區段或透過輸入 `yast2 snapper` 啓動Snapper模組。
2. 請確定目前組態設為根。除非您手動新增了自己的 Snapper 組態，否則應始終如此設定。
3. 從清單中選擇一對前快照和後快照。YaST 與 Zypper 快照對都屬於前與後類型。在描述欄中，YaST 快照標示為 `zypp(y2base)`，Zypper 快照標示為 `zypp(zypper)`。



4. 按一下顯示變更以開啓兩個快照之間不同的檔案清單。



5. 檢閱檔案清單。若要顯示檔案的前版本與後版本之間的「差異」，請從清單中選取它。



6. 要還原一個或多個檔案，請勾選相應的核取方塊來選取相關的檔案或目錄。按一下還原選定項目，然後按一下是確認該動作。

要還原單個檔案，請按一下其名稱以啓動差異比對檢視。按一下從第一個還原，然後按一下是確認您的選擇。

程序 7.2 使用 `snapper` 指令復原變更

1. 執行 `snapper list -t pre-post` 取得 YaST 和 Zypper 快照的清單。在描述欄中，YaST 快照標示為 `yast MODULE_NAME`；Zypper 快照標示為 `zypp (zypper)`。

```
tux > sudo snapper list -t pre-post
Pre # | Post # | Pre Date                          | Post Date                         | Description
------+--------+-----------------------------------+-----------------------------------+--------------
311   | 312    | Tue 06 May 2014 14:05:46 CEST | Tue 06 May 2014 14:05:52 CEST | zypp(y2base)
340   | 341    | Wed 07 May 2014 16:15:10 CEST | Wed 07 May 2014 16:15:16 CEST | zypp(zypper)
342   | 343    | Wed 07 May 2014 16:20:38 CEST | Wed 07 May 2014 16:20:42 CEST | zypp(y2base)
344   | 345    | Wed 07 May 2014 16:21:23 CEST | Wed 07 May 2014 16:21:24 CEST | zypp(zypper)
346   | 347    | Wed 07 May 2014 16:41:06 CEST | Wed 07 May 2014 16:41:10 CEST | zypp(y2base)
348   | 349    | Wed 07 May 2014 16:44:50 CEST | Wed 07 May 2014 16:44:53 CEST | zypp(y2base)
350   | 351    | Wed 07 May 2014 16:46:27 CEST | Wed 07 May 2014 16:46:38 CEST | zypp(y2base)
```

2. 使用下列指令取得快照對的已變更檔案清單：`snapper status 前..後`。含有內容變更的檔案以 `c` 標示，新增的檔案以 `+` 標示，刪除的檔案以 `-` 標示。

```
tux > sudo snapper status 350..351
+..... /usr/share/doc/packages/mikachan-fonts
+..... /usr/share/doc/packages/mikachan-fonts/COPYING
+..... /usr/share/doc/packages/mikachan-fonts/dl.html
```

```
c..... /usr/share/fonts/truetype/fonts.dir
c..... /usr/share/fonts/truetype/fonts.scale
+..... /usr/share/fonts/truetype/みかちゃん-p.ttf
+..... /usr/share/fonts/truetype/みかちゃん-pb.ttf
+..... /usr/share/fonts/truetype/みかちゃん-ps.ttf
+..... /usr/share/fonts/truetype/みかちゃん.ttf
c..... /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_64.cache-4
c..... /var/lib/rpm/Basenames
c..... /var/lib/rpm/Dirnames
c..... /var/lib/rpm/Group
c..... /var/lib/rpm/Installtid
c..... /var/lib/rpm/Name
c..... /var/lib/rpm/Packages
c..... /var/lib/rpm/Providename
c..... /var/lib/rpm/Requirename
c..... /var/lib/rpm/Sha1header
c..... /var/lib/rpm/Sigmd5
```

3. 若要顯示特定檔案的差異，請執行 `snapper diff` 前..後 檔案名稱。如果不指定檔案名稱，將會顯示所有檔案的差異。

```
tux > sudo snapper diff 350..351 /usr/share/fonts/truetype/fonts.scale
--- /.snapshots/350/snapshot/usr/share/fonts/truetype/fonts.scale       2014-04-23
 15:58:57.000000000 +0200
+++ /.snapshots/351/snapshot/usr/share/fonts/truetype/fonts.scale       2014-05-07
 16:46:31.000000000 +0200
@@ -1,4 +1,4 @@
-1174
+1486
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso10646-1
 ds=y:ai=0.2:luximr.ttf -b&h-luxi mono-bold-i-normal--0-0-0-0-c-0-iso8859-1
[...]
```

4. 若要還原一或多個檔案，請執行 `snapper -v undochange` 前..後 檔案名稱。如果不指定檔案名稱，將會還原所有已變更的檔案。

```
tux > sudo snapper -v undochange 350..351
     create:0 modify:13 delete:7
     undoing change...
     deleting /usr/share/doc/packages/mikachan-fonts
     deleting /usr/share/doc/packages/mikachan-fonts/COPYING
     deleting /usr/share/doc/packages/mikachan-fonts/dl.html
     deleting /usr/share/fonts/truetype/みかちゃん-p.ttf
     deleting /usr/share/fonts/truetype/みかちゃん-pb.ttf
     deleting /usr/share/fonts/truetype/みかちゃん-ps.ttf
     deleting /usr/share/fonts/truetype/みかちゃん.ttf
     modifying /usr/share/fonts/truetype/fonts.dir
     modifying /usr/share/fonts/truetype/fonts.scale
     modifying /var/cache/fontconfig/7ef2298fde41cc6eeb7af42e48b7d293-x86_64.cache-4
     modifying /var/lib/rpm/Basenames
```

```
    modifying /var/lib/rpm/Dirnames
    modifying /var/lib/rpm/Group
    modifying /var/lib/rpm/Installtid
    modifying /var/lib/rpm/Name
    modifying /var/lib/rpm/Packages
    modifying /var/lib/rpm/Providename
    modifying /var/lib/rpm/Requirename
    modifying /var/lib/rpm/Sha1header
    modifying /var/lib/rpm/Sigmd5
    undoing change done
```

警告：回復使用者新增

不建議透過使用 Snapper 回復變更的方式回復使用者新增。因為快照中排除了某些目錄，屬於這些使用者的檔案將保留在檔案系統中。如果使用已刪除使用者的使用者 ID 建立使用者，則此使用者將繼承這些檔案。因此，強烈建議您使用 YaST使用者和群組管理工具來移除使用者。

7.2.2 使用 Snapper 還原檔案

除了安裝與管理快照之外，Snapper 還會建立時間軸快照。您可以使用這些備份快照來還原不小心刪除的檔案或還原舊版檔案。利用 Snapper 的差異比對功能，您還可以瞭解在特定時間點執行了哪些修改。

還原檔案功能對於預設不會建立快照的子磁碟區或分割區上的資料尤其有用。例如，要從主目錄還原檔案，可以為自動建立時間軸快照的 `/home` 建立單獨的 Snapper 組態。如需指示，請參閱第 7.4 節 「建立和修改 Snapper 組態」。

警告：還原檔案與復原的比較

從根檔案系統（由 Snapper 的根組態定義）建立的快照可用於執行系統復原。建議您使用從快照開機然後執行復原的方式執行此復原。如需詳細資料，請參閱第 7.3 節 「透過從快照開機來執行系統復原」。

也可使用從根檔案系統快照還原所有檔案的方式執行復原（如下文所述）。但我們不建議採用這種方法。您可以還原單個檔案（如 `/etc` 目錄中的組態檔案），但不能從快照還原整份檔案清單中的檔案。

此限制僅影響從根檔案系統建立的快照！

程序 7.3 使用 YAST SNAPPER 模組還原檔案

1. 從 YaST 的其他區段或透過輸入 `yast2 snapper` 啓動Snapper模組。
2. 選擇要從中選擇快照的目前組態。
3. 選取要從中還原檔案的時間軸快照，並選擇顯示變更。時間軸快照屬於單一類型，以時間軸描述。
4. 按一下檔案名稱，從文字方塊中選取一個檔案。隨即顯示快照版本與目前系統之間的差異。勾選該核取方塊以選取要還原的檔案。對所有您要還原的檔案執行此操作。
5. 按一下還原選定項目，然後按一下是確認該動作。

程序 7.4 使用 `snapper` 指令還原檔案

1. 執行以下指令獲取特定組態之時間軸快照的清單：

   ```
   tux > sudo snapper -c CONFIG list -t single | grep timeline
   ```

   `CONFIG` 需要以現有的 Snapper 組態取代。使用 `snapper list-configs` 顯示清單。

2. 執行以下指令獲取指定快照之已變更檔案的清單：

   ```
   tux > sudo snapper -c CONFIG status SNAPSHOT_ID..0
   ```

   以您要從中還原檔案之快照的 ID 取代 `SNAPSHOT_ID`。

3. (選擇性)透過執行以下指令，列出目前檔案版本與快照中之檔案的差異

   ```
   tux > sudo snapper -c CONFIG diff SNAPSHOT_ID..0 FILE NAME
   ```

   如果您未指定 `<檔案名稱>`，則會顯示所有檔案的差異。

4. 要還原一或多個檔案，請執行

   ```
   tux > sudo snapper -c CONFIG -v undochange SNAPSHOT_ID..0 FILENAME1 FILENAME2
   ```

   如果未指定檔案名稱，將會還原所有已變更的檔案。

7.3 透過從快照開機來執行系統復原

SUSE Linux Enterprise Server 上包含的 GRUB 2 版本能夠從 Btrfs 快照開機。除 Snapper 的復原功能外，它還可復原設定有誤的系統。只有針對預設 Snapper 組態（`root`）建立的快照才可開機。

重要：支援的組態

從 SUSE Linux Enterprise Server 12 SP5 開始，僅當根分割區的預設子磁碟區組態未變更時，才支援系統復原。

將快照開機時，該快照中包含之檔案系統的部分會以唯讀模式掛接，而從快照中排除的所有其他檔案系統以及該檔案系統的排除部分會以讀寫模式掛接並且可修改。

重要：復原變更與復原的比較

使用快照還原資料時，您必須知道 Snapper 可以處理兩種完全不同的案例。

復原變更
: 如第 7.2 節 「使用 Snapper 復原變更」中所述復原變更時，系統會比較兩個快照並回復這兩個快照之間的變更。使用此方法還可以將選取的檔案明確排除在還原之外。

復原
: 如下文所述執行復原時，系統會重設回建立快照當時的狀態。

若要從可開機快照執行復原，必須符合以下要求。執行預設安裝時，系統會進行相應設定。

從可開機快照執行復原的要求

- 根檔案系統必須是 Btrfs。不支援從 LVM 磁碟區快照開機。
- 根檔案系統必須位於單一裝置、單一分割區以及單一子磁碟區上。如 `/srv` 之類從快照中排除的目錄（如需完整的清單，請參閱第 7.1.2 節 「從快照中排除的目錄」）可以位於單獨的分割區上。
- 系統必須能透過已安裝的開機載入程式開機。

要從可開機的快照執行復原，請按如下步驟執行：

1. 將系統開機。在開機功能表中，選擇可開機快照並選取您要開機的快照。系統會按日期列出快照清單，最近的快照最先列出。
2. 登入系統。仔細檢查是否一切都如預期般運作。請注意，您無法對快照中包含的任何目錄執行寫入。但無論您接下來執行什麼操作，您寫入到其他目錄的資料都不會遺失。
3. 根據您是否要執行復原操作，選擇下一步動作：

   a. 如果您不想對目前狀態的系統執行復原，請重新開機進入目前的系統狀態。然後，您便可選擇另一個快照，或是啓動救援系統。

   b. 若要進行復原，請執行

   ```
   tux > sudo snapper rollback
   ```

   並在之後重新開機。在開機螢幕上，選擇預設開機項目以重新開機到已恢復的系統。系統即會建立復原前檔案系統狀態的快照。一個全新的讀寫快照即會取代根的預設子磁碟區。如需詳細資料，請參閱第 7.3.1 節 「復原後的快照」。

   透過 `-d` 選項新增快照的描述非常實用。例如：

   ```
   New file system root since rollback on DATE TIME
   ```

提示：復原至特定的安裝狀態

如果安裝期間未停用快照，將在初始系統安裝結束時建立初始可開機快照。您隨時可以透過將此快照開機，返回到該狀態。**安裝後** 可依描述識別該快照。

開始對 Service Pack 或新的主要版本進行系統升級時，也會建立可開機快照（前提是快照未停用）。

7.3.1 復原後的快照

在執行復原之前，將會建立一個執行中檔案系統的快照。快照描述會參考復原中所還原快照的 ID。

對於透過復原建立的快照，其 `Cleanup` 屬性的值會設為 `number`。因此，復原快照會在達到設定的快照數後自動刪除。如需詳細資訊，請參閱第 7.6 節 「自動快照清理」。如果快照包含重要資料，請在系統移除快照之前從快照中擷取資料。

7.3.1.1 復原快照範例

例如，在全新安裝之後，系統上存在以下可用的快照：

```
root # snapper --iso list
Type   | # |     | Cleanup | Description           | Userdata
-------+---+ ... +---------+-----------------------+-------------
single | 0 |     |         | current               |
single | 1 |     |         | first root filesystem |
single | 2 |     | number  | after installation    | important=yes
```

執行 `sudo snapper rollback` 之後，將會建立快照 `3`，它包含執行復原前系統的狀態。快照 `4` 是新的預設 Btrfs 子磁碟區，因此是重新開機之後的系統。

```
root # snapper --iso list
Type   | # |     | Cleanup | Description           | Userdata
-------+---+ ... +---------+-----------------------+-------------
single | 0 |     |         | current               |
single | 1 |     | number  | first root filesystem |
single | 2 |     | number  | after installation    | important=yes
single | 3 |     | number  | rollback backup of #1 | important=yes
single | 4 |     |         |                       |
```

7.3.2 存取和識別快照開機項目

若要從快照開機，請重新開機並選擇從唯讀快照啓動開機載入程式。此時會開啓一個螢幕，其中列出了所有可開機的快照。最近的快照列在最前面，最舊的快照列在最後面。使用 ↓ 和 ↑ 導覽，然後按 `Enter` 啓動所選的快照。從開機功能表啓動快照不會立即重新開機，而是開啓所選快照的開機載入程式。

圖形 7.1 開機載入程式：快照

開機載入程式中的每個快照項目遵循一種可方便您識別快照的命名規劃：

```
[*]❶OS❷ (KERNEL❸,DATE❹TTIME❺,DESCRIPTION❻)
```

❶ 如果快照標示為**重要**，該項將帶有 `*` 標記。

❷ 作業系統標籤。

❹ 採用 `YYYY-MM-DD` 格式的日期。

❺ 採用 `HH:MM` 格式的時間。

❻ 此欄位包含快照的描述。對於手動建立的快照，這是使用選項 `--description` 建立的字串，或自訂字串（請參閱提示：為開機載入程式快照項目設定自訂描述）。對於自動建立的快照，這是呼叫的工具，例如 `zypp(zypper)` 或 `yast_sw_single`。根據開機螢幕的大小，可能會截斷較長的描述。

提示：為開機載入程式快照項目設定自訂描述

可以用自訂字串來取代快照描述欄位中的預設字串。例如，如果自動建立的描述不能充分描述快照，或者使用者提供的描述太長，則這種做法會很有用。若要為快照*編號*設定自訂字串*字串*，請使用以下指令：

```
tux > sudo snapper modify --userdata "bootloader=STRING" NUMBER
```

描述的長度不應超過 25 個字元 – 超過此大小的內容無法在開機畫面上正常顯示。

7.3.3 限制

無法進行完整系統復原，即將整個系統還原到與建立快照時完全相同的狀態。

7.3.3.1 從快照中排除的目錄

根檔案系統快照並不包含所有目錄。如需詳細資料和原因，請參閱第 7.1.2 節 「從快照中排除的目錄」。因此，這些目錄中的資料不會還原，也就形成以下限制。

執行復原之後，附加產品和協力廠商軟體可能回無法使用。

如果從快照中排除的子磁碟區（如 `/opt`）中的應用程式還有其他部分資料安裝在該快照中包含的子磁碟區中，則應用程式及附加產品的安裝資料在復原後可能無法使用。重新安裝應用程式或附加產品可解決此問題。

檔案存取問題

如果應用程式在快照以及目前系統之間變更了檔案許可權和/或擁有權，則復原後該應用程式可能無法存取這些檔案。請在復原之後重設受影響檔案的許可權和/或擁有權。

不相容的資料格式

如果服務或應用程式在快照和目前系統之間建立了新的資料格式，則復原之後該應用程式可能無法讀取受影響的資料檔案。

混合代碼和資料的子磁碟區

`/srv` 等子磁碟區可能同時包含代碼和資料。復原可能導致代碼不起作用。例如，降級 PHP 版本可能導致該 Web 伺服器之 PHP 程序檔損毀。

使用者資料

如果復原從系統移除了使用者，但這些使用者所擁有的資料存在於快照排除的目錄中，則這些資料不會予以移除。如果使用相同使用者 ID 建立使用者，則此使用者將繼承這些檔案。使用 `find` 等工具尋找並移除孤立的檔案。

7.3.3.2 不復原開機載入程式資料

無法復原開機載入程式，因為開機載入程式的所有「階段」必須組合在一起共同作用。而執行 `/boot` 復原則無法保證這一要求。

7.4 建立和修改 Snapper 組態

每個分割區或 `Btrfs` 子磁碟區都有一個專用的組態檔案用於定義 Snapper 的行為方式。這些組態檔案位於 `/etc/snapper/configs/` 下。

如果根檔案系統足夠大（大約有 12 GB），安裝時將自動對根檔案系統 `/` 啓用快照。相應的預設組態命名為 `root`。該組態可建立和管理 YaST 及 Zypper 快照。如需預設值清單，請參閱第 7.4.1.1 節 「組態資料」。

注意：啓用快照所需的最小根檔案系統大小

如第 7.1 節 「預設設定」中所述，若要啓用快照，根檔案系統中需要有額外的可用空間。所需空間取決於所安裝的套件數量以及快照中包括的磁碟區變更量，另外還取決於快照頻率和歸檔的快照數。

若要在安裝期間自動啓用快照，需要符合最小根檔案系統大小。此大小約為 12 GB。將來，依據基礎系統的架構和大小的不同，此值可能會發生變化。它取決於安裝媒體內 `/control.xml` 檔案中以下標記的值：

```
<root_base_size>
<btrfs_increase_percentage>
```

該值透過下面的公式計算得出：`ROOT_BASE_SIZE` * (1 + `BTRFS_INCREASE_PERCENTAGE`/100)

請記住，此值是最小大小。請考慮分給根檔案系統更多空間。一般而言，兩倍於未啓用快照時所需的大小即可。

您可以為其他格式為 `Btrfs` 的分割區或 `Btrfs` 分割區上的現有子磁碟區建立您自己的組態。在下面的範例中，我們將設定 Snapper 組態以用於備份掛接於 `/srv/www` 且採用 `Btrfs` 格式之獨立分割區上的 Web 伺服器資料。

建立組態後，您可以使用 `snapper` 自身或 YaST Snapper 模組從這些快照中還原檔案。在 YaST 中，您需要選取目前組態，同時還需要使用全域參數 `-c` 指定 `snapper` 的組態（例如 `snapper -c myconfig list`）。

若要建立新的 Snapper 組態，請執行 `snapper create-config`：

```
tux > sudo snapper -c www-data❶ create-config /srv/www❷
```

❶ 組態檔案的名稱。

❷ 要建立快照之分割區或 `Btrfs` 子磁碟區的掛接點。

此指令將會使用 `/etc/snapper/config-templates/default` 中的合理預設值建立一個新的組態檔案 `/etc/snapper/configs/www-data`。如需如何調整這些預設值的指示，請參閱第 7.4.1 節 「管理現有的組態」。

提示：組態預設值

新組態的預設值取自 `/etc/snapper/config-templates/default`。若要使用您自己的一組預設值，請在同一目錄中建立此檔案的副本，然後根據您的需求進行調整。若要使用該副本，請使用 `create-config` 指令指定 `-t` 選項：

```
tux > sudo snapper -c www-data create-config -t MY_DEFAULTS /srv/www
```

7.4.1 管理現有的組態

`snapper` 提供多個用於管理現有組態的子指令。您可以列出、顯示、刪除及修改它們：

列出組態

使用指令 `snapper list-configs` 可獲取所有現有組態：

```
tux > sudo snapper list-configs
Config | Subvolume
-------+----------
root   | /
usr    | /usr
local  | /local
```

顯示組態

使用子指令 `snapper -c CONFIG get-config` 可顯示指定的組態。`Config` 需要使用 `snapper list-configs` 所顯示的組態名稱取代。如需有關組態選項的詳細資訊，請參閱第 7.4.1.1 節 「組態資料」。

若要顯示預設組態，請執行

```
tux > sudo snapper -c root get-config
```

修改組態

使用子指令 `snapper -c CONFIG set-config OPTION=VALUE` 可修改指定組態中的選項。`Config` 需要使用 `snapper list-configs` 所顯示的組態名稱取代。`OPTION` 及 `VALUE` 的可能的值列於第 7.4.1.1 節 「組態資料」中。

刪除組態

使用子指令 `snapper -c CONFIG delete-config` 可刪除組態。`Config` 需要使用 `snapper list-configs` 所顯示的組態名稱取代。

7.4.1.1 組態資料

每個組態都包含可以從指令行修改的選項清單。以下清單提供每個選項的詳細資料。若要變更某個值，請執行 `snapper -c 組態 set-config "索引鍵=值"`。

`ALLOW_GROUPS` 、 `ALLOW_USERS`

授予一般使用者快照的使用權限。如需相關資訊，請參閱第 7.4.1.2 節 「以一般使用者身分使用 Snapper」。

預設值為 `""`。

`BACKGROUND_COMPARISON`

定義建立前後快照後是否應在背景中對它們進行比較。

預設值為 `"yes"`。

`EMPTY_*`

為包含相同「前」快照和「後」快照的快照組定義清理演算法。如需詳細資料，請參閱第 7.6.3 節 「清理無差異的快照組」。

`FSTYPE`

分割區的檔案系統類型。請勿進行變更。

預設值為 "btrfs"。

NUMBER_*
: 為安裝快照與管理快照定義清理演算法。如需詳細資料，請參閱第 7.6.1 節 「清理編號快照」。

QGROUP / SPACE_LIMIT
: 將定額支援新增至清理演算法。如需詳細資料，請參閱第 7.6.5 節 「新增磁碟定額支援」。

SUBVOLUME
: 要建立快照之分割區或子磁碟區的掛裝點。請勿進行變更。
預設值為 "/".

SYNC_ACL
: 如果一般使用者要使用 Snapper（請參閱第 7.4.1.2 節 「以一般使用者身分使用 Snapper」），他們必須能存取 .snapshot 目錄，並且能讀取其中的檔案。如果 SYNC_ACL 設定為 yes，Snapper 會使用 ACL 自動允許 ALLOW_USERS 或 ALLOW_GROUPS 項目中的使用者和群組存取該目錄及其中的檔案。
預設值為 "no"。

TIMELINE_CREATE
: 如果設定為 yes，則會每小時建立一個快照。有效值：yes 和 no.
預設值為 "no"。

TIMELINE_CLEANUP / TIMELINE_LIMIT_*
: 為時間軸快照定義清理演算法。如需詳細資料，請參閱第 7.6.2 節 「清理時間軸快照」。

7.4.1.2 以一般使用者身分使用 Snapper

依預設，Snapper 只能由 root 使用者使用。但是，在下列情況中，某些群組或使用者需要能夠建立快照或透過回復至快照來復原變更：

- 想要建立 /srv/www 快照的網站管理員
- 想要建立其主目錄快照的使用者

為了以上這些目的，您可以建立用於授予使用者或/和群組權限的 Snapper 組態。相應的 `.snapshots` 目錄必須可由指定的使用者讀取和存取。實現這一目的最簡單的方法是將 SYNC_ACL 選項設定為 `yes`。

程序 7.5 讓一般使用者可以使用 SNAPPER

請注意，此程序中的所有步驟都必須由 `root` 使用者執行。

1. 若不存在，請為使用者應該能夠在其上使用 Snapper 的分割區或子磁碟區建立 Snapper 組態。如需指示，請參閱第 7.4 節 「建立和修改 Snapper 組態」。範例：

   ```
   tux > sudo snapper --config web_data create /srv/www
   ```

2. 組態檔案將在 `/etc/snapper/configs/CONFIG` 下建立，其中 CONFIG 是您在上一步中使用 `-c/--config` 指定的值（如 `/etc/snapper/configs/web_data`）。根據您的需求調整該值；如需詳細資料，請參閱第 7.4.1 節 「管理現有的組態」。

3. 設定 `ALLOW_USERS` 和/或 `ALLOW_GROUPS` 的值，以分別向使用者和/或群組授予權限。多個項目需要用 `Space` 加以分隔。例如，若要向使用者 `www_admin` 授予權限，請執行：

   ```
   tux > sudo snapper -c web_data set-config "ALLOW_USERS=www_admin" SYNC_ACL="yes"
   ```

4. 現在，指定的使用者和/或群組便可以使用給定的 Snapper 組態。您可以使用 `list` 指令進行測試，例如：

   ```
   www_admin:~ > snapper -c web_data list
   ```

7.5 手動建立和管理快照

Snapper 的功能並不僅限於依照組態自動建立和管理快照；您還可以使用指令行工具或 YaST 模組手動建立快照組（「前快照與後快照」）或單一快照。

所有 Snapper 操作都會對現有組態進行（如需詳細資料，請參閱第 7.4 節 「建立和修改 Snapper 組態」）。您只能為存在組態的分割區或磁碟區建立快照。預設會使用系統組態（根）。如果您要為您自己的組態建立或管理快照，則必須明確選擇它。使用 YaST 中的目前組態下拉式方塊，或在指令行上指定 `-c` 選項（`snapper -c MYCONFIG COMMAND`）。

7.5.1 快照中繼資料

每個快照都包含快照本身和一些中繼資料。建立快照時，您還需要指定中繼資料。修改快照即表示變更其中繼資料 — 您無法修改其內容。使用 `snapper list` 顯示現有快照及其中繼資料：

`snapper --config home list`
: 列出組態 `home` 的快照。若要列出預設組態（root）的快照，請使用 `snapper -c root list` 或 `snapper list`。

`snapper list -a`
: 列出所有現有組態的快照。

`snapper list -t pre-post`
: 列出預設（`root`）組態的所有「前」快照與「後」快照組。

`snapper list -t single`
: 列出預設（`root`）組態的所有 `single` 類型快照。

下列中繼資料適用於每個快照：

- 類型：快照類型，請參閱第 7.5.1.1 節 「快照類型」 以取得詳細資料。此資料無法變更。
- 編號：快照的唯一編號。此資料無法變更。
- 前快照編號：指定相應前快照的編號。僅適用於類型為後的快照。此資料無法變更。
- 描述：快照的描述。
- 使用者資料：延伸描述，您可在其中以逗號分隔之「鍵=值」清單的形式指定自訂資料：`reason=testing, project=foo`。此欄位還用於將快照標示為重要（`important=yes`）並列出建立該快照的使用者（user=tux）。
- 清理演算法：快照的清理演算法，請參閱第 7.6 節 「自動快照清理」 以取得詳細資料。

7.5.1.1 快照類型

Snapper 知道三種不同類型的快照：前、後與單一。實際上，它們並無差異，但是 Snapper 會以不同的方式處理它們。

前
: 修改前檔案系統的快照。每個**前**快照都對應一個**後**快照。例如，用於自動建立 YaST/Zypper 快照。

後
: 修改後檔案系統的快照。每個**後**快照都對應一個**前**快照。例如，用於自動建立 YaST/Zypper 快照。

單一
: 獨立快照。例如，用於自動按小時建立快照。這是建立快照時的預設類型。

7.5.1.2 清理演算法

Snapper 提供了三種用於清理舊快照的演算法。cron 日常工作中會執行這些演算法。可以定義要在 Snapper 組態中保留的不同類型的快照數量（如需詳細資料，請參閱第 7.4.1 節 「管理現有的組態」）。

數量
: 當達到特定的快照計數時刪除舊快照。

時間軸
: 刪除經過特定期限的舊快照，但會保留若干每小時、每日、每月和每年快照。

空-前-後
: 刪除無差異的前/後快照對。

7.5.2 建立快照

可透過執行 `snapper create` 或在 YaST 模組 Snapper 中按一下建立建立快照。下列範例說明如何從指令行建立快照。透過 YaST 介面可便於採用這兩種建立方法。

提示：快照描述

您應始終指定有意義的描述，以便日後能夠識別其用途。透過使用者資料選項可以指定更多資訊。

`snappercreate --description "2014 年第 2 週的快照"`

為預設（根）組態建立獨立快照（類型為單一）並提供描述。因為未指定清理演算法，所以一律不自動刪除快照。

`snapper --config home create --description "在 ~tux 中清理"`

為名為 home 的自訂組態建立獨立快照（類型為單一）並提供描述。因為未指定清理演算法，所以一律不自動刪除快照。

`snapper --config home create --description "每日資料備份" --cleanup-algorithm timeline`>

為名為 home 的自訂組態建立獨立快照（類型為單一）並提供描述。當檔案符合特定於組態中時間軸清理演算法的準則時，將會自動刪除該檔案。

`snapper create --type pre--print-number--description "在 Apache 組態清理之前"--userdata "important=yes"`

建立類型為前的快照並列印快照編號。需要第一個指令才能建立用於儲存「前」和「後」狀態的快照對。快照會標示為重要。

`snapper create --type post--pre-number 30--description "在 Apache 組態清理之前"--userdata "important=yes"`

建立類型為後並與前快照編號 30 配對的快照。需要第二個指令才能建立用於儲存「前」和「後」狀態的快照對。快照會標示為重要。

`snapper create --command COMMAND--description "在指令前後"`

在執行指令前後自動建立快照對。僅當在指令行上使用 snapper 時，此選項才可用。

7.5.3 修改快照中繼資料

Snapper 可讓您修改快照的描述、清理演算法和使用者資料，所有其他中繼資料則無法變更。下列範例說明如何從指令行修改快照。透過 YaST 介面可便於採用這兩種建立方法。

若要透過指令行修改快照，您需要知道其編號。使用 `snapper list` 可以顯示所有快照及其編號。

YaST Snapper 模組已列出所有快照。從清單中選擇一個快照，然後按一下修改。

`snapper modify --cleanup-algorithm "時間軸"` 10

修改預設（**根**）組態之快照 10 的中繼資料。清理演算法設為**時間軸**。

`snapper --config home modify --description "每日備份" -cleanup-algorithm "時間軸"120`

修改名為 `home` 的自訂組態之快照 120 的中繼資料。將會設定新的描述並取消設定清理演算法。

7.5.4 刪除快照

若要使用 YaST Snapper 模組刪除快照，請從清單中選擇快照，然後按一下刪除。

若要使用指令行工具刪除快照，您需要知道其編號。請透過執行 `snapper list` 來取得該編號。若要刪除快照，請執行 `snapper delete` *編號*。

不允許刪除目前的預設子磁碟區快照。

使用 Snapper 刪除快照時，在背景中執行的 Btrfs 程序將會回收已釋放的空間。因此，可用空間的可見度與可用性將會延遲。如果您希望在刪除快照後立即可以使用釋放的空間，請結合選項 `--sync` 使用 delete 指令。

提示：刪除快照對

刪除**前**快照時，您應一律刪除其對應的**後**快照（反之亦然）。

`snapper delete 65`

刪除預設（**根**）組態的快照 65。

`snapper -c home delete 89 90`

刪除名為 `home` 之自訂組態的快照 89 和 90。

`snapper delete --sync 23`

刪除預設（**根**）組態的快照 23，並使釋放的空間立即可用。

提示：刪除未參考的快照

有時，雖然 Btrfs 快照存在，但卻缺少包含 Snapper 中繼資料的 XML 檔案。這種情況表示快照對 Snapper 不可見，需要手動刪除該快照：

```
btrfs subvolume delete /.snapshots/SNAPSHOTNUMBER/snapshot
rm -rf /.snapshots/SNAPSHOTNUMBER
```

提示：舊快照佔用較多磁碟空間

如果您要刪除快照以釋放硬碟上的空間，請確定先刪除舊快照。快照越舊，它佔用的磁碟空間就越多。

也可以透過 cron 日常工作自動刪除快照。如需詳細資訊，請參閱第 7.5.1.2 節 「清理演算法」。

7.6 自動快照清理

快照會佔用磁碟空間，一段時間後，快照可能會佔用大量的磁碟空間。為了防止磁碟上的空間耗盡，Snapper 提供了演算法用於自動刪除舊快照。這些演算法依據時間軸快照和編號快照（管理快照與安裝快照組）而異。您可以指定要為每種類型保留的快照數量。

除此之外，您可以指定一個快照空間定額，用於定義快照可佔用的最大磁碟空間量。系統還可以自動刪除無相異的「前」快照與「後」快照組。

清理演算法一律系繫結到單一 Snapper 組態，因此您需要為每個組態指定演算法。若要防止自動刪除特定的快照，請參閱問：。

預設設定（root）設定為清理編號快照以及空的「前」快照與「後」快照組。已啓用定額支援 - 快照佔用的空間不可超過根分割區可用磁碟空間的 50%。時間軸快照預設為停用，因此，時間軸清理演算法亦已停用。

7.6.1 清理編號快照

Snapper 組態的以下參數控制編號快照（管理快照與安裝快照組）的清理。

NUMBER_CLEANUP

啓用或停用安裝快照與管理快照組的清理。如果已啓用，則在快照總數超過 NUMBER_LIMIT 和/或 NUMBER_LIMIT_IMPORTANT 指定的數字，以及 NUMBER_MIN_AGE 指定的時限時，會刪除快照組。有效值：yes（啓用）、no（停用）。

預設值為 "yes"。

用於變更或設定值的範例指令：

```
tux > sudo snapper -c CONFIG set-config "NUMBER_CLEANUP=no"
```

NUMBER_LIMIT / NUMBER_LIMIT_IMPORTANT

定義要保留多少個普通和/或重要安裝快照與管理快照組。只會保留最新的快照。如果 NUMBER_CLEANUP 設定為 "no"，則忽略此參數。

NUMBER_LIMIT 的預設值為 "2-10"，NUMBER_LIMIT_IMPORTANT 的預設值為 "4-10"。

用於變更或設定值的範例指令：

```
tux > sudo snapper -c CONFIG set-config "NUMBER_LIMIT=10"
```

重要：範圍值與常數值的比較

如果已啓用定額支援（請參閱第 7.6.5 節 「新增磁碟定額支援」），則需要將限制指定為最小值-最大值範圍，例如 2-10。如果已停用定額支援，則需要提供常數值，例如 10，否則清理將會失敗並出現錯誤。

NUMBER_MIN_AGE

定義快照在可供自動刪除之前的最短保留期限（秒）。期限小於此處指定值的快照不會刪除，不論存在多少個這樣的快照。

預設值為 "1800"。

用於變更或設定值的範例指令：

```
tux > sudo snapper -c CONFIG set-config "NUMBER_MIN_AGE=864000"
```

注意：限制和期限

NUMBER_LIMIT、NUMBER_LIMIT_IMPORTANT 和 NUMBER_MIN_AGE 始終會予以評估。只有在滿足全部條件時，才會刪除快照。

如果您希望總是保留 NUMBER_LIMIT* 所定義數量的快照而不考慮其期限，請將 NUMBER_MIN_AGE 設定為 0。

下面的範例顯示了保留最近 10 個重要和 10 個普通快照（不論保留期限）的組態：

```
NUMBER_CLEANUP=yes
NUMBER_LIMIT_IMPORTANT=10
NUMBER_LIMIT=10
NUMBER_MIN_AGE=0
```

不過，如果您不想保留超過特定期限的快照，請將 NUMBER_LIMIT* 設定為 0，並使用 NUMBER_MIN_AGE 提供期限。

下面的範例顯示了只保留十天以內的快照的組態：

```
NUMBER_CLEANUP=yes
NUMBER_LIMIT_IMPORTANT=0
NUMBER_LIMIT=0
NUMBER_MIN_AGE=864000
```

7.6.2 清理時間軸快照

Snapper 組態的以下參數控制時間軸快照的清理。

TIMELINE_CLEANUP

啓用或停用時間軸快照的清理。如果已啓用，則在快照總數超過 TIMELINE_LIMIT_* 指定的數量以及 TIMELINE_MIN_AGE 指定的期限時，會刪除快照。有效值：yes 和 no.

預設值為 "yes"。

用於變更或設定值的範例指令：

```
tux > sudo snapper -c CONFIG set-config "TIMELINE_CLEANUP=yes"
```

TIMELINE_LIMIT_DAILY、TIMELINE_LIMIT_HOURLY、TIMELINE_LIMIT_MONTHLY、TIMELINE_LIMIT_WEEKLY、TIMELINE_LIMIT_YEARLY

要以小時、天、月、週和年保留的快照數量。

每個項目的預設值為 "10"，但 TIMELINE_LIMIT_WEEKLY 除外（預設為 "0"）。

`TIMELINE_MIN_AGE`

定義快照在可供自動刪除之前的最短保留期限（秒）。

預設值為 `"1800"`。

範例 7.1 時間軸組態範例

```
TIMELINE_CLEANUP="yes"
TIMELINE_CREATE="yes"
TIMELINE_LIMIT_DAILY="7"
TIMELINE_LIMIT_HOURLY="24"
TIMELINE_LIMIT_MONTHLY="12"
TIMELINE_LIMIT_WEEKLY="4"
TIMELINE_LIMIT_YEARLY="2"
TIMELINE_MIN_AGE="1800"
```

此組態範例啓用了會自動清理的每小時快照。系統始終會同時評估 `TIMELINE_MIN_AGE` 和 `TIMELINE_LIMIT_*`。在此範例中，快照在可供刪除之前的最短期限設定為 30 分鐘（1800 秒）。因為我們建立了每小時快照，所以這可確保僅保留最新的快照。如果將 `TIMELINE_LIMIT_DAILY` 設為非零值，這表示同時會保留當天的第一個快照。

要保留的快照

- 每小時：已建立的最後 24 個快照。
- 每日：保留最近 7 天內建立的第一個每日快照。
- 每月：保留最近 12 個月內當月最後一天建立的第一個快照。
- 每週：保留最近 4 週內當週最後一天建立的第一個快照。
- 每年：保留最近 2 年內當年最後一天建立的第一個快照。

7.6.3 清理無差異的快照組

如第 7.1.1 節 「快照類型」中所述，每當您執行 YaST 模組或執行 Zypper 時，會在啓動時建立一個「前」快照，在結束時建立一個「後」快照。如果您做出任何變更，則「前」快照與「後」快照沒有差異。在 Snapper 組態中設定以下參數可自動刪除此類「空白」快照組：

`EMPTY_PRE_POST_CLEANUP`

如果設定為 `yes`，將會刪除前後快照相同的快照對。

預設值為 `"yes"`。

`EMPTY_PRE_POST_MIN_AGE`

定義前後快照相同的快照對在自動刪除之前必須保留的最短期限（以秒為單位）。預設值為 `"1800"`。

7.6.4 清理手動建立的快照

Snapper 未針對手動建立的快照提供自訂清理演算法。但是，您可以向手動建立的快照指定 number 或 timeline 清理演算法。如果您這麼做，該快照會加入所指定演算法的「清理佇列」。可以在建立快照時，或者透過修改現有快照來指定清理演算法：

`snapper create --description "Test" --cleanup-algorithm number`

為預設（root）組態建立獨立快照（單一類型）並指定 `number` 清理演算法。

`snapper modify --cleanup-algorithm "時間軸" 25`

使用數字 25 修改快照，並指定清理演算法 `timeline`。

7.6.5 新增磁碟定額支援

除了上述 number 和/或 timeline 清理演算法之外，Snapper 還支援定額。您可以定義允許快照佔用的可用空間百分比。此百分比值一律套用至相應 Snapper 組態中定義的 Btrfs 子磁碟區。

如果在安裝期間啓用了 Snapper，則會自動啓用定額支援。如果在安裝後的某個時間手動啓用 Snapper，則可以透過執行 `snapper setup-quota` 來啓用定額支援。這需要提供有效的組態（如需詳細資訊，請參閱第 7.4 節 「建立和修改 Snapper 組態」）。

定額支援由 Snapper 組態的以下參數控制。

`QGROUP`

Snapper 使用的 Btrfs 定額群組。如果未設定，請執行 `snapper setup-quota`。如果已設定，則僅當您熟悉 `man 8 btrfs-qgroup` 時方可對其進行變更。此值是使用 `snapper setup-quota` 設定的，請勿變更。

SPACE_LIMIT

允許快照使用的空間限制，以 1（100%）的分數表示。有效值範圍為 0 到 1（0.1 = 10%，0.2 = 20%...）。

需遵循以下限制和指導方針：

- 只能在已啓用現有 number 和/或 timeline 清理演算法的前提下才能啓用定額。如果未啓用任何清理演算法，則無法套用定額限制。
- 啓用定額支援後，Snapper 會視需要執行兩輪清理。第一輪清理套用針對編號快照和時間軸快照指定的規則。僅當完成這一輪清理後超出定額時，才會在第二輪清理中套用定額特定的規則。
- 即使已啓用定額支援，Snapper 也永遠會保留 NUMBER_LIMIT* 和 TIMELINE_LIMIT* 值指定的快照數量，而不論是否超出了定額。因此，建議為 NUMBER_LIMIT* 和 TIMELINE_LIMIT* 指定範圍值（MIN-MAX），以確定可以套用定額。

  例如，如果設定了 NUMBER_LIMIT=5-20，Snapper 將執行第一輪清理，並將普通的編號快照數量減至 20 個。如果這 20 個快照超出定額，Snapper 將在第二輪清理中刪除最舊的快照，直到符合定額限制。永遠會保留最少五個快照，不論這些快照佔用了多少空間。

7.7 常見問題解答

問： 為何 Snapper 從不顯示 /var/log、/tmp 及其他目錄中的變更?

答： 對於我們確定要從快照中排除的某些目錄，請參閱第 7.1.2 節 「從快照中排除的目錄」獲取清單及排除原因。為了將某路徑從快照中排除，我們為該路徑建立了子磁碟區。

問： 快照使用了多少磁碟空間? 如何釋放磁碟空間?

答： 目前，Btrfs 工具無法顯示快照配置的磁碟空間量。但是，如果您啓用了定額，則可以判斷在刪除所有快照後可以釋放多少空間：

1. 取得定額群組 ID（在以下範例中為 1/0）：

```
tux > sudo snapper -c root get-config | grep QGROUP
```

```
QGROUP                  | 1/0
```

2. 重新掃描子磁碟區定額：

```
tux > sudo btrfs quota rescan -w /
```

3. 顯示定額群組（在以下範例中為 `1/0`）的資料：

```
tux > sudo btrfs qgroup show / | grep "1/0"
1/0           4.80GiB    108.82MiB
```

   第三欄顯示刪除所有快照時可釋放的空間數量（`108.82MiB`）。

為了釋放包含快照之 `Btrfs` 分割區上的空間，您需要刪除不需要的快照，而不是檔案。與新快照相比，較舊的快照佔用的空間更多。如需詳細資料，請參閱第 7.1.3.4 節 「控制快照歸檔」。

從一個 Service Pack 升級到另一個 Service Pack 會導致快照佔用系統子磁碟區上的大量磁碟空間，這是因為有許多資料進行了變更（套件更新）。對於不再需要的快照，建議您手動將其刪除。如需詳細資料，請參閱第 7.5.4 節 「刪除快照」。

問： 能否從開機載入程式將快照開機?

答： 能，請參閱第 7.3 節 「透過從快照開機來執行系統復原」獲取詳細資訊。

問： 如何永久保留快照?

答： 目前，Snapper 無法防止手動刪除快照。但是，您可以防止清理演算法自動刪除快照。除非您使用 第 7.5.2 節 「建立快照」--cleanup-algorithm **指定清理演算法，否則不會為手動建立的快照（請參閱）**指定清理演算法。總是為自動建立的快照指定 `number` 或 `timeline` 演算法。若要從一或多個快照中移除此類指定，請執行以下步驟：

1. 列出所有可用的快照：

```
tux > sudo snapper list -a
```

2. 記住您要防止刪除的快照數。

3. 執行以下指令，並用您記住的數字取代數字預留位置：

```
tux > sudo snapper modify --cleanup-algorithm "" #1 #2 #n
```

4. 再次執行 `snapper list -a` 來檢查結果。在 `Cleanup` 欄中，與修改的快照相對應的項目現在應該為空白。

問: 何處可以取得有關 Snapper 的詳細資訊?

答: 請造訪 Snapper 首頁，網址為 http://snapper.io/。

8 透過 VNC 進行遠端存取

虛擬網路計算（VNC）可讓您透過圖形桌面環境控制遠端電腦（與遠端外圍程序存取相對）。VNC 獨立於平台，可讓您從任何作業系統存取遠端機器。

SUSE Linux Enterprise Server 支援兩種不同類型的 VNC 工作階段：一次性工作階段，只要用戶端 VNC 連接不中斷，該工作階段將一直處於「作用中」；永久工作階段，此工作階段將一直處於「作用中」，除非明確將其終止。

注意：工作階段類型

機器可以在不同的連接埠上同時提供這兩種工作階段，但是開啓的工作階段無法從一種類型轉換為另一種類型。

8.1 `vncviewer` 用戶端

若要連接到伺服器提供的 VNC 服務，需要使用一個用戶端。SUSE Linux Enterprise Server 中的預設用戶端是 `tigervnc` 套件提供的 `vncviewer`。

8.1.1 使用 vncviewer CLI 進行連接

若要啓動 VNC 查看器並發起與伺服器的工作階段，請使用以下指令：

```
tux > vncviewer jupiter.example.com:1
```

您也可以指定含兩個冒號的連接埠號碼，而不指定 VNC 顯示埠號碼：

```
tux > vncviewer jupiter.example.com::5901
```

注意：顯示號碼和連接埠號碼

您在 VNC 用戶端中指定的實際顯示號碼或連接埠號碼必須與在目標機器上透過 `vncserver` 指令選取的顯示號碼或連接埠號碼相同。如需更多資訊，請參閱第 8.4 節 「永久 VNC 工作階段」。

8.1.2 使用 vncviewer GUI 進行連接

執行 `vncviewer` 且不指定 `--listen` 或要連接到的主機會顯示一個視窗，要求您輸入連接詳細資料。按第 8.1.1 節 「使用 vncviewer CLI 進行連接」中所述在 VNC 伺服器欄位中輸入主機，然後按一下連接。

圖形 8.1 VNCVIEWER

8.1.3 連接未加密通知

VNC 協定支援不同類型的加密連接，請不要將這些連接與密碼驗證相混淆。如果某個連接未使用 TLS，VNC 檢視器的視窗標題中可能會出現 「(連接未加密！)」 文字。

8.2 Remmina：遠端桌面用戶端

Remmina 是功能豐富的新式遠端桌面用戶端。它支援多種存取方法，例如 VNC、SSH、RDP 或 Spice。

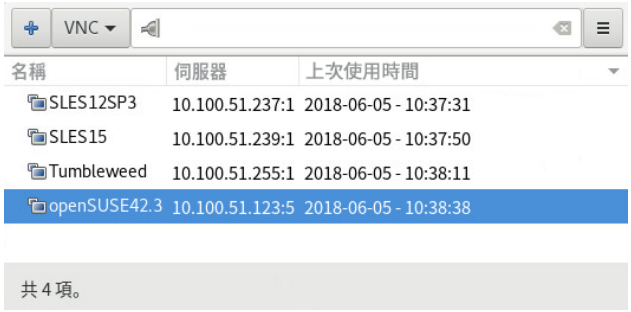
8.2.1 安裝

若要使用 Remmina，請檢查系統上是否安裝了 `remmina` 套件，若未安裝，請加以安裝。記得還要安裝適用於 Remmina 的 VNC 外掛程式：

```
root # zypper in remmina remmina-plugin-vnc
```

8.2.2 主視窗

透過輸入 `remmina` 指令執行 Remmina。

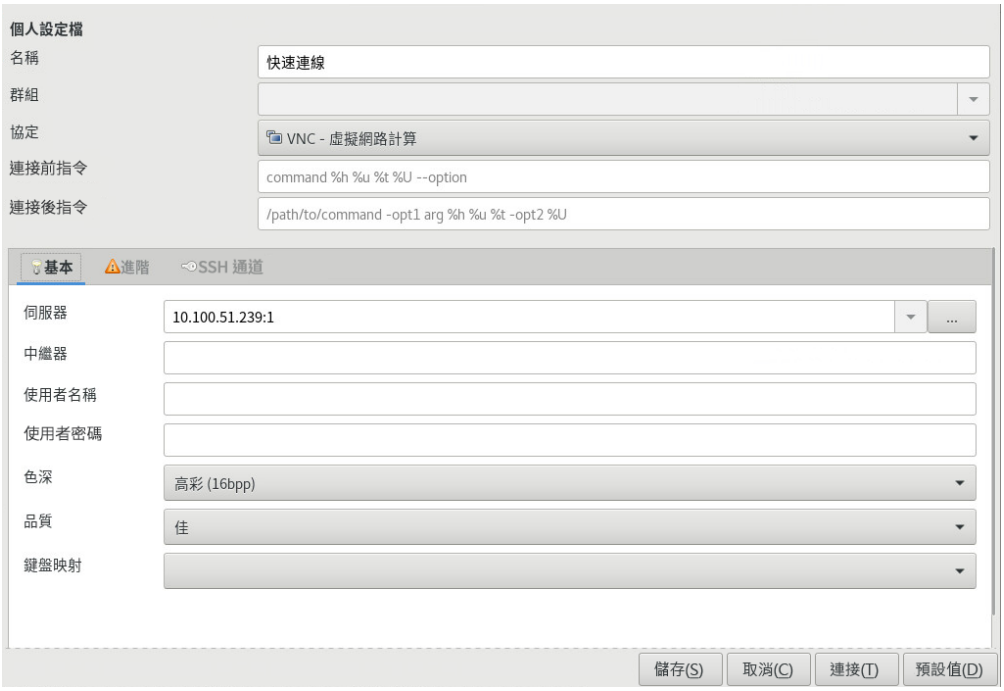


圖形 8.2 REMMINA 的主視窗

該應用程式主視窗顯示儲存的遠端工作階段清單。在這裡，您可以新增和儲存新遠端工作階段、快速啓動新工作階段而不儲存、啓動先前儲存的工作階段，或設定 Remmina 的全域優先設定。

8.2.3 新增遠端工作階段

若要新增和儲存新遠端工作階段，請按一下主視窗左上方的 。遠端桌面優先設定視窗即會開啓。



圖形 8.3 遠端桌面優先設定

在用於指定剛才新增的遠端工作階段設定檔的欄位中填寫資訊。最重要的技術包括:

名稱
: 設定檔的名稱，將列於主視窗中。

通訊協定
: 連接到遠端工作階段時要使用的通訊協定，例如 VNC。

伺服器
: 遠端伺服器的 IP 或 DNS 位址和顯示號碼。

使用者名稱、密碼
: 將用於進行遠端驗證的身分證明。保留為空白表示不進行驗證。

色彩深度、品質
: 根據連接速度和品質選取最佳選項。

選取進階索引標籤可輸入更具體的設定。

提示：停用加密

如果用戶端與遠端伺服器之間的通訊不加密，請啓用停用加密，否則連接會失敗。

選取 SSH 索引標籤可顯示進階 SSH 通道和驗證選項。

請按儲存加以確認。新的設定檔將列在主視窗中。

8.2.4 啓動遠端工作階段

您可以啓動先前儲存的工作階段，也可以快速啓動一個遠端工作階段而不儲存連接詳細資料。

8.2.4.1 快速啓動遠端工作階段

若要快速啓動遠端工作階段而不真正新增和儲存連接詳細資料，請使用主視窗頂部的下拉式方塊和文字欄位。

圖形 8.4 快速啓動

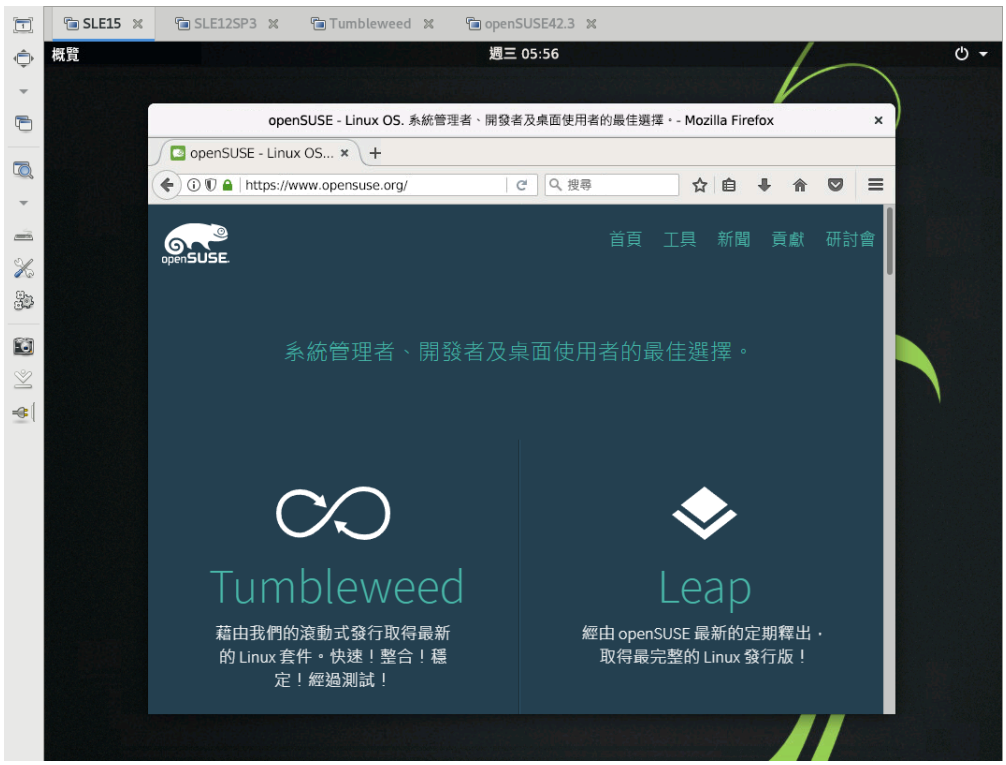
從下拉式方塊中選取通訊協定（例如「VNC」），然後輸入 VNC 伺服器 DNS 或 IP 位址，後跟一個冒號和顯示號碼，然後按 Enter 確認。

8.2.4.2 開啓儲存的遠端工作階段

若要開啓特定的遠端工作階段，請從工作階段清單中連按兩下該工作階段。

8.2.4.3 遠端工作階段視窗

遠端工作階段會在新視窗的索引標籤中開啓。每個索引標籤代管一個工作階段。視窗左側的工具列可協助您管理視窗/工作階段，例如切換全螢幕模式、調整視窗大小以適應工作階段的顯示大小、將特定按鍵動作傳送到工作階段、擷取工作階段的螢幕畫面，或設定影像品質。



圖形 8.5 正在檢視 SLES 15 遠端工作階段的 REMMINA

8.2.5 編輯、複製和刪除儲存的工作階段

若要編輯儲存的某個遠端工作階段，請在 Remmina 主視窗中以滑鼠右鍵按一下該工作階段的名稱，然後選取編輯。如需相關欄位的描述，請參閱第 8.2.3 節 「新增遠端工作階段」。

若要複製儲存的某個遠端工作階段，請在 Remmina 主視窗中以滑鼠右鍵按一下該工作階段的名稱，然後選取複製。在遠端桌面優先設定視窗中，變更設定檔的名稱，（選擇性）調整相關選項，然後按一下儲存確認。

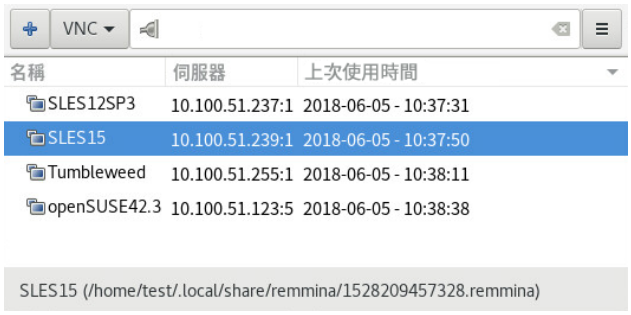
若要刪除儲存的某個遠端工作階段，請在 Remmina 主視窗中以滑鼠右鍵按一下該工作階段的名稱，然後選取刪除。在下一個對話方塊中，按一下是確認。

8.2.6 從指令行執行遠端工作階段

如果您需要從指令行或使用批次檔案開啓遠端工作階段，而不先開啓應用程式主視窗，請使用以下語法：

```
tux > remmina -c profile_name.remmina
```

Remmina 的設定檔儲存在您主目錄下的 `.local/share/remmina/` 目錄中。若要確定哪個設定檔屬於您要開啓的工作階段，請執行 Remmina，在主視窗中按一下工作階段名稱，然後在視窗底部的狀態行中查看設定檔的路徑。



圖形 8.6 查看設定檔的路徑

如果 Remmina 未在執行，您可以將設定檔重新命名為更合理的檔案名稱，例如 `sle15.remmina`。您甚至可以將設定檔複製到自訂目錄，並從該目錄中使用 `remmina -c` 指令來執行它。

8.3 一次性 VNC 工作階段

一次性工作階段由遠端用戶端啓動。它會在伺服器上啓動圖形登入畫面。這樣，您就可以選擇啓動工作階段的使用者，如果登入管理員支援，還可以選擇桌面環境。終止此類 VNC 工作階段的用戶端連接時，在該工作階段內啓動的所有應用程式也會隨之終止。一次性 VNC 工作階段無法共用，不過在同一台主機上可以同時啓動多個工作階段。

程序 8.1 啓用一次性 VNC 工作階段

1. 啓動YaST › 網路服務 › 遠端管理（VNC）。
2. 核取允許進行遠端管理（不含工作階段管理）。
3. 如果您打算在網頁瀏覽器視窗中存取 VNC 工作階段，請啓用允許使用網路瀏覽器存取。
4. 若需要，還可核取在防火牆中開啓埠(例如，當網路介面設定為位於外部區域中時)。如果您有多個網路介面，請透過防火牆細節設定在特定介面開啓防火牆埠的限制。

5. 按下一步確認您的設定值。

6. 如果有些需要的套件尚未提供，您需要批准安裝缺失套件。

提示：重新啓動顯示管理員

YaST 對顯示管理員設定進行了變更。您需要登出目前的圖形工作階段並重新啓動顯示管理員，以使變更生效。

圖形 8.7 遠端管理

8.3.1 可用的組態

SUSE Linux Enterprise Server 上的預設組態以 16 位元色彩深度、1024x768 像素解析度顯示工作階段。使用「一般」的 VNC 檢視器（相當於 VNC 顯示埠 `1`）時，可在連接埠 `5901` 上檢視工作階段，而使用網頁瀏覽器時，則可在連接埠 `5801` 上檢視工作階段。

其他組態可在不同的連接埠上使用，請參閱第 8.3.3 節 「設定一次性 VNC 工作階段」。

在一次性工作階段中，VNC 顯示埠號碼和 X 顯示埠號碼是獨立的。VNC 顯示埠號碼需要手動指定給伺服器支援的每個組態（在上例中為 :1）。每次以其中一種組態啓動 VNC 工作階段時，工作階段會自動獲得可用的 X 顯示埠號碼。

依預設，VNC 用戶端與伺服器會嘗試透過安裝後產生的自行簽署 SSL 證書安全通訊。您可以使用預設的證書，也可以用自己的證書取代它。使用自行簽署的證書時，需在首次連接之前確認其簽名。

8.3.2 啓動一次性 VNC 工作階段

若要連接一次性 VNC 工作階段，必須安裝 VNC 檢視器，另請參閱第 8.1 節「vncviewer 用戶端」。

8.3.3 設定一次性 VNC 工作階段

如果您不需要或不想修改預設組態，可以跳過本小節。

一次性 VNC 工作階段透過 `systemd` 通訊端 `xvnc.socket` 啓動。其中預設會提供六個組態區塊：三個用於 VNC 檢視器（`vnc1` 到 `vnc3`），另外三個提供 Java Applet（`vnchttpd1` 到 `vnchttpd3`）。預設只有 `vnc1` 及 `vnchttpd1` 處於作用中狀態。

若要在開機時啓動 VNC 伺服器通訊端，請執行以下指令：

```
sudo systemctl enable xvnc.socket
```

若要立即啓動通訊端，請執行：

```
sudo systemctl start xvnc.socket
```

可透過 `server_args` 選項設定 `Xvnc` 伺服器。如需選項清單，請參閱 `Xvnc --help`。

新增自訂組態時，請確定它們沒有使用同一台主機上的其他組態、服務或現有的永久 VNC 工作階段已在使用的連接埠。

輸入以下指令啓用組態變更：

```
tux > sudo systemctl reload xvnc.socket
```

重要：防火牆與 VNC 埠

依照程序 8.1 「啓用一次性 VNC 工作階段」 中所述啓用遠端管理時，會在防火牆中開啓連接埠 `5801` 及 `5901`。如果提供 VNC 工作階段的網路介面受防火牆保護，則您在為 VNC 工作階段啓用其他連接埠時，需要手動開啓相應的連接埠。如需指示，請參閱《Security Guide》，第 15 章「Masquerading and Firewalls」。

8.4 永久 VNC 工作階段

從多個用戶端可以同時存取一個永久工作階段。此特性非常適合用於在一個用戶端具有完整存取權，而所有其他用戶端具有僅檢視存取權的場合進行演示操作。在訓練中，訓練員可能需要存取學員桌面的情況則是另一個使用案例。

提示：連接至永久 VNC 工作階段

若要連接至永久 VNC 工作階段，必須安裝 VNC 檢視器。如需更多詳細資料，請參閱第 8.1 節 「`vncviewer` 用戶端」。

持續 VNC 工作階段分為以下兩類：

- 使用 `vncserver` 啓動的 VNC 工作階段
- 使用 `vncmanager` 啓動的 VNC 工作階段

8.4.1 使用 `vncserver` 啓動的 VNC 工作階段

此類型的持續 VNC 工作階段在伺服器上啓動。不論用戶端的連接狀態為何，此工作階段以及在此工作階段中啓動的所有應用程式會一直執行，直至工作階段被終止。對永久工作階段的存取受到密碼保護，密碼可能為以下兩種類型：

- 授予完全存取權的一般密碼，或
- 授予非互動式（僅檢視）存取權限的選擇性僅檢視密碼。

一個工作階段可以同時擁有這兩種類型的多個用戶端連接。

程序 8.2 使用 `vncserver` 啓動持續 VNC 工作階段

1. 開啓外圍程序，並確定您以應當擁有 VNC 工作階段的使用者身分登入。
2. 如果提供 VNC 工作階段的網路介面受防火牆保護，則您需要在防火牆中手動開啓工作階段使用的連接埠。如果要啓動多個工作階段，可以開啓一系列連接埠。如需如何設定防火牆的詳細資料，請參閱《Security Guide》，第 15 章「Masquerading and Firewalls」。

   `vncserver` 對顯示埠 `:1` 使用連接埠 `5901`，對顯示埠 `:2` 使用連接埠 `5902`，依此類推。對於永久工作階段，VNC 顯示埠和 X 顯示埠的號碼通常都是相同的。
3. 若要以 1024x769 像素的解析度及 16 位元的色彩深度啓動工作階段，請輸入以下指令：

   ```
   vncserver -alwaysshared -geometry 1024x768 -depth 16
   ```

   `vncserver` 指令在未指定顯示埠號碼的情況下，會挑選未使用的號碼，並列印它的選擇。有關更多選項的資訊，請參閱 `man 1 vncserver`。

首次執行 `vncserver` 時，它會要求您輸入擁有工作階段完整存取權限的密碼。如果需要，您也可以提供對工作階段具有僅檢視存取權限的密碼。

此處提供的密碼還可供同一個使用者用來啓動以後的工作階段。使用 `vncpasswd` 指令可以變更這些密碼。

重要：安全性考量

請務必使用足夠長（8 個或更多字元）的嚴密密碼。請不要共用這些密碼。

若要終止工作階段，請從 VNC 檢視器關閉在 VNC 工作階段內部執行的桌面環境，就如同關閉一般本地 X 工作階段一樣。

如果您想要手動終止工作階段，請在 VNC 伺服器上開啓一個外圍程序，並務必以擁有您要終止之 VNC 工作階段的使用者身分登入。執行以下指令以終止在顯示埠 `:1` 上執行的工作階段：`vncserver -kill :1`

8.4.1.1 設定永久 VNC 工作階段

永久 VNC 工作階段可透過編輯 `$HOME/.vnc/xstartup` 來設定。依預設，此外圍程序程序檔會啓動它啓動時所處的同一個 GUI/視窗管理員。在 SUSE Linux Enterprise Server 中，此 GUI/視窗管理員為 GNOME 或 IceWM。如果想要使用您選擇的視窗管理員啓動工作階段，請設定變數 `WINDOWMANAGER`：

```
WINDOWMANAGER=gnome vncserver -geometry 1024x768
WINDOWMANAGER=icewm vncserver -geometry 1024x768
```

注意：每個使用者對應一個組態

永久 VNC 工作階段透過單個基於使用者的組態設定。由同一個使用者啓動的多個工作階段皆使用相同的啓動檔案和密碼檔案。

8.4.2 使用 vncmanager 啓動的 VNC 工作階段

程序 8.3 啓用持續 VNC 工作階段

1. 啓動 YaST › 網路服務 › 遠端管理（VNC）。
2. 啓用允許進行遠端管理（含工作階段管理）。
3. 如果您打算在網頁瀏覽器視窗中存取 VNC 工作階段，請啓用允許使用網路瀏覽器存取。
4. 若需要，還可核取在防火牆中開啓埠(例如，當網路介面設定為位於外部區域中時)。如果您有多個網路介面，請透過防火牆細節設定在特定介面開啓防火牆埠的限制。
5. 按下一步確認您的設定值。

6. 如果有些需要的套件尚未提供，您需要批准安裝缺失套件。

提示：重新啓動顯示管理員

YaST 對顯示管理員設定進行了變更。您需要登出目前的圖形工作階段並重新啓動顯示管理員，以使變更生效。

8.4.2.1 設定永久 VNC 工作階段

依程序 8.3 「啓用持續 VNC 工作階段」中所述啓用 VNC 工作階段管理後，便可以使用您喜歡的 VNC 檢視器（例如 `vncviewer` 或 Remmina）正常連接到遠端工作階段。此時將顯示登入畫面。當您登入後，桌面環境的系統匣中將出現「VNC」圖示。按一下該圖示可開啓 VNC 工作階段視窗。如果該圖示未出現，或者您的桌面環境不支援將圖示放在系統匣中，請手動執行 `vncmanager-controller`。

圖形 8.8 VNC 工作階段設定

有幾項設定會影響 VNC 工作階段的行為：

非持續，私人
: 這相當於一次性工作階段。其他使用者將看不見此類工作階段，它在您斷開連接後即會終止。如需相關資訊，請參閱第 8.3 節 「一次性 VNC 工作階段」。

持續，可見
: 其他使用者可以看見工作階段，它在您斷開連接後仍繼續執行。

工作階段名稱
: 您可以在此處指定持續工作階段的名稱，以便在重新連接時可以輕鬆識別它。

不需要密碼
: 任何人不必提供使用者身分證明登入即可存取工作階段。

需要使用者登入
: 需要使用有效的使用者名稱和密碼登入後，才能存取工作階段。該選項會在允許的使用者文字方塊中列出有效的使用者名稱。

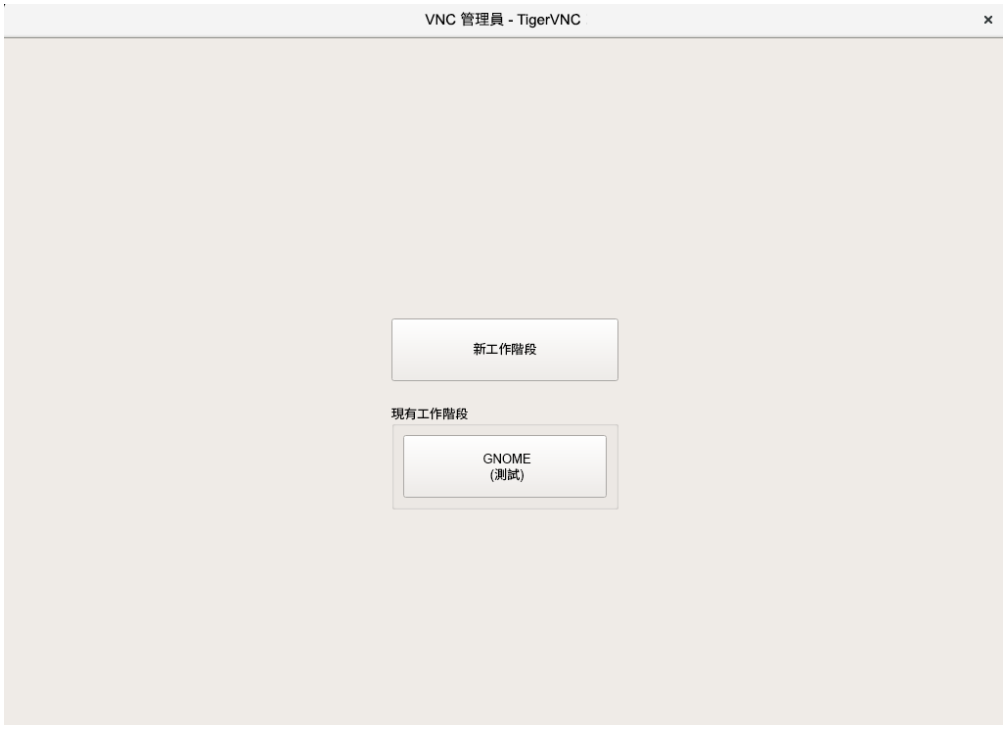
一次允許一個用戶端
: 禁止多個使用者同時加入工作階段。

一次允許多個用戶端
: 允許多個使用者同時加入持續工作階段。非常適合在遠端簡報或培訓這類場合中使用。

按一下確定加以確認。

8.4.2.2 加入持續 VNC 工作階段

依第 8.4.2.1 節 「設定永久 VNC 工作階段」中所述設定持續 VNC 工作階段後，可透過 VNC 檢視器加入它。當 VNC 用戶端連接到伺服器後，系統將提示您選擇是要建立新工作階段，還是加入現有工作階段：



圖形 8.9 加入持續 VNC 工作階段

當您按一下現有工作階段的名稱後，系統可能要求您輸入登入身分證明，具體取決於持續工作階段設定。

8.5 加密 VNC 通訊

如果 VNC 伺服器設定正確，則 VNC 伺服器與用戶端之間的所有通訊都會加密。驗證在工作階段開始時進行，實際的資料傳輸在驗證後開始。

無論是一次性還是持續 VNC 工作階段，都可透過 `server_args` 行中 `/usr/bin/Xvnc` 指令的 `-securitytypes` 參數設定安全性選項。`-securitytypes` 參數會選取驗證方法和加密。它的選項如下：

驗證

None、TLSNone、X509None
: 沒有驗證。

VncAuth、TLSVnc、X509Vnc
: 驗證使用自訂密碼。

Plain、TLSPlain、X509Plain

驗證使用 PAM 來驗證使用者的密碼。

加密

None、VncAuth、Plain

不加密。

TLSNone、TLSVnc、TLSPlain

匿名 TLS 加密。所有內容都會加密，但不驗證遠端主機。因此，您可以防護被動攻擊者，但不能防禦中間人攻擊者。

X509None、X509Vnc、X509Plain

使用證書進行 TLS 加密。如果使用自行簽署的證書，則在第一次連接時，系統將要求您驗證證書。在以後的連接中，僅當證書有變更時，系統才會向您發出警告。因此，在第一次連接時，您可以防禦中間人攻擊之外的所有其他攻擊（類似於使用一般的 SSH）。如果使用由證書管理中心簽署且與機器名稱相符的證書，您的安全將得到全面保障（類似於使用一般的 HTTPS）。

提示：證書和金鑰的路徑

如果您使用基於 X509 的加密，需要透過 `-X509Cert` 和 `-X509Key` 選項指定 X509 證書和金鑰的路徑。

如果您選取多種安全性類型（以逗號分隔），將會使用用戶端與伺服器都支援且允許的第一種安全性。如此，您便可在伺服器上設定隨機加密。如果您需要支援不支援加密的 VNC 用戶端，此功能將相當實用。

在用戶端上，您也可以指定允許的安全性類型，以防在您連接到已知啓用了加密的伺服器時遭到降級攻擊（雖然在該情況下，我們的 vncviewer 會發出「連接未加密！」訊息來警告您）。

9 使用 RSync 複製檔案

當今時代，使用者通常都會有數部電腦：家用機器和辦公機器、筆記型電腦、智慧型電話或平板電腦。因而，在多個裝置之間保持檔案和文件同步的任務就變得越發重要。

警告：資料遺失的風險

您在開始使用同步工具之前，應該先熟悉其特性和功能。請務必備份您的重要檔案。

9.1 概念綜覽

對於要透過慢速網路連接同步大量資料的情況，Rsync 提供了可靠的方法來僅傳輸檔案中的變更。此方法不僅適用於文字檔案，還適用於二進位檔案。為了偵測檔案之間的差異，Rsync 將檔案分為多個區塊，並計算它們的檢查總數。

偵測變更對運算能力有一定的要求。因此，請確定兩端的機器均具有足夠的資源，包括 RAM。

當需要定期傳輸大量僅包含微小變更的資料時，Rsync 特別實用。進行備份時就常常用到該工具。Rsync 也非常適合用來鏡像預備伺服器，此類伺服器將 Web 伺服器的完整目錄樹儲存到 DMZ 內的某部 Web 伺服器中。

Rsync 並不是同步工具，雖然它的名字看上去有些像。Rsync 工具一次只能在一個方向複製資料。它不會也不能反向複製資料。如果您需要既能同步來源又能同步目標的雙向工具，請使用 Csync。

9.2 基本語法

Rsync 是一個指令行工具，基本語法如下：

```
rsync [OPTION] SOURCE [SOURCE]... DEST
```

您可以在任何本地或遠端機器上使用 Rsync，前提是您擁有相應的存取權限和寫入權限。可以有多個 `SOURCE` 項目。`SOURCE` 和 `DEST` 預留位置可以是路徑和/或 URL。

下面介紹一些最常用的 Rsync 選項：

`-v`
: 輸出較詳細的文字

`-a`
: 歸檔模式；以遞迴方式複製檔案並保留時間戳記、使用者/群組擁有權、檔案權限和符號連結

`-z`
: 壓縮傳輸的資料

注意：結尾斜線計數

使用 Rsync 時，應特別注意結尾斜線。目錄後面的結尾斜線表示目錄的內容。沒有結尾斜線表示目錄自身。

9.3 在本地複製檔案和目錄

下面的描述假設目前的使用者擁有 `/var/backup` 目錄的寫入許可權。若要將單個檔案從機器上的一個目錄複製到另一個路徑，請使用以下指令：

```
tux > rsync -avz backup.tar.xz /var/backup/
```

檔案 `backup.tar.xz` 會複製到 `/var/backup/`，絕對路徑是 `/var/backup/backup.tar.xz`。

請勿忘記在 `/var/backup/` 目錄後面加上結尾斜線！如果不插入斜線，檔案 `backup.tar.xz` 會複製到 `/var/backup`（檔案）中，而不是 `/var/backup/` 目錄中！

複製目錄與複製單個檔案相似。下面的範例將目錄 `tux/` 及其內容複製到 `/var/backup/` 目錄中：

```
tux > rsync -avz tux /var/backup/
```

在絕對路徑 `/var/backup/tux/` 中可找到副本。

9.4 從遠端複製檔案和目錄

兩部機器上都需要有 Rsync 工具。若要從遠端目錄複製檔案或將檔案複製到遠端目錄，需要提供 IP 位址或網域名稱。如果本地機器和遠端機器上目前的使用者名稱相同，則可以不指定使用者名稱。

若要使用相同的使用者（在本地和遠端主機上）將檔案 `file.tar.xz` 從本地主機複製到遠端主機 `192.168.1.1`，請使用以下指令：

```
tux > rsync -avz file.tar.xz  tux@192.168.1.1:
```

依據您的個人偏好，也可以使用下面的指令，它們的作用相同：

```
tux > rsync -avz file.tar.xz 192.168.1.1:~
tux > rsync -avz file.tar.xz 192.168.1.1:/home/tux
```

在所有使用標準組態的情況下，系統會提示您輸入遠端使用者的密碼片語。此指令會將 `file.tar.xz` 複製到使用者 `tux` 的主目錄（通常為 `/home/tux`）。

從遠端複製目錄與在本地複製目錄相似。下面的範例將目錄 `tux/` 及其內容複製到 `192.168.1.1` 主機上的遠端目錄 `/var/backup/`：

```
tux > rsync -avz tux 192.168.1.1:/var/backup/
```

假設您在主機 `192.168.1.1` 上擁有寫入許可權，便可在絕對路徑 `/var/backup/tux` 中找到副本。

9.5 設定和使用 Rsync 伺服器

Rsync 可當成在用於內送連接的預設連接埠 873 上列出的精靈（`rsyncd`）執行。此精靈可以接收「複製目標」。

下面的描述介紹如何在 `jupiter` 上建立具有備份目標的 Rsync 伺服器。此目標可用於儲存您的備份。若要建立 Rsync 伺服器，請執行以下操作：

程序 9.1 設定 RSYNC 伺服器

1. 在 jupiter 上，建立用於儲存您所有備份檔案的目錄。在此範例中，我們使用 `/var/backup`：

   ```
   root # mkdir /var/backup
   ```

2. 指定擁有權。在此範例中，該目錄為使用者群組中的使用者 `tux` 所擁有：

```
root # chown tux.users /var/backup
```

3. 設定 `rsyncd` 精靈。
我們將組態檔案分割成一個主檔案，和一些用於存放您的備份目標的「模組」。如此，以後便可更輕鬆地新增其他目標。全域值可以儲存在 `/etc/rsyncd.d/*.inc` 檔案中，而模組放置在 `etc/rsyncd.d/*.conf` 檔案中：

   a. 建立目錄 `/etc/rsyncd.d/`：

```
root # mkdir /etc/rsyncd.d/
```

   b. 在主組態檔案 `/etc/rsyncd.conf` 中，新增以下幾行：

```
# rsyncd.conf main configuration file
log file = /var/log/rsync.log
pid file = /var/lock/rsync.lock

&merge /etc/rsyncd.d ❶
&include /etc/rsyncd.d ❷
```

      ❶ 將 `/etc/rsyncd.d/*.inc` 檔案中的全域值合併到主組態檔案中。

      ❷ 從 `/etc/rsyncd.d/*.conf` 檔案中載入任何模組（或目標）。這些檔案不應包含任何對全域值的參考。

   c. 在檔案 `/etc/rsyncd.d/backup.conf` 中透過以下幾行建立您的模組（您的備份目標）：

```
# backup.conf: backup module
[backup] ❶
   uid = tux ❷
   gid = users ❷
   path = /var/backup ❸
   auth users = tux ❹
   secrets file = /etc/rsyncd.secrets ❺
   comment = Our backup target
```

      ❶ 備份目標。可以使用您喜歡的任何名稱。但最好依照目標的用途來命名，並使用在 `*.conf` 檔案中所用的相同名稱。

      ❷ 指定在進行檔案傳輸時所用的使用者名稱或群組名稱。

      ❸ 定義用於儲存備份的路徑（從步驟 1 中）。

4 指定所允許使用者的逗號分隔清單。清單以最簡單的方式包含允許連接到此模組的使用者名稱。在我們的範例中，只允許使用者 `tux`。

5 指定包含使用者名稱和明文密碼的行所在檔案的路徑。

d. 建立包含以下內容的 `/etc/rsyncd.secrets` 檔案，並取代 `PASSPHRASE`：

```
# user:passwd
tux:PASSPHRASE
```

e. 確認該檔案只有 `root` 使用者可讀取：

```
root # chmod 0600 /etc/rsyncd.secrets
```

4. 透過以下指令啓動並啓用 `rsyncd` 精靈：

```
root # systemctl enable rsyncd
root # systemctl start rsyncd
```

5. 測試是否可存取 Rsync 伺服器：

```
tux > rsync jupiter::
```

您應該會看到類似如下的回應：

```
backup          Our backup target
```

若非如此，請檢查您的組態檔案、防火牆和網路設定。

上述步驟建立了 Rsync 伺服器，現在可以使用它來儲存備份。本範例也建立了監聽所有連接的記錄檔。這個檔案是儲存在 `/var/log/rsyncd.log`。如果您要對傳輸進行除錯，此檔案非常實用。

若要列出備份目標的內容，請使用以下指令：

```
rsync -avz jupiter::backup
```

此指令會列出伺服器上 `/var/backup` 目錄中存在的所有檔案。這個要求也會記錄在 `/var/log/rsyncd.log` 記錄檔中。若要開始實際傳輸，請提供來源目錄。請使用 `.` 來代表目前的目錄。例如，下面的指令會將目前的目錄複製到 Rsync 備份伺服器：

```
rsync -avz . jupiter::backup
```

依預設，Rsync 不會在執行時刪除檔案和目錄。若要允許刪除，必須另外指定選項 `--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。

9.6 更多資訊

CSync

雙向檔案同步器，請參閱 https://www.csync.org/。

RSnapshot

建立增量備份，請參閱 http://rsnapshot.org。

Unison

與 CSync 類似的檔案同步處理器，但具有圖形介面，請參閱 http://www.seas.upenn.edu/~bcpierce/unison/。

Rear

一個災難備援架構，請參閱 https://www.suse.com/documentation/sle-ha-12/ 上 SUSE Linux Enterprise High Availability Extension 的《管理指南》。

II Linux 系統開機

10 開機程序簡介

Linux 系統開機涉及多個元件和任務。完成韌體和硬體啓始化程序（取決於機器的架構）之後，系統將透過開機載入程式 GRUB 2 啓動核心。在此之後，開機程序完全由作業系統控制，並由 `systemd` 負責處理。`systemd` 會提供一組「目標」，用於啓動與日常使用、維護或緊急情況相關的組態。

10.1 術語

本章使用的術語可能有不同的解釋。為了理解本章中術語的用法，請閱讀以下定義：

`init`

有兩個不同的程序通常會命名為 「init」：

- 掛接根檔案系統的 `initramfs` 程序
- 從實際根檔案系統執行且用於啓動其他所有程序的作業系統程序

在這兩種情況下，`systemd` 程式都會處理此任務。首先會從 `initramfs` 執行此程序，以掛接根檔案系統。掛接成功後，從根檔案系統以初始程序的形式重新執行此程序。為了避免混淆這兩個 `systemd` 程序，我們將第一個程序稱為 `init on initramfs`，將第二個程序稱為 `systemd`。

`initrd` / `initramfs`

`initrd`（初始 RAM 磁碟）是一個影像檔案，內含核心所載入的並且做為暫存根檔案系統從 `/dev/ram` 掛接的根檔案系統影像。掛接此檔案系統需要使用檔案系統驅動程式。

從核心 2.6.13 開始，`initramfs`（初始 RAM 檔案系統）取代了 initrd，前者無需檔案系統驅動程式即可掛接。SUSE Linux Enterprise Server 只使用 `initramfs`。但是，由於 `initramfs` 做為 `/boot/initrd` 儲存，因此通常將其稱為 「initrd」。本章只使用名稱 `initramfs`。

10.2 Linux 開機程序

Linux 開機程序由數個階段組成，每個階段分別由不同的元件所代表：

1. 第 10.2.1 節 「 啓始化和開機載入程式階段 」
2. 第 10.2.2 節 「核心階段」
3. 第 10.2.3 節 「init on initramfs 階段」
4. 第 10.2.4 節 「systemd 階段」

10.2.1 啓始化和開機載入程式階段

在啓始化階段，將設定機器硬體並準備好裝置。此程序根據硬體架構的不同有很大的差別。

SUSE Linux Enterprise Server 在所有架構中使用開機載入程式 GRUB 2。根據架構和韌體，啓動 GRUB 2 開機載入程式的程序可能包括多個步驟。開機載入程式的用途是載入核心以及初始的 RAM 式檔案系統 (initramfs)。如需 GRUB 2 的詳細資訊，請參閱第 12 章 「開機載入程式 GRUB 2」。

10.2.1.1 AArch64 與 AMD64/Intel 64 上的啓始化和開機載入程式階段

開啓電腦之後，BIOS 或 UEFI 會啓始化螢幕和鍵盤，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。辨識開機媒體及其幾何尺寸之後，系統控制權將會從 BIOS/UEFI 轉到開機載入程式。

在配備傳統 BIOS 的機器上，只能載入開機磁碟第一個實體 512 位元組資料磁區（主開機記錄，MBR）中的程式碼。只有極少量的 GRUB 2 程式碼能夠裝入 MBR。開機載入程式的唯一作用就是從 MBR 與第一個分割區（MBR 分割區表）之間的間隙處，或是從 BIOS 開機分割區（GPT 分割區表）載入包含檔案系統驅動程式的 GRUB 2 核心影像。此影像包含檔案系統驅動程式，因此能夠存取根檔案系統中的 `/boot`。`/boot` 包含 GRUB 2 核心 (core) 的附加模組以及核心 (kernel) 和 initramfs 影像。取得此分割區的存取權之後，GRUB 2 會將核心和 initramfs 影像載入記憶體，並將控制權交接到核心。

從包含已加密分割區 `/boot` 的加密檔案系統將 BIOS 系統開機時，需要輸入解密密碼兩次。GRUB 2 使用第一次輸入的密碼來解密 `/boot`，`systemd` 使用第二次輸入的密碼來載入加密的磁碟區。

在配備 UEFI 的機器上，開機程序比配備傳統 BIOS 的機器要簡單得多。韌體能夠讀取包含 GPT 分割區表的磁碟的 FAT 格式化系統分割區。此 EFI 系統分割區（在執行中的系統上載入為 `/boot/efi`）可提供足夠的空間，用於代管由韌體直接載入和執行的完備 GRUB 2。

如果 BIOS/UEFI 支援網路開機，則也可以設定提供開機載入程式的開機伺服器。然後，可以透過 PXE 將系統開機。BIOS/UEFI 用做開機載入程式。它會從開機伺服器取得開機影像，然後啓動系統。這與本地硬碟完全無關。

10.2.1.2 IBM z Systems 上的啓始化和開機載入程式階段

在 IBM z Systems 上，必須透過名為 `zipl`（z initial program load，z 初始程式載入）的開機載入程式啓始化開機程序。雖然 `zipl` 支援讀取不同的檔案系統，但它不支援 SLE 預設檔案系統（Btrfs）或者從快照開機。因此，SUSE Linux Enterprise Server 使用兩階段的開機程序來確定開機時完全支援 Btrfs：

1. `zipl` 從 ext2 格式化分割區 `/boot/zipl` 開機。此分割區包含一個極簡的核心，以及一個載入記憶體中的 initramfs。initramfs 包含 Btrfs 驅動程式（及其他元件）和開機載入程式 GRUB 2。系統使用參數 `initgrub`（告知要啓動 GRUB 2）來啓動核心。
2. 核心會掛接根檔案系統，以使 `/boot` 可存取。現在，將從 initramfs 啓動 GRUB 2。GRUB 2 從 `/boot/grub2/grub.cfg` 讀取其組態，並從 `/boot` 載入最終的核心和 initramfs。現在，將透過 Kexec 載入新核心。

10.2.2 核心階段

開機載入程式轉交系統控制權後，所有架構中的開機程序均是相同的。開機載入程式會將核心和初始的 RAM 式檔案系統（`initramfs`）都載入記憶體中，而核心將接管控制權。

核心設定記憶體管理並偵測 CPU 類型及其功能之後，將啓始化硬體，並從記憶體中掛接使用 `initramfs` 載入的暫存根檔案系統。

10.2.2.1 initramfs 檔案

`initramfs`（初始 RAM 檔案系統）是一個小型 cpio 歸檔，可由核心載入 RAM 磁碟。該檔案位於 `/boot/initrd` 中。可以使用名為 `dracut` 的工具建立該檔案，如需詳細資料，請參閱 `man 8 dracut`。

`initramfs` 提供了一個極簡的 Linux 環境，可用於在掛接實際根檔案系統之前執行程式。BIOS 或 UEFI 常式會將最精簡的 Linux 環境載入記憶體，該環境只需要有足夠的記憶體，除此之外，沒有特定硬體需求。`initramfs` 歸檔必須始終提供一個名為 `init` 的可執行檔，該檔案會執行根檔案系統上的 `systemd` 精靈，使開機程序得以繼續。

在根目錄檔案系統能夠掛接以及作業系統可以啓動之前，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。`init` on `initramfs` 會載入根檔案系統所需的模組。當模組載入之後，`udev` 便會為 `initramfs` 提供所需的裝置。在後來的開機程序中，變更根檔案系統後，必須重新產生這些裝置。可以使用 `systemd` 單位 `systemd-udev-trigger.service` 來實現此目的。

10.2.2.1.1 重新產生 initramfs

由於 `initramfs` 包含多個驅動程式，因此，每當其中某個驅動程式有新版本推出時，都需要更新 initramfs。在安裝包含驅動程式更新的套件時可以自動完成這種更新。YaST 或 zypper 透過顯示用於產生 `initramfs` 的指令輸出來告知此狀態。但在某些情況下，您需要手動重新產生 `initramfs`：

- 由於更換硬體而需新增驅動程式
- 將系統目錄移至 RAID 或 LVM
- 將磁碟新增至包含根檔案系統的 LVM 群組/Btrfs RAID
- 變更核心變數

由於更換硬體而需新增驅動程式

如果需要更換硬體（例如硬碟），並且開機時此硬體需要核心中的不同驅動程式，則您必須更新 `initramfs` 檔案。

編輯 `/etc/dracut.conf.d/01-dist.conf`（如果該檔案不存在，則予以建立）並新增下面一行。

```
force_drivers+="DRIVER1"
```

以驅動程式的模組名稱取代 `DRIVER1`。如果您需要新增多個驅動程式，請逐一列出並以空格分隔。

```
force_drivers+="DRIVER1 DRIVER2"
```

繼續執行程序 10.1 「產生 initramfs」。

將系統目錄移至 RAID 或 LVM

每次您要將執行中系統上的交換檔案或系統目錄（例如 `/usr`）移至 RAID 或邏輯磁碟區時，都需要建立一個包含軟體 RAID 或 LVM 驅動程式支援的 `initramfs`。為此，請在 `/etc/fstab` 中建立相關的項目，並掛接新項目（例如，使用 `mount -a` 和/或 `swapon -a`）。

繼續執行程序 10.1 「產生 initramfs」。

將磁碟新增至包含根檔案系統的 LVM 群組/Btrfs RAID

每當您要在包含根檔案系統的邏輯磁碟區群組或者 Btrfs RAID 中新增（或移除）磁碟時，都需要建立一個支援增大的磁碟區的 `initramfs`。請遵循程序 10.1 「產生 initramfs」中的說明操作。

繼續執行程序 10.1 「產生 initramfs」。

變更核心變數

如果您在 `sysctl` 介面中透過編輯相關檔案（`/etc/sysctl.conf` 或 `/etc/sysctl.d/*.conf`）變更了核心變數的值，系統下次重新開機時，這項變更將會遺失。即使您在執行時使用 `sysctl --system` 載入這些值，變更也不會儲存到 initramfs 檔案中。您需要依照程序 10.1 「產生 initramfs」中所述更新該檔案。

程序 10.1 產生 INITRAMFS

請注意，您需要以 `root` 使用者身分執行以下程序中的所有指令。

1. 執行以下指令產生新的 `initramfs` 檔案

```
dracut MY_INITRAMFS
```

請以所選檔案名稱取代 `MY_INITRAMFS`。新的 `initramfs` 將會建立為 `/boot/MY_INITRAMFS`。

或者執行 `dracut -f`。這會覆寫目前使用的現有檔案。

2. （如果在上一步中執行了 `dracut -f`，請跳過此步驟）。為上一步中所建立的 `initramfs` 檔案建立連結：

```
(cd /boot && ln -sf MY_INITRAMFS initrd)
```

3. 在 IBM z Systems 架構中，另外還需執行 `grub2-install`。

10.2.3 init on initramfs 階段

由核心從 `initramfs` 掛接的暫存根檔案系統包含可執行檔案 `systemd`（下文稱為 `init` on `initramfs`，另請參閱第 10.1 節 「術語」）。此程式執行掛接正確根檔案系統所需的全部動作。它為所需的檔案系統提供核心功能，並為使用 `udev` 的大量儲存控制器提供裝置驅動程式。

`initramfs` 上的 `init` 主要用途是為掛接以及存取實際根檔案系統做好準備。根據您的系統組態，`initramfs` 上的 `init` 負責下列任務。

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。

提供區塊特殊檔案
: 核心根據載入的模組產生裝置事件。`udev` 會處理這些事件，並在 `/dev` 內的 RAM 檔案系統中產生所需的特殊區塊檔案。如果沒有這些專用檔案，便無法存取檔案系統和其他裝置。

管理 RAID 和 LVM 設定
: 如果您之前將系統設定為在 RAID 或 LVM 下存放根檔案系統，`initramfs` 上的 `init` 此時會設定 LVM 或 RAID，以便之後能夠存取根檔案系統。

管理網路組態
: 如果您之前將系統設定為使用網路掛接的根檔案系統（透過 NFS 掛接），那麼 `init` 此時必須確定是否已載入適當的網路驅動程式，並且這些驅動程式是否設定為允許存取根檔案系統。

如果檔案系統位於 iSCSI 或 SAN 這樣的網路區塊裝置上，`initramfs` 上的 `init` 還會設定與儲存伺服器間的連線。如果主要目標不可用，SUSE Linux Enterprise Server 支援從次要 iSCSI 目標開機。如需有關開機 iSCSI 目標的組態的更多詳細資料，請參閱《儲存管理指南》，第 14 章「IP 網路上的大型儲存裝置：iSCSI」，第 14.3.1 節「使用 YaST 設定 iSCSI 啓動器的組態」。

注意：處理掛接錯誤

如果開機檔案系統無法在開機環境中掛接，則必須對該系統進行檢查與修復，然後才能繼續開機。對於 Ext3 與 Ext4 檔案系統，檔案系統檢查程式會自動啓動。如果是 XFS 和 Btrfs 檔案系統，則不會自動開始修復程序，而是向使用者顯示有關可用於修復檔案系統的選項的資訊。成功修復檔案系統後，結束開機環境將會使系統重新嘗試掛接根檔案系統。如果掛接成功，將繼續正常開機。

10.2.3.1 安裝程序中的 init on initramfs 階段

若於安裝過程的啓始開機階段呼叫 `init on initramfs`，它執行的任務會與上述任務有所不同。請注意，安裝系統也不會從 `initramfs` 啓動 `systemd` — 這些任務由 `linuxrc` 執行。

尋找安裝媒體

當您啓動安裝程序時，機器會載入一個安裝核心以及一個包含 YaST 安裝程式的特殊 `init`。YaST 安裝程式在 RAM 檔案系統中執行，它必須知道安裝媒體的位置，才能存取該媒體來安裝作業系統。

啓始硬體辨識並載入適當核心模組

如第 10.2.2.1 節 「initramfs 檔案」中所述，開機程序從最少的一組驅動程式（可在大多數硬體組態中使用）開始。在 AArch64、POWER 和 AMD64/Intel 64 機器上，`linuxrc` 會啓動初始硬體掃描程序，以確定適合您硬體組態的驅動程式集。在 IBM z Systems 上，需要提供驅動程式及其參數的清單（例如，透過 linuxrc 或 parmfile 提供）。

這些驅動程式用來產生系統開機所需的自訂 `initramfs`。如果開機不需要這些模組，但是 coldplug 需要這些模組，則可以使用 `systemd` 載入這些模組；如需詳細資訊，請參閱第 13.6.4 節 「載入核心模組」。

載入安裝系統

系統正確識別硬體後，會立即載入相應的驅動程式。`udev` 程式會建立特殊的裝置檔案，`linuxrc` 將使用 YaST 安裝程式啓動安裝系統。

啓動 YaST

最後，`linuxrc` 啓動 YaST，後者則啓動套件安裝和系統組態。

10.2.4 systemd 階段

找到「實際的」根檔案系統後，對其進行錯誤檢查並加以掛接。若掛接成功，系統會清理 `initramfs`，並執行根檔案系統上的 `systemd` 精靈。`systemd` 是 Linux 的系統和服務管理員。它是做為 PID 1 啓動的父程序，用做 `init` 系統來啓動和維護使用者空間服務。如需詳細資料，請參閱第 13 章 「systemd 精靈」。

11 UEFI（整合可延伸韌體介面）

UEFI（整合可延伸韌體介面）是系統硬體隨附的韌體、系統的所有硬體元件與作業系統之間的介面。

UEFI 在 PC 系統上的應用越來越廣泛，正逐漸取代傳統的 PC-BIOS。例如，UEFI 可正確支援 64 位元系統並提供安全開機（「安全開機」，需要 2.3.1c 版或更高版本的韌體），這是它最重要的功能之一。最後，所有 x86 平台上藉由 UEFI 都將可以使用標準韌體。

UEFI 另外還提供下列優勢：

- 從具有 GUID 分割區表（GPT）的大型磁碟（超過 2 TiB）開機。
- 獨立於 CPU 的結構和驅動程式。
- 具有網路功能的彈性作業系統前環境。
- CSM（相容性支援模組）可支援透過類似於 PC-BIOS 模擬功能將舊版作業系統開機。

如需詳細資訊，請參閱 http://en.wikipedia.org/wiki/Unified_Extensible_Firmware_Interface。下列各節並不是一般性的 UEFI 綜覽，只是有關如何在 SUSE Linux Enterprise Server 中實作某些功能的提示。

11.1 安全開機

在 UEFI 的領域里，保護開機程序即表示建立一條信任鏈。「平台」是此信任鏈的根；在 SUSE Linux Enterprise Server 環境中，可將主機板和板載韌體視為「平台」。換言之，它就是硬體廠商，信任鏈從硬體廠商流向元件製造商、作業系統廠商等。

信任透過公用金鑰加密來表示。硬體廠商將所謂的「平台金鑰（PK）」放入韌體中，表示信任的根。與作業系統廠商及其他的信任關係透過使用「平台金鑰」簽署其金鑰加以記錄。

最後，要求任何程式碼必須由其中一個「信任」金鑰簽署後，韌體才將執行程式碼，以此構建安全性 — 可以是作業系統開始載入程式、位於某些 PCI Express 卡之快閃式記憶體內或磁碟上的一些驅動程式，也可以是韌體本身的更新。

若要使用安全開機，您需要以韌體信任的金鑰簽署您的作業系統載入程式，並且需要作業系統載入程式驗證其載入的核心是否可信。

金鑰交換金鑰（KEK）可新增至 UEFI 金鑰資料庫。這樣一來，您便可以使用其他證書，只要這些證書是以 PK 的私密部分簽署的即可。

11.1.1 在 SUSE Linux Enterprise Server 上實作

預設會安裝 Microsoft 的金鑰交換金鑰（KEK）。

注意：需要 GUID 分割區表（GPT）

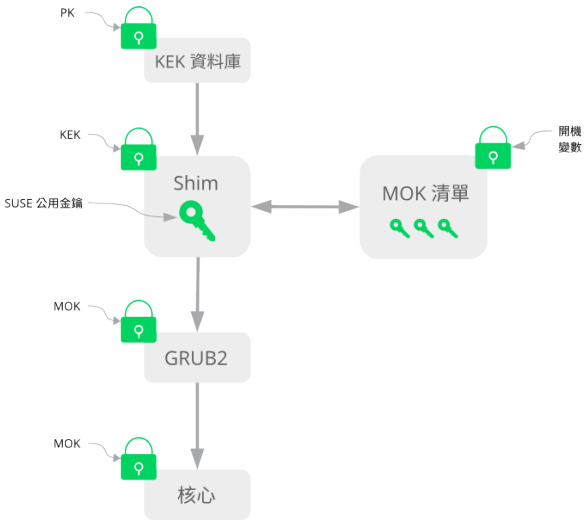
預設已在 UEFI/x86_64 安裝中啓用安全開機功能。可以在開機載入程式設定對話方區塊的開機碼選項索引標籤中找到啓用安全開機支援選項。該選項支援在韌體中啓用安全開機後開機，同時也支援在停用安全開機後開機。

圖形 11.1 安全開機支援

安全開機功能要求 GUID 分割區表（GPT）以主開機記錄（MBR）取代舊分割區。如果 YaST 在安裝期間偵測到 EFI 模式，它將會嘗試建立 GPT 分割區。UEFI 需要在 FAT 格式 EFI 系統分割區（ESP）上找到 EFI 程式。

支援 UEFI 安全開機本質上需要具有韌體辨識為信任金鑰之數位簽名的開機載入程式。該金鑰需要先天為韌體所信任（無需任何手動介入）。

實現此目的方法有兩種。一種方法是與硬體廠商協作，讓他們簽署一個 SUSE 金鑰，隨後 SUSE 會使用它來簽署開機載入程式。另一種方法是執行 Microsoft 的 Windows Logo Certification 程式認證開機載入程式，並讓 Microsoft 辨識 SUSE 簽署金鑰（即，使用其 KEK 進行簽署）。目前，SUSE 透過 UEFI 簽署服務（在此範例中為 Microsoft）來簽署載入程式。



圖形 11.2 UEFI：安全開機程序

SUSE 在實作層使用預設安裝的 `shim` 載入程式。這是一個智慧型解決方案，它可避免法律問題並大大簡化了認證和簽署步驟。`shim` 載入程式的工作是載入開機載入程式（例如 GRUB 2）並對它進行驗證；此開機載入程式接著會僅載入 SUSE 金鑰簽署的核心。SUSE 從全新安裝的 SLE11 SP3 開始提供此功能，並啓用 UEFI 安全開機。

信任使用者的類型有兩種：

- 第一種是擁有金鑰的使用者。平台金鑰（PK）幾乎允許所有作業。金鑰交換金鑰（KEK）允許 PK 允許的所有作業，變更 PK 除外。
- 第二種是擁有實際存取機器權限的任何人。擁有實際存取機器權限的使用者可以將機器重新開機並設定 UEFI。

UEFI 提供兩種類型的變數來滿足那些使用者的需求：

- 第一類變數即「已驗證的變數」，它們可從開機程序（即「開機服務環境」）和執行中作業系統中更新。僅當簽署變數新值的金鑰是用於簽署變數舊值的相同金鑰時，才能進行更新。並且它們只能附加至或變更為序號更高的值。
- 第二種稱為「僅開機服務變數」。開機程序期間執行的所有程式碼都可以存取這些變數。在開機程序結束之後並在作業系統啓動之前，開機載入程式必須呼叫 `ExitBootServices` 呼叫。此後，這些變數無法再存取，且作業系統無法更改它們。

各種 UEFI 金鑰清單屬於第一種類型，因為此類型允許線上更新與新增金鑰、驅動程式和韌體指紋，以及將它們列入黑名單。第二類變數即「僅開機服務變數」。該類變數可協助您以安全且支援開放原始碼的方式實作安全開機，因此符合 GPLv3 要求。

SUSE 首先啓動 `shim`，它是一個小而簡單的 EFI 開機載入程式，由 SUSE 和 Microsoft 簽署。

這允許 `shim` 載入並執行。

`shim` 隨後會繼續驗證它要載入的開機載入程式是否受信任。在預設情況下，`shim` 將會使用內嵌於其主體的獨立 SUSE 證書。此外，`shim` 將允許「註冊」額外的金鑰，並覆寫預設的 SUSE 金鑰。在下文中，我們稱它們為「機器擁有者金鑰」，或簡稱為 MOK。

接著，開機載入程式會在驗證核心後將其開機，核心繼而將對模組執行同樣的操作。

11.1.2 MOK（機器擁有者金鑰）

如果使用者（「機器擁有者」）要取代開機程序的任何元件，將會使用機器擁有者金鑰 (MOK)。`mokutils` 工具將會協助簽署元件並管理 MOK。

當 `shim` 載入時，註冊程序便會開始將機器重新開機並中斷開機程序（例如按某個鍵）。隨後，`shim` 將進入註冊模式，允許使用者以開機分割區上之檔案中的金鑰取代預設 SUSE 金鑰。如果使用者選擇這麼做，`shim` 隨後將會計算該檔案的雜湊，並將結果置於「僅開機服務」變數中。這允許 `shim` 偵測對開機服務以外之檔案所進行的任何變更，從而避免篡改使用者核准的 MOK 清單。

所有這些操作都在開機期間進行 — 現在僅執行已驗證的程式碼。因此，僅顯示於主控台中的使用者可以使用機器擁有者的金鑰組。它不能是可以遠端存取作業系統的惡意軟體或駭客，因為駭客或惡意軟體只能變更檔案，而無法變更儲存在「僅開機服務」變數中的雜湊。

開機載入程式一旦載入並經 `shim` 驗證，便會在要驗證核心時回呼 `shim`，以避免驗證碼重複。`Shim` 將對此使用相同的 MOK 清單，並通知開機載入程式是否可以載入該核心。

如此一來，您便可以安裝您自己的核心或開機載入程式。您只需要安裝一組新的金鑰，隨後真實存在於第一次重新開機期間，由此對它們進行授權。由於 MOK 清單並非單獨一個 MOK，因此您可以讓 `shim` 信任來自多個廠商的金鑰，從而允許從開機載入程式進行雙重或多重開機。

11.1.3 將自訂核心開機

下文以 http://en.opensuse.org/openSUSE:UEFI#Booting_a_custom_kernel 為基礎。

安全開機不會阻止您使用自我編譯的核心。您必須使用您自己的證書簽署該核心，並讓韌體或 MOK 知道該證書。

1. 建立用於簽署的自訂 X.509 金鑰和證書：

   ```
   openssl req -new -x509 -newkey rsa:2048 -keyout key.asc \
     -out cert.pem -nodes -days 666 -subj "/CN=$USER/"
   ```

   如需建立證書的詳細資訊，請參閱 http://en.opensuse.org/openSUSE:UEFI_Image_File_Sign_Tools#Create_Your_Own_Certificate。

2. 將金鑰和證書封裝成 PKCS#12 結構：

   ```
   openssl pkcs12 -export -inkey key.asc -in cert.pem \
     -name kernel_cert -out cert.p12
   ```

3. 產生 NSS 資料庫以與 `pesign` 配合使用：

   ```
   certutil -d . -N
   ```

4. 將 PKCS#12 中包含的金鑰和證書輸入到 NSS 資料庫中：

   ```
   pk12util -d . -i cert.p12
   ```

5. 使用 `pesign` 以新簽名「保護」核心：

   ```
   pesign -n . -c kernel_cert -i arch/x86/boot/bzImage \
   ```

```
  -o vmlinuz.signed -s
```

6. 列出核心影像上的簽名：

```
pesign -n . -S -i vmlinuz.signed
```

此時，您可以如一般方式在 `/boot` 中安裝核心。因為核心現在已有自訂簽名，所以需要將用於簽署的證書輸入到 UEFI 韌體或 MOK 中。

7. 將證書轉換為 DER 格式以便輸入到韌體或 MOK 中：

```
openssl x509 -in cert.pem -outform der -out cert.der
```

8. 將證書複製到 ESP 以方便存取：

```
sudo cp cert.der /boot/efi/
```

9. 使用 `mokutil` 自動啓動 MOK 清單。

   - a. 將證書輸入到 MOK 中：

     ```
     mokutil --root-pw --import cert.der
     ```

     `--root-pw` 選項可讓 `root` 使用者直接使用。

   b. 檢查準備就緒、即待註冊的證書清單：

   ```
   mokutil --list-new
   ```

   c. 將系統重新開機；`shim` 應會啓動 MokManager。您需要輸入 `root` 密碼以確認將證書輸入到 MOK 清單中。

   d. 檢查是否已註冊新輸入的金鑰。

   ```
   mokutil --list-enrolled
   ```

   - a. 或者，如果要手動啓動 MOK，請按照下面的程序操作：
     重新開機

   b. 在 GRUB 2 功能表中，按「`c`」鍵。

   c. 類型：

   ```
   chainloader $efibootdir/MokManager.efi
   ```

```
boot
```

d. 選取從磁碟註冊金鑰。

e. 導覽到 `cert.der` 檔案並按 `Enter`。

f. 遵循指示註冊金鑰。一般情況下，應按「`0`」後再按「`y`」進行確認。或者，韌體功能表可能提供將新金鑰新增至簽名資料庫的方法。

11.1.4 使用非內建的驅動程式

在啓用安全開機的情況下進行安裝的過程中，不支援新增非現成驅動程式（即，不是 SUSE Linux Enterprise Server 隨附的驅動程式）。預設不信任用於 SolidDriver/PLDP 的簽署金鑰。

您可以透過兩種不同的方式，在啓用安全開機的情況下在安裝期間安裝協力廠商驅動程式。在這兩種情況下，都要：

- 在安裝之前，透過韌體或系統管理工具將需要的金鑰新增至韌體資料庫。此選項取決於您具體所用的硬體。如需詳細資訊，請諮詢您的硬體廠商。
- 使用 https://drivers.suse.com/ 上或您的硬體廠商提供的可開機驅動程式 ISO，於第一次開機時在 MOK 清單中註冊需要的金鑰。

若要使用可開機驅動程式 ISO 在 MOK 清單中註冊驅動程式金鑰，請執行以下步驟：

1. 將上述 ISO 影像燒錄到空白 CD/DVD 媒體中。
2. 使用新的 CD/DVD 媒體開始安裝，並準備好標準的安裝媒體或網路安裝伺服器的 URL。
   如果您要進行網路安裝，請使用 `install=` 選項在指令行上輸入網路安裝來源的 URL。
   如果您是從光學媒體安裝，安裝程式會先從驅動程式套件開機，然後要求插入產品的第一張安裝光碟。
3. 安裝時將會使用包含已更新驅動程式的 initrd。

如需詳細資訊，請參閱 https://drivers.suse.com/doc/Usage/Secure_Boot_Certificate.html。

11.1.5 功能和限制

以安全開機模式開機時，將會套用以下功能：

- 安裝到 UEFI 預設的開機載入程式位置，這是為了保留或還原 EFI 開機項目而採用的機制。
- 透過 UEFI 重新開機。
- 如果沒有可回復到的傳統的 BIOS，Xen 監管程式將使用 UEFI 開機。
- 支援 UEFI IPv6 PXE 開機。
- UEFI 支援視訊模式，核心可以從 UEFI 取回視訊模式，以使用相同的參數設定 KMS 模式。
- UEFI 支援從 USB 裝置開機。

以安全開機模式開機時，必須遵守以下限制：

- 為了確保無法輕易規避安全開機，當在安全開機模式下執行時，系統會停用部分核心功能。
- 必須簽署開機載入程式、核心和核心模組。
- 將會停用 Kexec 和 Kdump。
- 將會停用休眠（暫停磁碟上的作業）。
- 無法存取 `/dev/kmem` 和 `/dev/mem`，即使以 root 使用者身分也不行。
- 無法存取 I/O 埠，即使以 root 使用者身分也不行。所有 X11 圖形驅動程式都必須使用核心驅動程式。
- 不允許透過 sysfs 存取 PCI BAR。
- 無法使用 ACPI 中的 `custom_method`。
- 無法使用適用於 asus-wmi 模組的 debugfs。
- `acpi_rsdp` 參數對核心沒有任何影響。

11.2 更多資訊

- http://www.uefi.org — UEFI 首頁，您可在其中找到最新的 UEFI 規格。
- Olaf Kirch 和 Vojtěch Pavlík 發佈的部落格文章（上述章節主要取自這些文章）：
  - http://www.suse.com/blogs/uefi-secure-boot-plan/
  - http://www.suse.com/blogs/uefi-secure-boot-overview/
  - http://www.suse.com/blogs/uefi-secure-boot-details/
- http://en.opensuse.org/openSUSE:UEFI — UEFI 與 openSUSE。

12 開機載入程式 GRUB 2

本章介紹如何設定 SUSE® Linux Enterprise Server 中使用的開機載入程式 GRUB 2。GRUB 是傳統 GRUB 開機載入程式（現在稱做「GRUB Legacy」）的後繼產品。從 SUSE® Linux Enterprise Server 12 版本開始，就已使用 GRUB 2 做為預設的開機載入程式。產品中提供了一個 YaST 模組來進行最重要的設定。整個開機程序簡述於第 10 章 「開機程序簡介」。如需關於 UEFI 機器安全開機支援的詳細資料，請參閱第 11 章 「UEFI（整合可延伸韌體介面）」。

12.1 GRUB Legacy 與 GRUB 2 之間的主要差異

- 組態儲存在不同的檔案中。
- 支援更多的檔案系統（例如 Btrfs）。
- 可以直接讀取 LVM 或 RAID 裝置上儲存的檔案。
- 使用者介面可翻譯，並可以改變主題。
- 包含一套用於載入模組的機制，以支援諸如檔案系統等的其他功能。
- 自動搜尋及產生其他核心與作業系統（例如 Windows）的開機項目。
- 包含一個類似於 Bash 的精簡主控台。

12.2 組態檔案結構

GRUB 2 的組態以下列檔案為基礎：

`/boot/grub2/grub.cfg`
: 此檔案包含 GRUB 2 功能表項目的組態。它取代了 GRUB Legacy 中的 `menu.lst`。`grub.cfg` 由 `grub2-mkconfig` 指令自動產生，不應該對其進行編輯。

`/boot/grub2/custom.cfg`

這個可選用檔案在開機時由 `grub.cfg` 直接獲取，可用於將自訂項目新增至開機功能表。從 SUSE Linux Enterprise Server 開始，這些項目也將會在使用 `grub-once` 時進行剖析。

`/etc/default/grub`

此檔案控制 GRUB 2 的使用者設定，通常包含背景和主題等其他環境設定。

`/etc/grub.d/` 下的程序檔

在執行 `grub2-mkconfig` 指令期間，將會讀取此目錄中的程序檔。主要組態檔案 `/boot/grub/grub.cfg` 中整合了這些程序檔的指示。

`/etc/sysconfig/bootloader`

在使用 YaST 設定開機載入程式以及每次安裝新核心時，會用到此組態檔案。它將經過 perl-bootloader 的評估，後者會相應地修改開機載入程式組態檔案（例如，GRUB 2 對應的組態檔案 `/boot/grub 2/grub.cfg`）。`/etc/sysconfig/bootloader` 不是特定於 GRUB 2 的組態檔案，其值會套用於 SUSE Linux Enterprise Server 上安裝的任何開機載入程式。

`/boot/grub2/x86_64-efi`、`/boot/grub2/power-ieee1275`、`/boot/grub2/s390x`

這些組態檔案包含特定於架構的選項。

GRUB 2 可以透過多種方式控制。現有組態啟動項目，可以從圖形功能表選取（開頭顯示畫面）。組態從基於其他組態檔案編譯的檔案 `/boot/grub2/grub.cfg` 載入（參閱下文）。所有 GRUB 2 組態檔案都視為系統檔案，編輯這些組態檔案需要有 `root` 權限。

注意：啓用組態變更

手動編輯 GRUB 2 組態檔案後，您需要執行 `grub2-mkconfig` 才能啓用變更。但使用 YaST 變更組態時就不需要如此，因為 YaST 會自動執行 `grub2-mkconfig`。

12.2.1 檔案 /boot/grub2/grub.cfg

帶有開機功能表的圖形開頭顯示畫面以 GRUB 2 組態檔案 `/boot/grub2/grub.cfg` 為基礎，它包含關於可以透過功能表開機之所有分割區或作業系統的資訊。

系統每次開機時，GRUB 2 會直接從檔案系統載入功能表檔案。因此，在變更組態檔案後不需要重新安裝 GRUB 2。安裝或移除核心後，系統會自動重建 `grub.cfg`。

`grub.cfg` 由 `grub2-mkconfig` 基於檔案 `/etc/default/grub` 以及 `/etc/grub.d/` 目錄中的程序檔編譯。因此，切勿手動編輯該檔案，而應該編輯相關的來源檔案，或者依照第 12.3 節 「使用 YaST 設定開機載入器」中所述，使用 YaST 的開機載入程式模組來修改組態。

12.2.2 檔案 /etc/default/grub

此檔案包含 GRUB 2 的更多一般選項，例如，顯示功能表的時間，或者要開機的預設作業系統。若要列出所有可用選項，請查看以下指令的輸出：

```
grep "export GRUB_DEFAULT" -A50 /usr/sbin/grub2-mkconfig | grep GRUB_
```

除了已定義的變數外，使用者還可以引入自己的變數，日後在 `/etc/grub.d` 目錄下的程序檔內使用。

編輯 `/etc/default/grub` 後，請執行 `grub2-mkconfig` 以更新主要組態檔案。

注意：範圍

此檔案中設定的所有選項都是會影響所有開機項目的一般選項。透過 GRUB_*_XEN_* 組態選項可以設定 Xen 核心或 Xen 監管程式的特定選項。如需詳細資訊，請參閱下文。

`GRUB_DEFAULT`

設定預設會啓動的開機功能表項目。它的值可以是數值、功能表項目的完整名稱，或者「saved」。

`GRUB_DEFAULT=2` 會啓動第三個（從零開始計數）開機功能表項目。

`GRUB_DEFAULT="2>0"` 會啓動第三個頂層功能表項目的第一個子功能表項目。

`GRUB_DEFAULT="Example boot menu entry"` 會啓動名為「Example boot menu entry」的功能表項目。

`GRUB_DEFAULT=saved` 會將 `grub2-once` 或 `grub2-set-default` 指令指定的項目開機。`grub2-reboot` 只設定下一次重新開機的預設開機項目，而 `grub2-set-default` 設定發生變更之前的預設開機項目。`grub2-editenv list` 列出下一個開機項目。

`GRUB_HIDDEN_TIMEOUT`

等待使用者按某個鍵的指定秒數。在此期間，除非使用者按下某個鍵，否則不顯示功能表。如果使用者在指定的時間內未按任何鍵，控制權將移交給 `GRUB_TIMEOUT`。`GRUB_HIDDEN_TIMEOUT=0` 首先會檢查是否按下了 `Shift`，如果是，則顯示開機功能表，否則會立即啓動預設的功能表項目。如果 GRUB 2 只識別了一個可開機作業系統，則預設行為就是如此。

`GRUB_HIDDEN_TIMEOUT_QUIET`

如果指定 `false`，那麼當啓動了 `GRUB_HIDDEN_TIMEOUT` 功能時，系統會在一個空白螢幕上顯示倒數計時器。

`GRUB_TIMEOUT`

在自動啓動預設開機項目之前，開機功能表顯示的時間期限，以秒為單位。如果按下某個鍵，則會取消逾時，GRUB 2 將等待您手動完成選擇。如果指定 `GRUB_TIMEOUT=-1`，則在您手動選取開機項目之前，功能表會一直顯示。

`GRUB_CMDLINE_LINUX`

此行中的項目將新增到正常和復原模式之開機項目的末尾。使用它可以將核心參數新增至開機項目。

`GRUB_CMDLINE_LINUX_DEFAULT`

與 `GRUB_CMDLINE_LINUX` 一樣，但只能在正常模式下附加項目。

`GRUB_CMDLINE_LINUX_RECOVERY`

與 `GRUB_CMDLINE_LINUX` 一樣，但只能在復原模式下附加項目。

`GRUB_CMDLINE_LINUX_XEN_REPLACE`

此項目將徹底取代所有 Xen 開機項目的 `GRUB_CMDLINE_LINUX` 參數。

`GRUB_CMDLINE_LINUX_XEN_REPLACE_DEFAULT`

與 `GRUB_CMDLINE_LINUX_XEN_REPLACE` 一樣，但它只會取代 `GRUB_CMDLINE_LINUX_DEFAULT` 的參數。

`GRUB_CMDLINE_XEN`

此項目只為 Xen 客體核心指定核心參數 — 工作原理與 `GRUB_CMDLINE_LINUX` 相同。

`GRUB_CMDLINE_XEN_DEFAULT`

與 `GRUB_CMDLINE_XEN` 一樣 — 工作原理與 `GRUB_CMDLINE_LINUX_DEFAULT` 相同。

`GRUB_TERMINAL`

啓用並指定輸入/輸出終端機裝置。可以是 `console`（PC BIOS 和 EFI 主控台）、`serial`（序列終端機）、`ofconsole`（開放韌體主控台）或預設值 `gfxterm`（圖形模式輸出）。用引號括住所需的多個選項可以啓用多個裝置，例如 `GRUB_TERMINAL="console serial"`。

`GRUB_GFXMODE`

`gfxterm` 圖形終端機使用的解析度。請注意，您只能使用您的圖形卡（VBE）支援的模式。預設值為「auto」，即嘗試選取偏好的解析度。在 GRUB 2 指令行中輸入 `videoinfo` 可顯示 GRUB 2 可使用的螢幕解析度。當 GRUB 2 開機功能表螢幕顯示時，輸入 `c` 可存取指令行。

您還可以在解析度設定後面附加一個值來指定色彩深度，例如 `GRUB_GFXMODE=1280x1024x24`。

`GRUB_BACKGROUND`

設定 `gfxterm` 圖形終端機的背景影像。該影像必須是 GRUB 2 在開機時可讀的檔案，並且必須以 `.png`、`.tga`、`.jpg` 或 `.jpeg` 字尾結尾。必要時，系統會縮放該影像以適合螢幕大小。

`GRUB_DISABLE_OS_PROBER`

如果將此選項設定為 `true`，將會停用自動搜尋其他作業系統的功能。系統只會偵測 `/boot/` 中的核心影像，以及 `/etc/grub.d/` 中您自己的程序檔內的選項。

`SUSE_BTRFS_SNAPSHOT_BOOTING`

如果將此選項設定為 `true`，GRUB 2 可直接開機至 Snapper 快照。如需詳細資訊，請參閱第 7.3 節 「透過從快照開機來執行系統復原」。

如需選項的完整清單，請參閱 GNU GRUB 手冊 (http://www.gnu.org/software/grub/manual/grub.html#Simple-configuration)↗。如需可能參數的完整清單，請參閱 http://en.opensuse.org/Linuxrc↗。

12.2.3 /etc/grub.d 中的程序檔

系統在執行 `grub2-mkconfig` 指令期間，將讀取此目錄中的程序檔，這些程序檔的指示都整合到 `/boot/grub2/grub.cfg` 中。`grub.cfg` 中功能表項目的順序由此目錄中檔案的執行順序來決定。具有前置編號的檔案先執行，從最小的編號開始。`00_header` 在 `10_linux` 之前執行，而後者又在 `40_custom` 之前執行。如果存在採用字母名稱的檔案，這些檔案將在採用編號命名的檔案之後執行。在執行 `grub2-mkconfig` 期間，只有可執行檔才能在 `grub.cfg` 中產生輸出。依預設，`/etc/grub.d` 目錄中的所有檔案都是可執行檔。

提示：將自訂內容永久儲存在 grub.cfg 中

由於每次執行 `grub2-mkconfig` 時都會重新編譯 `/boot/grub2/grub.cfg`，因此所有自訂內容都會遺失。如果要將您的行直接插入到 `/boot/grub2/grub.cfg` 中，並且希望在執行 `grub2-mkconfig` 之後它們不會遺失，請將其插入到下面兩行之間：

```
### BEGIN /etc/grub.d/90_persistent ###
```

和

```
### END /etc/grub.d/90_persistent ###
```

行。`90_persistent` 程序檔可確保此類內容會保留下來。

下面列出了最重要的程序檔：

`00_header`

設定環境變數，例如系統檔案位置、顯示設定、主題和以前儲存的項目。它還可以輸入 `/etc/default/grub` 中儲存的優先設定。通常，您不需要變更此檔案。

`10_linux`

識別根裝置上的 Linux 核心，並建立相關的功能表項目，其中包括關聯的復原模式選項（如果已啓用）。主功能表頁面中只顯示最新核心，其他核心包含在子功能表中。

`30_os-prober`

此程序檔使用 `OS-prober` 來搜尋 Linux 和其他作業系統，並將結果放入 GRUB 2 功能表。其中有些區段用於識別其他特定作業系統，例如 Windows 或 macOS。

`40_custom`

使用此檔案可以方便地在 `grub.cfg` 中包含自訂開機項目。切勿變更開頭的 `exec tail -n +3 $0` 部分。

處理順序依據前置編號確定，編號最小的程序檔最先執行。如果多個程序檔的前置編號相同，則按整個名稱的字母順序來決定執行順序。

提示：/boot/grub2/custom.cfg

如果您建立了 `/boot/grub2/custom.cfg` 並在其中填入了內容，則開機時系統會自動將其包含到 `/boot/grub40/grub.cfg` 中緊接在 40_custom 後面的位置。

12.2.4 BIOS 磁碟機與 Linux 裝置之間的映射

在 GRUB Legacy 中，`device.map` 組態檔案用於依據 BIOS 磁碟機代號衍生 Linux 裝置名稱。不一定總能猜對 BIOS 磁碟機與 Linux 裝置之間的映射。例如，如果在 BIOS 組態中交換了 IDE 和 SCSI 驅動器的開機順序，那麼 GRUB Legacy 就會使用錯誤的順序。

GRUB 2 在產生 `grub.cfg` 時使用裝置 ID 字串 (UUID) 或檔案系統標籤，因此避免了此問題。GRUB 2 公用程式會即時建立一個暫存裝置對應，這通常足以滿足需要，在單磁碟系統中更是如此。

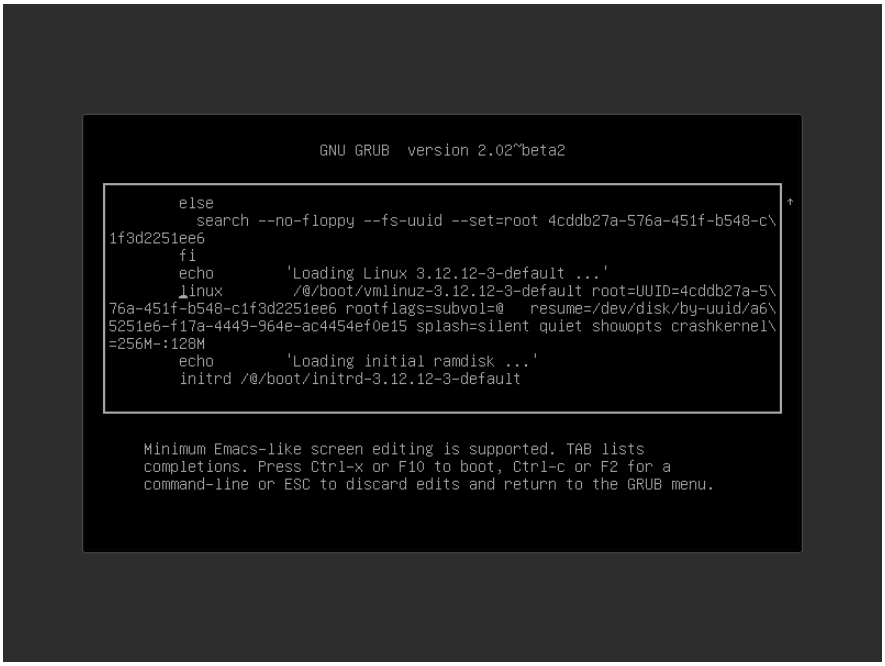
但是，如果您需要覆寫 GRUB 2 的自動裝置對應機制，請建立自訂對應檔案 `/boot/grub2/device.map`。下面的範例將變更映射，使 `DISK 3` 成為開機磁碟。請注意，GRUB 2 分割區編號以 `1` 開始，而不是像 GRUB Legacy 中那樣以 `0` 開始。

```
(hd1)  /dev/disk-by-id/DISK3 ID
(hd2)  /dev/disk-by-id/DISK1 ID
(hd3)  /dev/disk-by-id/DISK2 ID
```

12.2.5 在開機程序期間編輯功能表項目

當系統由於組態錯誤而不再能夠開機時，如果能夠直接編輯功能表項目，就會很有幫助。使用功能表編輯器還可以在不變更系統組態的情況下測試新設定。

1. 在圖形開機功能表中，使用方向鍵選取要編輯的項目。
2. 按 E 開啓文字型編輯器。
3. 使用方向鍵移到您要編輯的行。



圖形 12.1 GRUB 2 開機編輯器

現在，您可以採取以下兩種做法：

a. 將用空格分隔的參數新增到以 `linux` 或 `linuxefi` 開頭的行的末尾，以編輯核心參數。http://en.opensuse.org/Linuxrc 上提供了完整的參數清單。

b. 或者編輯一般選項，以變更核心版本等。按 →| 鍵會顯示所有可能的補齊建議。

4. 按 F10 使用您所做的變更啓動系統，或者按 Esc 放棄編輯，並返回 GRUB 2 功能表。

透過這種方式進行的變更只會套用到目前的開機過程，而不會永久儲存。

重要：開機程序期間的鍵盤配置

US 鍵盤配置是啓動時唯一可以使用的鍵盤配置。請參閱圖形 40.2 「美國鍵盤配置」。

注意：安裝媒體中的開機載入程式

在使用傳統 BIOS 的系統上，安裝媒體的開機載入程式仍是 GRUB Legacy。若要新增開機選項，請選取一個項目，然後開始輸入。在安裝開機項目中新增的內容將永久儲存在安裝的系統中。

注意：在 z Systems 上編輯 GRUB 2 功能表項目

IBM z Systems 上的游標移動方式和編輯指令有所不同，如需詳細資料，請參閱第 12.4 節 「z Systems 上終端機使用方式的差異」。

12.2.6 設定啓動密碼

即使在作業系統開機之前，GRUB 2 也支援對檔案系統的存取。沒有 root 許可權的使用者可以存取 Linux 系統中的檔案，而在系統開機後，他們將無權存取這些檔案。若要阻擋此類型的存取，或防止使用者啓動特定的功能表項目，請設定開機密碼。

重要：開機需要密碼

如果設定了開機密碼，則每次開機時都需要輸入該密碼，這意味著系統不會自動開機。

按如下步驟設定開機密碼。或者使用 YaST（使用密碼保護開機載入程式 ）。

1. 使用 `grub2-mkpasswd-pbkdf2` 來加密密碼：

```
tux > sudo grub2-mkpasswd-pbkdf2
```

```
Password: ****
Reenter password: ****
PBKDF2 hash of your password is grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

2. 將產生的字串連同 `set superusers` 指令一起貼到檔案 `/etc/grub.d/40_custom` 中。

```
set superusers="root"
password_pbkdf2 root grub.pbkdf2.sha512.10000.9CA4611006FE96BC77A...
```

3. 執行 `grub2-mkconfig` 以將變更輸入到主要組態檔案中。

在重新開機後，當您嘗試啓動某個功能表項目時，系統會提示您輸入使用者名稱和密碼。輸入 `root` 以及您在執行 `grub2-mkpasswd-pbkdf2` 指令期間輸入的密碼。如果身分證明正確，系統將啓動選定的開機項目。

如需詳細資訊，請參閱https://www.gnu.org/software/grub/manual/grub.html#Security。

12.3 使用 YaST 設定開機載入器

在 SUSE Linux Enterprise Server 系統中，設定開機載入程式一般選項最簡單的方法是使用 YaST 模組。在 YaST 控制中心內，選取系統 › 開機載入程式。模組會顯示系統目前的開機載入程式組態，並允許您進行變更。

使用開機碼選項索引標籤可以檢視和變更關於類型、位置和進階載入程式設定的設定。您可以選擇要在標準模式還是 EFI 模式下使用 GRUB 2。

圖形 12.2 開機碼選項

重要：EFI 系統要求使用 GRUB2-EFI

如果您使用的是 EFI 系統，則只能安裝 GRUB2-EFI，否則您的系統不再能夠開機。

重要：重新安裝開機載入程式

若要重新安裝開機載入程式，請確定變更 YaST 中的一個設定，然後將其復原。例如，若要重新安裝 GRUB2-EFI，請先選取 GRUB2，然後立即重新切換至 GRUB2-EFI。

否則，開機載入程式可能只會部分重新安裝。

注意：自訂開機載入程式

若要使用此處未列出的開機載入程式，請選取請勿安裝任何開機載入程式。請先詳細閱讀開機載入程式的說明文件，再選取這個選項。

12.3.1 開機載入程式位置和開機碼選項

開機載入程式的預設位置為主開機記錄（MBR）或 `/` 分割區的開機磁區，具體取決於分割區設定。若要修改開機載入程式的位置，請執行下列步驟：

程序 12.1 變更開機載入程式位置

1. 選取開機碼選項索引標籤，然後為開機載入程式位置選擇以下其中一個選項：

   從主開機記錄開機
   : 選取此選項將在包含 `/boot` 目錄的磁碟 MBR 中安裝開機載入程式。通常，這將是掛接到 `/` 的磁碟，但如果 `/boot` 掛接到其他磁碟上的獨立分割區中，則將會使用該磁碟的 MBR。

   從根分割區開機
   : 這會在 `/` 分割區的開機磁區中安裝開機載入程式。

   自訂開機分割區
   : 這個選項可讓您手動指定開機載入程式的位置。

2. 按一下確定套用您的變更。

圖形 12.3 代碼選項

開機碼選項索引標籤包含以下其他選項：

在分割區表中為開機分割區設定使用中旗標

啓動包含 `/boot` 目錄的分割區（PReP 分割區）。此選項用於具有舊 BIOS 的系統和/或舊式作業系統，因為它們可能無法從非使用中的分割區開機。您可以放心地啓用此選項。

將一般開機碼寫入 MBR

如果 MBR 包含自訂的「非 GRUB」代碼，此選項會以不受作業系統限制的泛型代碼取代該代碼。如果您停用此選項，系統可能變得無法開機。

啓用受信任的開機支援

啓動支援受信任之計算功能的 TrustedGRUB2（受信任的平台模組（Trusted Platform Module，TPM））。如需詳細資訊，請參閱 https://github.com/Sirrix-AG/TrustedGRUB2 。

12.3.2 調整磁碟順序

如果您的電腦有多個硬碟，您可以指定磁碟的開機順序。如果從 MBR 開機，將在清單中的第一個磁碟中安裝 GRUB 2。預設在該磁碟中安裝 SUSE Linux Enterprise Server。清單的其餘部分是有關 GRUB 2 的裝置對應程式的提示（請參閱第 12.2.4 節「BIOS 磁碟機與 Linux 裝置之間的映射」）。

警告：無法開機的系統

通常情況下，預設值幾乎對所有部署都有效。如果您錯誤地變更了磁碟的開機順序，系統下次重新開機時可能無法開機。例如，如果清單中的第一個磁碟不在 BIOS 開機序列中，並且清單中的其他磁碟有空白 MBR，系統將無法開機。

程序 12.2 設定磁碟順序

1. 開啓開機碼選項索引標籤。
2. 按一下編輯磁碟開機順序。
3. 如果列出超過一個磁碟，請選擇一個磁碟，然後按一下向上或向下重新排列顯示磁碟的順序。

4. 按兩次確定儲存變更。

12.3.3 設定進階選項

可透過開機載入程式選項索引標籤來設定進階開機選項。

12.3.3.1 開機載入程式選項索引標籤

圖形 12.4 開機載入程式選項

開機載入程式逾時

輸入新值或者用滑鼠按住相應的方向鍵，以變更逾時秒數的值。

偵測外來作業系統

如果選取該選項，開機載入程式將會搜尋其他系統（例如 Windows）或其他 Linux 安裝。

開機時隱藏功能表

隱藏開機功能表並使用預設項目開機。

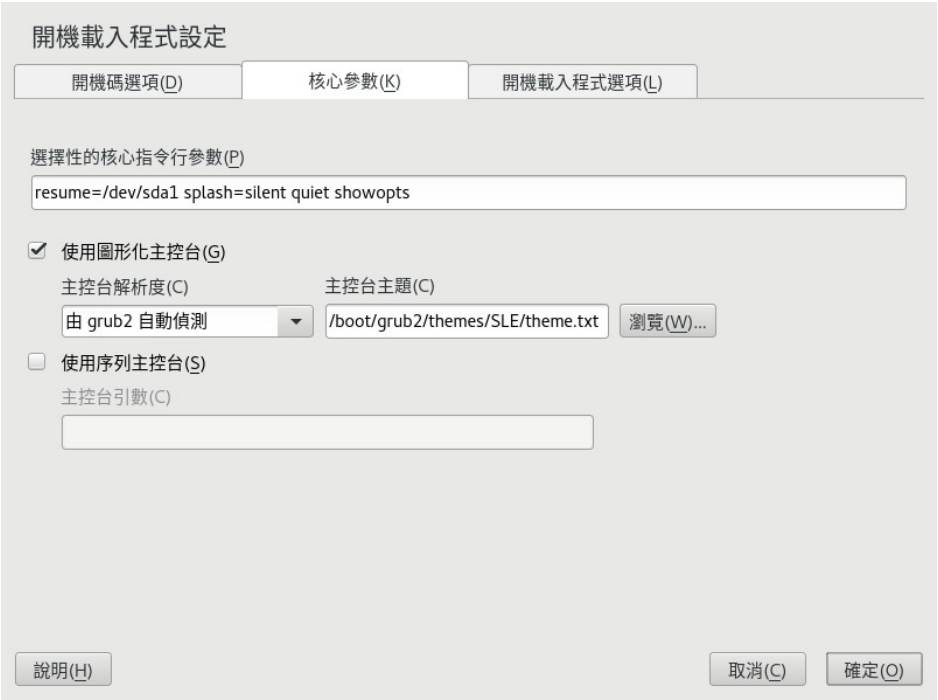
調整預設開機項目

從「預設開機區段」清單中選取所需項目。請注意，開機項目名稱中的「>」符號用於分隔開機區段及其子區段。

使用密碼保護開機載入程式

使用一個附加的密碼保護開機載入程式和系統。如需詳細資訊，請參閱第 12.2.6 節 「設定啓動密碼」。

12.3.3.2 核心參數索引標籤



圖形 12.5 核心參數

主控台解析度

主控台解析度選項指定開機過程中的預設螢幕解析度。

核心指令行參數

在預設參數的末尾新增選擇性核心參數。如需所有可用參數的清單，請參閱 http://en.opensuse.org/Linuxrc 。

使用圖形主控台

如果核取該選項，則開機功能表會顯示在圖形開頭顯示畫面中，而不是以文字模式顯示。此時，您可以透過主控台解析度清單設定開機螢幕的解析度，並使用主控台主題檔案選擇器指定圖形主題定義檔案。

使用序列控制台

如果您的機器是透過序列控制台控制的，請啓用此選項並指定以何速度來使用哪一個 COM 埠。請參閱 `info grub` 或 http://www.gnu.org/software/grub/manual/grub.html#Serial-terminal

12.4 z Systems 上終端機使用方式的差異

在 3215 和 3270 終端機上，游標的移動方式以及在 GRUB 2 中發出編輯指令的方式存在一些差異和限制。

12.4.1 限制

互動

互動性存在嚴重的限制。輸入時通常不能獲得直觀的回饋。若要查看游標所在的位置，請輸入底線（_）。

注意：3270 與 3215 的比較

與 3215 終端機相比，3270 終端機在顯示和重新整理螢幕方面要好得多。

游標移動方式

無法進行「傳統的」游標移動操作。Alt、Meta、Ctrl 和游標鍵不起作用。若要移動游標，請使用第 12.4.2 節 「按鍵組合」中列出的按鍵組合。

插入記號

插入記號（^）用做控制字元。若要輸入文字 ^ 後再輸入一個字母，請依序輸入 ^、^ 和字母。

輸入

將無法使用 Enter 鍵，請改用 ^—J。

12.4.2 按鍵組合

常用的替代按鍵：	^—J	確認（「Enter」）
	^—L	中止，返回前一「狀態」
	^—I	Tab 鍵補齊（在編輯模式與外圍程序模式下）
功能表模式下可用的按鍵：	^—一個	到第一個項目
	^—E	到最後一個項目
	^—P	到上一個項目
	^—N	到下一個項目
	^—G	向上一頁
	^—C	向下一頁
	^—F	將選定的項目開機或進入子功能表（與 ^—J 的作用相同）
	E	編輯選定的項目
	C	進入 GRUB-She11
編輯模式下可用的按鍵：	^—P	向上移動一行
	^—N	向下移動一行
	^—B	向左移動一格
	^—F	向右移動一格
	^—一個	到行首

	^—E	到行尾
	^—H	退格
	^—D	刪除
	^—K	刪除到行尾
	^—Y	恢復刪除
	^—O	插入新行
	^—L	重新整理螢幕
	^—X	開機項目
	^—C	進入 GRUB-She11
指令行模式下可用的按鍵：	^—P	上一個指令
	^—N	歷程中的下一個指令
	^—一個	到行首
	^—E	到行尾
	^—B	向左移動一格
	^—F	向右移動一格
	^—H	退格
	^—D	刪除
	^—K	刪除到行尾
	^—U	刪除行
	^—Y	恢復刪除

12.5 實用的 GRUB 2 指令

`grub2-mkconfig`

依據 `/etc/default/grub` 以及 `/etc/grub.d/` 中的程序檔產生新的 `/boot/grub2/grub.cfg`。

範例 12.1 GRUB2-MKCONFIG 用法

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

提示：語法檢查

執行不帶任何參數的 `grub2-mkconfig` 會將組態列印至 STDOUT 以供使用者檢查。在寫入 `/boot/grub2/grub.cfg` 後使用 `grub2-script-check` 可以檢查其語法。

重要：grub2-mkconfig 無法修復 UEFI 安全開機表

如果您使用的是 UEFI 安全開機，但您的系統無法再正常連接 GRUB 2，則您可能需要另行重新安裝 Shim 並重新產生 UEFI 開機表，為此請使用：

```
root # shim-install --config-file=/boot/grub2/grub.cfg
```

`grub2-mkrescue`

為安裝的 GRUB 2 組態建立一個可開機的救援影像。

範例 12.2 GRUB2-MKRESCUE 用法

```
grub2-mkrescue -o save_path/name.iso iso
```

`grub2-script-check`

檢查指定檔案中有無語法錯誤。

範例 12.3 GRUB2-SCRIPT-CHECK 用法

```
grub2-script-check /boot/grub2/grub.cfg
```

`grub2-once`

僅設定下次開機的預設開機項目。若要取得可用開機項目的清單，請使用 `--list` 選項。

範例 12.4 GRUB2-ONCE 用法

```
grub2-once number_of_the_boot_entry
```

提示：`grub2-once` 說明

不使用任何選項呼叫該程式可以取得所有可能選項的完整清單。

12.6 更多資訊

如需關於 GRUB 2 更進一步的資訊，請參閱 http://www.gnu.org/software/grub/。另請參閱 `grub` info 頁面。您也可以在「技術資訊搜尋」中搜尋關鍵字「GRUB 2」，取得關於特殊問題的資訊，網址是 http://www.suse.com/support。

13 systemd 精靈

程式 systemd 是程序 ID 為 1 的程序。負責以所需的方式啓始化系統。systemd 由核心直接啓動，訊號 9（通常會終止程序）對它不起作用。所有其他程式不是由 systemd 直接啓動，就是由它的子程序啓動。

從 SUSE Linux Enterprise Server 12 起，systemd 取代了常用的 System V init 精靈。systemd 與 System V init 完全相容（透過支援 init 程序檔）。systemd 的其中一個主要優點在於它透過積極主動的平行服務啓動，使開機速度顯著加快。另外，systemd 僅在切實需要服務時才會啓動該服務。它在開機時不會無條件地啓動精靈，而是在第一次需要時予以啓動。systemd 還支援「核心控制群組」(cgroup)，對系統狀態拍攝快照並還原等等。如需詳細資料，請參閱 http://www.freedesktop.org/wiki/Software/systemd/。

13.1 systemd 概念

本節將詳細討論 systemd 背後的概念。

13.1.1 systemd 是什麼

systemd 是適用於 Linux 的系統和工作階段管理員，它與 System V 及 LSB init 程序檔相容。主要功能如下：

- 提供積極主動的平行化功能
- 使用插槽及 D-Bus 啓用方式來啓動服務
- 提供精靈的隨需啓動
- 使用 Linux cgroup 追蹤程序
- 支援對系統狀態拍攝快照並還原
- 維護掛接點和自動掛接點
- 實作事務相關型複雜的服務控制邏輯

13.1.2 單位檔案

單位組態檔案包含有關以下項目的資訊：服務、通訊端、裝置、掛接點、自動掛接點、交換檔案或分割區、啓動目標、監控的檔案系統路徑、受 systemd 控制和監督的計時器、暫時系統狀態快照、資源管理片段，或一組外部建立的程序。「 單位檔案」是 systemd 用於表示下列項目的通用術語：

- 服務： 程序相關資訊（例如執行精靈）；檔案名以 .service 結尾
- 目標： 用於將單位分組以及在啓動期間用作同步點；檔案名以 .target 結尾
- 插槽： IPC 或網路插槽或檔案系統 FIFO 的相關資訊，適用於插槽型啓動（如 `inetd`）；檔案名以 .socket 結尾
- 路徑： 用於觸發其他單位（例如，在檔案變更時執行服務）；檔案名以 .path 結尾
- 計時器： 受控計時器的相關資訊，適用於計時器型啓動；檔案名以 .timer 結尾
- 掛接點： 通常由 fstab 產生器自動產生；檔案名以 .mount 結尾
- 自動掛接點： 檔案系統自動掛接點的相關資訊；檔案名以 .automount 結尾
- Swap： 用於記憶體分頁之交換裝置或檔案相關資訊；檔案名以 .swap 結尾
- 裝置： sysfs/udev(7) 裝置樹中所展示之裝置的相關資訊；檔案名以 .automount 結尾
- 範圍/片段： 分階層管理程序群組之資源的概念；檔案名以 .scope/.slice 結尾

如需有關 systemd.unit 的詳細資訊，請參閱 http://www.freedesktop.org/software/systemd/man/systemd.unit.html

13.2 基本用法

System V init 系統使用若干個指令來處理服務 - init 程序檔、`insserv`、`telinit` 及其他。systemd 可以簡化服務管理，因為對於大部分處理服務的任務，只需記住一條指令：`systemctl`。它使用「 指令加子指令」表示法，與 `git` 或 `zypper` 相似：

```
systemctl GENERAL OPTIONS SUBCOMMAND SUBCOMMAND OPTIONS
```

如需完整的手冊，請參閱 `man 1 systemctl`。

提示：終端機輸出和 Bash 完成法

如果輸出進入終端機（而不是進入管線或檔案之類），依預設，systemd 指令會將長輸出傳送到切換程式。使用 `--no-pager` 選項可關閉切換模式。

systemd 還支援 bash 完成法，它可讓您輸入子指令的第一個字母，然後按 →| 自動填全子指令。此功能僅可用於 `bash` 外圍程序，並且需要安裝套件 `bash-completion`。

13.2.1 管理正在執行的系統中的服務

用於管理服務的子指令與透過 System V init 管理服務的子指令相同（`start`、`stop`、...）。下面列出了服務管理指令的通用語法：

systemd

```
systemctl reload|restart|start|status|stop|... MY_SERVICE(S)
```

System V init

```
rcMY_SERVICE(S) reload|restart|start|status|stop|...
```

systemd 可讓您一次管理多個服務。它不是像 System V init 那樣依次執行 init 程序檔，而是執行類似如下的指令：

```
systemctl start MY_1ST_SERVICE MY_2ND_SERVICE
```

若要列出系統上所有可用的服務：

```
systemctl list-unit-files --type=service
```

下表列出了 systemd 和 System V init 的最重要的服務管理指令：

表格 13.1 服務管理指令

任務	systemd 指令	System V init 指令
啓動：	`start`	`start`
停止：	`stop`	`stop`
重新啓動： 關閉服務，然後啓動這些服務。如果某項服務並未執行，則會將其啓動。	`restart`	`restart`
有條件地重新啓動： 如果服務目前正在執行中，則予以重新啓動。對於未在執行中的服務，則不執行任何動作。	`try-restart`	`try-restart`
重新載入： 指示服務重新載入它們的組態檔案，而不中斷操作。使用案例：指示 Apache 重新載入修改過的 `httpd.conf` 組態檔案。請注意，並非所有服務都支援重新載入。	`reload`	`reload`
重新載入或重新啓動： 如果服務支援重新載入，則重新載入服務，否則重新啓動服務。如果某項服務並未執行，則會將其啓動。	`reload-or-restart`	`n/a`
有條件地重新載入或重新啓動： 如果服務支援重新載入，則重新載入服務，否則重新啓動那些目前正在執行的服務。對於未在執行中的服務，則不執行任何動作。	`reload-or-try-restart`	`n/a`
取得詳細的狀態資訊： 列出服務狀態的相關資訊。`systemd` 指令顯示詳細資料，例如描述、可執行檔、狀態、cgroup 及服務發出的最新訊息（請參閱第 13.6.8 節 「服務除錯」）。使用 System V init 顯示的詳細資料級別因服務而異。	`status`	`status`

任務	systemd 指令	System V init 指令
取得簡要的狀態資訊。 顯示服務是否處於使用中狀態。	`is-active`	`status`

13.2.2 永久啓用/停用服務

上一節中提及的服務管理指令可讓您操作目前工作階段的服務。systemd 還可讓您永久啓用或停用服務，使之可以按要求自動啓動，或者始終無法使用。此操作可以透過 YaST 或在指令行上執行。

13.2.2.1 在指令行上啓用/停用服務

下表列出了 systemd 和 System V init 的啓用和停用指令：

重要：服務啓動

在指令行上啓用服務時，服務不會自動啓動。系統將其排定為下一次系統啓動或執行層級/目標變更時啓動。若要在啓用服務之後立即啓動它，請明確執行 `systemctl start MY_SERVICE` 或 `rc MY_SERVICE start`。

表格 13.2 用於啓用和停用服務的指令

任務	`systemd` 指令	System V init 指令
啓用：	`systemctl enable MY_SERVICE(S)`	`insserv MY_SERVICE(S)` 、 `chkconfig -a MY_SERVICE(S)`
停用：	`systemctl disable MY_SERVICE(S).service`	`insserv -r MY_SERVICE(S)` 、 `chkconfig -d MY_SERVICE(S)`

任務	`systemd` 指令	System V init 指令
檢查：　顯示是否已啓用某個服務。	`systemctl is-enabled MY_SERVICE`	`chkconfig MY_SERVICE`
重新啓用：　與重新啓動服務相似，此指令先停用服務，然後再啓用該服務。若要使用服務的預設值重新啓用服務，可使用此任務。	`systemctl reenable MY_SERVICE`	無
遮罩：　「停用」某項服務之後，仍然可以手動啓動它。若要徹底停用服務，您需要予以遮罩。使用須謹慎。	`systemctl mask MY_SERVICE`	無
取消遮罩：　遮罩某項服務之後，惟有先將其取消遮罩，才能再次予以使用。	`systemctl unmask MY_SERVICE`	無

13.3 系統啓動和目標管理

啓動系統和關閉系統的整個程序由 init 維護。依此觀點，核心可以視為背景程序，以維護所有其他程序，並根據其他程式的要求來調整 CPU 時間和硬體存取。

13.3.1 目標與執行層級的比較

使用 System V init 時，系統將開機進入「執行層級」。執行層級定義了系統的啓動方式，以及在所執行的系統中可以使用哪些服務。執行層級標有編號；最常見的執行層級是 `0`（關閉系統）、`3`（多重使用者，包含網路）和 `5`（多重使用者，包含網路及顯示管理員）。

systemd 透過使用「 目標單位」引入新的概念。不過，它仍然與執行層級概念完全相容。目標單位是有名稱而不是有編號的，它有多個作用。例如，目標 `local-fs.target` 和 `swap.target` 掛接本地檔案系統和交換空間。

目標 `graphical.target` 提供包含網路和顯示管理員功能的多重使用者系統，與執行層級 5 相當。複雜的目標，例如 `graphical.target` 透過結合其他目標的子集用作「 中繼」目標。因為 systemd 能夠組合現有目標，便於使用者更便利地建立自訂目標，因此提供了可觀的靈活性。

下列清單顯示了最重要的 systemd 目標單位。如需完整清單，請參閱 `man 7 systemd.special`。

選定的 SYSTEMD 目標單位

`default.target`
: 預設開機的目標。這並非「 真實」目標，而是一個符號連結，指向 `graphic.target` 之類的另一個目標。可透過 YaST 永久變更（請參閱第 13.4 節 「使用 YaST 管理 服務」）。若要為某個工作階段變更它，請在開機提示處使用核心參數 `systemd.unit=MY_TARGET.target`。

`emergency.target`
: 在主控台上啓動緊急外圍程序。請僅在開機提示符處以如下格式使用它：`systemd.unit=emergency.target`。

`graphical.target`
: 啓動包含網路、多重使用者支援和顯示管理員功能的系統。

`halt.target`
: 關閉系統。

`mail-transfer-agent.target`
: 啓動傳送和接收郵件所需的所有服務。

`multi-user.target`
: 啓動包含網路的多重使用者系統。

`reboot.target`
: 系統重新開機。

`rescue.target`
: 啓動不包含網路的單一使用者系統。

為了保持與 System V init 執行層級系統相容，systemd 提供了名為 `runlevelX.target` 的特殊目標，可映射至編號為 `X` 的相應執行層級。

如果您要知道目前的目標，請使用指令：`systemctl get-default`

表格 13.3 SYSTEM V 執行層級和 systemd 目標單位

System V 執行層級	systemd 目標	用途
0	`runlevel0.target`、`halt.target`、`poweroff.target`	關閉系統
1, S	`runlevel1.target`、`rescue.target`、	單一使用者模式
2	`runlevel2.target`、`multi-user.target`、	本地多重使用者，不包含遠端網路
3	`runlevel3.target`、`multi-user.target`、	完整的多重使用者，包含網路
4	`runlevel4.target`	未使用/使用者定義
5	`runlevel5.target`、`graphical.target`、	完整的多重使用者，包含網路及顯示管理員
6	`runlevel6.target`、`reboot.target`、	系統重新開機

重要：systemd 忽略 /etc/inittab

System V init 系統中的執行層級在 `/etc/inittab` 中設定。systemd 不使用此組態。如需如何建立您自己的可開機目標的指示，請參閱第 13.5.3 節 「建立自訂目標」。

13.3.1.1 用於變更目標的指令

請使用下列指令來操作目標單位：

任務	systemd 指令	System V init 指令
變更目前的目標/執行層級	`systemctl isolate` `MY_TARGET`.target	`telinit` `X`
變更為預設目標/執行層級	`systemctl default`	無

取得目前的目標/執行層級	`systemctl list-units --type=target`
	對 systemd 而言，使用中的目標一般不止一個。該指令列出目前處於使用中狀態的所有目標。
		`who -r`
		或
		`runlevel`

永久性變更預設的執行層級	使用服務管理員或執行下列指令： `ln -sf /usr/lib/systemd/system/` `MY_TARGET`.target /etc/systemd/system/default.target	使用服務管理員或變更以下行 `id:` `X`:initdefault: （位於 `/etc/inittab` 中）
變更目前開機程序的預設執行層級	在開機提示的選項中輸入下列文字： `systemd.unit=` `MY_TARGET`.target	在開機提示中輸入所需的執行層級編號。
顯示目標/執行層級的相依性	`systemctl show -p "Requires"` `MY_TARGET`.target `systemctl show -p "Wants"` `MY_TARGET`.target 「Requires」 會列出硬相依性（必須解析的相依性），而 「Wants」 則列出軟相依性（可行時解析的相依性）。	無

13.3.2 系統啓動除錯

systemd 針對系統啓動過程提供了分析方法。您可以查看所有服務及其狀態的清單（而不必剖析 `/varlog/`）。systemd 還允許您掃描啓動程序，以瞭解每項服務耗費多長時間啓動。

13.3.2.1 檢閱服務啓動

若要檢閱自從系統開機以來已啓動的完整服務清單，請輸入指令 `systemctl`。這將列出所有使用中的服務，如下方所述（已縮短）。若要獲得特定服務的詳細資訊，請使用 `systemctl status MY_SERVICE`。

範例 13.1 列出使用中的服務

```
root # systemctl
UNIT                          LOAD   ACTIVE SUB       JOB DESCRIPTION
[...]
iscsi.service                 loaded active exited    Login and scanning of iSC+
kmod-static-nodes.service     loaded active exited    Create list of required s+
libvirtd.service              loaded active running   Virtualization daemon
nscd.service                  loaded active running   Name Service Cache Daemon
ntpd.service                  loaded active running   NTP Server Daemon
polkit.service                loaded active running   Authorization Manager
postfix.service               loaded active running   Postfix Mail Transport Ag+
rc-local.service              loaded active exited    /etc/init.d/boot.local Co+
rsyslog.service               loaded active running   System Logging Service
[...]
LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

161 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

若要限制為輸出無法啓動的服務，請使用 `--failed` 選項：

範例 13.2 列出失敗的服務

```
root # systemctl --failed
UNIT                    LOAD   ACTIVE SUB    JOB DESCRIPTION
apache2.service         loaded failed failed     apache
NetworkManager.service loaded failed failed     Network Manager
plymouth-start.service loaded failed failed     Show Plymouth Boot Screen

[...]
```

13.3.2.2 啓動時間除錯

為了對系統啓動時間除錯，systemd 提供了 `systemd-analyze` 指令。它會顯示總啓動時間以及按啓動時間排序的服務清單，還可以產生 SVG 圖，其中顯示各服務相對於其他服務所耗費的啓動時間。

列出系統啓動時間

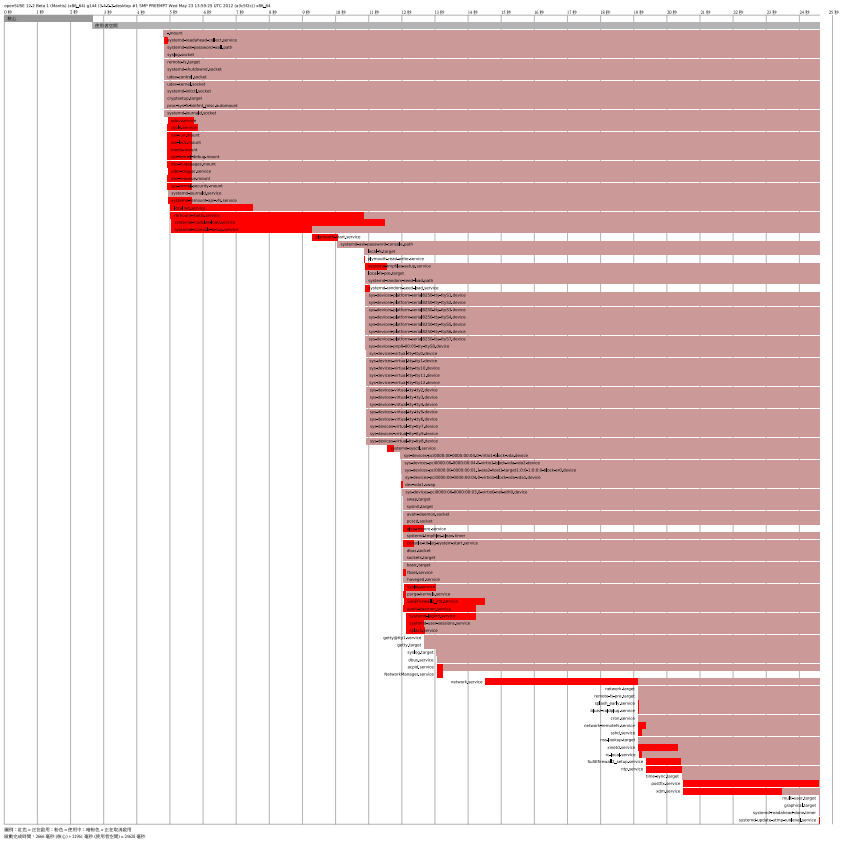
```
root # systemd-analyze
Startup finished in 2666ms (kernel) + 21961ms (userspace) = 24628ms
```

列出服務啓動時間

```
root # systemd-analyze blame
  6472ms systemd-modules-load.service
  5833ms remount-rootfs.service
  4597ms network.service
  4254ms systemd-vconsole-setup.service
  4096ms postfix.service
  2998ms xdm.service
  2483ms localnet.service
  2470ms SuSEfirewall2_init.service
  2189ms avahi-daemon.service
  2120ms systemd-logind.service
  1210ms xinetd.service
  1080ms ntp.service
[...]
    75ms fbset.service
    72ms purge-kernels.service
    47ms dev-vda1.swap
    38ms bluez-coldplug.service
    35ms splash_early.service
```

服務啓動時間圖

```
root # systemd-analyze plot > jupiter.example.com-startup.svg
```



13.3.2.3 檢閱完成的啓動程序

上述指令可用於檢閱已啓動的服務及其啓動所耗費的時間。如果您需要知道更多詳細資料，可以在開機提示中輸入下列參數，指示 `systemd` 詳細記錄完整的啓動程序：

```
systemd.log_level=debug systemd.log_target=kmsg
```

現在，`systemd` 會將記錄訊息寫入核心環緩衝區。該緩衝區可透過 `dmesg` 檢視：

```
dmesg -T | less
```

13.3.3 System V 相容性

Systemd 與 System V 相容，因此，您仍可以使用現有的 System V init 程序檔。但是，至少有一個已知問題會導致 System V init 程序檔不能依原樣與 Systemd 配合使用：透過 init 程序檔中的 `su` 或 `sudo` 以其他使用者身分啓動服務，會導致程序檔失敗，從而產生「拒絕存取」錯誤。

使用 `su` 或 `sudo` 變更使用者時，會啓動 PAM 工作階段。完成 init 程序檔後會終止此工作階段。因此，init 程序檔啓動的服務也會終止。若要解決此問題，請執行下列步驟：

1. 建立與 init 程序檔同名、副檔名為 `.service` 的服務檔案包裝程式。

```
[Unit]
Description=DESCRIPTION
After=network.target

[Service]
User=USER
Type=forking (1)
PIDFile=PATH TO PID FILE (1)
ExecStart=PATH TO INIT SCRIPT start
ExecStop=PATH TO INIT SCRIPT stop
ExecStopPost=/usr/bin/rm -f PATH TO PID FILE (1)

[Install]
WantedBy=multi-user.target (2)
```

   以適當的值取代 `UPPERCASE LETTERS` 中寫入的所有值。

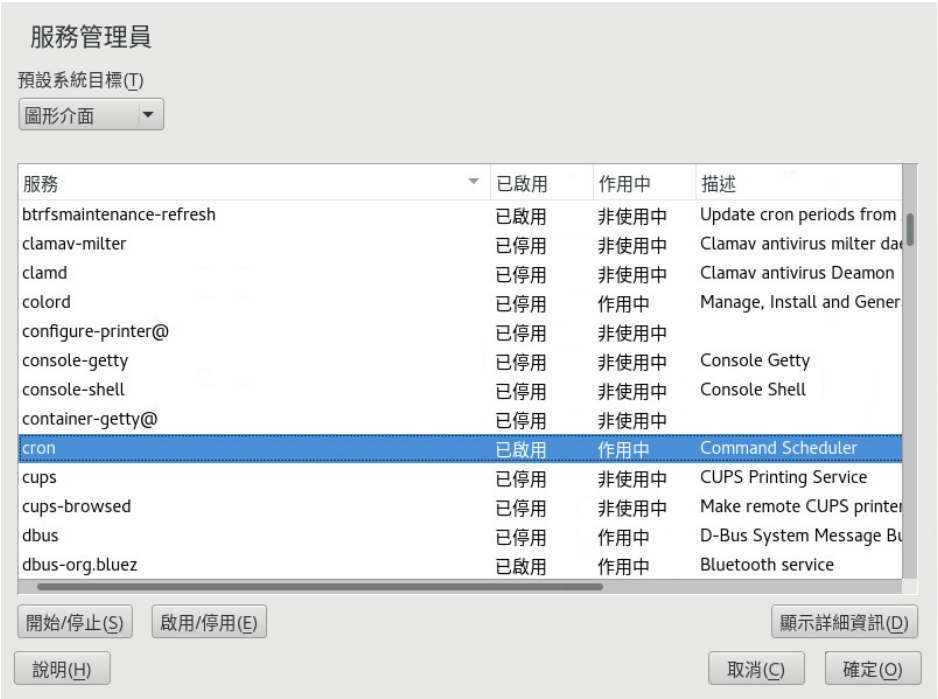
   (1) 選擇性 — 僅當 init 程序檔啓動精靈時才使用。

   (2) `multi-user.target` 在開機到 `graphical.target` 時也會啓動 init 程序檔。如果只應在開機到顯示管理員時才將它啓動，請在此處使用 `graphical.target`。

2. 使用 `systemctl start 應用程式` 啓動精靈。

13.4 使用 YaST 管理 服務

基本服務管理也可以透過 YaST 服務管理員模組實現。該模組不僅支援啓動、停止、啓用和停用服務，還可用於顯示服務的狀態以及變更預設目標。若要啓動 YaST 模組，請選取YaST › 系統 › 服務管理員。



圖形 13.1 服務管理員

變更預設系統目標

若要變更系統開機進入的目標，請從預設系統目標下拉方塊中選擇目標。最常用的目標是圖形介面（啓動圖形登入畫面）和多重使用者（以指令行模式啓動系統）。

啓動或停止服務

從表中選取服務。使用中欄顯示它目前是在執行中（使用中）還是未執行（非使用中）。其狀態可透過選擇啓動/停止進行切換。

如果啓動或停止服務，會變更其對目前執行中工作階段而言的狀態。若要在整個重新開機期間變更服務的狀態，您需要啓用或停用服務。

啓用或停用服務

從表中選取服務。已啓用欄顯示它目前是已啓用還是已停用。其狀態可透過選擇啓用/停用進行切換。

透過啓用或停用服務，可設定在開機期間是否啓動該服務（已啓用 或已停用）。此設定不影響目前的工作階段。若要變更該服務在目前工作階段中的狀態，您需要予以啓動或停止。

檢視狀態訊息

若要檢視服務的狀態訊息，請從清單中選取該服務，然後選擇 顯示詳細資料。您看到的輸出與 `systemctl -l status MY_SERVICE` 指令產生的輸出完全相同。

警告：錯誤的執行層級設定可能會造成系統損害

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

13.5 自訂 systemd

下列各節列出了 `systemd` 自訂的一些範例。

警告：避免覆寫自訂

請務必在 `/etc/systemd/` 中而絕非 `/usr/lib/systemd/` 中自訂。否則，下次更新 systemd 時會覆寫您的變更。

13.5.1 自訂服務檔案

systemd 服務檔案位於 `/usr/lib/systemd/system` 中。如果您要自訂服務檔案，請執行下列步驟：

1. 將要修改的檔案從 `/usr/lib/systemd/system` 複製到 `/etc/systemd/system`。保持檔案名稱不變。
2. 根據需要修改 `/etc/systemd/system` 中的副本。
3. 如需組態變更概觀，請使用 `systemd-delta` 指令。它會比較並識別哪些組態檔案覆寫其他組態檔案。如需詳細資料，請參閱 `sleha-init` man 頁面。

/etc/systemd 中修改過的檔案優先於 /usr/lib/systemd/system 中的原始檔案，前提是它們的檔案名稱相同。

13.5.2 建立「放入式」檔案

如果您只想在組態檔案中新增若干行或修改一小部分，可以使用「放入式」檔案。放入式檔案可讓您延伸單位檔案的組態，而不必編輯或覆寫單位檔案本身。

例如，若要變更位於 /usr/lib/systemd/system/FOOBAR.SERVICE 中 FOOBAR 服務的一個值，請依照以下步驟操作：

1. 建立名為 /etc/systemd/system/MY_SERVICE.service.d/ 的目錄。
   注意字尾為 .d。該目錄必須命名為要透過所放入之檔案修補的服務。
2. 在該目錄中，建立 WHATEVERMODIFICATION.conf 檔案。
   確保該檔案僅包含待修改值所在的行。
3. 將您所做的變更儲存到檔案中。它將用作原始檔案的延伸。

13.5.3 建立自訂目標

在 System V init SUSE 系統上並未使用執行層級 4，便於管理員自行建立執行層級組態。systemd 可讓您建立任意個自訂目標。建議您在開始時先在 graphical.target 等現有的目標上調整。

1. 將組態檔案 /usr/lib/systemd/system/graphical.target 複製到 /etc/systemd/system/MY_TARGET.target，並依據需要調整該檔案。
2. 上一步中複製的組態檔案已涵蓋該目標的必要的（「硬」）相依性。
   若要一併納入需要的（「軟」）相依項，請建立目錄 /etc/systemd/system/MY_TARGET.target.wants。
3. 對每個需要的服務，建立從 /usr/lib/systemd/system 連到 /etc/systemd/system/我的目標.target.wants 的符號連結。
4. 目標設定完畢後，重新載入 systemd 組態以便能夠使用新目標：

```
systemctl daemon-reload
```

13.6 進階用法

下列各節涵蓋進階主題，適用於系統管理員。如需更為進階的 systemd 文件，請參閱 Lennart Pöttering 針對管理員撰寫的 systemd 系列文章，網址為 http://0pointer.de/blog/projects↗。

13.6.1 清理暫存目錄

`systemd` 支援定期清理暫存目錄。將會自動移轉並啓用前一系統版本中的組態。`tmpfiles.d`（負責管理暫存檔案）將從 `/etc/tmpfiles.d/*.conf`、`/run/tmpfiles.d/*.conf` 和 `/usr/lib/tmpfiles.d/*.conf` 檔案中讀取其組態。`/etc/tmpfiles.d/*.conf` 中的組態將會覆寫其他兩個目錄中的相關組態（`/usr/lib/tmpfiles.d/*.conf` 是套件將其組態檔案儲存到的位置）。

組態格式為每個路徑一行，該行包含動作與路徑、（選擇性）模式、擁有權、期限和引數欄位，具體視動作而定。以下範例將取消連結 X11 鎖定檔案：

```
Type Path               Mode UID  GID  Age Argument
r    /tmp/.X[0-9]*-lock
```

若要取得 tmpfile 計時器的狀態：

```
systemctl status systemd-tmpfiles-clean.timer
systemd-tmpfiles-clean.timer - Daily Cleanup of Temporary Directories
 Loaded: loaded (/usr/lib/systemd/system/systemd-tmpfiles-clean.timer; static)
 Active: active (waiting) since Tue 2014-09-09 15:30:36 CEST; 1 weeks 6 days ago
   Docs: man:tmpfiles.d(5)
         man:systemd-tmpfiles(8)

Sep 09 15:30:36 jupiter systemd[1]: Starting Daily Cleanup of Temporary Directories.
Sep 09 15:30:36 jupiter systemd[1]: Started Daily Cleanup of Temporary Directories.
```

如需處理暫存檔案的詳細資訊，請參閱 `man 5 tmpfiles.d`。

13.6.2 系統記錄

第 13.6.8 節 「服務除錯」說明如何檢視給定服務的記錄訊息。然而，記錄的訊息顯示並不局限為服務記錄。您還可以存取和查詢 `systemd` 寫入的完整記錄訊息 — 亦即「日誌」。使用 `journalctl` 指令可顯示從最舊項目開始的完整記錄訊息。如需套用過濾器或變更輸出格式等選項的資訊，請參閱 `man 1 journalctl`。

13.6.3 快照

您可以使用 `isolate` 子指令將 `systemd` 的目前狀態儲存到指定的快照，日後可以回復到該狀態。此功能在測試服務或自訂目標時非常有用，因為它允許您隨時回到定義的狀態。快照僅在目前工作階段中可用，重新開機時將自動刪除。快照名稱必須以 `.snapshot` 結尾。

建立快照

```
systemctl snapshot MY_SNAPSHOT.snapshot
```

刪除快照

```
systemctl delete MY_SNAPSHOT.snapshot
```

檢視快照

```
systemctl show MY_SNAPSHOT.snapshot
```

啓動快照

```
systemctl isolate MY_SNAPSHOT.snapshot
```

13.6.4 載入核心模組

使用 `systemd`，可透過 `/etc/modules-load.d` 中的組態檔案，在開機時自動載入核心模組。該檔案應命名為 `MODULE.conf` 並包含以下內容：

```
# load module MODULE at boot time
MODULE
```

如果某個套件安裝了用於載入核心模組的組態檔案，該檔案將安裝到 `/usr/lib/modules-load.d`。如果存在兩個同名的組態檔案，將優先使用 `/etc/modules-load.d` 中的組態檔案。

如需詳細資訊，請參閱 `modules-load.d(5)` 線上文件。

13.6.5 載入服務之前執行必要動作

使用 System V 時，需要在載入服務之前執行的 init 動作必須在 `/etc/init.d/before.local` 中指定。systemd 不再支援此程序。如果您需要在啓動服務之前執行動作，請執行以下步驟：

載入核心模組

在 `/etc/modules-load.d` 目錄中建立一個 drop-in 檔案（如需語法，請參閱 `man modules-load.d`）

建立檔案或目錄，清理目錄，變更擁有權

在 `/etc/tmpfiles.d` 中建立一個 drop-in 檔案（如需語法，請參閱 `man tmpfiles.d`）

其他任務

從以下範本建立一個系統服務檔案，例如 `/etc/systemd/system/before.service`：

```
[Unit]
Before=NAME OF THE SERVICE YOU WANT THIS SERVICE TO BE STARTED BEFORE
[Service]
Type=oneshot
RemainAfterExit=true
ExecStart=YOUR_COMMAND
# beware, executable is run directly, not through a shell, check the man pages
# systemd.service and systemd.unit for full syntax
[Install]
# target in which to start the service
WantedBy=multi-user.target
#WantedBy=graphical.target
```

建立服務檔案後，應執行以下指令（以 `root` 身分）：

```
systemctl daemon-reload
systemctl enable before
```

每次修改服務檔案時，都需要執行：

```
systemctl daemon-reload
```

13.6.6 核心控制群組（cgroup）

在傳統 System V init 系統上不一定能將程序明確指派給繁衍它的服務。有些服務（例如 Apache）會繁衍許多協力廠商程序（例如 CGI 或 Java 程序），這些程序本身又會繁衍許多程序。這導致您很難明確指派，甚至根本無法明確指派。另外，服務在不當終止後，可能殘留部分子項保持活動狀態。

systemd 將每個服務放入它自己的 cgroup 中，從而解決此問題。cgroup 是一項核心功能，允許將程序及其所有子程序聚合至分層組織的群組中。systemd 根據相應的服務為每個 cgroup 命名。由於程序未經特許不得「 離開」其 cgroup，因此這樣可以有效地使用服務名稱標記該服務繁衍的所有程序。

若要列出屬於服務的所有程序，請使用指令 `systemd-cgls`。結果類似於以下範例（已縮短）：

範例 13.3 列出屬於服務的所有程序

```
root # systemd-cgls --no-pager
├─1 /usr/lib/systemd/systemd --switched-root --system --deserialize 20
├─user.slice
│ └─user-1000.slice
│   ├─session-102.scope
│   │ ├─12426 gdm-session-worker [pam/gdm-password]
│   │ ├─15831 gdm-session-worker [pam/gdm-password]
│   │ ├─15839 gdm-session-worker [pam/gdm-password]
│   │ ├─15858 /usr/lib/gnome-terminal-server

[...]

└─system.slice
  ├─systemd-hostnamed.service
  │ └─17616 /usr/lib/systemd/systemd-hostnamed
  ├─cron.service
  │ └─1689 /usr/sbin/cron -n
  ├─ntpd.service
  │ └─1328 /usr/sbin/ntpd -p /var/run/ntp/ntpd.pid -g -u ntp:ntp -c /etc/ntp.conf
  ├─postfix.service
  │ ├─ 1676 /usr/lib/postfix/master -w
  │ ├─ 1679 qmgr -l -t fifo -u
  │ └─15590 pickup -l -t fifo -u
  ├─sshd.service
  │ └─1436 /usr/sbin/sshd -D

[...]
```

如需 cgroup 的詳細資訊，請參閱《System Analysis and Tuning Guide》，第 9 章「Kernel Control Groups」。

13.6.7 終止服務（傳送信號）

如第 13.6.6 節 「核心控制群組（cgroup）」中所述，在 System V init 系統中不一定能將程序指派給其父服務，導致難以終止服務及其所有子項。未終止的子程序將保留為廢止程序。

systemd 的理念在於將每個服務限制在 cgroup 中，從而得以明確識別服務的所有子程序，因此可讓您傳送信號給這些程序中的每個程序。可使用 `systemctl kill` 將信號傳送給服務。如需可用信號清單，請參閱 `man 7 signals`。

將 `SIGTERM` 傳送給服務

`SIGTERM` 是傳送的預設信號。

```
systemctl kill MY_SERVICE
```

將信號傳送給服務

可使用 `-s` 選項指定應傳送的信號。

```
systemctl kill -s SIGNAL MY_SERVICE
```

選取程序

依預設，`kill` 指令會將信號傳送給指定 cgroup 的 `all` 程序。您可以將傳送目標限制為 `control` 或 `main` 程序。後者非常實用，如下例透過傳送 `SIGHUP` 強制服務重新載入其組態所示：

```
systemctl kill -s SIGHUP --kill-who=main MY_SERVICE
```

警告：不支援終止或重新啓動 D-Bus 服務

D-Bus 服務是 systemd 用戶端與做為 pid 1 執行的 systemd 管理員之間進行通訊的訊息匯流排。雖然 `dbus` 是個獨立的精靈，但它也是 init 基礎架構的組成部分。

在執行中的系統中終止或重新啓動 `dbus` 的效果類似於嘗試終止或重新啓動 pid 1。此操作將中斷 systemd 用戶端與伺服器間的通訊，並使大部分 systemd 功能不可用。

因此，不建議也不支援終止或重新啓動 `dbus`。

13.6.8 服務除錯

systemd 依預設不會過度記錄詳細資料。如果服務啓動成功，則不會產生任何輸出。如果啓動失敗，則會顯示簡短的錯誤訊息。不過，`systemctl status` 可讓您以不同方式對服務的啓動和作業進行除錯。

systemd 隨附自己的記錄機製（「 日誌」），可以記錄系統訊息，便於您一併顯示服務訊息與狀態訊息。`status` 指令的工作方式與 `tail` 相似，也可以採用不同的格式顯示記錄訊息，因此成為功能强大的除錯工具。

顯示服務啓動失敗

每當服務啓動失敗時，使用 `systemctl status MY_SERVICE` 可獲得詳細的錯誤訊息：

```
root # systemctl start apache2
Job failed. See system journal and 'systemctl status' for details.
root # systemctl status apache2
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
   Active: failed (Result: exit-code) since Mon, 04 Jun 2012 16:52:26 +0200; 29s ago
   Process: 3088 ExecStart=/usr/sbin/start_apache2 -D SYSTEMD -k start (code=exited,
 status=1/FAILURE)
   CGroup: name=systemd:/system/apache2.service

Jun 04 16:52:26 g144 start_apache2[3088]: httpd2-prefork: Syntax error on line
205 of /etc/apache2/httpd.conf: Syntax error on li...alHost>
```

顯示最後 `N` 條服務訊息

`status` 子指令的預設行為是顯示服務發出的最近 10 條訊息。若要變更要顯示的訊息數，請使用 `--lines=N` 參數：

```
systemctl status ntp
systemctl --lines=20 status ntp
```

以附加模式顯示服務訊息

若要顯示服務訊息的「 即時串流」，請使用 `--follow` 選項，其工作方式與 `tail -f` 相似：

```
systemctl --follow status ntp
```

訊息輸出格式

`--output=模式` 參數可讓您變更服務訊息的輸出格式。最重要的可用模式如下：

`short`

預設格式。顯示記錄訊息及易於理解的時戳。

`verbose`

完整輸出所有欄位。

`cat`

精簡輸出，不含時戳。

13.7 更多資訊

如需 systemd 的詳細資訊，請參閱下列線上資源：

首頁

http://www.freedesktop.org/wiki/Software/systemd

管理員的 systemd

Lennart Pöttering 是 systemd 的原著者之一，他撰寫了一系列部落格文章（寫本章時已有 13 篇），其網址為 http://0pointer.de/blog/projects 。

III 系統

14 64 位元系統環境的 32 位元和 64 位元應用程式

SUSE® Linux Enterprise Server 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 Enterprise Server 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 Enterprise Server 平台上的執行方式。

適用於 64 位元平台 POWER、z Systems 和 AMD64/Intel 64 的 SUSE Linux Enterprise Server 可讓現有的 32 位元應用程式「無需額外設定」 即可在 64 位元環境中執行。對應的 32 位元平台為：用於 POWER 的 ppc 和用於 AMD64/Intel 64 的 x86。。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。目前的 POWER 系統以 32 位元模式執行大部分應用程式，不過您可以執行 64 位元應用程式。

注意：不支援建構 32 位元應用程式

SUSE Linux Enterprise Server 不支援編譯 32 位元應用程式，僅提供 32 位元二進位檔案的執行時期支援。

14.1 執行期間支援

重要：不同應用程式版本之間的衝突

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

此規則的一個例外是 PAM（可插入驗證模組）。SUSE Linux Enterprise Server 在驗證程序中使用 PAM 做為使用者與應用程式之間的溝通層。在另外還可執行 32 位元應用程式的 64 位元作業系統上，一律要安裝兩個版本的 PAM 模組。

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

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

所有 64 位元程式庫和物件檔案都位於名為 `lib64` 的目錄。以往儲存在 `/lib` 和 `/usr/lib` 下的 64 位元物件檔案，現在放在 `/lib64` 和 `/usr/lib64` 目錄下。這表示在 `/lib` 和 `/usr/lib` 之下，有預留空間給 32 位元程式庫使用，因而兩種版本的檔案名稱能夠保持不變。

32 位元 `/lib` 目錄的子目錄，如果包含不取決於字數的資料內容，則不會移動。此配置與 LSB (Linux Standards Base) 以及 FHS (File System Hierarchy Standard) 相容。

14.2 核心規格

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

32 位元系統模擬的 64 位元核心呼叫，不支援系統程式使用的所有 API。這要視平台而定。因此，有少數應用程式 (例如 `lspci`) 必須在非 POWER 平台上以 64 位元程式的形式編譯，才能正常運作。在 IBM z Systems 上，並非所有 ioctl 都在 32 位核心 ABI 中可用。

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：核心可載入模組

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。

15 journalctl：查詢 systemd 日誌

systemd 取代 SUSE Linux Enterprise 12 中的傳統 init 程序檔後（請參閱第 13 章 「systemd 精靈」），引入了自身的記錄系統日誌。由於所有系統事件都將寫入到日誌中，因此，使用者不再需要執行基於 syslog 的服務。

日誌本身是 systemd 管理的系統服務，完整名稱為 systemd-journald.service。它會根據從核心、使用者程序、標準輸入和系統服務錯誤收到的記錄資訊，維護結構化的索引記錄，藉以收集和儲存記錄資料。systemd-journald 服務預設處於開啓狀態。

```
# systemctl status systemd-journald
systemd-journald.service - Journal Service
   Loaded: loaded (/usr/lib/systemd/system/systemd-journald.service; static)
   Active: active (running) since Mon 2014-05-26 08:36:59 EDT; 3 days ago
     Docs: man:systemd-journald.service(8)
           man:journald.conf(5)
 Main PID: 413 (systemd-journal)
   Status: "Processing requests..."
   CGroup: /system.slice/systemd-journald.service
           └─413 /usr/lib/systemd/systemd-journald
[...]
```

15.1 將日誌設為永久

依預設，日誌在 /run/log/journal/ 中儲存記錄資料。由於 /run/ 目錄具有易失本性，因此，在重新開機時會遺失記錄資料。若要永久儲存記錄資料，/var/log/journal/ 目錄必須存在且具有正確的擁有權和許可權，如此，systemd-journald 服務便可在其中儲存其資料。systemd 將為您建立該目錄，如果您執行以下操作，它將會切換到永久記錄：

1. 以 root 身分開啓 /etc/systemd/journald.conf 進行編輯。

   ```
   # vi /etc/systemd/journald.conf
   ```

2. 取消註解包含 Storage= 的行，並將它變更為

   ```
   [...]
   [Journal]
   Storage=persistent
   #Compress=yes
   ```

```
[...]
```

3. 儲存該檔案，然後重新啓動 systemd-journald：

```
systemctl restart systemd-journald
```

15.2 journalctl 的有用參數

本節介紹了一些可用來增强 journalctl 預設行為的常見有用選項。journalctl 手冊頁 man 1 journalctl 中介紹了所有參數。

提示：與特定可執行檔相關的訊息

若要顯示與特定可執行檔相關的所有日誌訊息，請指定該可執行檔的完整路徑：

```
journalctl /usr/lib/systemd/systemd
```

-f

只顯示最近的日誌訊息，另外，在將新的記錄項目新增到日誌時會列印這些新項目。

-e

列印訊息並跳轉到日誌末尾，以便在頁面巡覽區中顯示最新的項目。

-r

以反向順序列印記錄訊息，使最新的項目列在最前面。

-k

只顯示核心訊息。這等同於欄位比對 _TRANSPORT=kernel（請參閱第 15.3.3 節 「依據欄位過濾」）。

-u

只顯示指定 systemd 單元的訊息。這等同於欄位比對 _SYSTEMD_UNIT=UNIT（請參閱第 15.3.3 節 「依據欄位過濾」）。

```
# journalctl -u apache2
[...]
Jun 03 10:07:11 pinkiepie systemd[1]: Starting The Apache Webserver...
```

```
Jun 03 10:07:12 pinkiepie systemd[1]: Started The Apache Webserver.
```

15.3 過濾日誌輸出

如果不結合任何參數呼叫 `journalctl`，它將顯示日誌的完整內容，最舊的項目列在最前面。可按特定的參數和欄位過濾輸出。

15.3.1 依據開機編號過濾

`journalctl` 可以依據特定的系統開機編號過濾訊息。若要列出所有可用的開機，請執行

```
# journalctl --list-boots
-1 097ed2cd99124a2391d2cffab1b566f0 Mon 2014-05-26 08:36:56 EDT—Fri 2014-05-30 05:33:44 EDT
 0 156019a44a774a0bb0148a92df4af81b Fri 2014-05-30 05:34:09 EDT—Fri 2014-05-30 06:15:01 EDT
```

第一欄列出開機偏移：`0` 表示目前的開機，`-1` 表示上一次開機，`-2` 表示再上一次的開機，依此類推。第二欄包含開機 ID，其後是特定開機的限制時間戳記。

顯示目前開機中的所有訊息：

```
# journalctl -b
```

如果需要查看上一次開機的記錄訊息，請新增一個偏移參數。下面的範例將輸出上一次開機的訊息：

```
# journalctl -b -1
```

另一種方法是依據開機 ID 列出開機訊息。要實現此目的，請使用 _BOOT_ID 欄位：

```
# journalctl _BOOT_ID=156019a44a774a0bb0148a92df4af81b
```

15.3.2 依據時間間隔過濾

可透過指定開始日期和/或結束日期來過濾 `journalctl` 的輸出。日期規格應採用「2014-06-30 9:17:16」這樣的格式。如果省略時間部分，則會假設為午夜。如果省略秒，則會假設為「:00」。如果省略日期部分，則會假設為目前日期。您也可以不採用

數字表示法，而是指定關鍵字「yesterday」、「today」或「tomorrow」。它們表示當日前一天、當日或者當日後一天的午夜。如果指定「now」，則表示目前時間。您還可以指定以 `-` 或 `+` 為字首的相對時間，分別表示目前時間之前或之後的特定時間。

僅顯示從現在開始產生的新訊息，並持續更新輸出：

```
# journalctl --since "now" -f
```

顯示從昨天午夜到 3:20AM 的所有訊息：

```
# journalctl --since "today" --until "3:20"
```

15.3.3 依據欄位過濾

您可以按特定的欄位過濾日誌輸出。要比對的欄位語法為 `FIELD_NAME=MATCHED_VALUE`，例如 `_SYSTEMD_UNIT=httpd.service`。您可以在單個查詢中指定多個比對條件，以更精確地過濾輸出訊息。如需預設欄位的清單，請參閱 `man 7 systemd.journal-fields`。

顯示特定程序 ID 產生的訊息：

```
# journalctl _PID=1039
```

顯示屬於特定使用者 ID 的訊息：

```
# journalctl _UID=1000
```

顯示來自核心環緩衝區的訊息（與 `dmesg` 產生的結果相同）：

```
# journalctl _TRANSPORT=kernel
```

顯示來自服務之標準輸出或錯誤輸出的訊息：

```
# journalctl _TRANSPORT=stdout
```

僅顯示指定服務產生的訊息：

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service
```

如果指定了兩個不同的欄位，則僅顯示同時與兩個運算式相符的項目：

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1488
```

如果兩個比對參考了同一個欄位，則顯示與兩個運算式中任意一個相符的所有項目：

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service _SYSTEMD_UNIT=dbus.service
```

可以使用「+」分隔符將兩個運算式組合成一個邏輯「OR」。下面的範例將顯示來自程序 ID 為 1480 之 Avahi 服務程序的所有訊息，以及來自 D-Bus 服務的所有訊息：

```
# journalctl _SYSTEMD_UNIT=avahi-daemon.service _PID=1480 + _SYSTEMD_UNIT=dbus.service
```

15.4 調查 systemd 錯誤

本節將介紹一個簡單的範例，說明如何找出並修復 `systemd` 在 `apache2` 啓動期間報告的錯誤。

1. 嘗試啓動 apache2 服務：

   ```
   # systemctl start apache2
   Job for apache2.service failed. See 'systemctl status apache2' and 'journalctl -xn' for
    details.
   ```

2. 我們來看看該服務的狀態如何：

   ```
   # systemctl status apache2
   apache2.service - The Apache Webserver
      Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
      Active: failed (Result: exit-code) since Tue 2014-06-03 11:08:13 CEST; 7min ago
     Process: 11026 ExecStop=/usr/sbin/start_apache2 -D SYSTEMD -DFOREGROUND \
              -k graceful-stop (code=exited, status=1/FAILURE)
   ```

   導致錯誤的程序 ID 為 11026。

3. 顯示與程序 ID 11026 相關的詳細訊息：

   ```
   # journalctl -o verbose _PID=11026
   [...]
   MESSAGE=AH00526: Syntax error on line 6 of /etc/apache2/default-server.conf:
   [...]
   MESSAGE=Invalid command 'DocumenttRoot', perhaps misspelled or defined by a module
   [...]
   ```

4. 改正 `/etc/apache2/default-server.conf` 中的錯字，啓動 apache2 服務，然後列印其狀態：

   ```
   # systemctl start apache2 && systemctl status apache2
   apache2.service - The Apache Webserver
   ```

```
   Loaded: loaded (/usr/lib/systemd/system/apache2.service; disabled)
   Active: active (running) since Tue 2014-06-03 11:26:24 CEST; 4ms ago
  Process: 11026 ExecStop=/usr/sbin/start_apache2 -D SYSTEMD -DFOREGROUND
           -k graceful-stop (code=exited, status=1/FAILURE)
Main PID: 11263 (httpd2-prefork)
   Status: "Processing requests..."
   CGroup: /system.slice/apache2.service
           ├─11263 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11280 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11281 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11282 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           ├─11283 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
           └─11285 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -D [...]
```

15.5 Journald 組態

可透過修改 `/etc/systemd/journald.conf` 來調整 systemd-journald 服務的行為。本節只介紹基本的選項設定。如需完整的檔案描述，請參閱 `man 5 journald.conf`。請注意，若要使變更生效，必須使用以下指令重新啓動日誌

```
# systemctl restart systemd-journald
```

15.5.1 變更日誌大小限制

如果將日誌記錄資料儲存到永久位置（請參閱第 15.1 節 「將日誌設為永久」），這些資料最多可佔用 `/var/log/journal` 所在檔案系統空間的 10%。例如，如果 `/var/log/journal` 位於一個 30 GB 的 `/var` 分割區中，則日誌最多可佔用 3 GB 磁碟空間。若要變更此限制，請變更（並取消註解） `SystemMaxUse` 選項：

```
SystemMaxUse=50M
```

15.5.2 將日誌轉遞到 /dev/ttyX

您可以將日誌轉遞到終端機裝置，以便在偏好的終端機螢幕（例如 `/dev/tty12`）上顯式相關的系統訊息。將以下 journald 選項變更為

```
ForwardToConsole=yes
TTYPath=/dev/tty12
```

15.5.3 將日誌轉遞到 Syslog 工具

Journald 與傳統的 syslog 實作（例如 `rsyslog`）回溯相容。請務必滿足以下條件：

- 已安裝 rsyslog。

```
# rpm -q rsyslog
rsyslog-7.4.8-2.16.x86_64
```

- 已啓用 rsyslog 服務。

```
# systemctl is-enabled rsyslog
enabled
```

- 已在 `/etc/systemd/journald.conf` 中啓用轉遞至 syslog。

```
ForwardToSyslog=yes
```

15.6 使用 YaST 過濾 systemd 記錄

過濾 systemd 記錄的簡單方法（無需處理 journalctl 語法）是使用 YaST 記錄模組。使用 `sudo zypper in yast2-journal` 安裝該模組後，請在 YaST 中選取 系統 › Systemd 記錄 啓動該模組。也可以在指令行中輸入 `sudo yast2 journal` 來啓動該模組。

圖形 15.1 YAST SYSTEMD 記錄

模組將在表中顯示記錄項目。使用頂部的搜尋方塊可以搜尋包含特定字元的項目，這類似於使用 `grep`。若要依日期和時間、單位、檔案或優先程度過濾項目，請按一下變更過濾器，然後設定相應的選項。

16 基本網路功能

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。可以透過 YaST 設定使用網路卡進行的網路存取。也可使用手動方式來設定組態。本章僅討論基本機制及相關的網路組態檔案。

Linux 及其他 Unix 作業系統使用 TCP/IP 通訊協定。它不是單一網路通訊協定，而是能夠提供各種服務的網路通訊協定家族的一員。TCP/IP 通訊協定家族中的數種通訊協定中所列的通訊協定用於透過 TCP/IP 在兩個機器之間交換資料。由 TCP/IP 組合而成的各個網路形成了一個跨國網路，也稱為「網際網路」。

RFC 代表要求建議 (Request for Comments)。RFC 是描述作業系統及其應用程式的各種網際網路通訊協定和執行程序的文件。RFC 文件描述網際網路通訊協定的設定。如需有關 RFC 的詳細資訊，請參閱 http://www.ietf.org/rfc.html 。

TCP/IP 通訊協定家族中的數種通訊協定

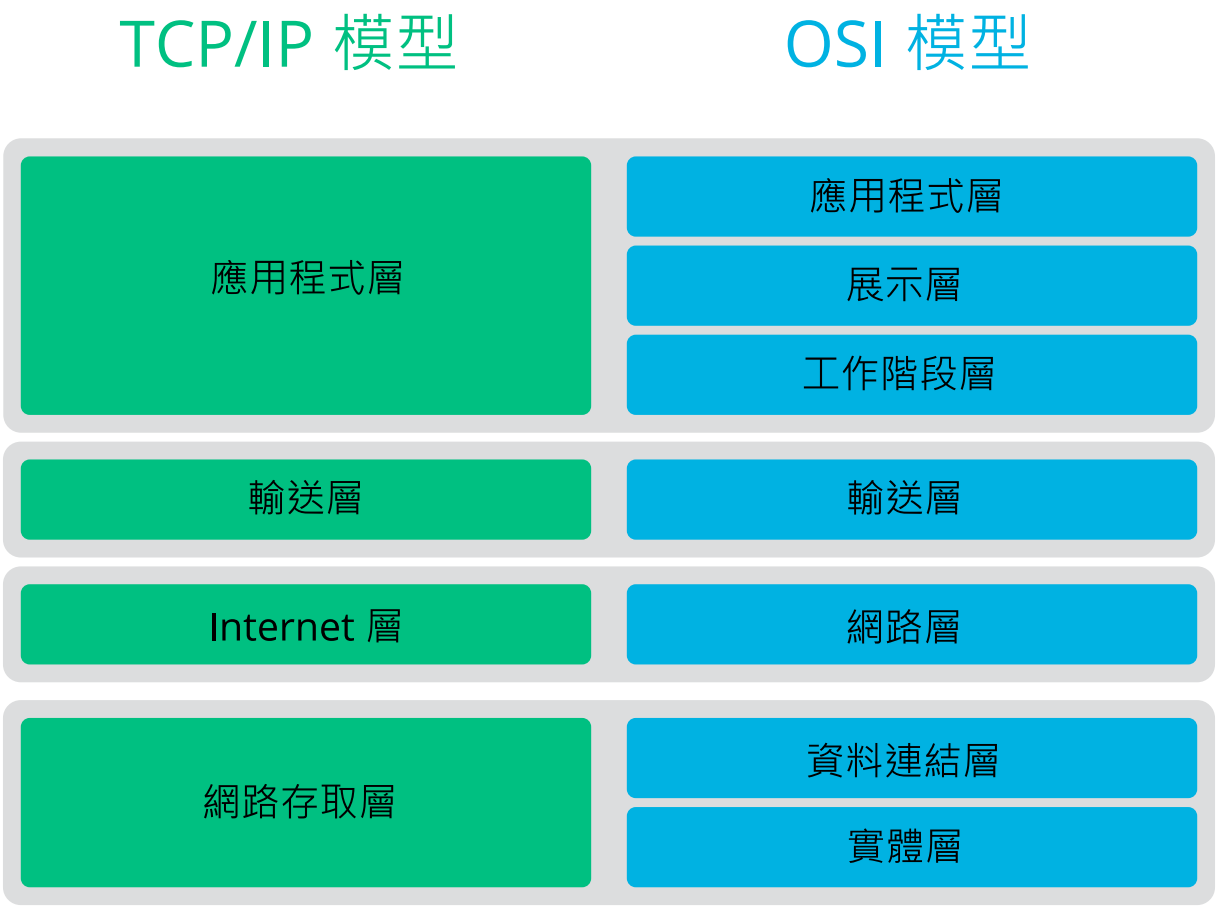
TCP
: 傳輸控制通訊協定：連接導向的安全通訊協定。傳輸的資料首先由應用程式當做資料流傳送出去，然後再由作業系統轉換為適當格式。資料送達目的地主機的相關應用程式時，使用的仍是最初傳送的原始資料流格式。TCP 可以判斷在傳輸期間是否有遺失或打亂了任何資料。只要是資料順序很重要的地方，就會執行 TCP。

UDP
: 使用者資料包通訊協定：無連接、不安全的通訊協定。要傳送的資料以應用程式產生的封包形式加以傳送。不會保證資料抵達接收者時的順序，而且可能會發生資料遺失的情況。UDP 適用以記錄為導向的應用程式。它的特點是延遲時間比 TCP 短。

ICMP
: 網際網路控制訊息通訊協定：這不是用於最終使用者的通訊協定，而是用來發出錯誤報告的特殊控制通訊協定，能夠控制參與 TCP/IP 資料傳輸的機器的行為。此外，它還提供特殊的回音模式，可以使用 ping 程式檢視。

IGMP
: 網際網路群組管理通訊協定：此通訊協定在實做 IP 多點廣播時控制機器行為。

如 圖形 16.1 「TCP/IP 的簡化層模型」 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 層由基礎硬體相依通訊協定提供支援，例如乙太網路。



圖形 16.1 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。資料連結層及實體層代表所使用的實體網路（例如乙太網路）。

幾乎所有的硬體通訊協定都是採用封包導向模式。要傳輸的資料會收集在封包中（無法一次全部傳送）。TCP/IP 封包的大小上限約為 64 KB。由於網路硬體可能有所限制，因此封包一般都很小。乙太網路上資料封包的大小上限約為 1500 個位元組。在乙太網路上傳送資料時，TCP/IP 封包的大小受此數量限制。如果傳送更多資料，則需要由作業系統傳送更多資料封包。

因為每層有自己指定的功能，關於每層的其他資訊必須儲存於資料封包中。這些資訊放在封包的「標頭」中。每層皆在產生的封包前端附加小的資料區塊，稱為通訊協定標頭。有關在乙太網路纜線上傳送的 TCP/IP 資料封包範例，請參閱圖形 16.2 「TCP/IP 乙太網路封包」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 16.2 TCP/IP 乙太網路封包

當應用程式在網路上傳送資料時，資料會經過每一層，除實體層外，全部在 Linux 核心執行。每層都負責準備資料使其能夠傳送到下一層。最底層最後要負責傳送資料。接收到資料時則反轉執行整個程序。就如同洋蔥的層級一般，在每層中，會從已傳輸的資料上移除通訊協定標頭。最後，傳輸層負責讓目的地端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 網路還是透過 56-Kbit/s 數據機纜線進行傳輸，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。

16.1 IP 位址與路由

在此節中的討論僅限於 IPv4 網路。如需有關 IPv6 通訊協定（IPv4 的後繼者）的資訊，請參閱第 16.2 節 「IPv6 --下一代的網際網路」。

16.1.1 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）通常按範例 16.1 「寫入 IP 位址」中第二列所述寫入。

範例 16.1 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。每個 IP 位址在全球範圍內只能使用一次。此規則有例外狀況，但下文中並未提及。

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，事實證明此系統太過死板，因此已停止使用。現在，則是使用無類別路由 (classless routing)，即 CIDR (無類別網域間路由，classless interdomain routing)。

16.1.2 網路遮罩與路由

網路遮罩用於定義子網路的位址範圍。如果兩台主機位於相同的子網路內，它們之間可直接連接。如果它們不在同一個子網路內，則需要用於處理子網路所有流量之閘道的位址。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 16.2 「連結 IP 位址到網路遮罩」。網路遮罩由 32 位元組成，可辨認 IP 位址有多少屬於網路。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。對於所有值為 `0` 的位元，標示其屬於子網路內。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` 位元組成，也可以計算網路遮罩內的位元數。在範例 16.2 「連結 IP 位址到網路遮罩」中，第一個 24 位元的網路也可寫成 `192.168.0.0/24`。

範例 16.2 連結 IP 位址到網路遮罩

```
IP address (192.168.0.20):  11000000 10101000 00000000 00010100
```

```
Netmask   (255.255.255.0):  11111111 11111111 11111111 00000000
----------------------------------------------------------------
Result of the link:         11000000 10101000 00000000 00000000
In the decimal system:           192.     168.       0.       0

IP address (213.95.15.200): 11010101 10111111 00001111 11001000
Netmask    (255.255.255.0): 11111111 11111111 11111111 00000000
----------------------------------------------------------------
Result of the link:         11010101 10111111 00001111 00000000
In the decimal system:           213.      95.      15.       0
```

舉另外一個例子：使用相同乙太網路纜線連接的所有機器，通常位於同一個子網路中，而且可以直接存取。即使以交換器或橋接器實際分配子網路時，仍然可以直接連接這些主機。

位於本地子網路外的 IP 位址只能在設定目標網路的閘道時，才能與本地通訊。在大部分的狀況下，只能有一個閘道來處理所有對外的通訊。但是，您也可以為不同的子網路，設定多個閘道。

如果已經設定閘道，所有的外部 IP 封包會傳送到適當的閘道。然後此閘道會試圖以同樣方式傳送封包--主機對主機--直到連結到目標主機或封包 TTL（持續時間）過期。

特定位址

基本網路位址
: 這是網路遮罩「及」網路中的任何位址，如 `Result` 下的範例 16.2 「連結 IP 位址到網路遮罩」 所顯示。此位址不能指定給任何主機。

廣播位址
: 這可以解釋為：「存取此子網路中的所有主機。」。若要產生此位址，網路遮罩會以二進位格式反轉，連結到具有邏輯 OR 的基本網路位址。因此以上範例會得到 192.168.0.255。此位址無法指派給任何主機。

本地主機
: 位址 `127.0.0.1` 是指定到每個主機上的「迴路裝置」(loopback device)。使用此位址以及完整迴路網路 `127.0.0.0/8` 中的所有位址（使用 IPv4 定義），可以設定與您自己機器之間的連接。如果使用 IPv6，則只有一個迴路位址（`::1`）。

因為 IP 位址在全世界必須是唯一的，您不能選取隨機位址。如果要設立私人 IP 結構的網路，有三種位址網域可以使用。這些將無法從其他網際網路取得連結，因為他們無法透過網路傳送。這些位址網域在 RFC 1597 指定並列於 表格 16.1 「私人 IP 位址網域」 中。

表格 16.1 私人 IP 位址網域

網路/網路遮罩	網域
10.0.0.0/255.0.0.0	10.x.x.x
172.16.0.0/255.240.0.0	172.16.x.x — 172.31.x.x
192.168.0.0/255.255.0.0	192.168.x.x

16.2 IPv6 --下一代的網際網路

重要：IBM z Systems：IPv6 支援

IBM z Systems 硬體的 CTC 和 IUCV 網路連接不支援 IPv6。

由於全球資訊網（WWW）的出現，過去十五年內，越來越多的電腦透過 TCP/IP 進行通訊，網際網路的規模發生了爆炸性的增長。自從 CERN 的 Tim Berners-Lee（http://public.web.cern.ch）於 1990 年發明 WWW 以來，網際網路主機的數量從幾千台成長為幾百萬台。

如前面所述，IPv4 位址僅由 32 個位元組成。而且，損失了一些 IP 位址 — 由於組織網路的方式，使得這些 IP 位址無法使用。您的子網路中可用的位址數量是位元數的平方減 2。例如，子網路有 2 個、6 個或 14 個位址可用。例如，如果要連接 128 個主機到網際網路，則子網路需要 256 個 IP 位址，但是其中只有 254 個可用，因為子網路結構本身需要用掉兩個 IP 位址：廣播與基本網路位址。

在目前的 IPv4 通訊協定之下，DHCP 或 NAT（網路位址轉譯，Network Address Translation）是典型的機制，可用來避免位址可能不足的問題。搭配保持私人和公用位址空間分開的方式，能夠減輕短少的情形。其中產生的問題是在於其組態，設定麻煩且難於維護。若要在 IPv4 網路中設定主機，需要一些位址項目，如主機自己的 IP 位址、子網路遮罩、閘道位址，可能還需要名稱伺服器位址。您必須知道所有這些項目，且無法從其他地方取得。

透過 IPv6，位址短少及繁複組態的情形應該都成為過去式了。以下小節說明更多 IPv6 改善的部分及它帶來的好處，還有關於從舊通訊協定轉移到新通訊協定的資訊。

16.2.1 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 個位元。這樣提供了數以千兆的 IP 位址。

然而，IPv6 位址不僅是在長度方面與之前的位址不同；這些位置的內部結構也不同，可能包含有關系統及其所屬網路的更明確資訊。有關 IPv6 的詳細資訊，可以在第 16.2.2 節 「定址類型與結構」 中找到。

以下列出了新通訊協定的其他優勢：

自動設定

IPv6 讓網路能夠「隨插即用」(plug and play)，表示新設定的系統不需經過任何手動設定，即可整合到（區域）網路。新主機使用其自動設定組態機制，從鄰近的路由器上可用的資訊取得自己的位址，依賴的是稱為「網路芳鄰探查」(Neighbor Discovery, ND) 的通訊協定。這個方法不需要管理員的介入，而且不需要維護分配位址的中央伺服器，這是 IPv4 的另一個優勢，因為自動位址分配需要 DHCP 伺服器。

但是，如果路由器連接到交換器，則路由器應傳送具有旗標的週期性通告，告知網路中的主機彼此如何進行互動。如需詳細資訊，請參閱 RFC 2462、`radvd.conf(5)` man 頁面以及 RFC 3315。

機動性

IPv6 能夠同時將數個位址指定給一個網路介面。這使得使用者能輕鬆地存取多個網路，可媲美行動電話服務公司提供的國際漫遊服務。當您出國時，進入相應區域後行動電話會自動登入國外服務，因此無論您身在何處，別人都可以用同一個號碼聯絡到您，您也可以像在國內一樣撥打電話。

安全通訊

使用 IPv4，網路安全性是附加的功能。IPv6 包括 IPSec 為其中一個核心功能，允許系統在安全的通道上進行通訊，避免網際網路上的外人竊聽。

反向相容性

實際上，不可能一次將整個網際網路從 IPv4 切換到 IPv6。因此關鍵在於，兩個通訊協定不僅能夠共存於網際網路上，而且能夠共存於一個系統中。使用相容位址（IPv4 位址可以輕鬆轉換為 IPv6 位址）和多個通道可以確保這一點。請

參閱第 16.2.3 節 「IPv4 與 IPv6 的共存」。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。

透過多點傳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 使伺服器能夠透過多點傳播對主機定址（即將多個主機做為群組的一部分定址），因而提供了更精細的方法。這種方法與透過廣播對所有主機定址，或透過單點傳播個別對每個主機定址均不同。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（「所有名稱伺服器多點傳播群組」）或所有路由器（「所有路由器多點傳播群組」）。

16.2.2 定址類型與結構

如上所述，目前的 IP 通訊協定存在兩個重要限制：IP 位址日益短缺，並且設定網路、維護路由表的任務變得越來越複雜繁重。IPv6 透過擴充位址空間到 128 個位元解決了第一個問題。透過引入階層位址結構，結合尖端網路位址配置技術及多重定址功能(將數個位址指定給同一個裝置，進而支援對多個網路的存取)，第二個問題也得到緩解。

使用 IPv6 時，瞭解三種不同類型的位址是很有用的：

單點傳播

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。

多點傳播

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。多點傳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。位址與位址之間以冒號（:）分隔。指定欄位內的任何前導零位元組可以刪除，但是欄位內或尾端的零不能刪除。另一個慣例是多於四個連續的零位元組可能會摺疊成兩個冒號。然而，每個位址只允許一個 :: 。這類的簡略的標記法，顯示於範例 16.3 「範例 IPv6 位址」 中，其中三行都是代表相同的位址。

範例 16.3 範例 IPV6 位址

```
fe80 : 0000 : 0000 : 0000 : 0000 : 10 : 1000 : 1a4
fe80 :    0 :    0 :    0 :    0 : 10 : 1000 : 1a4
fe80 :                            : 10 : 1000 : 1a4
```

IPv6 位址的每個部分都有定義的功能。第一個位元組形成字首，指定位址類型。中間的部分是位址的網路部分，但是可能不會使用。位址的尾端形成主機部分。透過 IPv6，在位址尾端的斜線後表示字首的長度，可以定義網路遮罩。位址如 範例 16.4 「指定字首長度的 IPv6 位址」 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在另一個子網路。

範例 16.4 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。各種 IPv6 字首中列出了其中的一部分。

各種 IPV6 字首

`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，如迴路裝置的位址，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。類似 IPv4，可以指定介面來構成特定子網路的部分。目前有下列位址空間：`2001::/16`（產品品質位址空間）與 `2002::/16`（6to4 位址空間）。

`fe80::/10`
: 連結本地位址。不應路由具有這種字首的位址，因此僅能從相同的子網路內進行連接。

`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`

這些都是多點傳播位址。

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路由資訊。

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，由硬體製造商在裝置中編碼固定識別碼，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，就可以給沒有 MAC 的介面指定 `EUI-64` 記號，例如以 PPP 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）

在首次啓始化介面時（此時尚無法透過其他方法確定位址），主機會使用此位址做為其來源位址。

`::1`（迴路）

迴路裝置的位址。

IPv4 相容位址

IPv6 位址是由 IPv4 位址以及由 96 個零位元組成的字首形成的。這類相容性位址用於通道（請參閱第 16.2.3 節 「IPv4 與 IPv6 的共存」），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。不應將此類型之來源位址或目標位址的封包路由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路由到其他子網路，但是不能路由到更寬廣的網際網路 — 不得跨越組織內部網路。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 ID 以及指定子網路 ID 的 16 位元欄位。同樣地，餘下的將填入零位元組。

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。其中一個網路可以使用 MAC 和已知的字首以完全自動化的方式設定，在啓用 IPv6（使用連結本地位址）後可以連接區域網路上的所有主機。利用形成位址部分的 MAC，全球使用的任何 IP 位址都成為唯一的。位址的唯一變數部分，是指定網站拓撲和公用拓撲，該部分視主機目前正在操作的實際網路而定。

如果主機要在不同的網路之間往返，至少需要兩個位址。其中一個，即主位址，不僅包含了介面識別碼，也包含了其通常所屬之主網路（及其對應字首）的識別碼。主位址是靜態位址，因此它通常不會變更。儘管如此，預定要送到行動主機的所有封包，還是可以傳送到主位址，無論是在主網路或其他外部網路中操作。這可藉由 IPv6 全新功能來達成，如「無狀態自動設定」與「網路芳鄰探查」。除了其主位址外，行動主機也取得一或多個其他的位址，這些位址屬於漫遊的外部網路。這些外部網路稱為 care-of 位址。主網路具有封包在外部漫遊時轉寄預定要送到主機的裝置。在 IPv6 環境中，這個任務是由主代理程式執行的，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。

16.2.3 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」。但這仍然沒有解決啓用了 IPv6 的主機如何與 IPv4 主機通訊，以及應如何透過目前的網路（絕大部分都以 IPv4 為基礎）傳輸 IPv6 封包的問題。最佳的解決方案是提供通道及相容性位址（請參閱第 16.2.2 節 「定址類型與結構」）。

IPv6 主機或多或少孤立於（全球）IPv4 網路間，可透過通道通訊：IPv6 封包會被包成 IPv4 封包，在 IPv4 網路中移動。兩個 IPv4 主機之間的連接，稱為「通道」。為實現此功能，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，透過能夠多點傳播的 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，這個方法不能過多地延伸，而且不易推廣，因為 IP 多點傳播在網際網路上尚未普及。所以，它僅能為啓用多點傳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，用此方法在隔離的 IPv6 主機與網際網路之間通訊時存在一些問題。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。

16.2.4 設定 IPv6

若要設定 IPv6，通常不需要在個別工作站中做任何變更。IPv6 預設會開啓這個選項。若要在安裝的系統上停用或啓用 IPv6，請使用 YaST 網路設定模組。在全域選項索引標籤上，根據需要核取或取消核取啓用 IPv6選項。若要暫時啓用直至下次重新開機，請以 `root` 身分輸入 `modprobe -i ipv6`。載入 IPv6 模組後無法將其卸載。

由於 IPv6 的自動組態概念，網路卡會在連結本地網路中指定一個位址。工作站通常不會進行路由表格管理。工作站可使用「路由器通告通訊協定」，向網路路由器查詢應使用的前置號碼和閘道。可使用 radvd 程式來設定 IPv6 路由器。此程式會通知工作站該 IPv6 位址應使用的前置號碼和路由器。或者，也可使用 zebra/quagga 自動設定位址和路由的組態。

如需如何使用 `/etc/sysconfig/network` 檔案設定各種通道類型的資訊，請參閱 `ifcfg-tunnel` 的 man 頁面（`man ifcfg-tunnel`）。

16.2.5 更多資訊

上述綜覽沒有完整地涵蓋 IPv6 主題。如需更深入的探討這種新的通訊協定，請參閱以下線上文件和書籍：

http://www.ipv6.org/
: 所有有關 IPv6 的入門資訊。

http://www.ipv6day.org
: 啓動您 IPv6 網路所需的所有資訊。

http://www.ipv6-to-standard.org/
: 啓用 IPv6 產品的清單。

http://www.bieringer.de/linux/IPv6/
: 在此處可找到 Linux IPv6-HOWTO 和許多與此主題相關的連結。

RFC 2460
: 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）。

16.3 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)。所有名稱元件之間以句號分隔，它們共同組成一個階層系統。但是，名稱階層與上述的 IP 位址階層無關。

考慮使用完整名稱，如 `jupiter.example.com`，以 `hostname.domain` 格式來寫入。完整名稱，也稱為完全合格的網域名稱 (FQDN)，由主機名稱和網域名稱（`example.com`）組成。後者也包含了「最上層網域」(top level domain) 或 TLD（`com`）。

TLD 指定因為過去的緣故變得相當混淆。習慣上，美國使用三個字母的網域名稱。全世界的其他國家，則是使用兩個字母的 ISO 國際代碼為標準。除此之外，2000 年引入了較長的 TLD，代表特定活動範圍（例如，`.info`、`.name`、`.museum`）。

在早期的網際網路 (1990 年前)，是使用檔案 `/etc/hosts` 儲存網際網路上所有機器的代表名稱。這種方式，對於連接到網際網路、快速增長的電腦數量層面而言，很快就證實是不切實際的。基於此因素，又開發出分散式的資料庫，以廣泛分散的方式來儲存主機名稱。這種資料庫與名稱伺服器類似，沒有有關網際網路上所有主機的立即可用資料，但是可以分散要求到其他名稱伺服器。

階層的最上層是由「root 名稱伺服器」(root name server) 所使用。這些 root 名稱伺服器管理最上層網域，且由「網路資訊中心」(Network Information Center, NIC) 負責管理。每個 root 名稱伺服器知道負責指定最上層網域的名稱伺服器。有關最上層網域 NIC 的資訊可從 http://www.internic.net 取得。

DNS 的功能不只是解析主機名稱。名稱伺服器也知道哪個主機，即「郵件交換器」(Mail Exchanger, MX)，負責接收該領域的電子郵件。

若要讓您的機器能夠解析 IP 位址，它必須知道至少一個名稱伺服器及其 IP 位址。使用 YaST 可輕鬆指定此類名稱伺服器。如需使用 SUSE® Linux Enterprise Server 設定名稱伺服器存取組態的資訊，請參閱第 16.4.1.4 節 「設定主機名稱和 DNS」。關於設定您自己的名稱伺服器的資訊，請參閱第 25 章 「網域名稱系統」。

`whois` 通訊協定與 DNS 密切相關。利用此程式，可快速找出負責指定網域的伺服器。

注意：MDNS 和 .local 網域名稱

`.local` 最上層網域將被解析程式視為連結本地網域。DNS 要求將做為多點傳播 DNS 要求予以傳送，而非通常的 DNS 要求。如果已在名稱伺服器組態中使用了 `.local` 網域，則必須在 `/etc/host.conf` 中關閉此選項。如需詳細資訊，請參閱 `host.conf` 手冊頁。

如果要在安裝期間關閉 MDNS，請使用 `nomdns=1` 做為開機參數。

如需有關多路廣播 DNS 的詳細資訊，請參閱 http://www.multicastdns.org↗。

16.4 使用 YaST 手動設定網路連接

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。要更瞭解手動網路組態的綜覽，請參閱第 16.5 節 「手動設定網路連接」。

已建立連結的所有網路介面（已連接網路線纜）將自動進行設定。可隨時在安裝的系統上設定其他的硬體。以下幾節將說明 SUSE Linux Enterprise Server 支援之所有網路連接類型的網路組態。

提示：IBM z Systems：可熱插拔網路卡

IBM z Systems 平台支援可熱插拔網路卡，但不支援這些網路卡透過 DHCP 自動進行網路整合（與在 PC 上的情況相同）。完成偵測後，接著以手動設定介面。

16.4.1 使用 YaST 設定網路卡

若要在 YaST 中設定以太網路卡或 Wi-Fi/藍芽卡，請選取系統 › 網路設定。啓動模組後，YaST 將顯示網路設定對話方塊，其中包含四個索引標籤：全域選項、綜覽、主機名稱/DNS和路由。

使用全域選項索引標籤可設定一般網路選項，例如網路設定方法、IPv6 和一般 DHCP 選項。如需詳細資訊，請參閱第 16.4.1.1 節 「設定全域網路選項的組態」。

綜覽索引標籤包含有關已安裝網路介面與組態的資訊。此處會列出所有正確偵測到之網路卡的名稱。在此對話方塊中，您可以手動設定新網路卡、移除或變更其組態。若要手動設定未自動偵測到的網路卡，請參閱第 16.4.1.3 節 「設定未偵測到的網路卡」。若要變更已設定卡的組態，請參閱第 16.4.1.2 節 「變更網路卡組態」。

使用主機名稱/DNS索引標籤可設定機器的主機名稱以及要使用伺服器的名稱。如需詳細資訊，請參閱第 16.4.1.4 節 「設定主機名稱和 DNS」。

路由索引標籤用於設定路由組態。如需相關資訊，請參閱第 16.4.1.5 節 「設定路由」。

圖形 16.3 設定網路組態

16.4.1.1 設定全域網路選項的組態

使用 YaST 網路設定模組的全域選項索引標籤，可以設定重要的全域網路選項，例如使用 NetworkManager、IPv6 和 DHCP 用戶端選項。這些設定適用於所有網路介面。

注意：Workstation Extension 提供了 NetworkManager

現在，Workstation Extension 提供有 NetworkManager。若要安裝 NetworkManager，請啓用 Workstation Extension 儲存庫，然後選取 NetworkManager 套件。

在網路設定方法中，選擇管理網路連線的方式。若希望 NetworkManager 桌面 applet 管理所有介面的連線，請選擇NetworkManager 服務。NetworkManager 最適合用於在多個有線和無線網路之間進行切換。如果您執行的不是桌面環境，或者您的電腦是 Xen 伺服器、虛擬系統或者會在網路中提供 DHCP 或 DNS 等網路服務，請使用Wicked 服務方法。如果使用 NetworkManager，則應使用 `nm-applet` 設定網路選項，且網路設定模組的綜覽、主機名稱/DNS和路由索引標籤都會處於停用狀態。如需 NetworkManager 的詳細資訊，請參閱 SUSE Linux Enterprise Desktop 文件。

在IPv6 通訊協定設定中，選擇是否要使用 IPv6 協定。可以同時使用 IPv6 和 IPv4。預設會啓用 IPv6。但是，在不使用 IPv6 通訊協定的網路中，停用 IPv6 通訊協定時回應較快。若要停用 IPv6，請停用啓用 IPv6。如果停用了 IPv6，核心將不再自動載入 IPv6 模組。重新開機後會套用此設定。

在DHCP 用戶端選項中，設定 DHCP 用戶端的選項。在一個網路中，每個 DHCP 用戶端的DHCP 用戶端識別碼均不能相同。若將其留為空白，則預設會使用網路介面的硬體位址。但是，如果您要使用同一個網路介面執行多個虛擬機，因此會使用同一個硬體位址，則請在此處指定不限格式的唯一識別碼。

要傳送的主機名稱指定當 DHCP 用戶端將訊息傳送至 DHCP 伺服器時，主機名稱選項欄位所使用的字串。有些 DHCP 伺服器會根據此主機名稱（動態 DNS）更新名稱伺服器區域（正向和反向記錄）。此外，有些 DHCP 伺服器需要要傳送的主機名稱選項欄位包含用戶端傳送之 DHCP 訊息中的特定字串。如果保留 `AUTO`，將傳送目前的主機名稱（即 `/etc/HOSTNAME` 中定義的主機名稱）。將選項欄位留為空白則不會傳送主機名稱。

如果您不希望根據 DHCP 的資訊變更預設路由，請停用透過 DHCP 變更預設路由。

16.4.1.2 變更網路卡組態

若要變更網路卡的組態，請在 YaST 的網路設定 › 綜覽中偵測到的網路卡清單中選取網路卡，然後按一下編輯。網路卡設定對話方塊隨即出現，您可以使用一般、位址和硬體索引標籤調整網路卡的組態。

16.4.1.2.1 設定 IP 位址

在網路卡設定對話方塊的位址索引標籤中，可以設定網路卡的 IP 位址或確定其 IP 位址的方式。系統支援 IPv4 和 IPv6 兩種位址。您可以為網路卡設定無 IP 位址(適用於 Bonding 裝置)、靜態指定的 IP 位址(IPv4 或 IPv6)，也可以透過DHCP或/與Zeroconf為其指定動態位址。

若要使用動態位址，請選擇是使用僅限 DHCP 版本 4(適用於 IPv4)、僅限 DHCP 版本 6(適用於 IPv6)，還是DHCP 版本 4 與 6。

若情況適合，系統會將安裝時第一個可用的已連結網路卡自動設定為使用透過 DHCP 設定的自動位址。

注意：IBM z Systems 和 DHCP

在 IBM z Systems 平台上，只有具有 MAC 位址的網路卡才支援基於 DHCP 的位址組態。此情況只適用於 OSA 和 OSA 高速網路卡。

若您用的是 DSL 連線而非 ISP（網際網路服務提供者）指定的靜態 IP，還應該使用 DHCP。若您決定使用 DHCP，請在 YaST 網路卡組態模組中開啓網路設定對話方塊，於全域選項索引標籤的DHCP 用戶端選項中設定詳細資料。若您將虛擬主機設定為透過同一個介面與不同的主機進行通訊，則需要使用DHCP 用戶端識別碼來分辨它們。

DHCP 對於用戶端組態是不錯的選擇，但不適用於伺服器組態。若要設定靜態 IP 位址，請如下執行：

1. 在 YaST 網路卡組態模組的綜覽索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 在位址索引標籤中，選擇靜態指定的 IP 位址。
3. 輸入IP 位址。使用 IPv4 和 IPv6 位址都可以。在子網路遮罩中輸入網路遮罩。若使用 IPv6 位址，請以 `/64` 格式使用子網路遮罩做為字首長度。
   您還可以選擇為此位址輸入完全合法的主機名稱，它將會寫入 `/etc/hosts` 組態檔案中。
4. 按下一步。
5. 若要啓用組態，請按一下確定。

注意：介面啓動和連結偵測

在啓動網路介面期間，`wicked` 會檢查載體，並且只有在偵測到連結之後，才會套用 IP 組態。如果不論連結狀態如何，您都需要套用該組態（例如，您要測試監聽某個位址的服務），則可以在 `/etc/sysconfig/network/ifcfg` 內的介面組態檔案中新增變數 `LINK_REQUIRED=no`，以跳過連結偵測。

另外，您可以使用變數 `LINK_READY_WAIT=5` 來指定等待連結的逾時值（以秒為單位）。

如需 `ifcfg-*` 組態檔案的詳細資訊，請參閱第 16.5.2.5 節 「/etc/sysconfig/network/ifcfg-*」和 `man 5 ifcfg`。

若使用靜態位址，系統將不會自動設定名稱伺服器和預設閘道。若要設定名稱伺服器，請依第 16.4.1.4 節 「設定主機名稱和 DNS」 中的說明進行。若要設定閘道，請依第 16.4.1.5 節 「設定路由」 中的說明進行。

16.4.1.2.2 設定多個位址

一個網路裝置可擁有多個 IP 位址。

注意：別名是相容性功能

這些所謂的別名或標籤只能各自用於 IPv4。如果是 IPv6，則會被忽略。使用 `iproute2` 網路介面時可以使用一或多個地址。

若要使用 YaST 設定網路卡的其他地址，請執行以下步驟：

1. 在 YaST網路設定對話方塊的綜覽索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 在位址 › 其他位址索引標籤中，按一下新增。
3. 輸入IPv4 地址標籤、IP 位址和網路遮罩。別名中不要包含介面名稱。
4. 若要啓用組態，請確認設定。

16.4.1.2.3 變更裝置名稱和 Udev 規則

可以在網路卡正在使用中時變更它的裝置名稱。也可以決定網路卡是否應由 udev 透過其硬體（MAC）位址或透過匯流排 ID 識別。後者更適合大型伺服器，便於熱插拔網路卡。若要使用 YaST 設定這些選項，請執行下列步驟：

1. 在 YaST網路設定對話方塊的綜覽索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 移至硬體索引標籤。目前的裝置名稱顯示在Udev 規則中。按一下變更。
3. 選擇 udev 是應透過網路卡的MAC 位址還是透過匯流排 ID來識別網路卡。網路卡目前的 MAC 位址和匯流排 ID 將顯示在對話方塊中。
4. 若要變更裝置名稱，請選取變更裝置名稱選項，然後編輯名稱。
5. 若要啓用組態，請確認設定。

16.4.1.2.4 變更網路卡核心驅動程式

有些網路卡可能有多個核心驅動程式可供使用。如果網路卡已設定，YaST 可讓您從可用且適合的驅動程式清單中選取要使用的核心驅動程式。還可以為核心驅動程式指定選項。若要使用 YaST 設定這些選項，請執行下列步驟：

1. 在 YaST 網路設定模組的綜覽索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 移至硬體索引標籤。
3. 在模組名稱中選取要使用的核心驅動程式。在選項中以 `=` `=VALUE` 格式為所選驅動程式輸入任何選項。若要使用多個選項，應以空格將其隔開。
4. 若要啓用組態，請確認設定。

16.4.1.2.5 啓動網路裝置

若使用結合 `wicked` 的方法，則可以將裝置設定為在開機時、連接纜線時、偵測到網路卡時啓動，或以手動方式啓動，或永不啓動。若要變更裝置啓動，請執行下列步驟：

1. 在 YaST 的系統 › 網路設定中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 在一般索引標籤中，從裝置啓用中選擇希望的項目。
   選擇開機時可在系統開機時啓動裝置。若使用電纜連接，系統會監控介面，探查是否有實體的連接。若使用熱插拔時，會在介面可用時對其進行設定。它與開機時選項類似，唯一的區別在於如果開機時介面不存在，則不會發生錯誤。選擇手動可以透過 `ifup` 手動控制介面。選擇永不則不會啓動裝置。在 NFSroot 時與開機時類似，區別是使用 `systemctl stop wicked.service` 指令不會關閉介面；如果 `wicked` 處於使用中狀態，則 `network` 服務也會處理 `wicked` 服務。若您使用的是 NFS 或 iSCSI 根檔案系統，請使用此選項。
3. 若要啓用組態，請確認設定。

提示：用做根檔案系統的 NFS

在透過網路將根分割區掛接為 NFS 共用的（無磁碟）系統中，設定 NFS 共用可供存取的網路裝置時請保持謹慎。

將系統關閉或重新開機時，預設的處理順序是先關閉網路連接，然後卸載根分割區。對於 NFS 根分割區，這種順序會造成問題，因為在尚未與 NFS 共用啓動網路連接的情況下，根分割區無法完全卸載。為防止系統停用相關的網路裝置，請依第 16.4.1.2.5 節 「啓動網路裝置」中所述開啓網路裝置組態索引標籤，然後在裝置啓動窗格中選取在 NFSroot 時。

16.4.1.2.6 設定最大傳送單位大小

您可以設定介面的最大傳送單位（MTU）。MTU 指允許的最大封包大小（以位元組計）。MTU 越高，頻寬效率就越高。但是，大型封包可能會將慢速介面阻擋一段時間，這會加劇後續封包的延遲。

1. 在 YaST 的系統 › 網路設定中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 在一般索引標籤中，從設定 MTU清單中選取所需的項目。
3. 若要啓用組態，請確認設定。

16.4.1.2.7 PCIe 多功能裝置

支援 LAN、iSCSI 與 FCoE 的多功能裝置受支援。YaST FCoE 用戶端（`yast2 fcoe-client`）會在額外的欄中顯示私人旗標，讓使用者可以選取用於 FCoE 的裝置。YaST 網路模組（`yast2 lan`）會在網路組態中排除「僅供儲存的裝置」。

如需 FCoE 的詳細資訊，請參閱《儲存管理指南》，第 15 章「乙太網路光纖通道儲存：FCoE」，第 15.3 節「使用 YaST 管理 FCoE 服務」。

16.4.1.2.8 IP-over-InfiniBand (IPoIB) 的 Infiniband 組態

1. 在 YaST 的系統 › 網路設定中選取 InfiniBand 裝置，然後按一下編輯。
2. 在一般索引標籤中，選取一種IP-over-InfiniBand(IPoIB) 模式：已連接(預設)或資料包。
3. 若要啓用組態，請確認設定。

如需有關 InfiniBand 的詳細資訊，請參閱 `/usr/src/linux/Documentation/infiniband/ipoib.txt`。

16.4.1.2.9 設定防火牆

您不必依《Security Guide》，第 15 章「Masquerading and Firewalls」，第 15.4.1 節「Configuring the Firewall with YaST」中所述輸入詳細的防火牆設定，只需在設定裝置的過程中決定裝置的基本防火牆組態。請執行下列步驟：

1. 開啓 YaST 的系統 › 網路設定模組。在綜覽索引標籤中，於偵測到的網路卡清單中選取一個網路卡，然後按一下編輯。
2. 進入網路設定對話方塊的一般索引標籤。
3. 決定您要將介面指定至的防火牆區域。可用的選項如下：

   停用防火牆
   : 只有當防火牆已停用或未執行時，此選項才可用。唯有您的機器位於受外部防火牆保護的更大網路中時，才能使用此選項。

自動指定區域

只有當防火牆已啓用時，此選項才可用。防火牆會執行，並且介面會自動指定至防火牆區域。包含關鍵字 `any` 的區域或外部區域將會用於這類介面。

內部區域（未保護）

防火牆會執行，但不强制執行任何保護此介面的規則。如果您的機器位於受外部防火牆保護的更大網路中，請使用此選項。如果機器具有多個網路介面，此選項也適用於連接到內部網路的介面。

廢除區域

廢除區域是內部網路與（有潛在風險的）網際網路之前的另一道防線。從內部網路與網際網路都可連接到指派至此區域的主機，但主機無法連存取內部網路。

外部區域

防火牆在此介面上執行，且會完全保護其抵禦其他可能有害的網路流量。此為預設選項。

4. 若要啓用組態，請確認設定。

16.4.1.3 設定未偵測到的網路卡

如果未能正確偵測到某個網路卡，該卡將不會包含在已偵測到的網路卡清單中。若您確定您的系統具備網路卡的驅動程式，可手動設定。也可設定特殊的網路裝置類型，如橋接、Bond、TUN 或 TAP。若要設定未偵測到的網路卡或特殊裝置，請執行下列步驟：

1. 在 YaST 的系統 › 網路設定 › 綜覽對話方塊中，按一下新增。
2. 在硬體對話方塊中，從可用的選項中設定介面的裝置類型和組態名稱。如果網路卡是 PCMCIA 或 USB 裝置，請啓用個別的核取方塊並使用下一步來結束對話方塊。否則，您可以根據需要定義要用於網路卡的核心模組名稱以及卡的選項。
   在Ethtool 選項中，可以設定 `ifup` 用於介面的 `ethtool` 選項。如需可用選項的資訊，請參閱 `ethtool` 手冊頁。
   如果選項字串以 `-` 開頭（例如，`-K INTERFACE_NAME rx on`），則會用目前的介面名稱取代字串中的第二個單字。否則（例如，`autoneg off speed 10`），`ifup` 會在開頭加上 `-s INTERFACE_NAME`。

3. 按一下下一步。
4. 在一般、位址和硬體索引標籤中設定介面的所有必要選項，例如 IP 位址、裝置啓動或防火牆區域。如需組態選項的詳細資訊，請參閱第 16.4.1.2 節 「變更網路卡組態」。
5. 若您介面裝置類型選擇無線，請在下一個對話設定無線連接。
6. 若要啓用新的網路組態，請確認設定。

16.4.1.4 設定主機名稱和 DNS

若您在安裝期間未變更網路組態，且已有乙太網路卡可用，則系統會自動為您的電腦產生主機名稱並啓動 DHCP。同時也會自動產生您主機要整合至網路環境所需的的名稱服務資訊。若網路位址設定使用 DHCP，則網域名稱伺服器清單會自動填入適當的資料。若您希望使用靜態設定，請手動設定數值。

若要變更您電腦的名稱並調整名稱伺服器搜尋清單，請如下執行：

1. 在 YaST 的系統 › 模組中，移至網路設定主機名稱/DNS索引標籤。
2. 輸入主機名稱並根據需要輸入網域名稱。如果機器是郵件伺服器，網域就格外重要。請注意，主機名稱是全域的，會套用於所有已設定的網路介面。
   若您要使用 DHCP 獲取 IP 位址，則電腦的主機名稱將由 DHCP 自動設定。若要連接到其他網路，應禁止此行為，因為其他網路可能會指定其他主機名稱，而且在執行時期變更主機名稱會混淆圖形桌面。若要停止使用 DHCP 獲取 IP 位址，請停用透過 DHCP 變更主機名稱。
   將主機名稱指定至迴路 IP會將您的主機名稱與 `/etc/hosts` 中的 `127.0.0.2`（迴路）IP 位址相關聯。如果您希望主機名稱始終能得到解析（即使是沒有使用中網路的情況下），則可以使用此選項。
3. 在修改 DNS 組態中，選取修改 DNS 組態（名稱伺服器、搜尋清單、`/etc/resolv.conf` 檔案的內容）的方式。
   若選取使用預設規則選項，則組態由 `netconfig` 程序檔來處理，這樣會將靜態定義的資料（使用 YaST 或在組態檔案中）與從 DHCP 用戶端或 NetworkManager 動態取得的資料合併。一般情況均可採用此預設規則。
   若選取僅手動選項，則無法使用 `netconfig` 修改 `/etc/resolv.conf` 檔案。但是可以手動編輯此檔案。

若選取自訂規則選項，則應指定定義合併規則的自訂規則字串。該字串包含要視為設定之有效來源的介面名稱清單（以逗號分隔）。除完整的介面名稱外，還允許使用基本萬用字元來對應多個介面。例如，`eth* ppp?` 將先找到所有 eth，然後找到所有 ppp0-ppp9 介面。以下為指定如何套用 `/etc/sysconfig/network/config` 檔案中所定義之靜態設定的兩個特定規則值：

`STATIC`
: 靜態設定需要與動態設定合併在一起。

`STATIC_FALLBACK`
: 只有動態組態不可用時才會使用靜態設定。

如需詳細資訊，請參閱 `netconfig`(8) 的 man 頁面（`man 8 netconfig`）。

4. 輸入名稱伺服器並填寫網域搜尋清單。名稱伺服器必須透過 IP 位址（例如 192.168.1.116）而非主機名稱指定。在網域搜尋索引標籤中指定的名稱就是用於解析主機名稱（沒有指定網域）的網域名稱。如果使用多個網域搜尋，請以逗號或空格將網域隔開。
5. 若要啓用組態，請確認設定。

還可以使用 YaST 透過指令行編輯主機名稱。YaST 執行的變更會立即生效（手動編輯 `/etc/HOSTNAME` 檔案的情況除外）。若要變更主機名稱，請使用下列指令：

```
yast dns edit hostname=HOSTNAME
```

若要變更名稱伺服器，請使用下列指令：

```
yast dns edit nameserver1=192.168.1.116
yast dns edit nameserver2=192.168.1.117
yast dns edit nameserver3=192.168.1.118
```

16.4.1.5 設定路由

若要讓您的電腦與其他電腦和其他網路通訊，必須提供路由資訊，以讓網路流量採取正確的路徑。若使用 DHCP，會自動提供此資訊。若使用靜態設定，必須手動新增此資料。

1. 在 YaST 中，移至網路設定 › 路由。

2. 輸入預設閘道的 IP 位址（必要時可以輸入 IPv4 和 IPv6）。預設閘道會比對每個可能的目的地，但是如果某個符合所需位址的路由表項目已經存在，就會使用該項目，而非透過預設閘道使用預設路由。

3. 可以在路由表中輸入更多項目。輸入目的地網路 IP 位址、閘道IP 位址和網路遮罩。選取要透過其將流量路由至所定義網路的裝置(減號表示任意裝置)。若要省略這些值的任何一個，請使用減號「`-`」。若要在表中輸入預設閘道，請在目的地欄位中使用「`default`」。

   注意：路由優先程度

   若使用了多個預設路由，則可以指定權值選項確定哪個路由的優先程度較高。若要指定度量選項，請在選項中輸入 `- metric NUMBER`。權值最高的路由做為預設路由。如果該網路裝置已解除連接，將會移除其路由，使用下一個路由。但是，目前的核心不會在靜態路由中使用權值，只有 `multipathd` 等路由精靈使用。

4. 如果系統是路由器，請視需要在網路設定中啓用IPv4 轉遞和IPv6 轉遞。

5. 若要啓用組態，請確認設定。

16.4.2 IBM z Systems：設定網路裝置

SUSE Linux Enterprise Server for IBM z Systems 支援多種類型的網路介面。您可使用 YaST 設定所有類型。

16.4.2.1 qeth-hsi 裝置

若要將 `qeth-hsi` (Hipersockets) 介面新增到安裝好的系統，請啓動 YaST 中的系統 › 網路設定模組。選取其中一個標示為Hipersocket的裝置做為「讀取」裝置位址，然後按一下編輯。輸入讀取通道、寫入通道和控制通道的裝置編號（裝置編號格式的範例：`0.0.0800`）。然後按「下一步」。在網路位址設定對話方塊中，為新介面指定 IP 位址和網路遮罩，然後按下一步和確定結束網路組態。

16.4.2.2 qeth-ethernet 裝置

若要將 `qeth-ethernet`（IBM OSA Express 乙太網路卡）介面新增到安裝好的系統，請啓動 YaST 中的系統 › 網路設定模組。選取其中一個標記為IBM OSA 高速乙太網路卡的裝置來做為「讀取」裝置位址，然後按一下編輯。輸入讀取通道、寫入通道和控制通道的裝置編號（裝置編號格式的範例：`0.0.0700`）。輸入所需的連接埠名稱、連接埠號碼（如果適用）、一些其他選項（請參閱 http://www.ibm.com/developerworks/linux/linux390/documentation_suse.html 中的《Linux for IBM z Systems: Device Drivers, Features, and Commands》（Linux for IBM z Systems：裝置驅動程式、功能和指令）參考手冊）、您的 IP 位址和相應的網路遮罩。按下一步與確定結束網路組態。

16.4.2.3 ctc 裝置

若要將 `ctc`（IBM 平行埠 CTC 介面卡）介面新增到安裝好的系統，請啓動 YaST 中的系統 › 網路設定模組。選取一個標示為IBM 平行埠 CTC 介面卡的裝置做為您的讀取通道，然後按一下設定。選擇適合您裝置的裝置設定(通常為相容模式)。指定您和遠端合作夥伴的 IP 位址。如有需要，可透過進階 › > 細節設定來調整 MTU 的大小。按下一步與確定結束網路組態。

警告：CTC 不再受支援

不建議使用此介面。未來的 SUSE Linux Enterprise Server 版本將不支援此介面。

16.4.2.4 lcs 裝置

若要將 `lcs`（IBM OSA-2 介面卡）介面新增到安裝好的系統，請啓動 YaST 中的系統 › 網路設定模組。選取標示為IBM OSA-2 介面卡的裝置，然後按一下設定。輸入所需的連接埠號碼、一些其他選項（請參閱 http://www.ibm.com/developerworks/linux/linux390/documentation_suse.html 中的《Linux for IBM z Systems: Device Drivers, Features, and Commands》（Linux for IBM z Systems：裝置驅動程式、功能和指令）參考手冊）、您的 IP 位址和相應的網路遮罩。按下一步與確定結束網路組態。

16.4.2.5 IUCV 裝置

若要將 `iucv`（IUCV）介面新增到安裝好的系統，請啓動 YaST 中的系統 › 網路設定模組。選取標示為IUCV的裝置，並按一下編輯。YaST 會提示您提供 IUCV 合作夥伴的名稱（對等）。輸入名稱（本項目區分大小寫）然後選取下一步。指定合作夥伴的IP 位址及遠端 IP 位址。如果需要，在一般索引標籤中設定 MTU大小。按下一步與確定結束網路組態。

警告：IUCV 不再受支援

不建議使用此介面。未來的 SUSE Linux Enterprise Server 版本將不支援此介面。

16.5 手動設定網路連接

手動設定網路軟體應為最後採取的替代方案。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。

16.5.1 `wicked` 網路組態

名為 `wicked` 的工具和程式庫提供了一個新的架構用於設定網路。

傳統網路介面管理面臨的其中一項挑戰是，各種不同的網路管理層混雜在一個程序檔中，最多在兩個不同的程序檔中。這些程序檔彼此之間如何互動沒有明確的定義。這會導致出現無法預測的問題、模糊的條件約束和慣例等情況。針對各種不同的情境部署多個特殊入侵層增大了維護負擔。所用的位址組態通訊協定透過 dhcpcd 等精靈來實作，而這些精靈與基礎架構中其他元件的互動相當不通暢。為了持續識別介面，引入了新潮的介面命名規劃，這一規劃需要繁重的 udev 支援。

wicked 的構想是透過多種方法分解問題。它沒有採用任何全新的理念，而是嘗試將不同專案中的想法集中在一起，以期形成一個更好的整體解決方案。

實現此目的的方法之一是使用用戶端/伺服器模型。此方法可讓 wicked 為位址組態等項目定義能夠很好地整合到整個架構中的標準化功能。例如，使用特定的位址組態時，管理員可能要求應該透過 DHCP 或 IPv4 zeroconf 設定介面。在這種情況下，位址組態服務只會從它的伺服器獲得租用，並傳遞到安裝了所要求位址和路由的 wicked 伺服器程序。

分解問題的另一個方法是強制執行分層機製。對於任何類型的網路介面，都可以定義一個 dbus 服務，用於設定網路介面的裝置層 — VLAN、橋接器、結合或半虛擬化裝置。位址組態等常用功能將透過在這些裝置特定服務頂層的聯合服務來實作，而無需專門實作。

wicked 架構使用各種 dbus 服務來實現這兩個方面的功能，這些服務將根據其類型連接至網路介面。本文提供了 wicked 中當前物件階層的簡要綜覽。

每個網路介面以 `/org/opensuse/Network/Interfaces` 的子物件表示。子物件的名稱由其 ifindex 指定。例如，通常 ifindex 為 1 的迴路介面是 `/org/opensuse/Network/Interfaces/1`，註冊的第一個乙太網路介面是 `/org/opensuse/Network/Interfaces/2`。

每個網路介面都有一個關聯的「類別」，用於選取該介面支援的 dbus 介面。依預設，每個網路介面的類別為 `netif`，wickedd 將自動連接與此類別相容的所有介面。在目前實作中，包括以下介面：

org.opensuse.Network.Interface
: 一般網路介面功能，例如，開啓或關閉連結、指定 MTU 等

org.opensuse.Network.Addrconf.ipv4.dhcp,
org.opensuse.Network.Addrconf.ipv6.dhcp,
org.opensuse.Network.Addrconf.ipv4.auto
: 適用於 DHCP、IPv4 zeroconf 等的位址組態服務

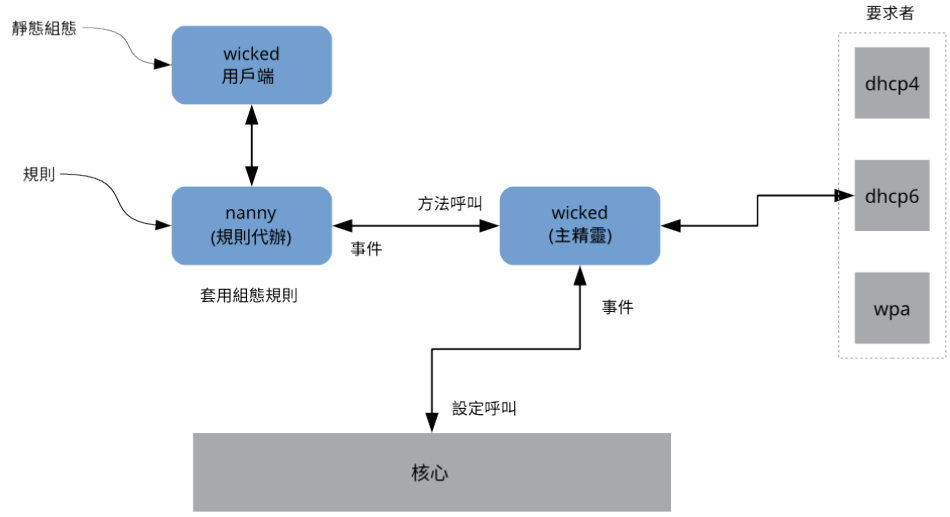
除此之外，網路介面可能還需要或者提供特殊的組態機制。例如，對於某個乙太網路裝置，您應該能夠控制連結速度和檢查總數卸載等。為了實現此目的，乙太網路裝置都有一個名為 `netif-ethernet` 的自己的類別，該類別屬於 `netif` 的子類別。因此，指定給乙太網路介面的 dbus 介面具有上面列出的所有服務，以及 `org.opensuse.Network.Ethernet` 服務，後者僅可用於屬於 `netif-ethernet` 類別的物件。

同樣，橋接器、VLAN、結合裝置或 infiniband 等介面類型也存在適用類別。

您要如何與某個首先需要建立的介面（例如 VLAN，它實際上是位於乙太網路裝置上的虛擬網路介面）互動呢？為此，wicked 定義了出廠介面，例如 `org.opensuse.Network.VLAN.Factory`。這種出廠介面只提供一個功能，就是讓您建立所需類型的介面。這些出廠介面將連接至 `/org/opensuse/Network/Interfaces` 清單節點。

16.5.1.1 wicked 架構與功能

`wicked` 服務包含多個部份，如圖形 16.4 「wicked 架構」中所述。



圖形 16.4 wicked 架構

`wicked` 目前支援以下內容：

- 使用組態檔案後端來剖析 SUSE 樣式的 `/etc/sysconfig/network` 檔案。
- 使用內部組態後端以 XML 格式表示網路介面組態。
- 開啓和關閉「一般」網路介面，例如乙太網路或 InfiniBand、VLAN、橋接器、結合裝置、tun、tap、虛構裝置、macvlan、macvtap、hsi、qeth、iucv 和無線（目前限制為一個 wpa-psk/eap 網路）裝置。
- 內建 DHCPv4 用戶端和內建 DHCPv6 用戶端。

- 預設啓用的 nanny 精靈有助於在裝置可用（介面熱插入）時自動啓動設定的介面，以及在偵測到連結（載體）時設定 IP 組態。如需相關資訊，請參閱第 16.5.1.3 節 「Nanny」。
- `wicked` 實作為一組與 systemd 相整合的 DBus 服務。因此，常用的 `systemctl` 指令都將適用於 `wicked`。

16.5.1.2 使用 wicked

在 SUSE Linux Enterprise 上，預設會執行 `wicked`。如果您要檢查目前啓用了哪個元件，以及該元件是否正在執行，請呼叫：

在 openSUSE Leap 上，`wicked` 預設在桌上型或伺服器硬體上執行。在行動硬體上，NetworkManager 預設會執行。如果您要檢查目前啓用了哪個元件，以及該元件是否正在執行，請呼叫：

```
systemctl status network
```

如果已啓用 `wicked`，則會看到類似下面的行：

```
wicked.service - wicked managed network interfaces
    Loaded: loaded (/usr/lib/systemd/system/wicked.service; enabled)
    ...
```

如果執行的是其他元件（例如 NetworkManager）並且您想要切換到 `wicked`，請先停止正在執行的元件，然後啓用 `wicked`：

```
systemctl is-active network && \
systemctl stop      network
systemctl enable --force wicked
```

如此可啓用 wicked 服務、建立從 `network.service` 到 `wicked.service` 的別名連結，並在下次開機時啓動網路。

啓動伺服器程序：

```
systemctl start wickedd
```

這會啓動 `wickedd`（主要伺服器）和關聯的要求者：

```
/usr/lib/wicked/bin/wickedd-auto4 --systemd --foreground
/usr/lib/wicked/bin/wickedd-dhcp4 --systemd --foreground
/usr/lib/wicked/bin/wickedd-dhcp6 --systemd --foreground
```

```
/usr/sbin/wickedd --systemd --foreground
/usr/sbin/wickedd-nanny --systemd --foreground
```

然後開啓網路：

```
systemctl start wicked
```

或者使用 `network.service` 別名：

```
systemctl start network
```

這些指令使用 `/etc/wicked/client.xml` 中定義的預設組態來源或系統組態來源。

若要啓用除錯，請在 `/etc/sysconfig/network/config` 中設定 `WICKED_DEBUG`，例如：

```
WICKED_DEBUG="all"
```

或者，若要省略一些資訊：

```
WICKED_DEBUG="all,-dbus,-objectmodel,-xpath,-xml"
```

使用用戶端公用程式顯示所有介面的介面資訊，或者顯示以 `IFNAME` 指定的介面的介面資訊：

```
wicked show all
wicked show IFNAME
```

XML 格式的輸出：

```
wicked show-xml all
wicked show-xml IFNAME
```

開啓一個介面：

```
wicked ifup eth0
wicked ifup wlan0
...
```

由於未指定組態來源，wicked 用戶端將會檢查 `/etc/wicked/client.xml` 中為它定義的預設組態來源：

1. `firmware`：iSCSI 開機韌體表（iBFT）
2. `compat`： `ifcfg` 檔案 — 為相容性而實作

將會套用 `wicked` 從指定介面的這些來源中取得的任何設定。預期的重要性順序為 `firmware`、`compat` – 將來這種順序可能會發生變更。

如需詳細資訊，請參閱 `wicked` 的 man 頁面。

16.5.1.3 Nanny

Nanny 是一個由事件與規則驅動的精靈，負責熱插拔裝置等非同步或被動性案例。因此，nanny 精靈可幫助啓動或者重新啓動延遲的裝置，或臨時消失的裝置。Nanny 會監視裝置和連結的變化，並整合目前規則集定義的新裝置。由於指定的逾時條件約束的原因，即使 `ifup` 已結束，Nanny 仍會繼續設定。

依預設，nanny 精靈在系統上處於使用中狀態。可以在 `/etc/wicked/common.xml` 組態檔案中啓用該精靈：

```
<config>
  ...
  <use-nanny>true</use-nanny>
</config>
```

如果使用此設定，ifup 和 ifreload 會將包含有效組態的規則套用至 nanny 精靈；然後，nanny 將會設定 `wickedd`，從而確定支援熱插拔。nanny 將在背景中等待事件或變更（例如，開啓新的裝置或載體）。

16.5.1.4 開啓多個介面

對於結合裝置和橋接器，有效的做法是在一個檔案（ifcfg-bondX）中定義整個裝置拓撲，並一次性將它開啓。然後，當您指定（橋接器或結合裝置的）頂層介面名稱時，wicked 可以開啓整個組態：

```
wicked ifup br0
```

此指令會依適當的順序自動設定橋接器及其相依項，而無需分別列出相依項（連接埠等）。

若要在一個指令中開啓多個介面：

```
wicked ifup bond0 br0 br1 br2
```

若要開啓所有介面：

```
wicked ifup all
```

16.5.1.5 通道與 Wicked 配合使用

如果需要將通道與 Wicked 配合使用，可以使用 `TUNNEL_DEVICE`。它可用於指定可選的裝置名稱，讓通道繫結到該裝置。通道式封包只會透過此裝置進行路由。

如需詳細資訊，請參閱 `man 5 ifcfg-tunnel`。

16.5.1.6 處理增量變更

有了 `wicked`，當您要重新設定某個介面時，並不需要真正將它關閉（除非核心有此要求）。例如，若要將另一個 IP 位址或路由新增到靜態設定的網路介面，請將該 IP 位址新增到介面定義，然後再次執行「ifup」操作。伺服器會儘量做到只更新那些已變更的設定。這適用于連結級別的選項，例如裝置 MTU 或 MAC 位址；也適用於網路級別的設定，例如位址、路由，甚至位址組態模式（例如，從靜態組態轉到 DHCP 時）。

當然，對於合併了多個真實裝置（例如橋接器或結合裝置）的虛擬介面，事情會變得有些棘手。對於結合裝置，當裝置運作時，您無法變更某些參數，否則會導致出錯。

但是，您仍可以新增或移除結合裝置或橋接器的子裝置，或者選擇結合裝置的主要介面。

16.5.1.7 Wicked 延伸：位址組態

`wicked` 設計為可使用外圍程序檔延伸。這些延伸可在 `config.xml` 檔案中定義。

目前支援數種類別的延伸：

- 連結組態：這些程序檔負責根據用戶端提供的組態來設定裝置的連結層，以及負責將連結層再次拆開。
- 位址組態：這些程序檔負責管理裝置的位址組態。通常，位址組態和 DHCP 由 `wicked` 自身管理，但是，可借助延伸來執行。
- 防火牆延伸：這些程序檔可以套用防火牆規則。

通常，延伸中包含一個啓動指令和一個停止指令、一個選擇性的「pid 檔案」，以及要傳遞給程序檔的一組環境變數。

為了說明此延伸的工作原理，請查看 `etc/server.xml` 中定義的防火牆延伸：

```
<dbus-service interface="org.opensuse.Network.Firewall">
 <action name="firewallUp"   command="/etc/wicked/extensions/firewall up"/>
 <action name="firewallDown" command="/etc/wicked/extensions/firewall down"/>

 <!-- default environment for all calls to this extension script -->
 <putenv name="WICKED_OBJECT_PATH" value="$object-path"/>
 <putenv name="WICKED_INTERFACE_NAME" value="$property:name"/>
 <putenv name="WICKED_INTERFACE_INDEX" value="$property:index"/>
</dbus-service>
```

此延伸會附加至 `<dbus-service>` 標記，並定義針對此介面的動作而要執行的指令。此外，宣告可以定義並啓始化傳遞給動作的環境變數。

16.5.1.8 Wicked 延伸：組態檔案

您也可以使用程序檔來延伸組態檔案的處理。例如，`extensions/resolver` 程序檔根據 `server.xml` 中設定的行為來最終處理租用中的 DNS 更新：

```
<system-updater name="resolver">
 <action name="backup" command="/etc/wicked/extensions/resolver backup"/>
 <action name="restore" command="/etc/wicked/extensions/resolver restore"/>
 <action name="install" command="/etc/wicked/extensions/resolver install"/>
 <action name="remove" command="/etc/wicked/extensions/resolver remove"/>
</system-updater>
```

當 `wickedd` 中收到更新時，系統更新程式常式將剖析租用，並呼叫解析程式程序檔中的適當指令（`backup`、`install` 等）。此後便可使用 `/sbin/netconfig` 或者透過手動寫入 `/etc/resolv.conf`（做為錯誤回復）來設定 DNS 設定。

16.5.2 組態檔案

本節提供網路組態檔的綜覽，並說明其用途和使用的格式。

16.5.2.1 /etc/wicked/common.xml

`/etc/wicked/common.xml` 檔案包含所有應用程式都應使用的通用定義。它源自或包含在位於同一目錄下的其他組態檔案中。儘管您可以使用此檔案允許在所有 `wicked` 元件間除錯，但建議使用 `/etc/wicked/local.xml` 檔案來實現此目的。執行維護更新後，`/`

`etc/wicked/common.xml` 可能會被覆寫，因此所做的變更可能會遺失。`/etc/wicked/common.xml` 檔案包含預設安裝中的 `/etc/wicked/local.xml`，因此通常不需要修改 `/etc/wicked/common.xml`。

若要透過將 `<use-nanny>` 設定為 `false` 來停用 `nanny`，可重新啓動 `wickedd.service`，然後執行以下指令來套用所有組態與規則：

```
wicked ifup all
```

注意：組態檔案

如果 `wickedd`、`wicked` 或 `nanny` 程式自己的組態檔案不存在，則會嘗試讀取 `/etc/wicked/common.xml`。

16.5.2.2 /etc/wicked/server.xml

`wickedd` 伺服器程序會在啓動時讀取 `/etc/wicked/server.xml` 檔案。該檔案將延伸儲存於 `/etc/wicked/common.xml`。除此之外，此檔案還會設定解析器的處理方式以及從 `addrconf` 要求者（例如 DHCP）接收資訊的方式。

建議將對此檔案所需的變更新增至一個由 `/etc/wicked/server.xml` 納入的單獨檔案 `/etc/wicked/server-local.xml`。使用單獨的檔案可以避免所做的變更在更新維護期間遭到覆寫。

16.5.2.3 /etc/wicked/client.xml

`/etc/wicked/client.xml` 用於 `wicked` 指令。該檔案指定探查由 `ibft` 管理的裝置時所用程序檔的位置，並可設定網路介面組態的位置。

建議將對此檔案所需的變更新增至一個由 `/etc/wicked/server.xml` 納入的單獨檔案 `/etc/wicked/client-local.xml`。使用單獨的檔案可以避免所做的變更在更新維護期間遭到覆寫。

16.5.2.4 /etc/wicked/nanny.xml

/etc/wicked/nanny.xml 設定連結層的類型。建議將特定的組態新增至一個單獨的檔案 /etc/wicked/nanny-local.xml，以免在維護更新期間遺失所做的變更。

16.5.2.5 /etc/sysconfig/network/ifcfg-*

這些檔案包含網路介面的傳統組態。在 °¸SUSE Linux Enterprise 11°¸ 中，這是除 iBFT 韌體以外唯一支援的格式。

注意：wicked 和 ifcfg-* 檔案

如果您指定 compat: 字首，wicked 會讀取這些檔案。根據 /etc/wicked/client.xml 中 SUSE Linux Enterprise Server 的預設組態，wicked 將嘗試先讀取這些檔案，然後再讀取 /etc/wicked/ifconfig 中的 XML 組態檔案。

提供的 --ifconfig 參數主要用於測試。如果指定該參數，則不會套用 /etc/wicked/ifconfig 中定義的預設組態來源。

ifcfg-* 檔案包含啓動模式和 IP 位址等資訊。可以使用的參數請參閱 ifup 的手冊頁。此外，如果一個一般設定只能用於一個介面，則檔案 dhcp 和 wireless 中的大多數變數在 ifcfg-* 檔案中都可以使用。但是，大多數 /etc/sysconfig/network/config 變數都是全域變數，在 ifcfg 檔案中不能將它們覆寫。例如，NETCONFIG_* 變數就是全域變數。

若要設定 macvlan 和 macvtab 介面，請參閱 ifcfg-macvlan 和 ifcfg-macvtap 的 man 頁面。例如，對於 macvlan 介面，請提供使用以下設定的 ifcfg-macvlan0：

```
STARTMODE='auto'
MACVLAN_DEVICE='eth0'
#MACVLAN_MODE='vepa'
#LLADDR=02:03:04:05:06:aa
```

如需 ifcfg.template 的相關資訊，請參閱第 16.5.2.6 節 「/etc/sysconfig/network/config、/etc/sysconfig/network/dhcp 和 /etc/sysconfig/network/wireless」。

System z IBM z Systems 不支援 USB。介面檔案的名稱和網路別名包含特定於 z Systems 的元素，例如 qeth。

16.5.2.6 /etc/sysconfig/network/config、/etc/sysconfig/network/dhcp 和 /etc/sysconfig/network/wireless

檔案 `config` 包含 `ifup`、`ifdown` 和 `ifstatus` 行為的一般設定。`dhcp` 包含無線 LAN 卡之 DHCP 和 `wireless` 的設定。所有三個組態檔中的變數都已被註解。`/etc/sysconfig/network/config` 中的某些變數也可以在 `ifcfg-*` 檔案中使用，而且在這些檔案中它們的優先程度更高。`/etc/sysconfig/network/ifcfg.template` 檔案列出了可在永久介面中指定的變數。但是，大多數 `/etc/sysconfig/network/config` 變數都是全域變數，在 ifcfg 檔案中不能將它們覆寫。例如，`NETWORKMANAGER` 或 `NETCONFIG_*` 變數就是全域變數。

注意：使用 DHCPv6

在 SUSE Linux Enterprise 11 中，即使是在未正確設定 IPv6 路由器廣播 (RA) 的網路中，DHCPv6 一向也能正常運作。從 SUSE Linux Enterprise 12 開始，DHCPv6 將適當地要求網路中至少有一個路由器發出 RA，用於指示此網路是由 DHCPv6 管理。

對於無法在其中正確設定路由器的網路，使用者可透過在 `ifcfg` 檔案中指定 `DHCLIENT6_MODE='managed'`，使用 `ifcfg` 選項來覆寫此行為。您也可以在安裝系統中使用開機參數來實現這種解決方案：

```
ifcfg=eth0=dhcp6,DHCLIENT6_MODE=managed
```

16.5.2.7 /etc/sysconfig/network/routes 和 /etc/sysconfig/network/ifroute-*

TCP/IP 封包的靜態路由是由 `/etc/sysconfig/network/routes` 和 `/etc/sysconfig/network/ifroute-*` 檔案確定的。您可在 `/etc/sysconfig/network/routes` 中指定各種系統任務所需的全部靜態路由，包括前往主機的路由、透過閘道前往主機的路由，以及前往網路的路由。對於需要個別路由的介面，請定義額外的組態檔案：`/etc/sysconfig/network/ifroute-*`。以介面的名稱取代萬用字元 (`*`)。在路由組態檔中的項目看起來就像這樣：

```
# Destination     Gateway           Netmask           Interface  Options
```

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。應該以 CIDR 表示法（地址加上關聯的路由字首長度）寫入網路，例如 10.10.0.0/16（對於 IPv4 路由）或 fc00::/7（對於 IPv6 路由）。關鍵字 `default` 表示該路由是與閘道位於相同位址系列中的預設閘道。對於沒有閘道的裝置，請使用明確的 0.0.0.0/0 或 ::/0 目的地。

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。

第三欄已廢棄；該欄用於包含目的地的 IPv4 網路遮罩。對於 IPv6 路由、預設路由，或者如果在第一欄中使用了字首長度（CIDR 表示法），請在此處輸入破折號（`-`）。

第四欄包含介面名稱。如果使用破折號（`-`）將它保留空白，可能會導致 `/etc/sysconfig/network/routes` 出現非預期的行為。如需詳細資訊，請參閱 `routes` 的 man 頁面。

第五欄（選擇性）可用於指定特殊選項。如需詳細資訊，請參閱 `routes` 的 man 頁面。

範例 16.5 通用網路介面和部分靜態路由

```
# --- IPv4 routes in CIDR prefix notation:
# Destination     [Gateway]         -                   Interface
127.0.0.0/8       -                 -                   lo
204.127.235.0/24  -                 -                   eth0
default           204.127.235.41    -                   eth0
207.68.156.51/32  207.68.145.45     -                   eth1
192.168.0.0/16    207.68.156.51     -                   eth1

# --- IPv4 routes in deprecated netmask notation"
# Destination     [Dummy/Gateway]   Netmask             Interface
#
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

# --- IPv6 routes are always using CIDR notation:
# Destination     [Gateway]               -           Interface
2001:DB8:100::/64 -                       -           eth0
```

```
2001:DB8:100::/32 fe80::216:3eff:fe6d:c042 -          eth0
```

16.5.2.8 /etc/resolv.conf

主機所屬的網域在 `/etc/resolv.conf` 中指定（關鍵字 `search`）。使用 `search` 選項最多可以指定六個網域，總共 256 個字元。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。使用 `nameserver` 選項最多可以指定 3 個名稱伺服器，一行指定一個。註解以井號或分號（`#` 或 `;`）開頭。如需取得範例說明，請參閱範例 16.6 「/etc/resolv.conf」。

不過，您不可手動編輯 `/etc/resolv.conf`。它是由 `netconfig` 程序檔產生的。若要定義靜態 DNS 組態而不使用 YaST，請在 `/etc/sysconfig/network/config` 檔案中手動編輯適當的變數：

`NETCONFIG_DNS_STATIC_SEARCHLIST`
: 用於主機名稱查詢的 DNS 網域名稱清單

`NETCONFIG_DNS_STATIC_SERVERS`
: 用於主機名稱查詢的名稱伺服器 IP 位址清單

`NETCONFIG_DNS_FORWARDER`
: 需要設定的 DNS 轉遞者名稱，例如 `bind` 或 `resolver`

`NETCONFIG_DNS_RESOLVER_OPTIONS`
: 可寫入 `/etc/resolv.conf` 的任意選項，例如：

```
debug attempts:1 timeout:10
```

如需詳細資訊，請參閱 `resolv.conf` 的 man 頁面。

`NETCONFIG_DNS_RESOLVER_SORTLIST`
: 最多 10 個項目的清單，例如：

```
130.155.160.0/255.255.240.0 130.155.0.0
```

如需詳細資訊，請參閱 `resolv.conf` 的 man 頁面。

若要使用 netconfig 停用 DNS 組態，請設定 `NETCONFIG_DNS_POLICY=''`。如需有關 `netconfig` 的詳細資訊，請參閱 `netconfig(8)` 的 man 頁面（`man 8 netconfig`）。

範例 16.6 /etc/resolv.conf

```
# Our domain
search example.com
#
# We use dns.example.com (192.168.1.116) as nameserver
nameserver 192.168.1.116
```

16.5.2.9 /sbin/netconfig

`netconfig` 是一個模組化工具，用於管理其他網路組態設定。它根據預先定義的規則，將靜態定義的設定與自動組態機制（如 DHCP 或 PPP）提供的設定進行合併。透過呼叫負責修改組態檔案和重新啓動服務或類似動作的 netconfig 模組，將必要的變更套用至系統。

`netconfig` 可以辨識三個主要動作。DHCP 或 PPP 等精靈使用 `netconfig modify` 與 `netconfig remove` 指令提供或移除 netconfig 的設定。使用者僅可使用 `netconfig update` 指令：

`modify`

`netconfig modify` 指令會修改目前的介面和服務特定的動態設定，並更新網路組態。Netconfig 會從標準輸入或從使用 `--lease-file FILENAME` 選項指定的檔案中讀取設定，並將其儲存於內部，直到系統重新開機（或者執行下一個修改或移除動作）為止。系統會覆寫同一個介面與服務組合的現有設定。該介面由 `-i INTERFACE_NAME` 參數指定。該服務由 `-s SERVICE_NAME` 參數指定。

`remove`

`netconfig remove` 指令會移除修改動作為指定介面和服務組合提供的動態設定，並更新網路組態。該介面由 `-i INTERFACE_NAME` 參數指定。該服務由 `-s SERVICE_NAME` 參數指定。

`update`

`netconfig update` 指令會使用目前的設定更新網路組態。當規則或靜態組態變更時可以使用此指令。如果只想更新指定的服務（`dns`、`nis` 或 `ntp`），請使用 `-m MODULE_TYPE` 參數。

netconfig 規則和靜態組態設定可透過手動方式定義，或使用 YaST 在 `/etc/sysconfig/network/config` 檔案中定義。自動組態工具（如 DHCP 或 PPP）提供的動態組態設定經由這些工具，透過 `netconfig modify` 和 `netconfig remove` 動作直接傳送。NetworkManager 啓用時，netconfig（在 `auto` 規則模式中）只會使用 NetworkManager 設定，而忽略使用傳統 ifup 方法設定的任何其他介面的設定。如果 NetworkManager 未提供任何設定，則使用靜態設定做為備用設定。系統不支援同時使用 NetworkManager 與 `wicked` 方法。

如需 `netconfig` 的詳細資訊，請參閱 `man 8 netconfig`。

16.5.2.10 /etc/hosts

在此檔案中（請參閱範例 16.7 「/etc/hosts」），將為主機名稱指定 IP 位址。如果沒有執行任何名稱伺服器，將使用此 IP 連接設定的所有主機將列示於此。對於每個主機，請在檔案中輸入一行資訊，其中包含 IP 位址、完全合格的主機名稱及主機名稱。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 16.7 /etc/hosts

```
127.0.0.1 localhost
192.168.2.100 jupiter.example.com jupiter
192.168.2.101 venus.example.com venus
```

16.5.2.11 /etc/networks

在此檔中，網路名稱會轉換為網路位址。格式與 `hosts` 檔案格式相似，但是網路名稱在位址前。請參閱範例 16.8 「/etc/networks」。

範例 16.8 /etc/networks

```
loopback    127.0.0.0
localnet    192.168.0.0
```

16.5.2.12 /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。每個參數都必須單獨佔用一行。註解的前面是 # 符號。表格 16.2 「/etc/host.conf 的參數」 顯示出可用的參數。/etc/host.conf 範例是顯示在 範例 16.9 「/etc/host.conf」 。

表格 16.2 /ETC/HOST.CONF 的參數

order hosts, bind	指定針對名稱解析存取服務的順序。可用的引數有（以空格或逗號分隔）：
	hosts：搜尋 /etc/hosts 檔案
	bind：存取名稱伺服器
	nis：使用 NIS
multi on/off	定義在 /etc/hosts 中所輸入的主機是否可以有多個 IP 位址。
nospoof on spoofalert on/off	這些參數會影響名稱伺服器 spoofing，但並不會對網路組態產生任何影響。
trim domainname	在主機名稱解析後，指定的網域名稱會與主機名稱分隔開來（前提是主機名稱包括網域名稱）。只有在本地網域分隔出來的名稱位於 /etc/hosts 檔案，但仍然使用附加的網域名稱進行辨識時，這個選項才有用。

範例 16.9 /etc/host.conf

```
# We have named running
order hosts bind
# Allow multiple address
multi on
```

16.5.2.13 /etc/nsswitch.conf

GNU C Library 2.0 的介紹隨附於名稱服務切換（NSS，Name Service Switch）的介紹。詳細資訊請參閱 `nsswitch.conf(5)` man 頁面和 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 16.10 「/etc/nsswitch.conf」。備註前面標有 `#` 符號。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`）（請參閱第 25 章 「網域名稱系統」）。

範例 16.10 /etc/nsswitch.conf

```
passwd:     compat
group:      compat

hosts:      files dns
networks:   files dns

services:   db files
protocols:  db files
rpc:        files
ethers:     files
netmasks:   files
netgroup:   files nis
publickey:  files

bootparams: files
automount:  files nis
aliases:    files nis
shadow:     compat
```

NSS 上可用的「資料庫」列示於 表格 16.3 「透過 /etc/nsswitch.conf 的可用資料庫」。

NSS 資料庫的組態選項將列於表格 16.4 「NSS「資料庫」的組態選項」。

表格 16.3 透過 /ETC/NSSWITCH.CONF 的可用資料庫

`aliases`	`sendmail` 所執行的郵件別名；請參閱 `man 5 aliases`。
`ethers`	乙太網路位址。
`netmasks`	網路及其子網路遮罩的清單。當您使用子網路時才需要。

group	getgrent 所使用的使用者群組。請參閱 group 的 man 頁面。
hosts	gethostbyname 及類似函數所使用的主機名稱與 IP 位址。
netgroup	在網路中用於控制存取權限的有效主機與使用者清單，請參閱 netgroup(5) man 頁面。
networks	getnetent 所使用的網路名稱與位址。
publickey	NFS 及 NIS+ 使用之 Secure_RPC 的公開金鑰和私密金鑰。
passwd	getpwent 所使用的使用者密碼；請參閱 passwd(5) man 頁面。
protocols	getprotoen 所使用的網路通訊協定；請參閱 protocols(5) man 頁面。
rpc	getrpcbyname 及類似功能所使用的遠端程序呼叫名稱與位址。
services	getservent 使用的網路服務。
shadow	getspnam 所使用的使用者遮蔽密碼；請參閱 shadow(5) man 頁面。

表格 16.4 NSS「資料庫」的組態選項

files	直接存取檔案，例如 /etc/aliases
db	透過資料庫存取
nis 、nisplus	NIS，請參閱《Security Guide》，第 3 章「Using NIS」

`dns`	只能做為 `hosts` 與 `networks` 的延伸
`compat`	只能做為 `passwd`、`shadow` 與 `group` 的延伸

16.5.2.14 /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)` man 頁面。依預設，`passwd`、`groups` 與 `hosts` 的系統項目會由 nscd 快取。這對 NIS 與 LDAP 等目錄服務的效能而言十分重要，因為存取名稱、群組或主機都不再需要網路連接。

如果啓用 `passwd` 的快取，通常需要 15 秒，才能辨識新增的本地使用者。使用以下指令重新啓動 nscd，縮短這段等待時間：

```
systemctl restart nscd
```

16.5.2.15 /etc/HOSTNAME

`/etc/HOSTNAME` 包含完全合格的主機名稱（FQHN）。完全合格的主機名稱是附加網域名稱的主機名稱。此檔案只能包含一行，其中設定了主機名稱。機器開機時會讀取此檔案。

16.5.3 測試與組態

將組態寫入您的組態檔案之前，可先進行測試。若要設定測試組態，請使用 `ip` 指令。若要測試連接，請使用 `ping` 指令。

`ip` 指令會直接變更網路組態，而不會將其儲存到組態檔案中。除非您將組態輸入正確的組態檔案，否則重新開機之後網路組態的變更就會遺失。

注意：`ifconfig` 和 `route` 已過時

`ifconfig` 和 `route` 工具已過時。請改用 `ip`。例如，`ifconfig` 會將介面名稱限制為 9 個字元。

16.5.3.1 使用 `ip` 設定網路介面

`ip` 是一項可顯示及設定網路裝置、路由、規則路由和通道的工具。

`ip` 是非常複雜的工具。它的常用語法為 `ip OPTIONS OBJECT COMMAND`。您可使用下列物件：

`link`
: 此物件代表網路裝置。

`address`
: 此物件代表裝置的 IP 位址。

`neighbor`
: 此物件代表 ARP 或 NDISC 快取項目。

`route`
: 此物件代表路由表格項目。

`rule`
: 此物件代表路由原則資料庫中的規則。

`maddress`
: 此物件代表多點傳播位址。

`mroute`
: 此物件代表多點傳播路由快取項目。

`tunnel`
: 此物件表示 IP 上的通道。

若未提供指令，會使用預設指令（通常是 `list`）。

使用 `ip link set DEVICE_NAME` 指令變更裝置的狀態。例如，若要停用裝置 eth0，請輸入 `ip link set eth0 down`。若要重新啓用，請使用 `ip link set eth0 up`。

啓用裝置之後，就可加以設定。若要設定 IP 位址，可使用 `ip addr add IP_ADDRESS + dev DEVICE_NAME`。例如，若要將介面 eth0 的 IP 位址以標準廣播（選項 `brd`）設定為 192.168.12.154/30，請輸入 `ip addr add 192.168.12.154/30 brd + dev eth0`。

若要具備作用中連接，必須設定預設閘道。若要為您的系統設定閘道，請輸入 `ip route add 閘道 IP 位址`。若要轉換某個 IP 位址，請使用 `nat`：`ip route add nat ip_address via other_ip_address`。

若要顯示所有裝置，請使用 `ip link ls`。若只希望顯示運作中介面，請使用 `ip link ls up`。若要列印裝置的介面統計值，請輸入 `ip -s link ls device_name`。若要檢是您裝置的位址，請輸入 `ip addr`。在 `ip addr` 的輸出中同時也可找到您裝置的 MAC 位址相關資訊。若要顯示所有路由，請使用 `ip route show`。

如需有關使用 `ip` 的詳細資訊，請輸入 `ip help` 或參閱 `ip(8)` man 頁面。`help` 選項也適用於所有 `ip` 子指令。例如，如果您需要 `ip addr` 的說明，請輸入 `ip addr help`。`/usr/share/doc/packages/iproute2/ip-cref.pdf` 中提供了有關 `ip` 的說明。

16.5.3.2 以 ping 測試連接

`ping` 指令是測試 TCP/IP 連接運作的標準工具。其使用 ICMP 通訊協定，將小型資料封包 ECHO_REQUEST 傳送至目的地主機，要求立即回應。如果成功，`ping` 將顯示表示這一結果的訊息。這表示網路連結正在作用。

`ping` 不僅會測試兩台電腦之間能否連接，還會提供一些關於連接品質的基本資訊。您可在 範例 16.11 「指令 ping 的輸出」 中看到 `ping` 輸出的一些範例。倒數第二行包含送出的封包數、封包遺失率以及執行 `ping` 總共花費的時間等資訊。

因此，可使用主機名稱或 IP 位址，例如 `ping example.com` 或 `ping 192.168.3.100`。程式會持續傳送封包，直到您按下 Ctrl—C 為止。

若您只需要檢查連接功能性，您可以 `-c` 選項限定封包數量。例如，若要將 ping 限制於三個封包，請輸入 `ping -c 3 example.com`。

範例 16.11 指令 PING 的輸出

```
ping -c 3 example.com
PING example.com (192.168.3.100) 56(84) bytes of data.
64 bytes from example.com (192.168.3.100): icmp_seq=1 ttl=49 time=188 ms
64 bytes from example.com (192.168.3.100): icmp_seq=2 ttl=49 time=184 ms
64 bytes from example.com (192.168.3.100): icmp_seq=3 ttl=49 time=183 ms
--- example.com ping statistics ---
```

```
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 183.417/185.447/188.259/2.052 ms
```

兩個封包之間的預設間隔為一秒。若要變更間隔，可以使用 ping 提供的選項 `-i`。例如，若要將 ping 間隔增加到十秒，請輸入 `ping -i 10 example.com`。

在具備多網路裝置的系統中，透過特定介面位址傳送 ping 非常實用。若要執行此動作，請使用 `-I` 選項以及所選裝置的名稱，例如 `ping -I wlan1 example.com`。

如需使用 ping 的選項與詳細資訊，請輸入 `ping -h` 或參閱 `ping (8)` man 頁面。

提示：Ping IPv6 位址

對於 IPv6 位址，請使用 `ping6` 指令。請注意，若要 ping 連結-本機位址，必須使用 `-I` 指定介面。如果該位址可透過 `eth1` 存取，則可以使用以下指令實現目的：

```
ping6 -I eth1 fe80::117:21ff:feda:a425
```

16.5.4 單位檔案和啓動程序檔

除了上述的組態檔案之外，還有一些在機器開機時載入網路服務的 systemd 單位檔案和各種程序檔。當系統切換為 `multi-user.target` 目標時，會啓動這些單位檔案和程序檔。網路程式的一些單位檔案和啓動程序檔中介紹了一些單位檔案和程序檔。如需有關 `systemd` 的詳細資訊，請參閱第 13 章 「systemd 精靈」；如需有關 `systemd` 目標的詳細資訊，請參閱 `systemd.special` 的 man 頁面（`man systemd.special`）。

網路程式的一些單位檔案和啓動程序檔

`network.target`

`network.target` 是網路的 systemd 目標，但其具體含義取決於系統管理員提供的設定。

如需詳細資訊，請參閱 http://www.freedesktop.org/wiki/Software/systemd/NetworkTarget/。

`multi-user.target`

`multi-user.target` 是包含全部所需網路服務之多使用者系統的 systemd 目標。

`xinetd`
: 啓動 xinetd。xinetd 可以用來讓伺服器服務能夠在系統上使用。例如，只要開啓 FTP 連接，它即可啓動 vsftpd。

`rpcbind`
: 啓動可將 RPC 程式號碼轉換為通用位址的 rpcbind 公用程式。NFS 伺服器等 RPC 服務需要用到。

`ypserv`
: 啓動 NIS 伺服器。

`ypbind`
: 啓動 NIS 用戶端。

`/etc/init.d/nfsserver`
: 啓動 NFS 伺服器。

`/etc/init.d/postfix`
: 控制後置程序。

16.6 基本路由器設定

路由器是一種網路裝置，可接收和傳送往來於多個網路的資料（網路封包）。路由器常用於將本地網路連接至遠端網路（網際網路）或連接多個本地網路區段。透過 SUSE Linux Enterprise Server，您可以建構一個具備 NAT（網路位址翻譯）或進階防火牆等功能的路由器。

下面是將 SUSE Linux Enterprise Server 轉變為路由器的基本步驟。

1. 啓用轉遞，例如在 `/etc/sysctl.d/50-router.conf` 中

   ```
   net.ipv4.conf.all.forwarding = 1
   net.ipv6.conf.all.forwarding = 1
   ```

   然後，為介面提供靜態 IPv4 與 IPv6 IP 設定。啓用轉遞會停用多項機制，例如 IPv6 不再接受 IPv6 RA（路由器廣播），這也會妨礙建立預設路由器。

2. 許多情況下（例如，當您可以透過多個介面連接同一個網路，或者通常使用的是 VPN 且已位於「正常的多宿主主機」上時），必須停用 IPv4 逆向路徑過濾（此功能目前不適用於 IPv6）：

```
net.ipv4.conf.all.rp_filter = 0
```

不過，您還可以改用防火牆設定進行過濾。

3. 若要接受來自路由器（位於外部、上行或 ISP 介面）的 IPv6 RA，並再次建立預設（亦或更具特定性）的 IPv6 路由，請設定：

```
net.ipv6.conf.${ifname}.accept_ra = 2
net.ipv6.conf.${ifname}.autoconf = 0
```

（注意：在以點分隔的 sysfs 路徑中，「`eth0.42`」 需要寫為 `eth0/42`。）

更多路由器行為和轉遞相依性的描述，可造訪 https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt。

為了在內部（DMZ）介面上提供 IPv6，並將您自己宣告為 IPv6 路由器，同時「將網路自動設定」到用戶端，請安裝並設定 `/etc/radvd.conf` 中的 `radvd`，例如：

```
interface eth0
{
    IgnoreIfMissing on;         # do not fail if interface missed

    AdvSendAdvert on;           # enable sending RAs
    AdvManagedFlag on;          # IPv6 addresses managed via DHCPv6
    AdvOtherConfigFlag on;      # DNS, NTP... only via DHCPv6

    AdvDefaultLifetime 3600;    # client default route lifetime of 1 hour

    prefix 2001:db8:0:1::/64    # (/64 is default and required for autoconf)
    {
        AdvAutonomous off;         # Disable address autoconf (DHCPv6 only)

        AdvValidLifetime 3600;     # prefix (autoconf addr) is valid 1 h
        AdvPreferredLifetime 1800; # prefix (autoconf addr) is prefered 1/2 h
    }
}
```

最後設定防火牆。在 SuSEfirewall2 中，您需要設定 `FW_ROUTE="yes"`（否則它還會再次重設轉遞 sysctl）並依據需要在 `FW_DEV_INT`、`FW_DEV_EXT`（和 `FW_DEV_DMZ`）區域變數中定義介面，可能還要設定 `FW_MASQUERADE="yes"` 和 `FW_MASQ_DEV`。

16.7 設定 Bonding 裝置

對於某些系統而言，實作的網路連接除了需要符合一般乙太網路裝置的標準資料安全性或可用性要求之外，還需要符合其他要求。在這些情況下，數個乙太網路裝置可以結集成單個 Bonding 裝置。

bonding 裝置的組態是透過 bonding 模組選項來設定，而其行為主要受 Bonding 裝置的模式影響。該模式預設為 `active-backup`，這表示如果使用中的從屬裝置失敗，另一個從屬裝置將變成使用中狀態。可用結合模式如下：

0 (balance-rr)
: 封包依次透過第一個到最後一個可用介面傳輸。提供容錯和負載平衡。

1 (active-backup)
: 只有一個網路介面處於使用中狀態。如果它失敗，另一個介面將變成使用中狀態。此設定是 SUSE Linux Enterprise Server 的預設設定。提供容錯。

2 (balance-xor)
: 根據以下規則在所有可用介面間拆分流量：`[ (source MAC address XOR'd with destination MAC address XOR packet type ID) modulo slave count]` 需要交換器的支援。提供容錯和負載平衡。

3 (broadcast)
: 在所有介面上廣播所有流量。需要交換器的支援。提供容錯。

4 (802.3ad)
: 將介面聚合成共用相同速度和雙工設定的群組。需要介面驅動程式中的 `ethtool` 支援，以及支援 IEEE 802.3ad 動態鏈結聚合並進行了相應設定的交換器。提供容錯和負載平衡。

5 (balance-tlb)
: 調適性傳輸負載平衡。需要介面驅動程式中的 `ethtool` 支援，但不需要交換器支援。提供容錯和負載平衡。

6 (balance-alb)
: 調適性負載平衡。需要介面驅動程式中的 `ethtool` 支援，但不需要交換器支援。提供容錯和負載平衡。

如需各種模式的詳細描述，請參閱 https://www.kernel.org/doc/Documentation/networking/bonding.txt。

提示：Bonding 和 Xen

Bonding 裝置僅適用於具有多個實際網路卡的機器。在大多數組態中，這表示您只應該在 Dom0 中使用結合組態。此外，只有在您將多個網路卡指定給 VM 客體系統的情況下，在 VM 客體中設定結合才有用。

若要設定 bonding 裝置，請執行以下程序：

1. 執行 YaST › 系統 › 網路設定。
2. 使用新增，然後將裝置類型變更為Bond。按下一步繼續。

3. 選取為結合裝置指定 IP 位址的方法。有三種方法可供您選擇：

   - 無 IP 位址
   - 動態位址（透過 DHCP 或 Zeroconf）
   - 靜態指定的 IP 位址

   請使用適合您環境的方法。

4. 在Bond 從屬索引標籤中，透過啓用相關的核取方塊來選取應包含在 Bond 中的乙太網路裝置。
5. 編輯結合驅動程式選項並選取結合模式。

6. 確認參數 `miimon=100` 已新增至Bond 驅動程式選項。若沒有此參數，就無法定期檢查資料的完整性。

7. 按下一步，然後按一下確定離開 YaST，以建立裝置。

16.7.1 Bonding 從屬的熱插拔

在特定網路環境（例如高可用性）中，有時候您需要將 Bonding 從屬介面取代成其他介面。原因可能在於網路裝置不斷發生故障。解決方案是設定 Bonding 從屬的熱插拔。

請依一般方式設定 Bond（根據 `man 5 ifcfg-bonding`），例如：

```
ifcfg-bond0
          STARTMODE='auto' # or 'onboot'
          BOOTPROTO='static'
          IPADDR='192.168.0.1/24'
          BONDING_MASTER='yes'
          BONDING_SLAVE_0='eth0'
          BONDING_SLAVE_1='eth1'
          BONDING_MODULE_OPTS='mode=active-backup miimon=100'
```

使用 `STARTMODE=hotplug` 和 `BOOTPROTO=none` 指定從屬：

```
ifcfg-eth0
          STARTMODE='hotplug'
          BOOTPROTO='none'

ifcfg-eth1
          STARTMODE='hotplug'
          BOOTPROTO='none'
```

`BOOTPROTO=none` 會使用 `ethtool` 選項（若提供），但不會使用 `ifup eth0` 設定連結。這是因為從屬介面是由 Bond 主要裝置所控制的。

`STARTMODE=hotplug` 會使從屬介面在可用時自動加入 Bond。

需要變更 `/etc/udev/rules.d/70-persistent-net.rules` 中的 `udev` 規則，以便依匯流排 ID（udev `KERNELS` 關鍵字等同於 `hwinfo --netcard` 中的「SysFS BusID」）比對裝置，而不是依 MAC 位址比對。如此將允許更換有缺陷的硬體（位於同一插槽但 MAC 不同的網路卡），並避免在結合變更其所有從屬裝置的 MAC 位址時出現混淆。

例如：

```
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*",
```

```
KERNELS=="0000:00:19.0", ATTR{dev_id}=="0x0", ATTR{type}=="1",
KERNEL=="eth*", NAME="eth0"
```

開機時，systemd `network.service` 不會等待熱插拔從屬，但會等待結合就緒（至少需要一個可用的從屬）。當從系統移除其中一個從屬介面（從 NIC 驅動程式解除結合、NIC 驅動程式的 `rmmod` 或實際 PCI 熱插拔移除）時，核心會自動將它從 Bond 中移除。當將新卡新增至系統時（更換插槽中的硬體），`udev` 會使用匯流排型永久命名規則將它重新命名為從屬的名稱，然後為它呼叫 `ifup`。`ifup` 呼叫會自動將它加入 Bond。

16.8 設定用於網路組合的組合裝置

「連結彙總」一詞是描述結合（或聚合）網路連接以提供邏輯層的一般性詞彙。有時，您會看到「通道組合」、「乙太網路結合」和「連接埠截斷」等詞彙。它們都是同義詞，指的是同一個概念。

這一概念便是廣為人知的「結合」，並且最初整合於 Linux 核心之內（請參閱第 16.7 節 「設定 Bonding 裝置」瞭解其原始實作）。網路組合一詞指的是這一概念的新型實作。

結合與網路組合的主要不同之處在於，組合提供的是一組小型核心模組，它們負責為 teamd 例項提供介面。其餘所有操作都在使用者空間內處理。這與原始的結合實作不同，原始的結合實作單獨在其核心內包含了其所有功能。如需兩者的比較，請參閱表格 16.5 「結合與組合的功能比較」。

表格 16.5 結合與組合的功能比較

特性	結合	組合
廣播、輪替 TX 規則	是	是
使用中備份 TX 規則	是	是
LACP (802.3ad) 支援	是	是
以雜湊為基礎的 TX 規則	是	是
使用者可以設定雜湊函數	否	是
TX 負載平衡支援 (TLB)	是	是

特性	結合	組合
針對 LACP 的 TX 負載平衡支援	否	是
Ethtool 連結監控	是	是
ARP 連結監控	是	是
NS/NA (IPV6) 連結監控	否	是
針對 TX/RX 路徑的 RCU 鎖定	否	是
連接埠優先程度和粘性	否	是
單獨的依連接埠連結監控設定	否	是
多連結監控設定	受限制	是
VLAN 支援	是	是
多裝置堆疊	是	是

來源: http://libteam.org/files/teamdev.pp.pdf

結合與網路組合兩種實作可以同時使用。網路組合是現有結合實作的一個備用方案。它不會取代結合。

網路組合可用於多種用途。下文中將介紹兩種最重要的用途，分別是：

- 實現不同網路裝置之間的負載平衡。
- 當其中一個網路裝置出現故障時，容錯移轉至另一個裝置。

目前還沒有支援建立組合裝置的 YaST 模組。您需要手動設定網路組合。一般程序如下所示，該程序適用於所有網路組合組態：

程序 16.1 一般程序

1. 確定您已安裝所有必要的套件。安裝套件 `libteam-tools`、 `libteamdctl0` 和 `python-libteam`。

2. 在 `/etc/sysconfig/network/` 下建立組態檔案。通常將是 `ifcfg-team0`。如果需要多部網路組合裝置，請為它們加上數字並依次遞增。
   此組態檔案包含一些變數，在 man 頁面中有相關說明（請參閱 `man ifcfg` 和 `man ifcfg-team`）。系統內的 `/etc/sysconfig/network/ifcfg.template` 檔案中提供了範例組態。

3. 移除將用於組合裝置的介面的組態檔案（通常為 `ifcfg-eth0` 和 `ifcfg-eth1`）。建議製做一個備份並移除兩個檔案。Wicked 將重新建立這些組態檔案，並收入適用於組合的必要參數。

4. （選擇性）檢查 Wicked 組態檔案中是否已包含所需的一切內容。

   ```
   wicked show-config
   ```

5. 啓動網路組合裝置 `team0`：

   ```
   wicked ifup all team0
   ```

   如需其他除錯資訊，請在 `all` 子指令之後使用 `--debug all` 選項。

6. 檢查網路組合裝置的狀態。此動作可藉由下列指令來完成：

   - 從 Wicked 取得 teamd 實例的狀態：

     ```
     wicked ifstatus --verbose team0
     ```

   - 取得整個實例的狀態：

     ```
     teamdctl team0 state
     ```

   - 取得 teamd 實例的 systemd 狀態：

     ```
     systemctl status teamd@team0
     ```

   三者顯示的視圖依您的需求而定，略有不同。

7. 如果日後需要變更 `ifcfg-team0` 檔案中的內容，請使用以下指令重新載入其組態：

   ```
   wicked ifreload team0
   ```

不要使用 `systemctl` 來啓動或停止組合裝置！而應使用 `wicked` 指令，如上所述。

若要徹底移除組合裝置，請執行以下程序：

程序 16.2 移除組合裝置

1. 停止網路組合裝置 `team0`：

```
wicked ifdown team0
```

2. 將檔案 `/etc/sysconfig/network/ifcfg-team0` 重新命名為 `/etc/sysconfig/network/.ifcfg-team0`。在檔案名稱前面插入一個點，以使 `wicked`「看不到」它。如果您確實不再需要該組態，也可以移除該檔案。

3. 重新載入組態：

```
wicked ifreload all
```

16.8.1 使用案例：網路組合間的負載平衡

負載平衡用於提升頻寬。使用以下組態檔案可以建立具有負載平衡功能的網路組合裝置。繼續執行程序 16.1 「一般程序」中的步驟以設定裝置。透過 `teamdctl` 檢查輸出。

範例 16.12 實現網路組合間負載平衡的組態

```
STARTMODE=auto ❶
BOOTPROTO=static ❷
IPADDRESS="192.168.1.1/24" ❷
IPADDR6="fd00:deca:fbad:50::1/64" ❷

TEAM_RUNNER="loadbalance" ❸
TEAM_LB_TX_HASH="ipv4,ipv6,eth,vlan"
TEAM_LB_TX_BALANCER_NAME="basic"
TEAM_LB_TX_BALANCER_INTERVAL="100"

TEAM_PORT_DEVICE_0="eth0" ❹
TEAM_PORT_DEVICE_1="eth1" ❹

TEAM_LW_NAME="ethtool" ❺
TEAM_LW_ETHTOOL_DELAY_UP="10" ❻
TEAM_LW_ETHTOOL_DELAY_DOWN="10" ❻
```

❶ 控制組合裝置的啓動。`auto` 值表示介面會在網路服務可用時設定，並在每次重新開機時自動啓動。

如果您要自行控制裝置，而並不希望裝置自動啓動，請將 `STARTMODE` 設定為 `manual`。

2 設定靜態 IP 位址（此處的 IPv4 為 `192.168.1.1`，IPv6 為 `fd00:deca:fbad:50::1`）。
如果網路組合裝置應使用動態 IP 位址，請設定 `BOOTPROTO="dhcp"`，並移除（或備註）包含 `IPADDRESS` 和 `IPADDR6` 的行。

3 將 `TEAM_RUNNER` 設定為 `loadbalance` 以啓動負載平衡模式。

4 指定應聚合以建立網路組合裝置的一或多部裝置。

5 定義連結監控器以監控從屬裝置的狀態。使用預設值 `ethtool` 只會檢查裝置是否已啓動並可存取，因此檢查速度很快。但它不會檢查裝置是否可以真正地傳送或接收封包。
如果您需要進一步確信連接的可用性，請使用 `arp_ping` 選項。這樣會傳送 Ping 到任意主機（在 `TEAM_LW_ARP_PING_TARGET_HOST` 變數中設定）。只有在收到回覆後，才視為網路組合裝置已啓動。

6 以毫秒為單位定義連結建立（或斷開）與執行器收到通知之間的延遲。

16.8.2 使用案例：使用網路組合實現容錯移轉

容錯移轉引入了一個平行的備份網路裝置，用於確保關鍵網路組合裝置的高可用性。備份網路裝置將全天候執行，並在主要裝置出現故障時接管其工作。

使用以下組態檔案可以建立具有容錯移轉功能的網路組合裝置。繼續執行程序 16.1「一般程序」中的步驟以設定裝置。透過 `teamdctl` 檢查輸出。

範例 16.13 DHCP 網路組合裝置的組態

```
STARTMODE=auto ❶
BOOTPROTO=static ❷
IPADDR="192.168.1.2/24" ❷
IPADDR6="fd00:deca:fbad:50::2/64" ❷

TEAM_RUNNER=activebackup ❸
TEAM_PORT_DEVICE_0="eth0" ❹
TEAM_PORT_DEVICE_1="eth1" ❹

TEAM_LW_NAME=ethtool ❺
TEAM_LW_ETHTOOL_DELAY_UP="10" ❻
TEAM_LW_ETHTOOL_DELAY_DOWN="10" ❻
```

❶ 控制組合裝置的啓動。`auto` 值表示介面會在網路服務可用時設定，並在每次重新開機時自動啓動。
如果您要自行控制裝置，而並不希望裝置自動啓動，請將 `STARTMODE` 設定為 `manual`。

❷ 設定靜態 IP 位址（此處的 IPv4 為 `192.168.1.2`，IPv6 為 `fd00:deca:fbad:50::2`）。
如果網路組合裝置應使用動態 IP 位址，請設定 `BOOTPROTO="dhcp"`，並移除（或備註）包含 `IPADDRESS` 和 `IPADDR6` 的行。

❸ 將 `TEAM_RUNNER` 設定為 `activebackup` 以啓動容錯移轉模式。

❹ 指定應聚合以建立網路組合裝置的一或多部裝置。

❺ 定義連結監控器以監控從屬裝置的狀態。使用預設值 `ethtool` 只會檢查裝置是否已啓動並可存取，因此檢查速度很快。但它不會檢查裝置是否可以真正地傳送或接收封包。
如果您需要進一步確信連接的可用性，請使用 `arp_ping` 選項。這樣會傳送 Ping 到任意主機（在 `TEAM_LW_ARP_PING_TARGET_HOST` 變數中設定）。只有在收到回覆後，才視為網路組合裝置已啓動。

❻ 以毫秒為單位定義連結建立（或斷開）與執行器收到通知之間的延遲。

16.8.3 使用案例：組合裝置上的 VLAN

VLAN 是虛擬區域網路（Virtual Local Area Network）的縮寫。它允許多個邏輯（虛擬）乙太網路在一個單一實體乙太網路上執行，會在邏輯上將網路分割成不同的廣播網域，以便封包只可在為同一 VLAN 指定的連接埠之間進行交換。

下面的使用案例會在組合裝置的基礎上建立兩個靜態 VLAN：

- `vlan0`，結合到 IP 位址 `192.168.10.1`
- `vlan1`，結合到 IP 位址 `192.168.20.1`

請執行下列步驟：

1. 在交換器上啓用 VLAN 標記。如果您要針對組合裝置使用負載平衡，則交換器需要支援連結彙總控制通訊協定（LACP）（802.3ad）。如需詳細資料，請查閱硬體手冊。

2. 確定是否要針對組合裝置使用負載平衡或容錯移轉。依第 16.8.1 節 「使用案例：網路組合間的負載平衡」或第 16.8.2 節 「使用案例：使用網路組合實現容錯移轉」所述設定組合裝置。

3. 在 `/etc/sysconfig/network` 中，建立包含以下內容的 `ifcfg-vlan0` 檔案：

```
STARTMODE="auto"
BOOTPROTO="static" ❶
IPADDR='192.168.10.1/24' ❷
ETHERDEVICE="team0" ❸
VLAN_ID="0" ❹
VLAN='yes'
```

   ❶ 定義固定的 IP 位址（在 `IPADDR` 中指定）。

   ❷ 定義 IP 位址，此處包含其網路遮罩。

   ❸ 包含要用於 VLAN 介面的實際介面，此處是我們的組合裝置（`team0`）。

   ❹ 為 VLAN 指定唯一的 ID。檔案名稱和 `VLAN_ID` 最好與名稱 `ifcfg-vlanVLAN_ID` 對應。在我們的範例中，`VLAN_ID` 為 `0`，因而檔案名稱為 `ifcfg-vlan0`。

4. 將 `/etc/sysconfig/network/ifcfg-vlan0` 檔案複製到 `/etc/sysconfig/network/ifcfg-vlan1`，並變更以下值：

   - `IPADDR`，從 `192.168.10.1/24` 變更為 `192.168.20.1/24`。
   - `VLAN_ID`，從 `0` 變更為 `1`。

5. 啓動兩個 VLAN：

```
root # wicked ifup vlan0 vlan1
```

6. 檢查 `ifconfig` 的輸出：

```
root # ifconfig -a
[...]
vlan0     Link encap:Ethernet  HWaddr 08:00:27:DC:43:98
          inet addr:192.168.10.1 Bcast:192.168.10.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fedc:4398/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:816 (816.0 b)
```

```
vlan1     Link encap:Ethernet  HWaddr 08:00:27:DC:43:98
          inet addr:192.168.20.1 Bcast:192.168.20.255 Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fedc:4398/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:816 (816.0 b)
```

16.9 使用 Open vSwitch 的軟體定義網路

軟體定義網路（SDN）表示控制流量傳送目標的系統（控制平面）與轉遞流量至所選目的地的基礎系統（資料平面，也稱為轉遞平面）區分開來。這表示之前由單個且往往不太靈活的交換器執行的功能，現在可以分給交換器（資料平面）及其控制器（控制平面）共同執行。在此模式下，可以對控制器編寫程式，使其變得非常靈活，而且可以快速適應不斷變化的網路條件。

Open vSwitch 是實作與 OpenFlow 通訊協定相容的分散式虛擬多層交換器的軟體。OpenFlow 允許控制器應用程式修改交換器的組態。它會分層至 TCP 通訊協定，並對一定範圍的軟體和硬體實作。於是，單個控制器便可以驅動多個大相逕庭的交換器。

16.9.1 Open vSwitch 的優勢

使用 Open vSwitch 的軟體定義網路有多項優勢，尤其是與虛擬機器配合使用時：

- 可以輕鬆識別網路狀態。
- 網路及其即時狀態可以從一個主機移動至另一個主機。
- 可以追蹤網路動態，並啓用外部軟體對它們進行回應。
- 可以套用並處理網路封包中的標記以識別它們的來源或目標機器，並維護其他網路位置。可以設定及移轉標記規則。
- Open vSwitch 會實作 GRE 通訊協定（泛型路由封裝）。有了這點，舉例而言，您便可以將私人 VM 網路彼此相連。
- Open vSwitch 可以單獨使用，不過其設計是與網路硬體整合，可用於控制硬體交換器。

16.9.2 安裝 Open vSwitch

1. 安裝 Open vSwitch 及補充套件：

   ```
   root # zypper install openvswitch openvswitch-switch
   ```

   如果您想讓 Open vSwitch 與 KVM 監管程式配合使用，還需另外安裝 `tunctl`。如果您想讓 Open vSwitch 與 Xen 監管程式配合使用，還需另外安裝 `openvswitch-kmp-xen`。

2. 啓用 Open vSwitch 服務：

   ```
   root # systemctl enable openvswitch
   ```

3. 重新啓動電腦或使用 `systemctl` 立即啓動 Open vSwitch 服務：

   ```
   root # systemctl start openvswitch
   ```

4. 若要檢查 Open vSwitch 是否已正確啓動，請使用：

   ```
   root # systemctl status openvswitch
   ```

16.9.3 Open vSwitch 精靈與公用程式的綜覽

Open vSwitch 由一些元件組成，其中包括一個核心模組和不同的使用者空間元件。核心模組用於加速資料路徑，但 Open vSwitch 的精簡安裝並不需要該模組。

16.9.3.1 精靈

Open vSwitch 的中心可執行檔是其兩個精靈。啓動 `openvswitch` 服務時，它們也會間接啓動。

Open vSwitch 主要精靈（`ovs-vswitchd`）提供交換器的實作。Open vSwitch 資料庫精靈（`ovsdb-server`）為儲存 Open vSwitch 之組態和狀態的資料庫提供服務。

16.9.3.2 公用程式

Open vSwitch 還包含一些方便您配合使用的公用程式。以下列出了其中一部分，僅介紹了重要的指令。

`ovsdb-tool`
: 建立、升級、壓縮與查詢 Open vSwitch 資料庫。對 Open vSwitch 資料庫執行異動。

`ovs-appctl`
: 設定正在執行的 `ovs-vswitchd` 或 `ovsdb-server` 精靈。

`ovs-dpctl` 與 `ovs-dpctl-top`
: 建立、修改、視覺化與刪除資料路徑。使用此工具可能會干擾 `ovs-vswitchd`，後者也執行資料路徑管理。因此，它通常僅用於診斷。
`ovs-dpctl-top` 會建立 `top` 一類的資料路徑的視覺化。

`ovs-ofctl`
: 管理任何遵守 OpenFlow 通訊協定的交換器。`ovs-ofctl` 不僅可以與 Open vSwitch 互動。

`ovs-vsctl`
: 提供組態資料庫的概覽介面，可用於查詢及修改資料庫。實際上，它會顯示 `ovs-vswitchd` 的狀態，並可用於其設定。

16.9.4 使用 Open vSwitch 建立橋接器

以下範例組態使用 SUSE Linux Enterprise Server 上預設使用的 Wicked 網路服務。若要瞭解 Wicked 的詳細資訊，請參閱第 16.5 節 「手動設定網路連接」。

安裝並啓動 Open vSwitch 後，執行如下操作：

1. 若要設定虛擬機器使用的橋接器，請建立包含如下內容的檔案：

```
STARTMODE='auto'❶
BOOTPROTO='dhcp'❷
OVS_BRIDGE='yes'❸
OVS_BRIDGE_PORT_DEVICE_1='eth0'❹
```

1. 啓動網路服務時自動設定橋接器。
2. 設定 IP 位址時使用的通訊協定。
3. 將組態標記為 Open vSwitch 橋接器。
4. 選擇應新增至橋接器的一或多部裝置。若要新增多部裝置，請在文件中為每部裝置附加一行：

   ```
   OVS_BRIDGE_PORT_DEVICE_SUFFIX='DEVICE'
   ```

   SUFFIX 可以是任何英數字串。不過，為了避免之前的定義被覆寫，請確定每部裝置的 SUFFIX 均唯一。

將檔案儲存至 /etc/sysconfig/network 目錄，並命名為 ifcfg-br0。您也可以使用除 br0 以外自己喜歡的任何名稱。但檔案名稱必須以 ifcfg- 開頭。若要瞭解更多選項，請參閱 ifcfg 與 ifcfg-ovs-bridge 的 man 頁面（man 5 ifcfg 與 man 5 ifcfg-ovs-bridge）。

2. 現在，啓動橋接器：

   ```
   root # wicked ifup br0
   ```

   Wicked 完成後應該會輸出橋接器的名稱及狀態 up（顯示在名稱的旁邊）。

16.9.5 Open vSwitch 直接與 KVM 配合使用

如第 16.9.4 節 「使用 Open vSwitch 建立橋接器」中所述建立橋接器後，您便可使用 Open vSwitch 來管理透過 KVM/QEMU 建立的虛擬機器的網路存取。

1. 為了能夠發揮 Wicked 的最大效用，可以對之前設定的橋接器做進一步的變更。開啓之前建立的 /etc/sysconfig/network/ifcfg-br0 並為另一個連接埠裝置附加一行：

   ```
   OVS_BRIDGE_PORT_DEVICE_2='tap0'
   ```

   另外，將 BOOTPROTO 設定為 none。現在，此檔案的內容形式如下：

   ```
   STARTMODE='auto'
   BOOTPROTO='none'
   OVS_BRIDGE='yes'
   ```

```
OVS_BRIDGE_PORT_DEVICE_1='eth0'
OVS_BRIDGE_PORT_DEVICE_2='tap0'
```

新的連接埠裝置 `tap0` 將在下一步中進行設定。

2. 現在，為 `tap0` 裝置新增組態檔案：

```
STARTMODE='auto'
BOOTPROTO='none'
TUNNEL='tap'
```

將檔案儲存至 `/etc/sysconfig/network` 目錄，並命名為 `ifcfg-tap0`。

提示：允許其他使用者存取 TAP 裝置

若要透過以非 `root` 使用者身分啓動的虛擬機器使用此 TAP 裝置，請附加：

```
TUNNEL_SET_OWNER=USER_NAME
```

若要為整個群組授予存取權，請附加：

```
TUNNEL_SET_GROUP=GROUP_NAME
```

3. 最後，開啓定義為第一個 `OVS_BRIDGE_PORT_DEVICE` 的裝置的組態。如果之前未變更過名稱，應為 `eth0`。因此，開啓 `/etc/sysconfig/network/ifcfg-eth0`，並確定已設定以下選項：

```
STARTMODE='auto'
BOOTPROTO='none'
```

如果檔案尚不存在，請予以建立。

4. 使用 Wicked 重新啓動橋接器介面：

```
root # wicked ifreload br0
```

此操作還會觸發系統重新載入新定義的橋接器連接埠裝置。

5. 若要啓動某虛擬機器，請使用類似如下的內容：

```
root # qemu-kvm \
-drive file=/PATH/TO/DISK-IMAGE① \
```

```
-m 512 -net nic,vlan=0,macaddr=00:11:22:EE:EE:EE \
-net tap,ifname=tap0,script=no,downscript=no ❷
```

❶ 您要啓動之 QEMU 磁碟影像的路徑。

❷ 使用之前建立的 TAP 裝置（`tap0`）。

如需使用 KVM/QEMU 的更多資訊，請參閱《Virtualization Guide》。

16.9.6 Open vSwitch 與 libvirt 配合使用

如第 16.9.4 節 「使用 Open vSwitch 建立橋接器」中所述建立橋接器之後，可以將該橋接器新增至使用 `libvirt` 進行管理的現有虛擬機器。由於 `libvirt` 對 Open vSwitch 橋接器提供部份支援，因此您可以直接使用第 16.9.4 節 「使用 Open vSwitch 建立橋接器」中建立的橋接器，無需進一步變更網路組態。

1. 開啓目標虛擬機器的網域 XML 檔案：

   ```
   root # virsh edit VM_NAME
   ```

   將 `VM_NAME` 取代為所需虛擬機器的名稱。預設的文字編輯器即會開啓。

2. 在文件中尋找網路區段，即尋找以 `<interface type="...">` 開頭並以 `</interface>` 結尾的區段。
   將現有區段取代為類似如下的網路區段：

   ```
   <interface type='bridge'>
     <source bridge='br0'/>
     <virtualport type='openvswitch'/>
   </interface>
   ```

   重要：`virsh iface-*` 和虛擬機器管理員與 Open vSwitch 的相容性

   目前，在使用 `virsh iface-*` 工具和虛擬機器管理員的情況下，Open vSwitch 與 `libvirt` 還不相容。如果您使用此類工具，組態可能會破壞。

3. 現在，您可以照常啓動或重新啓動虛擬機器。

如需使用 `libvirt` 的更多資訊，請參閱《Virtualization Guide》。

16.9.7 更多資訊

http://openvswitch.org/support/

Open vSwitch 專案網站的文件區段

https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

Open Networking Foundation 編寫的關於軟體定義網路和 OpenFlow 通訊協定的白皮書

17 印表機操作

SUSE® Linux Enterprise Server 支援以多種類型的印表機進行列印，包括遠端網路印表機。您可以手動設定印表機，也可以使用 YaST 進行設定。如需組態設定指示，請參閱《部署指南》，第 11 章「使用 YaST 設定硬體元件」，第 11.3 節「設定印表機」。圖形和指令行公用程式都可用來啓動和管理列印工作。如果您的印表機無法如預期般運作，請參閱第 17.8 節 「疑難排解」。

CUPS（Common Unix Printing System，通用 UNIX 列印系統）是 SUSE Linux Enterprise Server 中的標準列印系統。

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，請確認硬體可支援印表機的介面，並且印表機採用適合的語言。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。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。

  目前，PostScript 正逐漸被 PDF 取代，後者成為標準列印工作格式。可直接列印 PDF（而不是僅僅是 PostScript）的 PostScript+PDF 印表機已面世。對於傳統的 PostScript 印表機，需要在列印工作流程中將 PDF 轉換為 PostScript。

標準印表機（PCL 和 ESC/P 語言）
: 如果已知印表機語言，列印系統可以藉由 Ghostscript 將 PostScript 工作轉換為對應的印表機語言。此處理階段稱為解釋。最知名的語言是大多數 HP 印表機與類似產品使用的 PCL，以及 Epson 印表機使用的 ESC/P。這些印表機語言通常都受 Linux 支援，並可產生不錯的列印效果。Linux 可能無法提供一些特殊的印表機功能。除了 HP 和 Epson 之外，目前尚沒有其他印表機製造商開發 Linux 驅動程式，並透過開放原始碼授權將這些驅動程式提供給 Linux 套裝作業系統供應商。

專屬印表機（也稱為 GDI 印表機）
: 這些印表機並不支援任何一般的印表機語言。它們使用自己的印表機語言，而當有新型號發行，那些語言也可能有所變更。這些印表機通常指有 Windows 驅動程式。如需相關資訊，請參閱第 17.8.1 節 「沒有標準印表機語言模式支援的印表機」。

在購買新印表機之前，請參考下列來源以檢查您想要購買的印表機之支援性：

http://www.linuxfoundation.org/OpenPrinting/
: OpenPrinting 首頁，含印表機資料庫。資料庫會顯示最新的 Linux 支援狀態。但是，Linux 版本僅可與生產期間可用的驅動程式整合。因此，目前被評選為「完全支援」的印表機，在最新的 SUSE Linux Enterprise Server 版本發行之後，可能將失去此稱號。因此，資料庫不一定能指出正確狀態，而僅提供估計值。

http://pages.cs.wisc.edu/~ghost/
: Ghostscript 網頁。

`/usr/share/doc/packages/ghostscript/catalog.devices`
: 內建 Ghostscript 驅動程式清單

17.1 CUPS 工作流程

使用者會建立列印工作。列印工作由要列印的資料和有關線上同時週邊作業器的資訊組成。其中包括印表機的名稱或列印佇列的名稱，還有可能包括有關過濾器（例如印表機特定的選項）的資訊。

每一台印表機都至少有一個專屬的列印佇列。線上同時週邊作業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，線上同時週邊作業器會透過過濾器與後端，傳送資料至印表機。

過濾器會將列印應用程式所產生的資料（通常為 PostScript 或 PDF，但也會有 ASCII、JPEG 等）轉換為印表機特定資料（PostScript、PCL、ESC/P 等）。印表機的特性描述在 PPD 檔案中。PPD 檔案含有印表機特定選項以及在印表機上啓用它們所需的參數。過濾器系統可確保啓用使用者所選取的選項。

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

17.2 連接印表機的方法和通訊協定

有各種方法可將印表機連接到系統。CUPS 的組態無法辨識本地印表機與透過網路連接到系統的印表機。如需有關印表機連線的詳細資訊，請參閱 http://en.opensuse.org/SDB:CUPS_in_a_Nutshell 上的文章 CUPS in a Nutshell（CUPS 概述）。

System z CUPS 不支援 z/VM 提供的本地連接到 IBM z Systems 大型主機的印表機及類似裝置。在這些平台上，僅可透過網路列印。網路印表機的電纜必須根據印表機製造商的說明來安裝。

警告：在執行中的系統變更纜線連接

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

17.3 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。

若要設定 PostScript 印表機，最好的方法是取得適當的 PPD 檔。`manufacturer-PPDs` 與 `OpenPrintingPPDs-postscript` 套件中提供了許多 PPD 檔案。請參閱第 17.7.3 節「各種套件中的 PPD 檔案」和第 17.8.2 節「PostScript 印表機沒有可用的 PPD 檔案」。

新的 PPD 檔案可儲存在 `/usr/share/cups/model/` 目錄中，或依照《部署指南》，第 11 章「使用 YaST 設定硬體元件」，第 11.3.1.1 節「使用 YaST 新增驅動程式」中所述透過 YaST 新增到列印系統中。之後，便可在印表機設定期間選取該 PPD 檔案。

如果印表機製造商要求您安裝整個軟體套件，請謹慎處理。這種安裝類型可能導致 SUSE Linux Enterprise Server 提供的支援失效。另外，列印指令可能會以不同的方式運作，並且系統可能不再能夠對其他製造商的裝置定址。基於此原因，不建議安裝製造商軟體。

17.4 網路印表機

網路印表機可支援各種通訊協定，有些甚至可同時支援。雖然大部分受支援的通訊協定已標準化，但有一些製造商還是會修改標準。這樣，製造商只會針對一部分作業系統提供驅動程式。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，您可能需要試驗各種選項以實現功能性組態。

CUPS 支援 `socket`、`LPD`、`IPP` 和 `smb` 通訊協定。

socket
: socket 指將純文字列印資料直接傳送到 TCP 通訊端的連接。經常使用的幾個 socket 連接埠號碼為 `9100` 或 `35`。裝置 URI（資源識別字串）的語法為 socket://`IP.OF.THE.PRINTER`:`PORT`，例如 `socket://192.168.2.202:9100/`。

LPD（行列式印表機精靈，Line Printer Daemon）
: RFC 1179 中有對 LPD 通訊協定的詳細介紹。在此通訊協定下，部分工作相關資料（如列印佇列的 ID）會先於實際列印的資料傳送。因此，在設定 LPD 通訊協定時，必須指定列印佇列。雖然各家印表機製造商不同，但其實作方式足以靈活地接受任何名稱做為列印佇列。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 或類似名稱。LPD 服務的連接埠號碼為 `515`。裝置 URI 的範例為 `lpd://192.168.2.202/LPT1`。

IPP（網際網路列印通訊協定，Internet Printing Protocol）
: IPP 是相對較新的通訊協定（1999 年），它以 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。正確設定 IPP 必須要有列印佇列的名稱。IPP 的連接埠號碼為 `631`。裝置 URI 的範例為 `ipp://192.168.2.202/ps` 和 `ipp://192.168.2.202/printers/ps`。

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

必須在設定組態之前決定印表機支援的通訊協定。如果製造商未提供所需資訊，可使用 `nmap` 指令（隨附於 `nmap` 套件）來確定通訊協定。`nmap` 會檢查主機上是否有開啓的連接埠。例如：

```
nmap -p 35,137-139,515,631,9100-10000 IP.OF.THE.PRINTER
```

17.5 以指令行工具設定 CUPS

CUPS 可透過 `lpinfo`、`lpadmin` 與 `lpoptions` 等指令行工具來設定。您需要一個裝置 URI，其中包含後端（如 USB 和參數）。若要判斷系統中的裝置 URI 是否有效，請使用以下 `lpinfo -v | grep ":/"` 指令：

```
# lpinfo -v | grep ":/"
direct usb://ACME/FunPrinter%20XL
network socket://192.168.2.253
```

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

```
lpadmin -p QUEUE -v DEVICE-URI -P PPD-FILE -E
```

裝置（`-v`）便會以 `QUEUE`（`-p`）的形式供您使用，並使用指定的 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 URI。

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

```
lpadmin -p ps -v usb://ACME/FunPrinter%20XL -P \
/usr/share/cups/model/Postscript.ppd.gz -E
```

下列範例是設定網路印表機：

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

如需 `lpadmin` 的更多選項，請參閱 `lpadmin(8)` 的 man 頁面。

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

1. 首先，列出所有選項：

   ```
   lpoptions -p QUEUE -l
   ```

   範例：

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

   啓用的預設選項前面會加上星號（*），用以識別。

2. 以 `lpadmin` 變更選項：

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

3. 檢查新設定：

   ```
   lpoptions -p QUEUE -l

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

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

17.6 由指令行開始列印

若要從指令行列印，請輸入 `lp -d QUEUENAME FILENAME`，並以相應的名稱取代 `QUEUENAME` 和 `FILENAME`。

有些應用程式有賴 `lp` 指令來進行列印。在此情況下，請在應用程式的列印對話方塊中輸入正確的指令（通常無需指定 `FILENAME`），例如 `lp -d QUEUENAME`。

17.7 SUSE Linux Enterprise Server 中的特殊功能

部分 CUPS 功能已針對 SUSE Linux Enterprise Server 做出調整。此處涵蓋部份最重要的變更。

17.7.1 CUPS 與防火牆

執行 SUSE Linux Enterprise Server 的預設安裝後，SuSEfirewall2 隨即會啓用，且網路介面會設定為處於「**外部區域**」中，這會阻擋內送流量。如需 SuSEFirewall2 組態的詳細資訊，請參閱《Security Guide》，第 15 章「Masquerading and Firewalls」，第 15.4 節「SuSEFirewall2」及 http://en.opensuse.org/SDB:CUPS_and_SANE_Firewall_settings。

17.7.1.1 CUPS 用戶端

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

17.7.1.2 CUPS 伺服器

如果 CUPS 伺服器位於受防火牆保護的受信任網路環境中，則應將網路介面設定為處於防火牆的「**內部區域**」中。建議不要在不可信網路環境中設定 CUPS 伺服器，除非您確定該伺服器受到特殊防火牆規則和 CUPS 組態中的安全設定的保護。

17.7.2 瀏覽網路印表機

CUPS 伺服器會定期宣告透過網路共用之印表機的可用性及狀態資訊。用戶端可以存取此資訊，以便在列印對話方塊之類的地方顯示可用印表機清單。這稱為「瀏覽」。

CUPS 伺服器透過傳統的 CUPS 瀏覽協定或 Bonjour/DND-SD 宣告其在網路上的列印佇列。為了能夠瀏覽網路列印佇列，服務 `cups-browsed` 需要在透過 CUPS 伺服器列印的所有用戶端上執行。預設不會啓動 `cups-browsed`。若要為使用中的工作階段啓動該服務，請使用 `sudo systemctl start cups-browsed`。若要確保它在開機後會自動啓動，請在所有用戶端上透過 `sudo systemctl enable cups-browsed` 予以啓用。

如果在啓動 `cups-browsed` 之後瀏覽無法工作，CUPS 伺服器可能是透過 Bonjour/DND-SD 宣告網路列印佇列。在這種情況下，您需要另外安裝套件 `avahi`，然後在所有用戶端上透過 `sudo systemctl start avahi-daemon` 啓動相關聯的服務。

17.7.3 各種套件中的 PPD 檔案

YaST 印表機組態使用安裝於 `/usr/share/cups/model/` 中的 PPD 檔案來設定 CUPS 的佇列。為了尋找適合印表機型號的 PPD 檔案，YaST 會對照硬體偵測期間確定的廠商和型號比較所有 PPD 檔案內的廠商和型號。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。

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。例如，如果您擁有 PostScript 印表機，可直接將 PPD 檔案複製到 `/usr/share/cups/model`（如果這些檔案尚不存在於 `manufacturer-PPDs` 或 `OpenPrintingPPDs-postscript` 套件中），以取得印表機的最佳組態。

其他 PPD 檔案由下列套件提供：

- gutenprint：Gutenprint 驅動程式及其相符的 PPD
- splix：SpliX 驅動程式及其相符的 PPD
- OpenPrintingPPDs-ghostscript：Ghostscript 內建驅動程式的 PPD
- OpenPrintingPPDs-hpijs：適用於非 HP 印表機之 HPIJS 驅動程式的 PPD

17.8 疑難排解

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

17.8.1 沒有標準印表機語言模式支援的印表機

這些印表機不支援任何的一般印表機語言，且只有特殊的專屬控制序列才能處理。因此它們僅可在製造廠商針對其開發驅動程式的作業系統版本上使用。GDI 是 Microsoft* 為繪圖裝置所開發的程式設計介面。製造廠商通常只提供 Windows 適用的驅動程式，而由於 Windows 驅動程式使用 GDI 介面，因此這些印表機也稱為 GDI 印表機。問題實際並不是出在程式設計介面上，而是因這些印表機只能透過相應印表機型號的專用印表機語言來定址所造成。

部分 GDI 印表機可切換到 GDI 模式或某種標準印表機語言來操作。如果手邊有印表機手冊，可以參閱其中內容。某些型號需要特殊的 Windows 軟體才能進行切換（請注意，從 Windows 列印時，Windows 印表機驅動程式可能都會將印表機切換回 GDI 模式）。對於其他 GDI 印表機，則可以使用標準印表機語言的延伸模組。

部分製造廠商提供其 印表機的專用驅動程式。專用印表機驅動程式的壞處在於，其不保證可與安裝的列印系統配合使用，也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。

與其費時費力研究如何讓專用 Linux 驅動程式運作，購買一台支援標準印表機語言（最好是 PostScript）的印表機可能更符合成本效益。這樣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之驅動程式更新的需要。

17.8.2 PostScript 印表機沒有可用的 PPD 檔案

如果 `manufacturer-PPDs` 或 `OpenPrintingPPDs-postscript` 套件不包含適用於 PostScript 印表機的 PPD 檔案，通常可以使用印表機製造商提供的驅動程式 CD 中的 PPD 檔案，或從印表機製造商的網頁下載適合的 PPD 檔案。

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

17.8.3 網路印表機連接方式

查明網路問題

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

檢查 TCP/IP 網路

TCP/IP 網路和名稱解析必須可作用。

檢查遠端 `lpd`

使用以下指令測試是否可以與 `Host` 上的 `lpd`（連接埠 `515`）建立 TCP 連接：

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

如果無法建立到 `lpd` 的連接，可能是 `lpd` 不在作用中，或是有基本網路問題。如果相應的 `lpd` 處於使用中狀態，並且主機接受查詢，請以 `root` 身分執行以下指令，以查詢遠端 `HOST` 上 `QUEUE` 的狀態報告：

```
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。如果您收到類似範例 17.1 「來自 `lpd` 的錯誤訊息」 中的回應，問題可能是由遠端 `lpd` 所造成的。

範例 17.1 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`

CUPS 網路伺服器預設每 30 秒在 UDP 連接埠 `631` 上廣播一次其佇列。因而可以使用下面的指令來測試網路中是否有廣播 CUPS 網路伺服器。請確定在執行指令之前停止本地 CUPS 精靈。

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

如果廣播 CUPS 網路伺服器存在，輸出將如 範例 17.2 「來自 CUPS 網路伺服器的廣播」 中所示。

範例 17.2 來自 CUPS 網路伺服器的廣播

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

System z 請注意，IBM z Systems 乙太網路裝置預設不接收廣播。

以下指令可用於測試是否可以與 `HOST` 上的 `cupsd`（連接埠 `631`）建立 TCP 連接：

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

如果無法與 `cupsd` 建立連接，則可能是 `cupsd` 不在使用中，或者存在基本網路問題。如果相應的 `cupsd` 在使用中且主機接受查詢，`lpstat -h HOST -l -t` 會傳回 `HOST` 上所有佇列的狀態報告（可能非常長）。

下面的指令可用於測試 `HOST` 上的 `QUEUE` 是否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
```

對網路印表機或列印伺服器機器進行疑難排解

有時，在列印伺服器機器中執行的線上同時週邊作業器在處理多項列印工作時會產生問題。這是列印伺服器機器中的線上同時週邊作業器造成的，目前尚無法解決此問題。因應措施是，直接透過 TCP 通訊端對連接到列印伺服器機器的印表機進行定址，以繞過列印伺服器機器中的線上同時週邊作業器。請參閱第 17.4 節「網路印表機」。

如此，列印伺服器機器僅充當各種不同資料傳輸方式之間（TCP/IP 網路和本地印表機連接）的轉換器。若要使用此方法，您需要知道列印伺服器機器上的 TCP 連接埠。如果印表機連接到列印伺服器機器且已啓動，則通常可以在開啓列印伺服器機器電源一段時間後，使用 `nmap` 套件中的 `nmap` 公用程式確定此 TCP 連接埠。例如，`nmap IP-address` 可能會傳送列印伺服器機器的以下輸出：

```
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，可以在連接埠 `9100` 上透過 TCP 通訊端對連接到列印伺服器機器的印表機定址。根據預設，`nmap` 僅會檢查 `/usr/share/nmap/nmap-services` 中所列出之一般熟知的幾個連接埠。若要檢查所有可能的連接埠，請使用指令 `nmap -p FROM_PORT-TO_PORT IP_ADDRESS`。這可能會花費一些時間。如需詳細資訊，請參閱 `nmap` 的 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。

17.8.4 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，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，請選取更適合印表機的 PPD 檔案。

17.8.5 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`）。在報告資料傳輸失敗之前，允許失敗嘗試的次數，由後端決定。因為進一步的嘗試可能徒勞無功，`cupsd` 會停止對應佇列的列印。排除問題的起因之後，系統管理員必須以指令 `cupsenable` 重新啓動列印。

17.8.6 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`。當伺服器上的 `cupsd` 接受列印工作時，系統會為其指定一個新工作號碼。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。列印工作通常會立刻轉遞，所以無法以用戶端主機上的工作編號來刪除，因為用戶端 `cupsd` 將列印工作轉遞給伺服器 `cupsd` 後，便認為列印工作已完成。

若要刪除伺服器上的列印工作，請使用 `lpstat -h cups.example.com -o` 之類的指令來確定伺服器上的工作編號。此情況假設伺服器尚未完成該列印工作（即尚未完全將它傳送到印表機）。以如下方式使用獲得的工作編號來刪除伺服器上的列印工作：

```
cancel -h cups.example.com QUEUE-JOBNUMBER
```

17.8.7 損毀的列印工作與資料傳輸錯誤

如果您在列印過程中關閉印表機或電腦，列印工作將保留在佇列中。一旦電腦（或印表機）重新開啓，列印工作將繼續進行。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，或主機與印表機之間的通訊發生錯誤，印表機將無法正確處理資料，並列印出許多含有不明字元的紙張。若要修復該問題，請執行以下步驟：

1. 若要停止列印，請從噴墨印表機取出所有紙張，或是打開雷射印表機的紙匣。高品質的印表機會有按鈕可取消目前的列印成品。
2. 列印工作可能仍在佇列中，因為只有將工作完全傳送到印表機之後，才會移除。使用 `lpstat -o` 或 `lpstat -h cups.example.com -o` 檢查目前正在列印的佇列。使用 `cancel` `QUEUE-JOBNUMBER` 或 `cancel -h cups.example.com` `QUEUE-JOBNUMBER` 刪除列印工作。
3. 即使列印工作已從佇列刪除，部份資料可能仍會傳送到印表機。請檢查對應佇列的 CUPS 後端程序是否仍在執行中，並將它終止。
4. 將印表機關閉一段時間以完全重設印表機。然後裝入紙張並開啓印表機電源。

17.8.8 CUPS 除錯

使用以下標準程序找出 CUPS 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. 啓動 `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。

17.8.9 更多資訊

有關在 SUSE Linux 上列印的詳細資訊，請參閱 http://en.opensuse.org/Portal:Printing 上的 openSUSE 支援資料庫中。許多特定問題的解決方法在「SUSE 知識庫」（http://www.suse.com/support/）中都有說明。請搜尋文字 `CUPS` 找到相關文章。

18 X Window System

X Window System (X11) 是 UNIX 中既成現實標準的圖形使用者介面。X 採網路結構，可讓應用程式在一個主機上啓動而在透過任何種類的網路 (LAN 或網際網路) 連接的其他主機上顯示。本章提供 X 組態的基本資訊，以及在 SUSE® Linux Enterprise Server 中使用字型的背景資訊。

X Window System 一般不需要任何組態設定。X 啓動期間會動態偵測硬體。因此，`xorg.conf` 的使用已被取代。如果您仍需要指定自訂選項來變更 X 的行為方式，您仍然可以透過修改 `/etc/X11/xorg.conf.d/` 下的組態檔案實現變更。

提示：IBM z Systems：設定圖形使用者介面

IBM z Systems 沒有 X.Org 支援的任何輸入或輸出裝置。因此，本節所述的所有組態程序皆不適用。如需 IBM z Systems 的更多相關資訊，請參閱《部署指南》，第 4 章「在 IBM z Systems 上安裝」。

18.1 安裝與設定字型

Linux 中的字型可分為兩個部分：

描邊或向量字型

包含字符形狀的數學說明和繪圖說明。因此，每個字符都可以調整為任意大小而無損品質。在可以使用此類字型（或字符）之前，需要將數學說明轉換為點陣（網格）。此過程稱為字型點陣化。字型影射（內嵌於字型中）改進和最佳化特定大小的展示效果。點陣化和影射透過 FreeType 程式庫完成。

Linux 下的常用格式是 PostScript Type 1 和 Type 2、TrueType 及 OpenType。

點陣圖或字型

包含為特定字型大小設計的像素陣列。點陣圖字型的展示速度超快，而且相當簡單。然而，與向量字型相比，它無法在無損品質的情況下調整。因此，這些字型通常以不同的大小分佈。近期，點陣圖字型仍然在 Linux 主控台使用，有時也見於終端機。

Linux 旗下最常用的格式是可攜式編譯格式（PCF）或字符點陣圖影射格式（BDF）。

這些字型的外觀會受兩個主要方面影響：

- 選擇合適的字型系列，
- 採用特定演算法展示字型，讓讀者觀看起來最為舒適。

最後一點僅與向量字型相關。雖然上述兩點非常主觀，但是需要建立一些預設值。

Linux 字型展示系統由若干個程式庫及其各種關係組成。基本字型程式庫是 FreeType (http://www.freetype.org/)，它會將字型字符從受支援的格式轉換為最佳化的點陣圖字符。展示程序由演算法及其參數（可能受父問題影響）控制。

使用 FreeType 的每個程式或程式庫都應參考 Fontconfig (http://www.fontconfig.org/) 程式庫。此程式庫會從使用者及系統收集字型組態。當使用者修改其 Fontconfig 設定時，此變更將導致支援 Fontconfig 的套用。

Arabic、Han 或 Phags-Pa 等程序檔所需的更複雜的 OpenType 塑形，以及其他更高層級的文字處理透過 Harfbuzz (http://www.harfbuzz.org/) 或 Pango (http://www.pango.org/) 來進行。

18.1.1 顯示已安裝的字型

若要大致瞭解系統上已安裝哪些字型，請執行 `rpm` 或 `fc-list` 指令。二者均會出色回答，但有可能因根據系統和使用者組態傳回不同的清單。

`rpm`

呼叫 `rpm` 可查看系統上已安裝哪些包含字型的軟體套件：

```
rpm -qa '*fonts*'
```

每個字型套件應符合此表示式。然而，指令可能傳回誤報，例如 `fonts-config`（可能既不是也不包含字型）。

`fc-list`

呼叫 `fc-list`，大致瞭解哪些字型系列可供存取、是否已安裝在系統上或主目錄中：

```
fc-list ':' family
```

注意：指令 `fc-list`

指令 `fc-list` 是 Fontconfig 程式庫的包裝程式。它可以從 Fontconfig（更確切地說，從它的快取）查詢大量有趣的資訊。請參閱 `man 1 fc-list` 以取得詳細資料。

18.1.2 檢視字型

如果您要瞭解已安裝的字型有何外觀，請使用指令 `ftview`（套件 `ft2demos`）或造訪 http://fontinfo.opensuse.org/ 。例如，若要以 14 點顯示 FreeMono 字型，請依照下方所述使用 `ftview`：

```
ftview 14 /usr/share/fonts/truetype/FreeMono.ttf
```

如果您需要進一步的資訊，請造訪 http://fontinfo.opensuse.org/ 以瞭解哪些樣式（標準、粗體、斜體等）和語言受支援。

18.1.3 查詢字型

若要查詢給定模式時使用哪中字型，請使用 `fc-match` 指令。

例如，如果您的模式包含已安裝的字型，`fc-match` 會傳回檔案名稱、字型系列和樣式：

```
tux > fc-match 'Liberation Serif'
LiberationSerif-Regular.ttf: "Liberation Serif" "Regular"
```

如果所需字型在系統上不存在，Fontconfig 會調用相符規則嘗試找到最接近的可用字型。換言之，您的要求被取代為：

```
tux > fc-match 'Foo Family'
DejaVuSans.ttf: "DejaVu Sans" "Book"
```

Fontconfig 支援別名，即用另一個系列名稱取代原始名稱。常見的情況是通用名稱，例如 「sans-serif」、「serif」 和 「monospace」。這些別名可取代為實際的系列名稱或者甚至是系列名稱的偏好設定清單：

```
tux > for font in serif sans mono; do fc-match "$font" ; done
DejaVuSerif.ttf: "DejaVu Serif" "Book"
DejaVuSans.ttf: "DejaVu Sans" "Book"
DejaVuSansMono.ttf: "DejaVu Sans Mono" "Book"
```

結果可能因系統而異，具體視目前安裝的字型而定。

注意：視 Fontconfig 而定的相似度規則

Fontconfig 始終根據給定要求傳回最相似的實際系列（如果至少已安裝一個系列）。「相似度」視 Fontconfig 的內部測量以及使用者或管理員的 Fontconfig 設定而定。

18.1.4 安裝字型

若要安裝新字型，可採用下列幾種主要的方法：

1. 將 `*.ttf` 或 `*.otf` 等字型檔案手動安裝至已知字型目錄。如果要將字型套用至整個系統，請使用標準目錄 `/usr/share/fonts`。如果要安裝到主目錄中，請使用 `~/.config/fonts`。
   如果不想使用標準目錄，Fontconfig 可讓您選擇其他目錄。使用 `<dir>` 元素通知 Fontconfig 所用目錄，相關詳細資料請參閱第 18.1.5.2 節 「Fontconfig XML 深入介紹」。

2. 使用 `zypper` 安裝字型。許多字型做為套件提供，隨附在 SUSE 套裝作業系統或位於 M17N:fonts (http://download.opensuse.org/repositories/M17N:/fonts/) 儲存庫中。使用以下指令將儲存庫新增至清單。例如，若要為 SLE 12 新增儲存庫：

   ```
   sudo zypper ar
        http://download.opensuse.org/repositories/M17N:/fonts/SLE_12_SP5/
   ```

   若要搜尋 *字型系列名稱*，請使用下面此指令：

   ```
   sudo zypper se 'FONT_FAMILY_NAME*fonts'
   ```

18.1.5 設定字型外觀

結果不一定會令人滿意，具體視展示媒體和字型大小而定。例如，如今常規監視器解析度為 100dpi，導致像素太大，字型看上去粗陋難看。

有些演算法可用於應對低解析度，例如消除鋸齒（灰階平滑化）、影射（適合網格）或子像素展示（在一個方向使解析度增至三倍）。這些演算法還可能因字型格式而異。

重要：子像素展示的父問題

在 SUSE 套裝作業系統中未使用子像素展示。雖然 FreeType2 支援此演算法，但是所涉及的幾項專利將於 2019 年年末到期。因此，除非系統含有 FreeType2 程式庫並且該程式庫中已編譯子像素展示，否則在 Fontconfig 中設定子像素展示選項沒有任何效果。

透過 Fontconfig，可為每種字型個別選取展示演算法，也可為一組字型選取展示演算法。

18.1.5.1 透過 sysconfig 設定字型

SUSE Linux Enterprise Server 在 Fontconfig 上提供一個 `sysconfig` 層。您可以從此處入手體驗字型組態。若要變更預設設定，請編輯組態檔案 `/etc/sysconfig/fonts-config`。（或使用 YaST sysconfig 模組）。編輯該檔案之後，請執行 `fonts-config`。

```
sudo /usr/sbin/fonts-config
```

重新啓動應用程式以查看效果。請記住下列指示：

- 一些應用程式不需要重新啓動。例如，Firefox 會不時重新讀取 Fontconfig 組態。新建立或重新載入的標籤日後可取得新的字型組態。
- 系統會在安裝或移除每個套件後自動呼叫 `fonts-config` 程序檔（否則表示字型軟體套件有誤）。
- 可以使用 `fonts-config` 指令行選項暫時覆寫每個 sysconfig 變數。如需詳細資料，請參閱 `fonts-config --help`。

有數個 sysconfig 變數可以變更。請參閱 `man 1 fonts-config` 或 YaST sysconfig 模組的說明頁面。系統提供下列變數：

展示演算法的用法

考慮 `FORCE_HINTSTYLE`、`FORCE_AUTOHINT`、`FORCE_BW`、`FORCE_BW_MONOSPACE`、`USE_EMBEDDED_BITMAPS` 和 `EMBEDDED_BITMAP_LANGAGES`

一般別名的偏好設定清單

請使用 `PREFER_SANS_FAMILIES`、`PREFER_SERIF_FAMILIES`、`PREFER_MONO_FAMILIES` 和 `SEARCH_METRIC_COMPATIBLE`

下面的清單提供了一些組態範例，從「最適合閱讀」字型（對比度較高）到「最漂亮」（較平滑）排序。

點陣圖字型

透過 `PREFER_*_FAMILIES` 變數可對點陣圖字型設定偏好。對於這些變數，請按照說明部份中的範例操作。須知，這些字型以黑白且未平滑形式展示，並且點陣圖字型只有數種大小。考慮使用

```
SEARCH_METRIC_COMPATIBLE="no"
```

來停用度量相容性驅動的系統名稱取代。

以黑白展示的可調整字型

展示可調整字型時如果未消除鋸齒，可能導致效果類似於點陣圖字型，同時保持字型可調整性。使用示意良好的 Liberation 系列之類。遺憾的是，系統中缺少影射良好的字型。設定下列變數以強制採用此方法。

```
FORCE_BW="yes"
```

以黑白展示的等寬字型

僅以未消除鋸齒的方式展示等寬字型，否則使用預設設定：

```
FORCE_BW_MONOSPACE="yes"
```

預設值

展示所有字型時都消除鋸齒。使用位元組碼解譯器（BCI）展示影射良好的字型，使用自動影射器（`hintstyle=hintslight`）展示其他字型。讓所有相關 sysconfig 變數保持預設設定。

CFF 字型

以 CFF 格式使用字型。人們認為它們在 FreeType2 中進行最新改進之後，比預設的 TrueType 字型更適合閱讀。請按照 `PREFER_*_FAMILIES` 範例嘗試一下。可以透過

```
SEARCH_METRIC_COMPATIBLE="no"
```

將展示調暗調粗，因為它們依預設透過 `hintstyle=hintslight` 展示。還可以考慮使用：

```
SEARCH_METRIC_COMPATIBLE="no"
```

專用自動影射器

即使對於影射良好的字型，也可以使用 FreeType2 的自動影射器。這可能會導致字形變得更粗、對比度更低，有時還會變得更模糊。設定下列變數以啓用此項：

```
FORCE_AUTOHINTER="yes"
```

可使用 `FORCE_HINTSTYLE` 來控制影射層級。

18.1.5.2 Fontconfig XML 深入介紹

Fontconfig 的組態設定格式是可延伸標記語言 (XML)。下面的幾個範例不是完整參考，而是簡要概觀。詳細資料及其他靈感可在 `man 5 fonts-conf` 或 `/etc/fonts/conf.d/` 中找到。

中心 Fontconfig 組態檔案是 `/etc/fonts/fonts.conf`，它及其他作品包括整個 `/etc/fonts/conf.d/` 目錄。若要自訂 Fontconfig，可在兩個位置插入變更：

FONTCONFIG 組態檔案

1. 系統範圍變更：編輯檔案 `/etc/fonts/local.conf`（依預設包含一個空的 `fontconfig` 元素）。
2. 使用者特定變更：編輯檔案 `~/.config/fontconfig/fonts.conf`。將 Fontconfig 組態檔案置於 `~/.config/fontconfig/conf.d/` 目錄中。

使用者特定變更會覆寫任何系統範圍的設定。

注意：已取代的使用者組態檔案

檔案 `~/.fonts.conf` 標示為已取代，不應繼續使用。請改為使用 `~/.config/fontconfig/fonts.conf`。

每個組態檔案都需要有 `fontconfig` 元素。因此，最小的檔案外觀如下：

```
<?xml version="1.0"?>
   <!DOCTYPE fontconfig SYSTEM "fonts.dtd">
   <fontconfig>
   <!-- Insert your changes here -->
   </fontconfig>
```

如果預設目錄不足，請插入 `dir` 元素及相應的目錄：

```
<dir>/usr/share/fonts2</dir>
```

Fontconfig 會以遞迴方式搜尋字型。

字型展示演算法可透過下列 Fontconfig 片段選擇（請參閱範例 18.1 「指定展示演算法」）：

範例 18.1 指定展示演算法

```
<match target="font">
 <test name="family">
  <string>FAMILY_NAME</string>
 </test>
 <edit name="antialias" mode="assign">
  <bool>true</bool>
 </edit>
 <edit name="hinting" mode="assign">
  <bool>true</bool>
 </edit>
 <edit name="autohint" mode="assign">
  <bool>false</bool>
 </edit>
 <edit name="hintstyle" mode="assign">
  <const>hintfull</const>
 </edit>
</match>
```

可以測試的字型的各個內容。例如，`<test>` 元素可測試字型系列（如範例中所示）、字型間隔、間距、字型格式以及其他。完全棄用 `<test>` 時，系統會將所有 `<edit>` 元素套用至每個字型（全域變更）。

範例 18.2 別名和系列名稱取代

規則 1

```
<alias>
 <family>Alegreya SC</family>
 <default>
  <family>serif</family>
 </default>
</alias>
```

規則 2

```
<alias>
 <family>serif</family>
 <prefer>
  <family>Droid Serif</family>
 </prefer>
</alias>
```

規則 3

```
<alias>
 <family>serif</family>
 <accept>
  <family>STIXGeneral</family>
 </accept>
</alias>
```

範例 18.2 「別名和系列名稱取代」中的規則可產生已設定優先順序的系列清單 (PFL)。系統會根據元素執行不同的動作：

規則 1 中的 `<default>`
: 此規則會在 PFL 末尾新增 `serif` 系列名稱。

規則 2 中的 `<prefer>`
: 只要 PFL 中存在 `Alegreya SC`，此規則就會在 PFL 中的第一個 `serif` 之前新增「Droid Serif」 。

規則 3 中的 `<accept>`
: 此規則會在 PFL 中第一個 `serif` 系列名稱之後，緊貼著它新增「STIXGeneral」 系列名稱。

將此片段放置在一起，當片段按規則 1 - 規則 2 - 規則 3 的順序出現並且使用者要求 「Alegreya SC」 時，便會建立如表格 18.1 「從 Fontconfig 規則產生 PFL」中所述的 PFL。

表格 18.1 從 FONTCONFIG 規則產生 PFL

順序	目前 PFL
申請	`Alegreya SC`
規則 1	`Alegreya SC` 、 `serif`
規則 2	`Alegreya SC` 、 `Droid Serif` 、 `serif`
規則 3	`Alegreya SC` 、 `Droid Serif` 、 `serif`

在 Fontconfig 的度量中，系列名稱具有最高優先順序，高於樣式、大小等其他模式。Fontconfig 會檢查目前在系統上安裝了哪個系列。如果已安裝 「Alegreya SC」，則 Fontconfig 會傳回此名稱。如果未安裝，則系統會檢查 「Droid Serif」等。

請小心。在變更 Fontconfig 片段的順序時，Fontconfig 可能傳回不同的結果，如表格 18.2 「變更順序後從 Fontconfig 規則產生 PFL 的結果」中所述。

表格 18.2 變更順序後從 FONTCONFIG 規則產生 PFL 的結果

順序	目前 PFL	記事
申請	`Alegreya SC`	執行相同的要求。
規則 2	`Alegreya SC`	`serif` 未在 FPL 中，未取代任何內容
規則 3	`Alegreya SC`	`serif` 未在 FPL 中，未取代任何內容
規則 1	`Alegreya SC` 、 `serif`	`Alegreya SC` 存在於 FPL 中，執行取代

注意：隱含。

將 `<default>` 別名視為此群組的分類或包含（如果未安裝）。如範例所示，`<default>` 應該一律優先於該群組的 `<prefer>` 和 `<accept>` 別名。

`<default>` 分類不限於一般別名 `serif`、`sans-serif` 和 `monospace`。如需複雜的範例，請參閱 `/usr/share/fontconfig/conf.avail/30-metric-aliases.conf`。

範例 18.3 「別名和系列名稱取代」中的下列 Fontconfig 片段會建立 `serif` 群組。當前一種字型未安裝時，此群組中的每個系列可取代其他系列。

範例 18.3 別名和系列名稱取代

```
<alias>
 <family>Alegreya SC</family>
 <default>
  <family>serif</family>
 </default>
</alias>
<alias>
 <family>Droid Serif</family>
 <default>
  <family>serif</family>
 </default>
</alias>
<alias>
 <family>STIXGeneral</family>
 <default>
  <family>serif</family>
 </default>
</alias>
<alias>
 <family>serif</family>
 <accept>
  <family>Droid Serif</family>
  <family>STIXGeneral</family>
  <family>Alegreya SC</family>
 </accept>
</alias>
```

優先順序由 `<accept>` 別名中的順序提供。類似地，可以使用較強的 `<prefer>` 別名。

範例 18.4 「別名和系列名稱取代」擴展了範例 18.2 「別名和系列名稱取代」。

範例 18.4 別名和系列名稱取代

規則 4

```
<alias>
```

```
 <family>serif</family>
 <accept>
  <family>Liberation Serif</family>
 </accept>
</alias>
```

規則 5

```
<alias>
 <family>serif</family>
 <prefer>
  <family>DejaVu Serif</family>
 </prefer>
</alias>
```

範例 18.4 「別名和系列名稱取代」中擴展的組態將導致下列 PFL 演變：

表格 18.3 從 FONTCONFIG 規則產生 PFL 的結果

順序	目前 PFL
申請	Alegreya SC
規則 1	Alegreya SC、serif
規則 2	Alegreya SC、Droid Serif、serif
規則 3	Alegreya SC、Droid Serif、serif
規則 4	Alegreya SC、Droid Serif、serif、Liberation Serif、STIXGeneral
規則 5	Alegreya SC、Droid Serif、DejaVu Serif、serif、Liberation Serif、STIXGeneral

注意：隱含式。

- 如果同一個一般名稱存在多個 `<accept>` 宣告，則最後剖析的宣告「勝出」。若有可能，不要在使用者（`/etc/fonts/conf.d/*-user.conf`）之後使用 `<accept>`。
- 如果同一個一般名稱存在多個 `<accept>` 宣告，則最後剖析的宣告「勝出」。若有可能，在全系統範圍的設定中不要在使用者之前使用 `<prefer>`。
- 對於相同的一般名稱，每個 `<prefer>` 宣告會覆寫 `<accept>` 宣告。如果管理員不僅希望使用者能使用 `<prefer>`，甚至還想允許其使用 `<accept>`，就不應該在系統範圍的組態中使用 `<prefer>`。另一方面，因為使用者通常都是使用 `<prefer>`，這種做法應該不會產生任何不利影響。我們還發現在系統範圍的組態中使用 `<prefer>` 的情況。

18.2 更多資訊

請安裝 `-doc` 以取得更多更深入詳盡的 X11 相關資訊。`man 5 xorg.conf` 更詳細地說明了手動設定的格式（如果需要）。您可在專案的首頁 http://www.x.org 上找到關於 X11 開發的更多資訊。

驅動程式位於 `xf86-video-*` 套件中，例如 `xf86-video-nv`。相關手冊頁中詳細描述了這些套件隨附的很多驅動程式。例如，如果使用 `nv` 驅動程式，可以在 `man 4 nv` 中找到有關此驅動程式的詳細資訊。

有關協力廠商驅動程式的資訊可在 `/usr/share/doc/packages/<套件名稱>` 中找到。例如，安裝套件後，`x11-video-nvidiaG03` 的文件就位於 `/usr/share/doc/packages/x11-video-nvidiaG03` 中。

19 使用 FUSE 存取檔案系統

FUSE 是使用者空間中的檔案系統（file system in user space）的縮寫。這表示您可以非特權使用者的身分設定並掛接檔案系統。一般情況下，只有 `root` 才能執行此任務。FUSE 自身就是一個核心模組。將 FUSE 與外掛程式結合便能延伸其功能，幾乎可存取所有檔案系統，如遠端 SSH 連接、ISO 影像及其他。

19.1 設定 FUSE

您必須先安裝套件 `fuse` 才能使用 FUSE。是否需要以獨立套件形式提供的其他外掛程式，取決於要使用的檔案系統。

一般而言，您無需設定 FUSE。但建議您建立可將所有掛接點組合於其中的目錄。例如，可以建立目錄 `~/mounts` 並在該處插入不同檔案系統的子目錄。

19.2 裝載 NTFS 分割區

新技術檔案系統（NTFS, New Technology File System）是 Windows 的預設檔案系統。在一般情況下，由於非特權使用者無法使用外部 FUSE 程式庫掛接 NTFS 區塊裝置，因此下文所述的 Windows 分割區掛接程序需要 root 特權。

1. 切換為 `root` 身份，然後安裝套件 `ntfs-3g`。SUSE Linux Enterprise Workstation Extension 中提供了該套件。
2. 建立將用作掛接點的目錄，例如 `~/mounts/windows`。
3. 確定所需的 Windows 分割區。使用 YaST 並啓動磁碟分割程式模組，以檢視屬於 Windows 的分割區，但不要做任何修改。或者也可以切換為 `root` 身份，然後執行 `/sbin/fdisk -l`。尋找分割區類型為 `HPFS/NTFS` 的分割區。
4. 在讀寫模式下裝載分割區。使用相應的 Windows 分割區取代佔位符 `DEVICE`：

   ```
   ntfs-3g /dev/DEVICE MOUNT POINT
   ```

若要在唯讀模式下使用 Windows 分割區，請附加 `-o ro`：

```
ntfs-3g /dev/DEVICE MOUNT POINT -o ro
```

指令 `ntfs-3g` 使用目前的使用者（UID）與群組（GID）裝載指定裝置。若要對其他使用者設定寫入權限，請使用指令 `id USER` 以取得 UID 與 GID 的輸出。使用下列指令進行設定：

```
id tux
uid=1000(tux) gid=100(users) groups=100(users),16(dialout),33(video)
ntfs-3g /dev/DEVICE MOUNT POINT -o uid=1000,gid=100
```

可在 man 頁面中找到其他選項。

若要卸載資源，請執行 `fusermount -u` 掛接點。

19.3 更多資訊

如需詳細資訊，請參閱 FUSE 的首頁 http://fuse.sourceforge.net。

20 管理核心模組

雖然 Linux 屬於單核心，但可透過核心模組加以延伸。這些特殊物件可以插入到核心中，並可視需要移除。就實際角度而言，核心模組使新增和移除核心自身未包含的驅動程式和介面成為現實。Linux 提供了數個用於管理核心模組的指令。

20.1 使用 lsmod 和 modinfo 列出載入的模組

使用 `lsmod` 指令可檢視目前載入了哪些核心模組。該指令的輸出可能如下所示：

```
tux >  lsmod
Module                  Size  Used by
snd_usb_audio         188416  2
snd_usbmidi_lib        36864  1 snd_usb_audio
hid_plantronics        16384  0
snd_rawmidi            36864  1 snd_usbmidi_lib
snd_seq_device         16384  1 snd_rawmidi
fuse                  106496  3
nfsv3                  45056  1
nfs_acl                16384  1 nfsv3
```

輸出內容分為三欄：`Module` 欄列出所載入模組的名稱，`Size` 欄顯示各模組的大小。`Used by` 欄顯示參考模組的程序數及其名稱。請注意，此清單可能不完整。

若要檢視有關特定核心模組的詳細資訊，請使用 `modinfo MODULE_NAME` 指令。其中 `MODULE_NAME` 為所需核心模組的名稱。請注意，`modinfo` 二進位檔案位於使用者的 PATH 環境變數中未包含的 `/sbin` 目錄下。這意味著，當您以普通使用者身分執行 `modinfo` 指令時，必須指定該二進位檔案的完整路徑：

```
$ /sbin/modinfo kvm
filename:       /lib/modules/4.4.57-18.3-default/kernel/arch/x86/kvm/kvm.ko
license:        GPL
author:         Qumranet
srcversion:     BDFD8098BEEA517CB75959B
depends:        irqbypass
intree:         Y
vermagic:       4.4.57-18.3-default SMP mod_unload modversions
signer:         openSUSE Secure Boot Signkey
sig_key:        03:32:FA:9C:BF:0D:88:BF:21:92:4B:0D:E8:2A:09:A5:4D:5D:EF:C8
sig_hashalgo:   sha256
parm:           ignore_msrs:bool
parm:           min_timer_period_us:uint
parm:           kvmclock_periodic_sync:bool
```

```
parm:           tsc_tolerance_ppm:uint
parm:           lapic_timer_advance_ns:uint
parm:           halt_poll_ns:uint
parm:           halt_poll_ns_grow:int
parm:           halt_poll_ns_shrink:int
```

20.2 新增和移除核心模組

雖然可以使用 `insmod` 和 `rmmod` 分別新增和移除核心模組，但建議使用 `modprobe` 工具來執行這些操作。`modprobe` 具有多項重要優勢，包括自動解析相依項以及將核心模組加入黑名單。

如果不指定任何參數，使用 `modprobe` 指令會安裝指定的核心模組。必須使用 `root` 特權來執行 `modprobe`：

```
tux > sudo modprobe acpi
```

若要移除核心模組，請使用 `-r` 參數：

```
sudo modprobe -r acpi
```

20.2.1 開機時自動載入核心模組

您可以選擇不手動載入核心模組，而是使用 `system-modules-load.service` 服務在開機期間自動載入這些模組。若要啓用核心模組，請將 `.conf` 檔案新增到 `/etc/modules-load.d/` 目錄下。建議為組態檔案指定與模組相同的名稱，例如：

```
/etc/modules-load.d/rt2800usb.conf
```

組態檔案中必須包含所需核心模組的名稱（例如 `rt2800usb`）。

透過上述的這個技巧，無需指定任何參數即可載入核心模組。如果您需要使用特定選項載入核心模組，請將組態檔案新增到 `/etc/modprobe.d/` 目錄下。該檔案的副檔名必須為 `.conf`。檔案名稱必須符合以下命名慣例：`priority-modulename.conf`，例如：`50-thinkfan.conf`。組態檔案中必須包含核心模組名稱及所需參數。您可以使用以下範例指令來建立包含核心模組名稱及其參數的組態檔案：

```
echo "options thinkpad_acpi fan_control=1" | sudo tee /etc/modprobe.d/thinkfan.conf
```

注意：載入核心模組

當偵測到裝置或使用者空間要求特定功能時，系統會自動載入大多數核心模組。因此，很少需要手動將模組新增到 `/etc/modules-load.d/`。

20.2.2 使用 modprobe 將核心模組加入黑名單

將某個核心模組加入黑名單後，開機期間便不再會載入該模組。要停用您懷疑可能導致系統出現問題的某個模組時，此功能十分實用。請注意，您仍可透過使用 `insmod` 或 `modprobe` 工具來手動載入加入黑名單的核心模組。

若要將模組加入黑名單，請將 `blacklist MODULE_NAME` 一行新增到 `/etc/modprobe.d/50-blacklist.conf` 檔案中。例如：

```
blacklist nouveau
```

以 root 身分執行 `mkinitrd` 指令以產生新的 `initrd` 影像，然後將機器重新開機。可使用以下指令執行上述步驟：

```
su
echo "blacklist nouveau" >> /etc/modprobe.d/50-blacklist.conf && mkinitrd && reboot
```

如果只想暫時停用某個核心模組，可在開機期間即時將其加入黑名單。若要實現此目標，請在開機螢幕顯示時按 E 鍵。這樣，您會進入一個可供您修改開機參數的小編輯器。找到如下所示的行：

```
linux /boot/vmlinuz...splash= silent quiet showopts
```

將 `modprobe.blacklist=MODULE_NAME` 指令新增到該行結尾處。例如：

```
linux /boot/vmlinuz...splash= silent quiet showopts modprobe.blacklist=nouveau
```

按 F10 或 Ctrl—X 以依照指定的組態開機。

若要透過 GRUB 將某個核心模組永久加入黑名單，請開啓要編輯的 `/etc/default/grub` 檔案，將 `modprobe.blacklist=MODULE_NAME` 選項新增到 `GRUB_CMD_LINUX` 指令中。然後執行 `sudo grub2-mkconfig -o /boot/grub2/grub.cfg` 指令使變更生效。

21 使用 udev 進行動態核心裝置管理

核心可以新增或移除執行中系統內幾乎所有的裝置。裝置狀態的變更（無論裝置插入或移除）必須傳播至使用者空間。插入及識別裝置後需要對其進行設定。如果辨識到的裝置狀態發生任何變更，必須通知該裝置的使用者。udev 會提供所需的基礎結構，以便動態維護 /dev 目錄中的裝置節點檔案和符號連結。udev 規則能將外部工具插入核心裝置事件處理。因而，您可以透過新增在核心裝置處理過程中執行的特定程序檔，來自訂 udev 裝置處理方式，或者可以在裝置處理期間要求並輸入其他資料進行評估。

21.1 /dev 目錄

/dev 中的裝置節點可用來存取對應的核心裝置。透過 udev，/dev 目錄會反映核心的目前狀態。每個核心裝置都有一個對應的裝置檔案。如果裝置與系統的連接中斷，該裝置節點就會遭到移除。

/dev 目錄的內容保存在暫存檔案系統中，而且所有檔案都會在每次系統開機時顯示。根據系統設計，手動建立或修改的檔案在重新開機後都將遺失。無論可使用 systemd-tmpfiles 建立的對應核心裝置狀態為何，靜態檔案和目錄都必須在 /dev 目錄中。組態檔案可在 /usr/lib/tmpfiles.d/ 和 /etc/tmpfiles.d/ 中找到；如需詳細資訊，請參閱 systemd-tmpfiles(8) 線上文件。

21.2 核心 uevent 和 udev

sysfs 檔案系統會輸出必要的裝置資訊。每個核心已偵測和啓始化的裝置，都會建立包含其裝置名稱的目錄。其中會包含裝置特定的屬性內容。

每次新增或移除裝置時，核心都會傳送 uevent 來通知 udev 此變更。udev 精靈會在啓動時從 /usr/lib/udev/rules.d/*.rules 和 /etc/udev/rules.d/*.rules 檔案中讀取並剖析所有規則，然後將剖析結果保留在記憶體中。如果變更、新增或移除了規則檔案，精靈可以使用 udevadm control --reload 指令重新載入這些規則在記憶體中的表示。如需有關 udev 規則及其語法的詳細資訊，請參閱第 21.6 節 「透過 udev 規則影響核心裝置事件的處理」。

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

21.3 驅動程式、核心模組和裝置

核心匯流排驅動程式會查探裝置。核心（kernel）會為每個偵測到的裝置建立一個內部裝置結構，而驅動程式核心（core）會向 `udev` 精靈傳送一個 uevent。匯流排裝置會以特殊格式的 ID 識別本身，表明其為何種裝置。通常這些 ID 會包含廠商和產品 ID，以及其他子系統特定值。每個匯流排都會指定自己的 ID 配置，即所謂的 `MODALIAS`。核心會接收這些裝置資訊，並根據這些資訊設定 `MODALIAS` ID 字串，然後隨事件一起傳送該字串。例如，USB 滑鼠的 ID 字串將如下所示：

```
MODALIAS=usb:v046DpC03Ed2000dc00dsc00dp00ic03isc01ip02
```

每個裝置驅動程式都包含有裝置可處理的已知別名清單。清單會包含在核心模組檔案本身。depmod 程式會讀取 ID 清單，並且為目前所有可用模組在核心的 `/lib/modules` 目錄中建立 `modules.alias` 檔案。透過此基礎結構，模組載入方式就會像在每次出現帶有 `MODALIAS` 識別碼的事件時呼叫 `modprobe` 一樣容易。如果是呼叫 `modprobe $MODALIAS`，此次呼叫就會比對裝置的已組織裝置別名和模組指定別名。如果有找到符合項目，該模組就可載入。以上這一切都是由 `udev` 自動觸發。

21.4 開機和初始裝置設定

在 `udev` 精靈執行之前，於開機過程中發生的所有裝置事件都會遺失，這是因為處理這些事件的基礎結構位於根檔案系統中，在該階段無法使用。為彌補這一損失，核心在 sysfs 檔案系統中之每部裝置的裝置目錄中都提供了一份 `uevent` 檔案。使用 `add` 寫入該檔案，核心便可重新傳送與開機期間所遺失的相同事件。負責 `/sys` 中所有 `uevent` 檔案的簡易迴圈，可以再次觸發所有事件，建立裝置節點並執行裝置設定。

例如，開機期間出現的 USB 滑鼠可能無法由早期的開機邏輯啓始化，這是因為當時尚無法使用驅動程式。裝置探查事件遺失，而且無法找到裝置的核心模組。您無需手動搜尋連接的裝置，`udev` 會在根檔案系統可用後向核心要求所有裝置事件，這樣 USB 滑鼠裝置的事件就會再次執行。現在，它會在已掛接根目錄檔案系統中找到核心模組，並讓 USB 滑鼠完成啓始化。

從使用者空間的角度，執行期間的裝置冷插拔（Co1dP1ud）順序和裝置探查並沒有明顯的不同。這兩種情況都會使用相同規則來進行比對，而且會執行相同的設定程式。

21.5 監控執行中的 udev 精靈

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

```
UEVENT[1185238505.276660] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1 (usb)
UDEV  [1185238505.279198] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1 (usb)
UEVENT[1185238505.279527] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UDEV  [1185238505.285573] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0 (usb)
UEVENT[1185238505.298878] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/
input10 (input)
UDEV  [1185238505.305026] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/
input10 (input)
UEVENT[1185238505.305442] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/
input10/mouse2 (input)
UEVENT[1185238505.306440] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/
input10/event4 (input)
UDEV  [1185238505.325384] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/
input10/event4 (input)
UDEV  [1185238505.342257] add   /devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/
input10/mouse2 (input)
```

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

`udevadm monitor --env` 會顯示完整的事件環境：

```
ACTION=add
DEVPATH=/devices/pci0000:00/0000:00:1d.2/usb3/3-1/3-1:1.0/input/input10
SUBSYSTEM=input
SEQNUM=1181
NAME="Logitech USB-PS/2 Optical Mouse"
PHYS="usb-0000:00:1d.2-1/input0"
```

```
UNIQ=""
EV=7
KEY=70000 0 0 0 0
REL=103
MODALIAS=input:b0003v046DpC03Ee0110-e0,1,2,k110,111,112,r0,1,8,amlsfw
```

udev 也會將訊息傳送到 syslog。控制哪些訊息要傳送到 syslog 的預設 syslog 優先程度是在 udev 的組態檔 /etc/udev/udev.conf 中指定。可以使用 udevadm control --log_priority=LEVEL/NUMBER 變更執行中精靈的記錄優先程度。

21.6 透過 udev 規則影響核心裝置事件的處理

udev 規則可以比對核心新增至事件本身的任何內容，或者核心輸出到 sysfs 的任何資訊。規則也可向外部程式要求其他資訊。系統會將事件與目錄 /usr/lib/udev/rules.d/（適用於預設規則）和 /etc/udev/rules.d（系統專屬的組態）中提供的所有規則進行比對。

規則檔案中的每一行都包含至少一個鍵值組合。鍵類型共有兩種，包括比對和指定鍵。當所有比對鍵都與指定值相符時就會套用規則，而該指定值就會指定給指定鍵。相符規則可以指定裝置節點的名稱、新增指向該節點的符號連結，或是在事件處理過程中執行指定的程式。如果找不到任何符合規則，就會使用預設的裝置節點名稱來建立裝置節點。如需有關規則語法和系統提供之用於比對或輸入資料的鍵的詳細資訊，請參閱 udev man 頁面。以下範例規則提供了對 udev 規則語法的基本介紹。這些範例規則全部摘自 udev 預設規則集 /usr/lib/udev/rules.d/50-udev-default.rules。

範例 21.1 範例 udev 規則

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

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

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

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

主控台規則包含三個鍵：一個比對鍵（KERNEL）以及兩個指定鍵（MODE、OPTIONS）。KERNEL 比對規則用於搜尋類型為主控台的所有項目的裝置清單。只有完全符合的項目才有效，才會觸發此規則讓其執行。MODE 鍵用於將特殊權限指派給

裝置節點，在此案例中，僅此裝置的擁有者才會被指派讀取與寫入權限。OPTIONS 鍵用於將此規則做為要套用至所有此類型裝置的最後一條規則。符合此特殊裝置類型的任何後續規則都不會生效。

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

印表機規則可處理 USB 印表機，它包含兩個比對鍵，必須同時套用這兩個鍵才能套用整個規則（SUBSYSTEM 與 KERNEL）。三個指定鍵用於命名此裝置類型（NAME）、建立符號裝置連結（SYMLINK）以及對此裝置類型的成員進行分組（GROUP）。在 KERNEL 鍵中使用 * 萬用字元可使其符合多部 lp 印表機裝置。可以在 NAME 與 SYMLINK 鍵中使用替代項，透過內部裝置名稱延伸這些字串。例如，第一部 lp USB 印表機的符號連結會讀取 /dev/usblp0。

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

某些一般特性適用於所有規則：

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

21.6.1 在 udev 規則中使用運算子

建立金鑰時，您可以根據要建立的金鑰類型從多個運算子中進行選擇。比對鍵通常用於尋找符合或明顯不符合搜尋值的值。比對鍵可包含以下運算子：

`==`
: 比較是否相等。如果鍵包含搜尋模式，則所有符合此模式的結果均有效。

`!=`
: 比較是否不相等。如果鍵包含搜尋模式，則所有符合此模式的結果均有效。

指派鍵可使用以下運算子：

`=`
: 將某個值指派給鍵。如果鍵先前包含值清單，則此鍵將重設並僅指派單一值。

`+=`
: 將某個值新增至包含項目清單的鍵。

`:=`
: 指派最終值。不允許後續規則再做任何變更。

21.6.2 在 udev 規則中使用替代項

`udev` 規則支援使用佔位符與替代項。使用方式與任何其他程序檔中的方式類似。在 `udev` 規則中可以使用以下替代項：

`%r`，`$root`
: 依預設為裝置目錄 `/dev`。

`%p`，`$devpath`
: `DEVPATH` 的值。

`%k`，`$kernel`
: `KERNEL` 的值或內部裝置名稱。

`%n`，`$number`
: 裝置編號。

`%N`, `$tempnode`
: 裝置檔案的暫存名稱。

`%M`, `$major`
: 裝置的主要編號。

`%m`, `$minor`
: 裝置的次要編號。

`%s{ATTRIBUTE}`, `$attr{ATTRIBUTE}`
: `sysfs` 屬性的值（透過 `ATTRIBUTE` 指定）。

`%E{VARIABLE}`, `$env{VARIABLE}`
: 環境變數的值（透過 `VARIABLE` 指定）。

`%c`, `$result`
: `PROGRAM` 的輸出。

`%%`
: `%` 字元。

`$$`
: `$` 字元。

21.6.3 使用 udev 比對鍵

比對鍵定義要套用 `udev` 規則所必須滿足的條件。以下為可用的比對鍵：

`ACTION`
: 事件動作的名稱，例如新增或移除裝置時的 `add` 或 `remove`。

`DEVPATH`
: 事件裝置的裝置路徑，例如 `DEVPATH=/bus/pci/drivers/ipw3945`，用於搜尋與 ipw3945 驅動程式相關的所有事件。

`KERNEL`
: 事件裝置的內部（核心）名稱。

`SUBSYSTEM`
: 事件裝置的子系統，例如 `SUBSYSTEM=usb`，適用於與 USB 裝置相關的所有事件。

ATTR{FILENAME}

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

KERNELS

讓 udev 向上搜尋符合裝置名稱的裝置路徑。

SUBSYSTEMS

讓 udev 向上搜尋相符裝置子系統名稱的裝置路徑。

DRIVERS

讓 udev 向上搜尋相符裝置驅動程式名稱的裝置路徑。

ATTRS{FILENAME}

讓 udev 向上搜尋與 sysfs 屬性值相符之裝置的裝置路徑。

ENV{KEY}

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

PROGRAM

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

RESULT

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

21.6.4 使用 udev 指定鍵

與上述比對鍵不同，指定鍵不會說明必須滿足的條件，而是將值、名稱和動作指定給 udev 維護的裝置節點。

NAME

要建立之裝置節點的名稱。如果規則已設定了節點名稱，則將忽略適用於此節點的所有其他含 NAME 鍵的規則。

`SYMLINK`

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

`OWNER、GROUP、MODE`

新裝置節點的權限。在此處指定的值將覆寫已編譯的任何項目。

`ATTR{KEY}`

指定要寫入事件裝置之 `sysfs` 屬性的值。如果使用運算子 `==`，也會使用此鍵來比對 `sysfs` 屬性的值。

`ENV{KEY}`

告知 `udev` 將某變數輸出到環境中。如果使用運算子 `==`，也會使用此鍵來比對環境變數。

`RUN`

告知 `udev` 將某程式新增至要為此裝置執行的程式清單。對極短任務套用此項時要格外小心，以免封鎖此裝置的其他事件。

`LABEL`

在 `GOTO` 可以跳轉之處新增一個標籤。

`GOTO`

告知 `udev` 跳過數個規則，繼續執行 `GOTO` 鍵所參考標籤對應的規則。

`IMPORT{TYPE}`

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

- `program` 可告知 `udev` 執行外部程式並輸入其輸出。
- `file` 可告知 `udev` 輸入文字檔。
- `parent` 可告知 `udev` 輸入父代裝置中儲存的鍵。

`WAIT_FOR_SYSFS`

告知 `udev` 等待系統為特定裝置建立指定的 `sysfs` 檔案。例如，`WAIT_FOR_SYSFS="ioerr_cnt"` 通知 `udev` 等待，直到 `ioerr_cnt` 檔案建立。

OPTIONS

OPTION 鍵可以使用多個值：

- last_rule 告知 udev 忽略所有後續規則。
- ignore_device 告知 udev 完全忽略此事件。
- ignore_remove 告知 udev 忽略針對該裝置的所有後續移除事件。
- all_partitions 告知 udev 為區塊裝置上的所有可用分割區建立裝置節點。

21.7 永久裝置命名

動態裝置目錄和 udev 規則基礎架構讓系統可以為所有磁碟裝置提供固定名稱，無論裝置的辨識順序或所使用的連接為何。核心所建立的每個相應區塊裝置，都會採用針對特定匯流排、磁碟類型或檔案系統所設計的工具進行檢查。udev 會根據核心動態提供的裝置節點名稱，維護指向裝置的永久符號連結類別：

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

21.8 udev 使用的檔案

`/sys/*`
: 由 Linux 核心提供的虛擬檔案系統，可輸出所有目前已知裝置。`udev` 用此資訊在 `/dev` 中建立裝置節點。

`/dev/*`
: 動態建立的裝置節點和使用 `systemd-tmpfiles` 建立的靜態內容；如需詳細資訊，請參閱 `systemd-tmpfiles(8)` 線上文件。

下列檔案和目錄包含了 `udev` 基礎結構的重要元件：

`/etc/udev/udev.conf`
: `udev` 主組態檔。

`/etc/udev/rules.d/*`
: 系統專屬的 `udev` 事件比對規則。可在這裡新增自訂規則，以修改或覆寫 `/usr/lib/udev/rules.d/*` 中的預設規則。
系統依英數字元順序剖析檔案。檔案中優先程度較高的規則會修改或覆寫優先程度較低的規則。數值越小，優先程度越高。

`/usr/lib/udev/rules.d/*`
: 預設的 `udev` 事件比對規則。此目錄中的檔案由套件擁有，將在更新時覆寫。請勿在這裡新增、移除或編輯檔案，應該使用 `/etc/udev/rules.d`。

`/usr/lib/udev/*`
: `udev` 規則中呼叫的協助程式。

`/usr/lib/tmpfiles.d/` 和 `/etc/tmpfiles.d/`
: 針對靜態 `/dev` 內容。

21.9 更多資訊

如需關於 `udev` 基礎結構的詳細資訊，請參閱下列 man 頁面：

`udev`
: 關於 `udev`、鍵、規則和其他重要組態問題的一般資訊。

`udevadm`

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

`udevd`

關於 `udev` 事件管理精靈的資訊。

22 使用 kGraft 即時修補 Linux 核心

本文件介紹 kGraft 即時修補技術的基本原理，並提供 SLE Live Patching 服務的使用指導方針。

kGraft 是一項即時修補技術，使用它無需停止核心，就能在執行時期修補 Linux 核心。它可以最大程度地增大系統執行時間，從而提高系統可用性，這對於關鍵任務系統而言非常重要。該項技術還允許動態修補核心，鼓勵使用者安裝關鍵的安全性更新，不必將其推遲到排程的停機時間。

kGraft 修補程式是一個核心模組，旨在取代核心中的整個函數。kGraft 主要提供核心內基礎結構，用於在執行時期將修補的代碼與基本核心代碼相整合。

SLE Live Patching 是在定期 SUSE Linux Enterprise Server 維護基礎之上提供的服務。透過 SLE Live Patching 配送的 kGraft 補充了定期 SLES 維護更新。可以使用常用的更新堆疊和程序來部署 SLE Live Patching。

本文件中提供的資訊與 AMD64/Intel 64 和 POWER 架構相關。如果您使用的不是這些架構，則相關的程序可能有所不同。

22.1 kGraft 的優勢

若要對緊急情況（已知發生了應該儘快修復的嚴重弱點，或者已知的修復程式出現嚴重的系統穩定性問題）迅速做出回應，使用 kGraft 進行即時核心修補特別有用。該技術不適合用於非時間關鍵型的已排程更新。

kGraft 的一般用例包括：配有巨量 RAM，且開機時間經常長達 15 分鐘或以上的記憶體資料庫之類的系統、需要持續數周或數月不重新啓動的大規模擬真，或者向衆多消費者持續提供服務的基礎架構建置組塊。

kGraft 的主要優勢是它永不要求停止核心，哪怕是短暫停止。

kGraft 修補程式是 RPM 套件中的一個 `.ko` 核心模組。可以在安裝或更新 套件時，使用 `insmod` 指令將該模組插入核心。kGraft 會取代核心中的整個函數，即使這些函數正在執行。如果需要，可以使用更新的 kGraft 模組取代現有修補程式。

kGraft 也很精簡 - 因為利用了其他標準 Linux 技術，它只包含少量的程式碼。

22.2 kGraft 的低層級功能

kGraft 使用 ftrace 基礎結構執行修補。下面介紹了在 AMD64/Intel 64 架構上的實作。

為了修補某個核心函數，kGraft 要求該函數的開頭有一定的空間，以便插入指向新函數的跳躍點。此空間是在開啓函數評估的情況下，於核心編譯期間由 GCC 配置的。具體而言，將在核心函數的開頭注入一個 5 位元組呼叫指令。將此類經過檢測的核心開機時，評估呼叫將由 5 位元組 NOP（無操作）指令取代。

修補開始之後，第一個位元組將由 INT3（斷點）指令取代。這可以確定 5 位元組指令取代動作的不可部分完成性。其他四個位元組將由新函數的位址取代。最後，第一個位元組將由 JMP（長跳躍）OpCode 取代。

在整個過程中，將會使用處理器間不可遮罩岔斷（IPI NMI）來衝洗系統中其他 CPU 的理論式解碼佇列。這樣，無需停止核心（哪怕是短暫性的停止），就能切換到新的函數。IPI NMI 產生的岔斷可以毫秒為單位測量，並且不被視為服務岔斷，因為無論在哪種情況下，這些岔斷都是在核心執行時發生的。

永遠不會修補呼叫者。被呼叫者的 NOP 將由指向新函數的 JMP 取代。JMP 指令會永久保留。這種運作方式可以處理好函數指標（包括結構中的指標），並且不需要儲存任何舊資料就能取消修補。

但是，這些步驟本身並不足夠妥善：因為函數的取代可能會完成一部分，核心某個部分中修復的新函數可能仍會呼叫其他位置的某個舊函數，反之亦然。如果函數介面的語意在修補程式中發生變更，將會因應性地造成混亂。

因此，在取代所有函數之前，kGraft 使用以彈性機制為基礎且類似於 RCU（讀取-複製-更新）的方案，來確定每個使用者空間線串、核心線串和核心岔斷在全域檢視中都保持一致。將在每個核心入口和出口中，為每個線串設定一個旗標。這樣，一個舊函數總是會呼叫另一個舊函數，而一個新函數總是會呼叫另一個新函數。為所有程序設定「new universe」旗標之後，修補即告完成，此時，可以移除彈性機制，程式碼可以全速執行，且不會對效能產生影響，不過，每個修補的函數需要經歷超長時間的跳轉。

22.3 安裝 kGraft 修補程式

本節介紹如何啓用 SUSE Linux Enterprise Live Patching 延伸，以及如何安裝 kGraft 修補程式。

22.3.1 啓用 SLE Live Patching

若要在您的系統上啓用 SLE Live Patching，請遵循以下步驟：

1. 如果您的 SLES 系統尚未註冊，現在請註冊。可以在安裝系統期間完成註冊，或者以後再使用 YaST 產品註冊模組（`yast2 registration`）執行註冊。註冊後，按一下是查看可用線上更新的清單。
   如果您的 SLES 系統已註冊，但 SLE Live Patching 尚未啓用，請開啓 YaST 產品註冊模組（`yast2 registration`），然後按一下選取延伸。
2. 在可用延伸清單中選取 SUSE Linux Enterprise Live Patching 12，然後按下一步。
3. 確認授權條款並按下一步。
4. 輸入 SLE Live Patching 註冊代碼並按下一步。
5. 檢查安裝摘要和所選的模式。應該選取安裝模式 `Live Patching`。
6. 按一下接受完成安裝。如此就會在您的系統上安裝 kGraft 基本元件，以及初始的即時修補程式。

22.3.2 正在更新系統

1. SLE Live Patching 更新的配送形式允許使用標準 SLE 更新堆疊來套用修補程式。可以使用 `zypper patch`、YaST 線上更新或同等的方法來更新初始即時修補程式。
2. 核心將在安裝套件的過程中自動修補。但是，只有在喚醒並取出所有休眠程序之後，才能完全消除舊核心函數的呼叫。這樣可以節省大量的時間。儘管如此，我們並不認為使用舊核心函數的休眠程序存在安全性問題。不過，在最新的 kGraft 版本中，只有當所有程序都超出了核心使用者空間界限時，才可以套用另一個 kGraft 修補程式來停止使用前一修補程式已修補的功能。
   若要查看全域修補狀態，請檢查 `/sys/kernel/kgraft/in_progress` 中的旗標。值 `1` 表示存在仍需喚醒的休眠程序（修補仍在進行中）。值 `0` 表示所有程序都只使用了修補的函數，並且修補已經完成。或者，可以使用 `kgr status` 指令取得相同的資訊。

也可以根據每個程序檢查旗標。針對每個程序分別檢查 `/proc/PROCESS_NUMBER/kgr_in_progress` 中的數字。同樣，值 `1` 表示仍需喚醒的休眠程序。或者，可以使用 `kgr blocking` 指令輸出休眠程序的清單。

22.4 修補程式生命週期

可以使用 `zypper lifecycle` 來查看線上修補程式的過期日期。確認套件 `lifecycle-data-sle-live-patching` 已安裝。

```
tux > zypper lifecycle

Product end of support
Codestream: SUSE Linux Enterprise Server 12             2024-10-31
SUSE Linux Enterprise Server 12 SP2                     n/a*

Extension end of support
SUSE Linux Enterprise Live Patching                     2017-10-31

Package end of support if different from product:
SUSEConnect                              Now, installed 0.2.41-18.1, update available
 0.2.42-19.3.1
apache2-utils                            Now


*) See https://www.suse.com/lifecycle  for latest information
```

當到了修補程式的過期日期時，將不再提供此核心版本的更多線上修補程式。請在線上修補程式生命週期期限結束之前規劃核心更新。

22.5 移除 kGraft 修補程式

若要移除 kGraft 修補程式，請使用以下程序：

1. 首先使用 Zypper 移除修補程式本身：

   ```
   zypper rm kgraft-patch-3_12_32-25-default
   ```

2. 然後重新開機。

22.6 阻塞的核心執行線串

需要準備好核心線串才能處理 kGraft。協力廠商軟體不一定能夠配合 kGraft 使用，並且其核心模組可能會衍生大量的核心執行線串。這些線串會無限期阻擋修補程序。做為應急措施，kGraft 允許強行完成修補程序，而無需等待所有執行線串跨越安全檢查點。若要實現此目的，可以在 `/sys/kernel/kgraft/in_progress` 中寫入 `0`。在執行此程序之前，請先諮詢 SUSE 支援人員。

22.7 `kgr` 工具

使用 `kgr` 工具可以簡化許多 kGraft 管理任務。可用的指令為：

`kgr status`
: 顯示 kGraft 修補的總體狀態（`ready` 或 `in_progress`）。

`kgr patches`
: 顯示已載入 kGraft 修補程式的清單。

`kgr blocking`
: 列出阻止完成 kGraft 修補的程序。預設只會列出 PID。指定 `-v` 可以列顯指令行（如果可用）。再次指定 `-v` 還會顯示堆疊追蹤。

如需詳細資訊，請參閱 `man kgr`。

22.8 kGraft 技術的應用範圍

取代函數是 kGraft 的運作基礎。資料結構的變更只能透過 kGraft 間接完成。因此，變更核心資料結構時需要特別小心，如果變更幅度太大，可能需要重新開機。此外，kGraft 無法處理使用一個編譯器來編譯舊核心，同時使用另一個編譯器來編譯修補程式的情況。

由於 kGraft 的運作方式，對衍生大量核心線串之協力廠商模組的支援有限。

22.9 SLE Live Patching 的應用範圍

SLE Live Patching 的應用範圍包括 SUSE 通用弱點評分系統 (CVSS) 層級 7 以上弱點的修復，以及與系統穩定性和資料損毀相關的錯誤修復。無法針對滿足上述所有準則的所有修復類型產生即時修補程式。如果出於技術原因而無法產生核心即時修補程式，SUSE 有權不發佈修復。如需做為 SUSE CVSS 評級基礎的 CVSS 3.0 的詳細資訊，請參閱 https://www.first.org/cvss/。

22.10 使用支援流程與我們互動

在與 SUSE 技術部門合作解決技術難題的過程中，您可能會收到一個所謂的程式暫時修復 (PTF)。我們可能會針對各種套件（包括構成 SLE Live Patching 基礎的套件）發佈 PTF。

您可以像往常一樣安裝符合前一節中所述條件的 kGraft PTF，SUSE 會確定無需將有問題的系統重新開機，並且將來的即時更新可以正常套用。

針對基礎核心發佈的 PTF 會中斷即時修補程序。首先，安裝 PTF 核心意味著需要重新開機，因為在執行時期無法取代整個核心。其次，需要再次重新開機，以便將 PTF 取代為對其發佈了即時修補程式的任何定期維護更新。

可將 SLE Live Patching 中其他套件的 PTF 視為享有正常擔保的一般 PTF。

23 特殊系統功能

本章會提供各種軟體套件、虛擬主控台及鍵盤配置的相關資訊。還會介紹 `bash`、`cron` 和 `logrotate` 等軟體元件，因為這些元件較之上一版有所變更或加強。這些元件也許不很重要，但與系統的關係密切，使用者應該變更它們的預設行為。本章最後一節則會介紹語言與國家特定的設定（I18N 與 L10N）。

23.1 特殊軟體套件的資訊

程式 `bash`、`cron`、`logrotate`、`locate`、`ulimit` 和 `free`，對系統管理員和許多使用者而言十分重要。man 頁面和 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。

23.1.1 bash 套件與 /etc/profile

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

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

在 `~/.profile` 或 `~/.bashrc` 中進行自訂設定。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。

23.1.2 cron 套件

使用 cron 可在預先定義的時間自動在背景中執行指令。cron 使用格式有特別設定的時間表，並且該工具隨附了數個預設的時間表。使用者也可以視需要指定自訂的表。

cron 表格現在位於 `/var/cron/tabs`。`/etc/crontab` 做為整個系統的 cron 表格。在時間表格之後、指令之前，輸入要直接執行指令的使用者名稱。在 範例 23.1 「/etc/crontab 中的項目」 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` man 頁面（`man cron`）。

範例 23.1 /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，然後進行修改和儲存。

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` 目錄中，由 `/usr/lib/cron/run-crons` 控制其執行。`/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。

若要在自訂的時間執行 `hourly`、`daily` 或其他定期維護程序檔，請定期使用 `/etc/crontab` 項目移除時戳檔案（請參閱範例 23.2 「/etc/crontab：移除時戳檔案」，它可以在每個整點前移除 `hourly`，在每天凌晨的 2:14 移除 `daily` 等）。

範例 23.2 /ETC/CRONTAB：移除時戳檔案

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

或者，可以將 `/etc/sysconfig/cron` 中的 `DAILY_TIME` 設定為 `cron.daily` 啓動的時間。`MAX_NOT_RUN` 的設定可確定日常任務能夠觸發並加以執行，即使使用者在長時間內均未於指定的 `DAILY_TIME` 開啓電腦。`MAX_NOT_RUN` 的最大值為 14 天。

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

23.1.3 停止 Cron 狀態訊息

若要避免 cron 狀態訊息導致的郵件泛濫，在新安裝中請將 `/etc/sysconfig/cron` 中的 `SEND_MAIL_ON_NO_ERROR` 預設值設為「no」。即使將此設定設為「no」，系統仍會將 cron 資料輸出傳送至 `MAILTO` 位址，如 cron 手冊頁所述。

對於更新，建議根據需要設定這些值。

23.1.4 記錄檔：套件 logrotate

某些系統服務（精靈）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案中。這樣，管理員可以定期檢查某個時間點的系統狀態、找出錯誤或有問題的功能，並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。如需詳細資訊，請參閱《System Analysis and Tuning Guide》，第 3 章「Analyzing and Managing System Log Files」，第 3.3 節「Managing Log Files with `logrotate`」。

23.1.5 locate 指令

可以快速尋找檔案的 `locate` 指令，未包含在軟體標準安裝的範圍內。如果需要，請安裝套件 `mlocate`，它是套件 `findutils-locate` 的後續套件。updatedb 程序會在每晚自行啟動，或啟動系統後的 15 分鐘左右啟動。

23.1.6 ulimit 指令

利用 `ulimit`（user limits）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使用的記憶體特別有用。利用它，可以避免應用程式佔用過多的系統資源，降低作業系統效能，甚至讓系統當機。

`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 23.1「`ulimit`：設定使用者的資源」 中所列的選項。

表格 23.1 `ulimit`：設定使用者的資源

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

系統範圍的預設項目在 `/etc/profile` 中設定。建議不要直接編輯此檔案，因為在系統升級期間將會覆寫變更。若要自訂系統範圍的設定檔設定，請使用 `/etc/profile.local`。每個使用者的設定應該在 `~使用者/.bashrc` 中設定。

範例 23.3 ULIMIT：~/.BASHRC 中的設定

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

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

記憶體配置的單位必須為 KB。如需詳細資訊，請參閱 `man bash`。

重要：`ulimit` 支援

並非所有的外圍程序都支援 `ulimit` 指示詞。PAM（例如 `pam_limits`）做為 `ulimit` 的替代方法，提供了全面的調整功能。

23.1.7 free 指令

`free` 指令顯示系統中總的可用記憶體、已用實體記憶體和交換空間，以及核心佔用的緩衝區和快取。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，核心不會直接瞭解有關任何應用程式或使用者資料的資訊。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，它的某些部分會寫入交換分割區或檔案中，這樣，使用 `mmap` 指令便可一開始就從這些交換分割區或檔案中讀取這些部分（請參閱 `man mmap`）。

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

不過，如果您的目標是要得知目前使用了多少 RAM，請在 `/proc/meminfo` 中尋找此資訊。

23.1.8 man 頁面和資訊頁面

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

23.1.9 使用 `man` 指令選取 man 頁面

若要讀取 man 頁面，請輸入 `man MAN_PAGE`。如果不同區段中存在同名的 man 頁面，所有的這些 man 頁面都會列出，並會顯示對應的區段號碼。請選取要顯示的那個頁面。如果您未在幾秒內輸入區段號碼，將會顯示第一個 man 頁面。

若要將此行為變更為預設系統行為，請在外圍程序啓始化檔案（如 `~/.bashrc`）中設定 `MAN_POSIXLY_CORRECT=1`。

23.1.10 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 啓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。

啓動時，Emacs 會讀取多個檔案，其中包含使用者、系統管理員以及供應商的自訂設定或預設組態設定。啓始化檔案 `~/.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 Enterprise Server，`emacs` 套件可將檔案 `site-start.el` 安裝至目錄 `/usr/share/emacs/site-lisp` 中。`site-start.el` 檔案會在啓始化檔案 `~/.emacs` 前載入。此外，`site-start.el` 會確保那些以 Emacs 附加套件來散佈的特定組態檔案皆能自動載入，例如 `psgml`。此類型的組態檔案也位於 `/usr/share/emacs/site-lisp` 中，並且會以 `suse-start-` 為開頭。本地系統管理員可在 `default.el` 中指定整個系統的設定。

有關這些檔案的詳細資訊可在 Init File 下的 Emacs 資訊檔案中取得：`info:/emacs/InitFile`。關於如何在需要時停止載入這些檔案的資訊，也可在此找到。

Emacs 的元件分成數個套件：

- `emacs` 基本套件。
- `emacs-x11`（通常已安裝）：具有 X11 支援的程式。
- `emacs-nox`：沒有 X11 支援的程式。
- `emacs-info`：info 格式的線上文件。

- `emacs-el`：以 emacs lisp 編寫的未編譯程式庫檔案。執行期間用不到這類檔案。
- 需要時可安裝多種附加產品套件：`emacs-auctex`（LaTeX）、`psgml`（SGML 與 XML）、`gnuserv`（用戶端與伺服器作業）以及其他。

23.2 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。使用 Alt—F1 到 Alt—F6 可以在虛擬主控台之間進行切換。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。

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。

23.3 鍵盤配置

若要標準化程式的鍵盤配置，請變更下列的檔案：

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

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

在 X 下，可以按照 `/etc/X11/Xmodmap` 中的說明啓用組合鍵（複合鍵）。

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

提示：更多資訊

XKB 的相關資訊，可參閱 `/usr/share/doc/packages/xkeyboard-config` 中所列的文件（`xkeyboard-config` 套件的一部分）。

23.4 語言與國家專用的設定

本系統已在很大程度上進行了國際化，可以根據當地的需求進行修改。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量。

設定位於 `/etc/sysconfig/language` 中所定義的 `LC_` 變數。這不僅是指本地語言支援，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。這些類別中的每一種都可以使用自己的變數直接定義，或使用 `language` 檔案中的主變數間接定義（請參閱 `locale` man 頁面）。

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

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

`RC_LC_ALL`

此變數（如果設定）會覆寫先前所提到的變數值。

`RC_LANG`

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

`ROOT_USES_LANG`

有 `yes` 或 `no` 兩個變數。如果設為 `no`，`root` 始終可以在 POSIX 環境中作業。

變數可以用 YaST sysconfig 編輯器來設定。這樣的變數值中包含語言碼、國碼、編碼及輔助按鍵。個別的元件會以特定的字元來連接：

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

23.4.1 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 ISO 3166 中，請參閱 http://en.wikipedia.org/wiki/ISO_3166。

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

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

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

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

`LANG=en_IE@euro`
: 上方範例在語言設定中明確包括歐元符號。此項設定現已過時，因為 UTF-8 也涵蓋歐元符號。它只有在應用程式支援 ISO-8859-15 而不支援 UTF-8 時才有用。

對 `/etc/sysconfig/language` 所做的變更會透過以下程序鏈來啓用：

- 對於 Bash：`/etc/profile` 會讀取 `/etc/profile.d/lang.sh`，後者會分析 `/etc/sysconfig/language`。
- 對於 tcsh：在登入時，`/etc/csh.login` 會讀取 `/etc/profile.d/lang.csh`，後者會分析 `/etc/sysconfig/language`。

如此可以確保對 `/etc/sysconfig/language` 的任何變更在下次登入相應的外圍程序時即會生效，而不必手動將其啓用。

使用者可以適當地編輯自己的 `~/.bashrc` 來覆寫系統預設值。例如，如果不想對程式訊息使用系統範圍的 `en_US`，請加入 `LC_MESSAGES=es_ES`，這樣訊息將以西班牙語顯示。

23.4.2 ~/.i18n 中的地區設定

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

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

23.4.3 語言支援的設定

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

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

`LANGUAGE="br_FR:fr_FR"`

`LANGUAGE="gl_ES:es_ES:pt_PT"`

如有需要，請改用挪威文變體 Nynorsk 與 Bokmål（讓其他備用為 `no`）：

`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。

23.4.4 更多資訊

- The GNU C Library Reference Manual 的「Locales and Internationalization」一章。包含在 glibc-info 中。該套件可從 SUSE Linux Enterprise SDK 中取得。SDK 是適用於 SUSE Linux Enterprise 的模組，可以從 SUSE Customer Center 的線上通道取得，或者，轉到 http://download.suse.com/，然後搜尋並下載 SUSE Linux Enterprise Software Development Kit（SUSE Linux 軟體開發套件）。如需詳細資訊，請參閱《部署指南》，第 14 章「安裝模組、延伸和協力廠商附加產品」。
- 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。
- Bruno Haible 所寫的 Unicode-HOWTO，可從 http://tldp.org/HOWTO/Unicode-HOWTO-1.html 取得。

IV 服務

24 使用 NTP 進行時間同步化

NTP（網路時間協定）機制是一種協定，用於同步化網路上的系統時間。首先，機器可以從提供可靠時間來源的伺服器取得時間。其次，機器本身在網路中可以做為其他電腦的時間來源。這個目標是雙重的 — 即維護絕對正確的時間，並同步化網路內所有機器的系統時間。

維護精準的系統時間對於許多情況都非重要。內建的硬體時鐘通常無法符合資料庫或叢集等應用程式的要求。手動校正系統時間有可能會造成嚴重的問題，因為，例如時間倒退將可能造成重要應用程式無法正常運作。在網路中，通常需要同步所有機器中的系統時間，而手動調整時間的做法並不可取。NTP 提供了一種用於解決這些問題的機制。NTP 服務會依據網路中可靠的時間伺服器持續調整系統時間。它可以進一步管理本地參考的時鐘，例如收音機控制的時鐘。

24.1 使用 YaST 設定 NTP 用戶端

`ntp` 套件隨附的 NTP 精靈（`ntpd`）預先設定為使用本地電腦時鐘做為時間參考。不過，硬體時鐘僅供在沒有更精確的時間來源時備用。YaST 簡化了 NTP 用戶端的組態。

24.1.1 基本組態

YaST NTP 用戶端組態（網路服務 › NTP 組態）包含數個索引標籤。在一般設定索引標籤上設定 `ntpd` 的啓動模式和要查詢的伺服器。

僅手動
: 如果要手動啓動 `ntpd` 精靈，請選取僅手動。

同步但不啓動精靈
: 選取同步但不啓動精靈會定期設定系統時間，而不永久執行 `ntpd`。您可以設定同步時間間隔(分鐘)。

現在和開機時
: 選取現在和開機時，以便在系統開機時自動啓動 `ntpd`。建議您使用此設定。

24.1.2 變更基本組態

用戶端要查詢的伺服器以及其他時間來源會列在一般設定索引標籤的下半部。修改清單時，可依需要使用新增、編輯以及刪除。顯示記錄可用來檢視用戶端的記錄檔。

按一下新增以新增時間資訊的新來源。在下列對話方塊中，選取進行時間同步化的來源類型。可用的選項如下：

圖形 24.1 YAST：NTP 伺服器

伺服器

在選取下拉式清單（請參閱圖形 24.1 「YaST：NTP 伺服器」）中，指定是使用區域網路中的時間伺服器（本地 NTP 伺服器），還是使用可以處理您時區的網際網路時間伺服器（公用 NTP 伺服器）來設定時間同步。若需本地時間伺服器，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。選項可讓您為 `ntpd` 指定其他選項。

使用存取控制選項，您可以限制遠端電腦可以使用電腦上執行的精靈來執行的動作。當核取安全性設定索引標籤（請參閱圖形 24.2 「進階 NTP 組態：安全性設定」）中的將 NTP 服務限制為僅限已設定的伺服器後，此欄位才處於啓用狀態。

這些選項對應於 `/etc/ntp.conf` 中的 `restrict` 子句。例如，`nomodify notrap noquery` 不允許伺服器修改電腦的 NTP 設定，並且不允許使用 NTP 精靈的設陷裝置（遠端事件登入功能）。建議對超出您控制範圍（例如在網際網路上）的伺服器使用這些限制。

如需詳細資訊，請參閱 `/usr/share/doc/packages/ntp-doc`（`ntp-doc` 套件的一部份）。

點

點（peer），是指一台建立了對稱關係的機器：它可同時做為時間伺服器與用戶端。若要在相同的網路中使用點而非伺服器，請輸入系統的位址。其餘的對話方塊與伺服器對話方塊相同。

收音機時鐘

若要在系統中使用收音機時鐘來進行時間同步化，請在此對話方塊中輸入時鐘類型、單位編號、裝置名稱以及其他選項。按一下驅動程式校正，即可微調驅動程式。`/usr/share/doc/packages/ntp-doc/refclock.html` 中提供了關於本地無線電時鐘作業的詳細資訊。

外寄廣播

時間資訊與查詢也可以透過網路中的廣播傳輸。請在此對話方塊中輸入廣播所應傳送至的位址。除非您已經有類似收音機控制時鐘的可靠時間來源，否則請勿啓用廣播。

內送廣播

如果您想要讓用戶端透過廣播接收其資訊，請在這些欄位中輸入應該接受的個別封包位址。

圖形 24.2 進階 NTP 組態：安全性設定

在安全性設定索引標籤（請參閱圖形 24.2 「進階 NTP 組態：安全性設定」）中，指定 `ntpd` 是否應於 chroot jail 中啓動。預設不會啓動在 Chroot Jail 中執行 NTP 精靈。由於 Chroot Jail 選項可以防止攻擊者損毀整個系統，因此在 `ntpd` 遭受攻擊時，會有較高的安全性。

將 NTP 服務限制為僅限已設定的伺服器可增加系統安全性，方法是禁止遠端電腦檢視並修改電腦的 NTP 設定，並且禁止使用用於遠端事件登入的設陷裝置。啓用此選項後，這些限制會套用至所有遠端電腦，除非您覆寫一般設定索引標籤的時間來源清單中個別電腦的存取控制選項。對於所有其他遠端電腦，僅允許查詢本地時間。

如果 SuSEfirewall2 在作用中（預設情況下），請啓用在防火牆中開啓埠。如果您讓連接埠保持為關閉，就不可能對時間伺服器建立連接。

24.2 手動設定網路中的 NTP

在網路上使用時間伺服器的最簡單方式就是設定伺服器參數。例如，如果可以從網路存取名為 `ntp.example.com` 的時間伺服器，那麼，請新增以下行，將此伺服器的名稱新增到 `/etc/ntp.conf` 檔案：

```
server ntp.example.com
```

若要新增更多時間伺服器，請以關鍵字 伺服器 插入其他行。在使用 `systemctl start ntp` 指令啓始化 `ntpd` 後，大約需要一個小時來穩定時間以及建立漂移檔案以校正本地電腦時鐘。使用漂移檔案，可以在電腦開機時計算硬體時鐘的系統錯誤。它會立即使用校正，使系統時間具有更高的穩定性。

有兩種方法可以將 NTP 機制做為用戶端：首先，用戶端可固定在每段間隔時間後向已知伺服器查詢時間。隨著用戶端的增加，此方法可能造成伺服器的高負載。其次，用戶端可以等待網路中的廣播時間伺服器所送出的 NTP 廣播。此方法具有伺服器品質未知的缺點，而且伺服器送出錯誤的資訊可能造成嚴重的問題。

如果時間是經由廣播取得，就不需要伺服器名稱。在這樣的情形下，請在 `/etc/ntp.conf` 組態檔中輸入 `broadcastclient`。若要完全使用一或多個已知的時間伺服器，請輸入以 `servers` 開頭的名稱。

24.3 執行時期的動態時間同步

若系統開機後並無網路連接，`ntpd` 仍會啓動，但無法解析組態檔案中設定之時間伺服器的 DNS 名稱。在加密的 Wi-Fi 中使用 NetworkManager 時，可能會發生這種情況。

如果您希望 `ntpd` 在執行時期解析 DNS 名稱，必須設定 `dynamic` 選項。在開機後建立網路連接時，`ntpd` 會再次查詢名稱，並可以存取時間伺服器來取得時間資訊。

手動編輯 `/etc/ntp.conf`，並將 `dynamic` 新增至一或多個 `server` 項目：

```
server ntp.example.com dynamic
```

您也可以使用 YaST 並按以下步驟操作：

1. 在 YaST 中，按一下網路服務 › NTP 組態。
2. 選取要設定的伺服器。然後按一下編輯。
3. 啓動選項欄位，並新增 `dynamic`。如果已經輸入了其他選項，請使用空格分隔。
4. 按一下確定關閉編輯對話方塊。重複上述步驟以變更所有要變更的伺服器。
5. 最後按一下確定儲存設定。

24.4 設定本地參考時鐘

軟體套件 `ntpd` 包含與本機參照時鐘連接的驅動程式。`ntp-doc` 套件的 `/usr/share/doc/packages/ntp-doc/refclock.html` 檔案中提供了受支援時鐘的清單。每個驅動程式都與數字關聯。在 NTP 中，實際組態工作是透過虛擬 IP 位址來執行。把時鐘當成在網路中一樣，將它輸入 `/etc/ntp.conf` 檔案中。為此專門為它們指定了 `127.127.T.U` 格式的特殊 IP 位址。其中，`T` 代表時鐘的類型並可決定使用哪一個驅動程式，而 `U` 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。檔案 `/usr/share/doc/packages/ntp-doc/drivers/driverNN.html`（其中的 `NN` 為驅動程式的編號）提供了關於特定類型之時鐘的資訊。例如，「type 8」 時鐘（透過序列介面的收音機時鐘）需要其他可以更精確地指定時鐘的模式。例如，Conrad DCF77 接收器模組具有模式 5。若要使用此時鐘做為偏好的參考，請指定 `prefer` 關鍵字。Conrad DCF77 接收器模組的完整 `server` 行如下所示：

```
server 127.127.8.0 mode 5 prefer
```

其他的時鐘也使用相同的模式。安裝 `ntp-doc` 套件後，便可在 `/usr/share/doc/packages/ntp-doc` 目錄中找到 NTP 的文件。檔案 `/usr/share/doc/packages/ntp-doc/refclock.html` 提供了指向描述驅動程式參數之驅動程式頁面的連結。

24.5 將時鐘與外部時間參考（ETR）同步

支援將時鐘與外部時間參考（ETR）同步。外部時間參考每 2**20（2 的 20 次方）微秒傳送一次振盪器訊號與同步訊號，以將所有連結伺服器的 TOD 時鐘保持同步。

為了便利，兩個 ETR 裝置可連接到一個機器。如果時鐘偏差超出同步檢查容錯，則所有 CPU 都會收到一個機器記號，指出時鐘不同步。如果出現這種狀況，則所有啓用了 DASD I/O 到 XRC 的裝置就會停止運作，直到時鐘重新同步。

ETR 支援透過兩個 `sysfs` 屬性啓動；請以 `root` 身分執行下列指令：

```
echo 1 > /sys/devices/system/etr/etr0/online
echo 1 > /sys/devices/system/etr/etr1/online
```

25 網域名稱系統

必須使用 DNS（網域名稱系統）將網域和主機名稱解析為 IP 位址。例如，藉由這種方式，會給主機名稱 `jupiter` 指定 IP 位址 192.168.2.100。在設定您自己的名稱伺服器前，請參閱第 16.3 節 「名稱解析」 中有關 DNS 的一般資訊。以下組態範例使用預設 DNS 伺服器 – BIND。

25.1 DNS 詞彙

區域
: 網域名稱空間細分成一個個區域。例如，如果您擁有 `example.com`，您就擁有 `com` 網域的 `example` 區段（或區域）。

DNS 伺服器
: DNS 伺服器是為網域維護名稱和 IP 資訊的伺服器。您可以擁有用於主要區域的主要 DNS 伺服器、用於從屬區域的次要伺服器、或沒有任何區域處理快取功能的從屬伺服器。

  主要區域 DNS 伺服器
  : 主要區域包含您網路上的所有主機，而且 DNS 伺服器主要區域會儲存您的網域中所有主機最新的記錄。

  從屬區域 DNS 伺服器
  : 從屬區域是主要區域的副本。從屬區域 DNS 伺服器會用區域傳輸作業從其主伺服器取得區域資料。只要從屬區域 DNS 伺服器擁有有效（未過期）的區域資料，它就有權代表區域回應。如果從屬無法取得區域資料的新副本，它就會停止代表區域回應。

轉遞者
: 轉遞者是當您的 DNS 伺服器無法答覆查詢時，它應該將其傳送至的目標 DNS 伺服器。為了在同一個組態中啓用不同的組態來源，系統使用了 `netconfig`（並請參閱 `man 8 netconfig`）。

記錄

記錄是有關名稱和 IP 位址的資訊。有關支援的記錄及其語法，請參閱 BIND 文件中的說明。一些特殊的記錄如下：

NS 記錄

NS 記錄會告訴名稱伺服器哪些機器負責管理特定網域區域。

MX 記錄

MX（郵件交換）記錄會說明在網際網路上傳送郵件時要聯絡的機器。

SOA 記錄

SOA（起始授權）記錄是區域檔案中的第一筆記錄。SOA 記錄是在使用 DNS 同步化多部電腦之間的資料時使用。

25.2 安裝

若要安裝 DNS 伺服器，請啓動 YaST 並選取軟體 › 軟體管理。選擇檢視 › 模式，然後選取DHCP 和 DNS 伺服器。請確認安裝個別套件，以完成此安裝程序。

或者，在指令行中使用以下指令：

```
zypper in -t pattern dhcp_dns_server
```

25.3 利用 YaST 進行組態

使用 YaST DNS 模組可以設定區域網路的 DNS 伺服器。當您第一次啓動模組時，一個精靈會啓動，提示您指定有關伺服器管理的一些設定。完成此初步設定，即會產生基本的伺服器組態。使用進階模式可以處理更進階的組態任務，如設定 ACL、記錄、TSIG 金鑰和其他選項。

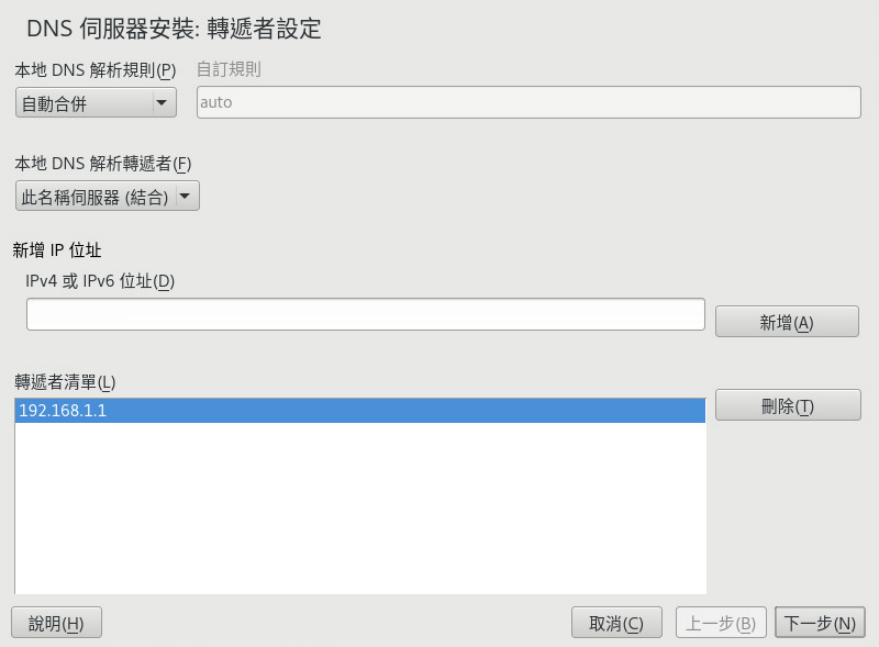
25.3.1 精靈組態

精靈包含三個步驟或對話方塊。在對話方塊的適當位置，您可以進入進階組態模式。

1. 第一次啓動模組時，會開啓圖形 25.1 「DNS 伺服器安裝：轉遞者設定」中所示的轉遞者設定對話方塊。本地 DNS 解析規則允許設定以下選項：

- 已停用合併轉遞者
- 自動合併
- 已啓用合併轉遞者
- 自訂組態 — 如果已選取自訂組態，則可以指定自訂規則；依預設，在已選取自動合併的情況下，自訂規則將設定為「`自動`」，但是，您可以在此處設定介面名稱，或者從 `STATIC` 與 `STATIC_FALLBACK` 這兩個特殊規則名稱中做出選擇。

在本地 DNS 解析轉遞者中，指定要使用的服務：使用系統名稱伺服器、此名稱伺服器 (bind) 或本地 dnsmasq 伺服器。
如需有關所有這些設定的詳細資訊，請參閱 `man 8 netconfig`。



圖形 25.1 DNS 伺服器安裝：轉遞者設定

轉遞者是指當您的 DNS 伺服器無法答覆查詢時，將查詢傳送至的目標 DNS 伺服器。請輸入它們的 IP 位址，然後按一下新增。

2. DNS 區域對話方塊包含數個部分，負責管理區域檔案，如第 25.6 節 「區域檔案」所述。對於新區域，請在名稱中提供其名稱。若要新增反向區域，名稱的結尾必須是 `.in-addr.arpa`。最後，選取類型(主要、從屬或轉遞)。請參閱圖形 25.2 「DNS 伺服器安裝：DNS 區域」。按一下編輯可設定現有區域的其他設定值。若要移除區域，按一下刪除。

圖形 25.2 DNS 伺服器安裝：DNS 區域

3. 在最後的對話方塊中，可按一下在防火牆中開啓埠，在防火牆中開啓 DNS 連接埠。然後決定是否在開機時啓動 DNS 伺服器（開啓或關閉）。您亦可啓用 LDAP 支援。請參閱圖形 25.3 「DNS 伺服器安裝：完成精靈」。

圖形 25.3 DNS 伺服器安裝：完成精靈

25.3.2 進階組態

啓動模組後，YaST 會開啓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：

25.3.2.1 啓動

在啓動下，定義是應在系統開機時啓動 DNS 伺服器還是手動啓動。若要立即啓動 DNS 伺服器，請按一下立即啓動 DNS 伺服器。若要停止 DNS 伺服器，請按一下立即停止 DNS 伺服器。若要儲存目前的設定，請選取立即儲存設定並重新載入 DNS 伺服器。您可以使用在防火牆中開啓埠開啓防火牆中的 DNS 埠，並使用防火牆詳細資訊修改防火牆設定。

若選取LDAP 主動支援，區域檔案將由 LDAP 資料庫來管理。DNS 伺服器重新啓動或由系統提示重新載入其組態時，會選用寫入到 LDAP 資料庫的任何區域資料變更。

25.3.2.2 轉遞者

如果您的本地 DNS 伺服器無法答覆要求，它會嘗試將要求轉遞至轉遞者(若做此設定)。可手動將此轉遞者新增至轉遞者清單。如果轉遞者與撥號連接同樣不是靜態的，則 netconfig 會處理組態。如需 netconfig 的詳細資訊，請參閱 `man 8 netconfig`。

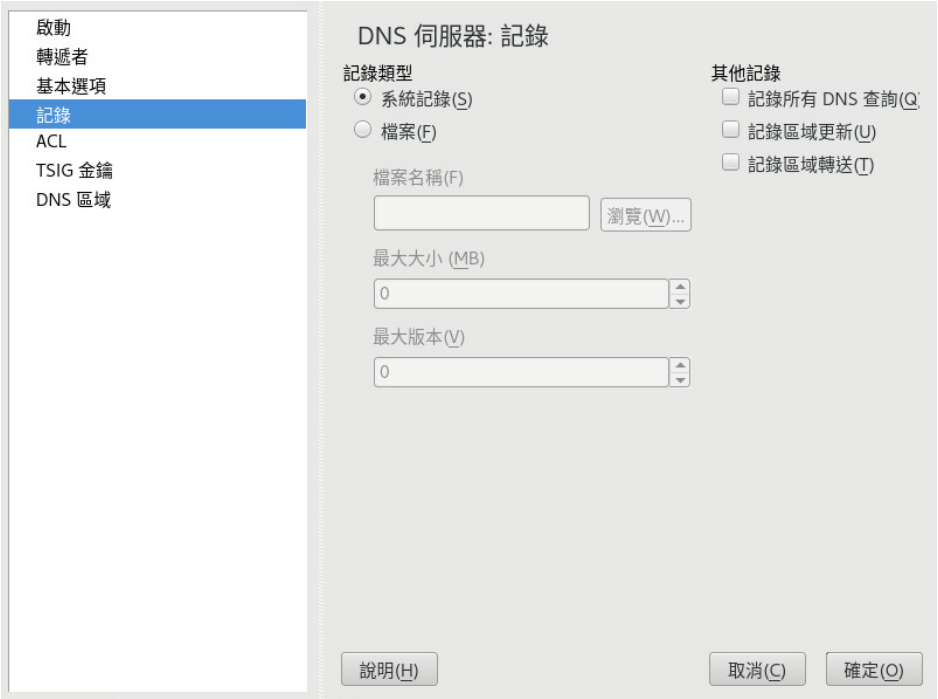
25.3.2.3 基本選項

在此區段中，可設定基本伺服器選項。從選項功能表中，選取所需項目，然後在相應的文字方塊中指定值。選取 新增 以包含新項目。

25.3.2.4 記錄

若要設定 DNS 伺服器應記錄的內容和記錄方式，請選取記錄。在記錄類型下，指定 DNS 伺服器應該寫入記錄資料的位置。選取系統記錄來使用全系統記錄，或選取檔案指定不同的檔案。如果使用後一種方式，還需另行指定名稱、最大檔案大小（以 MB 為單位）以及要儲存的記錄檔案版本數。

進一步選項可從其他記錄下存取。啓用記錄所有 DNS 查詢會記錄每個查詢，此選項會讓記錄檔變得非常大。所以，除了偵錯用途外，啓用此選項並不是理想的作法。若要記錄 DHCP 與 DNS 伺服器之間在區域更新期間的資料流量，請啓用記錄區域更新。若要記錄從主伺服器到從屬伺服器在區域傳輸期間的資料流量，請啓用記錄區域轉送。請參閱圖形 25.4 「DNS 伺服器：記錄」。



圖形 25.4 DNS 伺服器：記錄

25.3.2.5 ACL

使用此對話方塊可定義 ACL（存取控制清單）以強制執行存取限制。在名稱下提供獨特名稱後，在值下指定 IP 位址（有或沒有網路遮罩），格式如下：

```
{ 192.168.1/24; }
```

組態檔的語法要求位址以分號結尾，而且放置在大括號之間。

25.3.2.6 TSIG 金鑰

TSIG（交易簽章）的主要目的是保護 DHCP 與 DNS 伺服器之間的通訊。在第 25.8 節「安全交易」中有所描述。

若要產生 TSIG 金鑰，請在標籤為金鑰 ID的欄位中輸入特別的名稱，並指定用來儲存金鑰的檔案（檔案名稱）。按一下產生確認您的選擇。

若要使用之前建立的金鑰，請將金鑰 ID欄位保留空白，並在檔案名稱下選取儲存金鑰的檔案。接著，以 新增 確認您的選項。

25.3.2.7 DNS 區域（新增從屬區域）

若要新增從屬區域，請選取DNS 區域，選擇從屬區域類型，指定新區域的名稱，並按一下新增。

在區域編輯器子對話方塊中，於主 DNS 伺服器 IP下指定從屬伺服器應從中提取資料的主伺服器。若要限制對伺服器的存取，可以從清單選取其中一個 ACL。

25.3.2.8 DNS 區域（新增主要區域）

若要新增主要區域，請選取DNS 區域，選擇主要區域類型，寫入新區域的名稱，然後按一下新增。新增主要區域時，也需要新增反向區域。例如，如果新增指向子網路 `192.168.1.0/24` 中主機的 `example.com` 區域，則也應該新增涵蓋該 IP 位址範圍的反向區域。根據定義，該區域應該命名為 `1.168.192.in-addr.arpa`。

25.3.2.9 DNS 區域（編輯主要區域）

若要編輯主要區域，請選取DNS 區域，然後從表中選取主要區域，並按一下編輯。對話方塊由幾個頁面組成：基本（第一個開啓的頁面）、NS 記錄、MX 記錄、SOA 以及記錄。

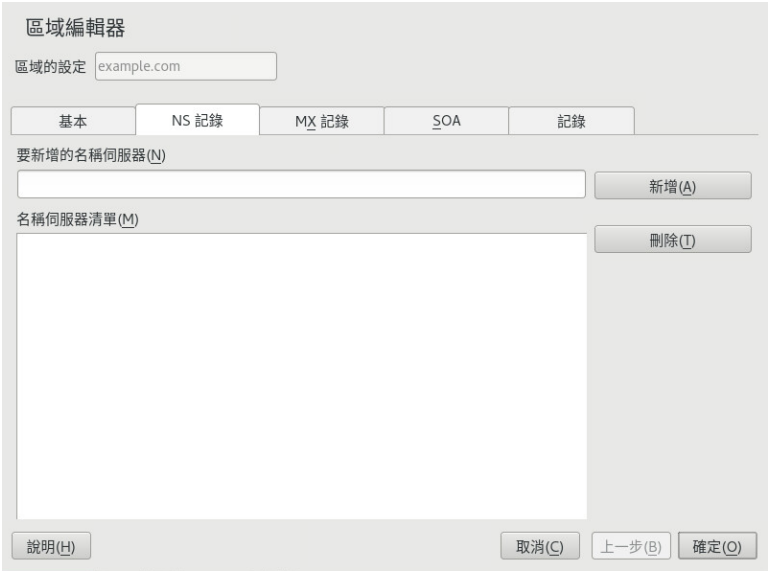
基本對話方塊（如圖形 25.5 「DNS 伺服器：區域編輯器（基本）」所示），可讓您定義動態 DNS 的設定以及到用戶端及從屬名稱伺服器之區域傳輸的存取選項。若要允許動態更新區域，請選取允許動態更新以及對應的 TSIG 金鑰。更新動作開始前，必須先定義金鑰。若要啓用區域傳輸，請選取對應的 ACL。必須先行定義 ACL。

在基本對話方塊中，選擇是否要啓用區域傳輸。使用列出的 ACL 來定義可以下載區域的人員。

圖形 25.5 DNS 伺服器：區域編輯器（基本）

區域編輯器（NS 記錄）

NS 記錄對話方塊可讓您為指定的區域定義替代名稱伺服器。請確定您自己的名稱伺服器包含於清單中。若要新增記錄，請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 25.6 「DNS 伺服器：區域編輯器（NS 記錄）」。



圖形 25.6 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，請輸入對應的位址及優先順序值。完成後，選取新增確認該動作。請參閱圖形 25.7 「DNS 伺服器：區域編輯器（MX 記錄）」。

圖形 25.7 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（起始授權）記錄。如需個別選項的說明，請參閱範例 25.6 「/var/lib/named/example.com.zone 檔案」。透過 LDAP 管理的動態區域，並不支援變更 SOA 記錄。

圖形 25.8 DNS 伺服器：區域編輯器（SOA）

區域編輯器（記錄）

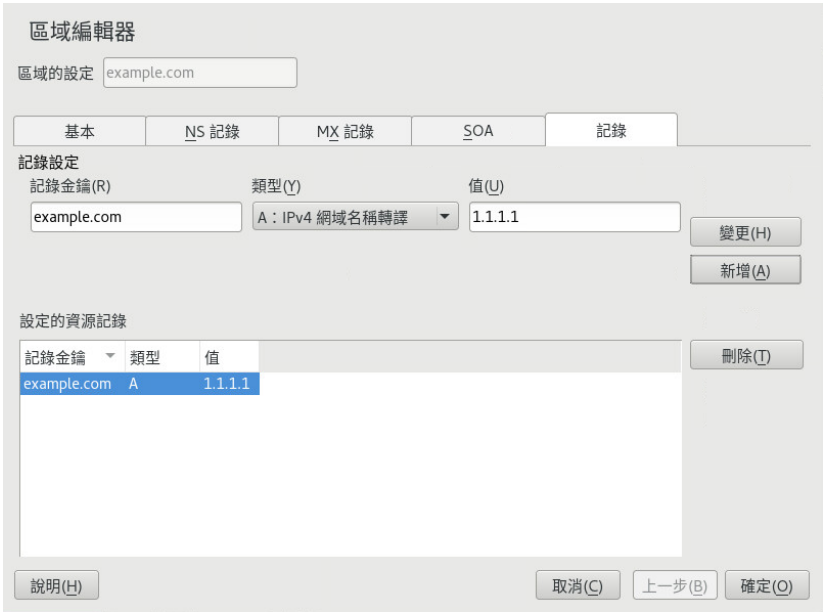
此對話方塊可管理名稱解析。在記錄金鑰中，輸入主機名稱，然後選取其類型。A類型表示主項目。此項目的值應為 IP 位址（IPv4）。請為 IPv6 位址使用AAAA。CNAME 是別名。使用 NS 與 MX 類型，可取得NS 記錄與MX 記錄標籤提供之資訊的詳細或部分擴充記錄。這三個類型都可以解析成現有的 A 記錄。PTR 是供反向區域所使用。它的內容與 A 記錄相反，例如：

```
hostname.example.com. IN A 192.168.0.1
1.0.168.192.in-addr.arpa IN PTR hostname.example.com.
```

25.3.2.9.1 新增反向區域

若要新增反向區域，請遵循以下程序：

1. 啓動 YaST › DNS 伺服器 › DNS 區域。
2. 如果您尚未新增主要正向區域，現在請新增並對其進行編輯。
3. 在記錄索引標籤中，填入相應的記錄金鑰和值，然後使用新增來新增記錄，並使用確定來確認。如果 YaST 告知某個名稱伺服器的記錄不存在，請在 NS 記錄索引標籤中新增該記錄。



圖形 25.9 為主要區域新增記錄

4. 回到 DNS 區域視窗，新增一個主要反向區域。

圖形 25.10 新增反向區域

5. 編輯該反向區域，然後在記錄索引標籤中，您可以看到 PTR：反向轉換記錄類型。新增相應的記錄金鑰和值，然後按一下新增並使用確定來確認。

圖形 25.11 新增反向記錄

視需要新增名稱伺服器記錄。

提示：編輯反向區域

新增正向區域後，返回到主功能表，然後選取要編輯的反向區域。在基本索引標籤中，啓用自動產生以下區域的記錄核取方塊，然後選取正向區域。這樣，反向區域中會自動更新正向區域的所有變更。

25.4 啓動 BIND 名稱伺服器

在 SUSE® Linux Enterprise Server 系統上，已預先設定名稱伺服器 BIND (Berkeley Internet Name Domain，柏克萊網際網路名稱網域)，因此在安裝後可以立即啓動此名稱伺服器，不會出現任何問題。一般而言，如果您能連接網際網路，並在 `/etc/resolv.conf` 中輸入了 `127.0.0.1` 做為 `localhost` 的名稱伺服器位址，則表示您已經有可以運作的名稱解析功能，因而無需知道提供者的 DNS。BIND 透過根名稱伺服器執行名稱解析，顯見處理程序較慢。一般而言，應該在 `forwarders` 下的組態檔 `/etc/named.conf` 中輸入提供者的 DNS 及其 IP 位址，以確保有效及安全的名稱解析。如果

目前此辦法可行，名稱伺服器會當成純粹的「僅快取」名稱伺服器執行。只有在您設定了名稱伺服器自己的區域後，它才會成為真正的 DNS。`/usr/share/doc/packages/bind/config` 中提供了一個簡單範例。

提示：自動使用名稱伺服器資訊

在某些類型的網際網路連接或網路連接下，名稱伺服器資訊可自動根據目前的情況進行調整。若要實現此目的，請將 `/etc/sysconfig/network/config` 檔案中的 `NETCONFIG_DNS_POLICY` 變數設定為 `auto`。

不過，在相關機構為您指派正式的網域之前，切勿設定網域。即使您有自己的網域而且是由提供者管理，最好也不要使用，否則 BIND 不會轉遞此網域的要求。例如，此網域將無法存取提供者的網頁伺服器。

若要啓動名稱伺服器，請以 `root` 身分輸入指令 `systemctl start named`。使用 `systemctl status named` 來檢查 named（在呼叫名稱伺服器程序時）是否已成功啓動。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` 做為預設伺服器，位址為 `127.0.0.1`。如果沒有傳回所需的結果，`/etc/resolv.conf` 可能包含不正確的名稱伺服器項目，或是檔案不存在。如果是第一次測試，請輸入 `host 127.0.0.1`，這通常都能成功。如果看到錯誤訊息，請使用 `systemctl status named` 檢查伺服器是否真的在執行。如果該名稱伺服器未啓動或者出現非預期的行為，請檢查 `journalctl -e` 的輸出。

若要使用提供者的名稱伺服器或網路上正在執行的名稱伺服器做為轉遞者，請在 `forwarders` 下的 `options` 區段中輸入對應的 IP 位址。範例 25.1 「named.conf 中的轉寄選項」中包含的位址只是範例。請根據您自己的設定調整這些項目。

範例 25.1 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; 192.168.1.116; };
        allow-query { 127/8; 192.168/16 };
        notify no;
};
```

`options` 項目後面跟著區域的項目：`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。對應的檔案無需修改，而且應該依其原狀運作。另外也請確定每個項目的末尾都有「;」，且大括號在正確的位置。變更

組態檔 `/etc/named.conf` 或區域檔後，可使用 `systemctl reload named` 告知 BIND 重新讀取這些檔案。使用 `systemctl restart named` 停止並重新啓動名稱伺服器可達成相同的效果。任何時候都可透過輸入 `systemctl stop named` 來停止伺服器。

25.5 /etc/named.conf 組態檔案

BIND 名稱伺服器本身的所有設定都儲存於檔案 `/etc/named.conf` 中。不過，要處理之網域的區域資料（包括主機名稱、IP 位址等）儲存於 `/var/lib/named` 目錄中各自的檔案中。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，另一個則是由個別網域的 `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 25.2 「基本的 /etc/named.conf」中顯示了最基本的 `/etc/named.conf`。

範例 25.2 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";
};
```

25.5.1 重要組態選項

`directory "FILENAME";`

指定 BIND 可以在其中找尋包含區域資料之檔案的目錄。通常是 `/var/lib/named`。

`forwarders { IP-ADDRESS; };`

指定無法直接解析 DNS 要求時應該將其轉遞至哪個名稱伺服器（一般屬於提供者）。以 IP 位址（例如 `192.168.1.116`）取代 `IP-ADDRESS`。

`forward first;`

將會在嘗試透過根名稱伺服器解析 DNS 要求之前先加以轉遞。除了 `forward first`，可以寫入 `forward only` 以轉遞所有要求，且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` 將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 哪個連接埠應該監聽 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 DNS 要求外送，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。以位址資訊（例如 `192.168.2.0/24`）取代 `NET`。尾部的 `/24` 是網路遮罩的縮寫表示式，在此例中為 `255.255.255.0`。

`allow-transfer ! *;;`

控制哪些主機可以要求區域傳輸。在範例中，這類要求是使用 `! *`. 如果沒有這個項目，就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 每小時都會在系統的日誌中產生數行統計資訊。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;
: 此選項定義 BIND 清除其快取記憶體的時間間隔。每次清除時觸發系統日誌中一個項目。時間規格單位為分鐘。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 僅會監聽啓動時偵測到的介面。如果不想出現這種情況，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啓動名稱伺服器時，`no` 會防止通知其他名稱伺服器。

如需可用選項的清單，請參閱 `man 5 named.conf` 的手冊頁。

25.5.2 記錄

記錄的內容、方式及位置皆可在 BIND 中詳細設定。一般而言，預設設定應該足夠。範例 25.3 「關閉記錄的項目」 顯示了這類項目的最簡單格式，而且完全停用了記錄。

範例 25.3 關閉記錄的項目

```
logging {
        category default { null; };
};
```

25.5.3 區域項目

範例 25.4 EXAMPLE.COM 的區域項目

```
zone "example.com" in {
      type master;
      file "example.com.zone";
      notify no;
```

```
};
```

在 `zone` 之後，指定要管理的網域名稱（`example.com`），後面跟上 `in` 以及大括號括住的相關選項區塊，如範例 25.4 「example.com 的區域項目」 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 25.5 「example.net 的區域項目」 中所示。

範例 25.5 EXAMPLE.NET 的區域項目

```
zone "example.net" in {
      type slave;
      file "slave/example.net.zone";
      masters { 10.0.0.1; };
};
```

區域選項：

type master;
: 藉由指定 `master`，可以告訴 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;
: 此區域傳輸自另一部名稱伺服器。必須與 `masters` 一起使用。

type hint;
: 區域 `.` 屬於 `hint` 類型，可用來設定根名稱伺服器。此區域定義可以維持原狀。

檔案 `example.com.zone` 或檔案 「slave/example.net.zone」；
: 此項目可指定網域之區域資料所在的檔案。從屬區域不需要此檔案，因為此資料是從另一個名稱伺服器提取。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。如果沒有此項目，將禁止區域更新。以下項目會產生相同的結果，因為 `! *` 可有效地禁止任何這類活動。

25.6 區域檔案

需要兩種類型的區域檔案。一個會給主機名稱指定 IP 位址，另一個的作用恰恰相反：為 IP 位址提供主機名稱。

提示：在區域檔案中使用點號（英文句號）

`.` 在區域檔案中具有重要意義。如果指定的主機名稱末尾沒有句點（`.`），則會附加區域。使用完整網域名稱指定的完整主機名稱必須以句點結尾（`.`），這是為了避免再次向它新增網域。缺少 `.` 或其位置錯誤往往是造成名稱伺服器組態錯誤的原因所在。

第一個要考慮的情況是負責網域 `example.com` 的區域檔案 `example.com.zone`，如範例 25.6 「/var/lib/named/example.com.zone 檔案」中所示。

範例 25.6 /VAR/LIB/NAMED/EXAMPLE.COM.ZONE 檔案

```
1.  $TTL 2D
2.  example.com. IN SOA      dns  root.example.com. (
3.               2003072441  ; serial
4.               1D          ; refresh
5.               2H          ; retry
6.               1W          ; expiry
7.               2D )        ; minimum
8.
9.               IN NS       dns
10.              IN MX       10 mail
11.
12. gate         IN A        192.168.5.1
13.              IN A        10.0.0.1
14. dns          IN A        192.168.1.116
15. mail         IN A        192.168.3.108
16. jupiter      IN A        192.168.2.100
17. venus        IN A        192.168.2.101
18. saturn       IN A        192.168.2.102
19. mercury      IN A        192.168.2.103
20. ntp          IN CNAME    dns
21. dns6         IN A6  0    2002:c0a8:174::
```

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（起始授權）控制記錄開始的地方：

- 最前面的 `example.com` 為要管理的網域名稱。名稱以 `.` 結尾，否則將再次附加區域。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，做為此區域的主伺服器。該名稱會從 `dns` 擴充為 `dns.example.com`，因為它沒有以 `.` 結尾。
- 後面跟著此名稱伺服器之負責人的電子郵件地址。因為 `@` 符號本身具有特殊意義，所以在此輸入 `.` 來代替。對於 `root@example.com`，項目必須讀做 `root.example.com.`。`.` 必須加在最後，以防止新增區域。
- `(` 會將 `)` 前面的所有行都包含在 SOA 記錄中。

行 3：

`serial number` 是任意號碼，每次此檔案變更時就會增加。通知次要名稱伺服器（從屬伺服器）發生變更，這是必要的。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` 指定在發生錯誤時次要名稱伺服器嘗試再次聯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無法重新取得與主要伺服器的聯絡時，在此時間範圍後丟棄快取資料。此例中為一週。

行 7：

SOA 記錄中的最後一個項目指定 `negative caching TTL`，亦即其他伺服器未解析 DNS 查詢之結果可以快取的時間。

行 9：

`IN NS` 指定負責此網域的名稱伺服器。`dns` 會擴充為 `dns.example.com`，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 記錄指定接受、處理和轉寄網域 `example.com` 之電子郵件的郵件伺服器。在此範例中，其為主機 `mail.example.com`。主機名稱前的號碼是優先設定值。如果有多個 MX 項目，則優先選用值最小的郵件伺服器。如果將郵件傳送到此伺服器失敗，則會使用下一個值更大的項目。

行 12—19：

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。此處列出的名稱不含 `.`，因為它們不包含其網域，所以會將 `example.com` 新增到所有名稱。系統會將兩個 IP 位址指定給主機 `gate`，因為它有兩張網路卡。若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，則會使用 `AAAA` 標示該項。

注意：IPv6 語法

IPv6 記錄與 IPv4 的語法稍有不同。因為可以分段，所以必須在位址前提供有關遺漏位元的資訊。若要使用所需的數字「}0」填寫 IPv6 位址，則在位址的正確位置新增兩個冒號。

```
pluto     AAAA 2345:00C1:CA11::1234:5678:9ABC:DEF0
pluto     AAAA 2345:00D2:DA11::1234:5678:9ABC:DEF0
```

行 20：

別名 `ntp` 可以用來定址 `dns`（`CNAME` 表示 canonical name（標準名稱））。

虛擬網域 `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` 會解析為 `168.192.in-addr.arpa`。請參閱範例 25.7「反向查詢」。

範例 25.7 反向查詢

```
1.  $TTL 2D
```

```
2.  168.192.in-addr.arpa.   IN SOA dns.example.com. root.example.com. (
3.                          2003072441      ; serial
4.                          1D              ; refresh
5.                          2H              ; retry
6.                          1W              ; expiry
7.                          2D )            ; minimum
8.
9.                          IN NS           dns.example.com.
10.
11. 1.5                     IN PTR          gate.example.com.
12. 100.3                   IN PTR          www.example.com.
13. 253.2                   IN PTR          cups.example.com.
```

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` 啓用反向查詢。假設區域稱為 `168.192.in-addr.arpa`，則不應新增到主機名稱。因此輸入的所有主機名稱都使用完整格式 — 附帶網域並以 `.` 做為結尾。其餘的項目與之前 `example.com` 範例中所述的項目對應。

行 3 至 7：

請參閱之前的 `example.com` 範例。

行 9：

同樣地，此行指定負責此區域的名稱伺服器。不過，這一次是以完整格式輸入名稱，即包含網域以及結尾的 `.`。

行 11 至 13：

這些是相關主機上 IP 位址的指標記錄提示。行的開頭僅輸入了 IP 位址的最後一部分，結尾沒有 `.`。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。

一般情況下，不同 BIND 版本之間的區域傳輸應該可以順利進行。

25.7 區域資料的動態更新

「動態更新」這個詞是指新增、變更或刪除主伺服器的區域檔案項目的作業。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，可為每個區域項目個別設定動態更新。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。如需此指令的完整語法，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 第 25.8 節 「安全交易」 所述。

25.8 安全交易

透過採用共享秘密金鑰（也稱為 TSIG 金鑰）的交易簽章（TSIG），可以實現安全交易。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，以及區域資料的動態更新，都需要安全交易。讓存取控制依靠金鑰比單純依靠 IP 位址要來得安全許多。

使用以下指令可產生 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，才能開啓 `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

請確定 `/etc/named.conf` 的許可權受到適當的限制。此檔案的預設值是 `0640`，擁有者為 `root` 及群組 `named`。另一種方法是，將金鑰移到具有特別限定許可權的其他檔案，然後將檔案從 `/etc/named.conf` 包含進來。若要包含外部檔案，請使用：

```
include  "filename"
```

將 **檔案名稱** 取代為金鑰所在檔案的絕對路徑。

若要讓伺服器 `host1` 能夠使用 `host2`（在此範例中位址為 `10.1.2.3`）的金鑰，伺服器的 `/etc/named.conf` 必須包含以下規則：

```
server 10.1.2.3 {
  keys { host1-host2. ;};
};
```

類比項目必須包含於 `host2` 的組態檔中。

針對為 IP 位址和位址範圍定義的任何 ACL（存取控制清單，切勿與檔案系統 ACL 混淆）新增 TSIG 金鑰，以確保交易安全性。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。

25.9 DNS 安全性

DNSSEC 或 DNS 安全性細述於 RFC 2535。DNSSEC 的可用工具在 BIND 手冊中有詳加討論。

與一或多個區域金鑰關聯的區域才是安全區域。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。

使用 `dnssec-signzone` 指令，可以建立產生的金鑰集（`keyset-` 檔案），將金鑰集安全地傳送至父區域，並對其簽名。這樣便會在 `/etc/named.conf` 中產生每個區域要包含的檔案。

25.10 更多資訊

如需詳細資訊，請參閱安裝於 `/usr/share/doc/packages/bind/arm` 下的 `bind-doc` 套件中的 BIND Administrator Reference Manual (BIND 管理員參考手冊)。另外也請參閱手冊參考的 RFC 以及 BIND 隨附的手冊頁。`/usr/share/doc/packages/bind/README.SUSE` 中包含有關 SUSE Linux Enterprise Server 中 BIND 的最新資訊。

26 DHCP

動態主機組態通訊協定（DHCP）的用途是從伺服器集中指定網路設定，這樣就不必在每一個工作站本地分別設定。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。如果您在用戶端使用 NetworkManager，則不必設定用戶端。這在環境多變、而且一次只使用一個介面的情況下非常有用。絕對不要在執行 DHCP 伺服器的機器上使用 NetworkManager。

提示：IBM z Systems：DHCP 支援

在 IBM z Systems 平台上，DHCP 僅可在使用 OSA 和 OSA Express 網路卡的介面上運作。這些網路卡是唯一使用 MAC 的網路卡，是 DHCP 自動組態功能的必要元件。

DHCP 伺服器的一種設定方式是識別每個使用網路卡硬體位址（大部分的情況下是固定的）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，也可以將 DHCP 設定成從專門設定的位址池，動態指派位址給每個相關用戶端。在後者的情形中，DHCP 伺服器在每次接到請求時會指派相同的位址給用戶端，即使經過較長的時間也是一樣。這只適用於網路用戶端數少於網路位址數的情況。

DHCP 可簡化系統管理員的工作。任何與位址及一般網路組態相關的變更（包括較大的變更）都可集中執行，只要編輯伺服器的組態檔。與重新設定眾多的工作站相比，這種方法要便利許多。此外，由於這些機器可以從位址池獲取 IP 位址，因此將機器（特別是新機器）整合到網路中也要容易得多。從 DHCP 伺服器擷取適當的網路設定這種方式，對於經常需要在不同網路中使用筆記型電腦的情況特別有用。

在本章中，DHCP 伺服器將在使用 `192.168.2.1` 做為閘道的工作站 `192.168.2.0/24` 所在的同一子網路中執行。DHCP 伺服器有固定的 IP 位址 `192.168.2.254`，提供的位址範圍有兩個：`192.168.2.10` 到 `192.168.2.20` 以及 `192.168.2.100` 到 `192.168.2.200`。

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，甚至是列印伺服器。

26.1 使用 YaST 設定 DHCP 伺服器

若要安裝 DHCP 伺服器，請啓動 YaST 並選取軟體 › 軟體管理。選擇過濾器 › 模式，然後選取DHCP 及 DNS 伺服器。請確認安裝個別套件，以完成此安裝程序。

重要：LDAP 支援

YaST DHCP 模組可以設定成將伺服器組態儲存在本地（即執行 DHCP 伺服器的主機上），或是由 LDAP 伺服器來管理其組態資料。若要使用 LDAP，請在設定 DHCP 伺服器前設定 LDAP 環境。

如需 LDAP 的詳細資訊，請參閱《Security Guide》，第 5 章「LDAP—A Directory Service」。

YaST DHCP 模組（`yast2-dhcp-server`）可讓您將自己的 DHCP 伺服器設定用於區域網路。該模組能夠以精靈模式或是進階組態模式執行。

26.1.1 初始組態（精靈）

首次啓動模組時，會啓動精靈，提示您對伺服器管理進行一些基本設定。完成此初始設定程序後會產生一個非常基本的伺服器組態，能夠發揮其基本功能。進階模式可以處理更多進階設定任務。請執行下列步驟：

1. 請從清單中選取 DHCP 伺服器應監聽的介面，然後按一下選取。接著選取開啓選取介面的防火牆，以開啓此介面的防火牆，然後按下一步。請參閱圖形 26.1「DHCP 伺服器：介面卡選項」。

圖形 26.1 DHCP 伺服器：介面卡選項

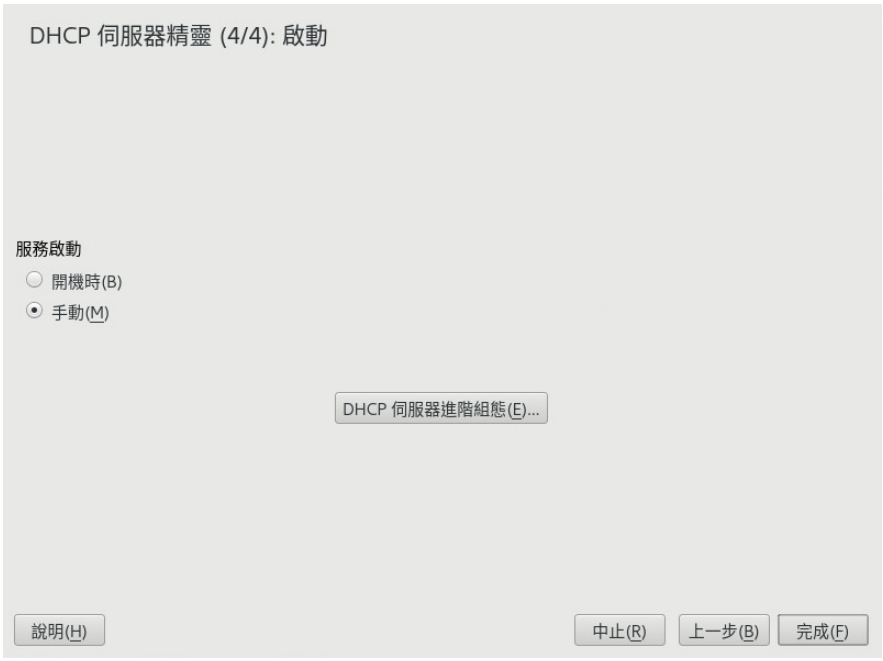
2. 使用核取方塊決定您的 DHCP 設定是否要由 LDAP 伺服器自動儲存。在文字方塊中提供 DHCP 伺服器應管理的所有用戶端的網路細節。這些細節包括網域名稱、時間伺服器的位址、主要及次要名稱伺服器的位址、列印和 WINS 伺服器的位址（供同時具有 Windows 與 Linux 用戶端的混和網路使用）、閘道位址以及租用時間。請參閱圖形 26.2 「DHCP 伺服器：全域設定」。

圖形 26.2 DHCP 伺服器：全域設定

3. 設定應該如何將動態 IP 位址指定給用戶端。若要這樣做，指定伺服器可以指派位址給 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。另外也指定租用時間，在此時間段內用戶端可以保留其 IP 位址，無需要求延續租用。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 26.3 「DHCP 伺服器：動態 DHCP」。

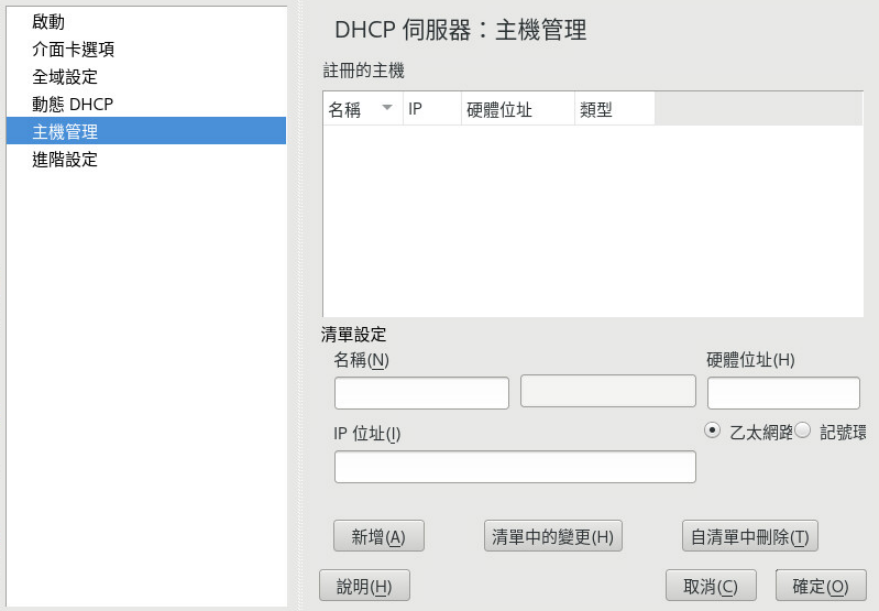
圖形 26.3 DHCP 伺服器：動態 DHCP

4. 定義應該如何啓動 DHCP 伺服器。指定您希望系統開機時自動啓動 DHCP 伺服器，還是在需要時手動啓動（例如，用於測試目的）。按一下完成完成伺服器的組態。請參閱圖形 26.4 「DHCP 伺服器：啓動」。



圖形 26.4 DHCP 伺服器：啓動

5. 除了以先前步驟描述的方式使用動態 DHCP 外，您也可以設定伺服器以準靜態方式指定位址。使用下半部所提供的文字方塊，指定要以這種方式進行管理之用戶端的清單。具體而言，就是提供要指定給這類用戶端的名稱和IP 位址，另外還要提供硬體位址和網路類型(記號環或乙太網路)。使用新增、編輯和從清單中刪除修改顯示於螢幕上半部分的用戶端清單。請參閱圖形 26.5 「DHCP 伺服器：主機管理」。



圖形 26.5 DHCP 伺服器：主機管理

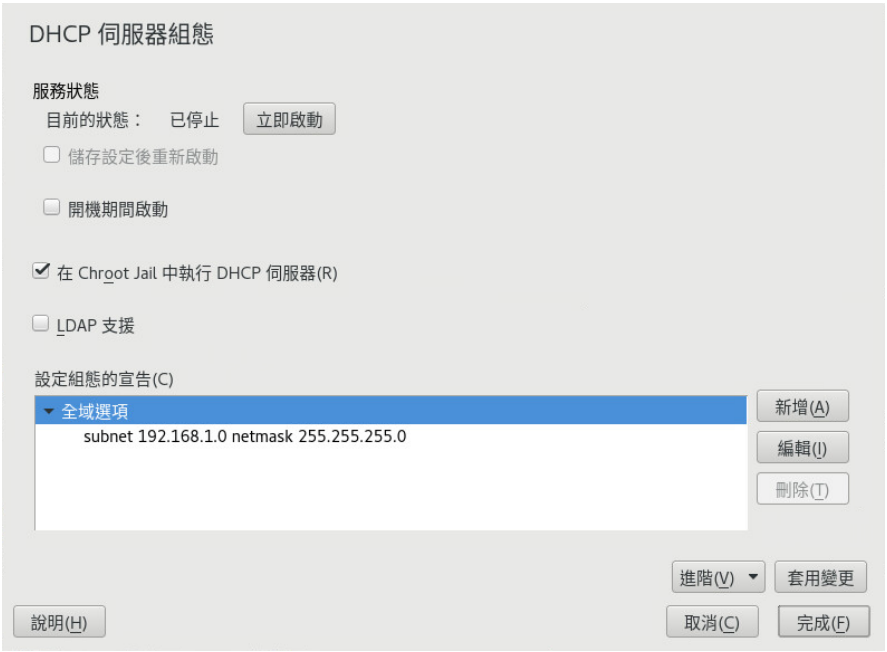
26.1.2 DHCP 伺服器組態（進階）

除了稍早討論的組態方法外，還有進階組態模式，可讓您變更 DHCP 伺服器設定的每個細節。啓動進階組態，方法是在啓動對話方塊中按一下DHCP 伺服器進階組態(請參閱圖形 26.4 「DHCP 伺服器：啓動」)。

Chroot 環境與宣告

在此第一個對話方塊中，選取啓動 DHCP 伺服器讓現有組態可以編輯。DHCP 伺服器行為的一個重要功能是它能夠在 chroot 環境或 chroot jail 中執行，以確保伺服器主機的安全。如果 DHCP 伺服器受到了外部攻擊，攻擊者仍將被封鎖在 chroot jail 中，從而可防止其存取系統的其他部分。對話方塊的下半部顯示已經定義之宣告的樹狀檢視。使用新增、刪除和編輯修改這些宣告。選取進階將帶

您到其他的進階對話方塊。請參閱圖形 26.6 「DHCP 伺服器：Chroot Jail 和宣告」。 選取 新增 後，定義要新增的宣告類型。使用 進階，檢視伺服器的記錄檔、設定 TSIG 金鑰管理，並根據 DHCP 伺服器的設定調整防火牆的組態。



圖形 26.6 DHCP 伺服器：CHROOT JAIL 和宣告

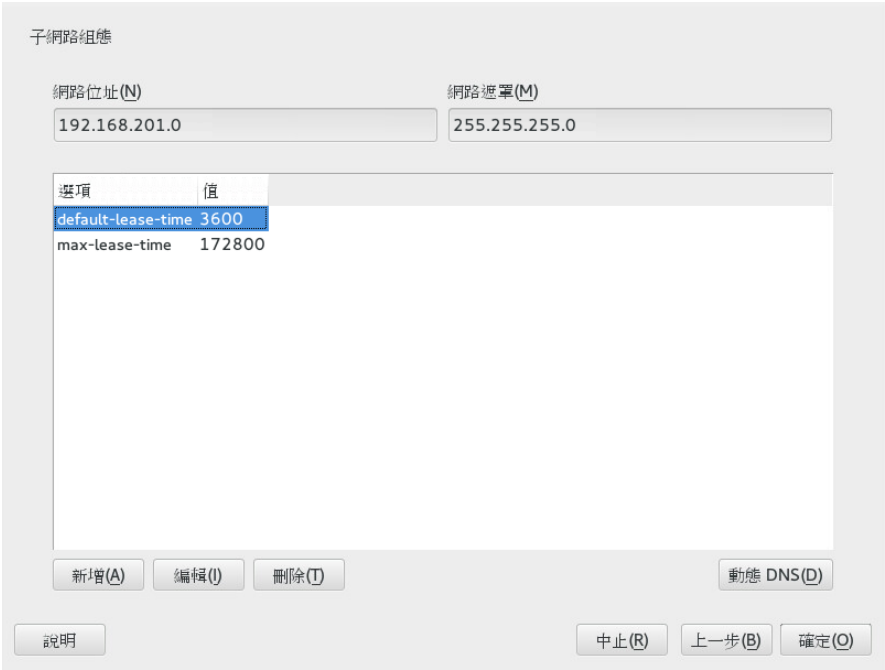
選取宣告類型

DHCP 伺服器的全域選項由數個宣告組成。此對話方塊可讓您設定宣告類型：子網路、主機、共享網路、群組、位址池和類別。此範例顯示新子網路的選項（請參閱圖形 26.7 「DHCP 伺服器：選取宣告類型」）。

圖形 26.7 DHCP 伺服器：選取宣告類型

子網路組態

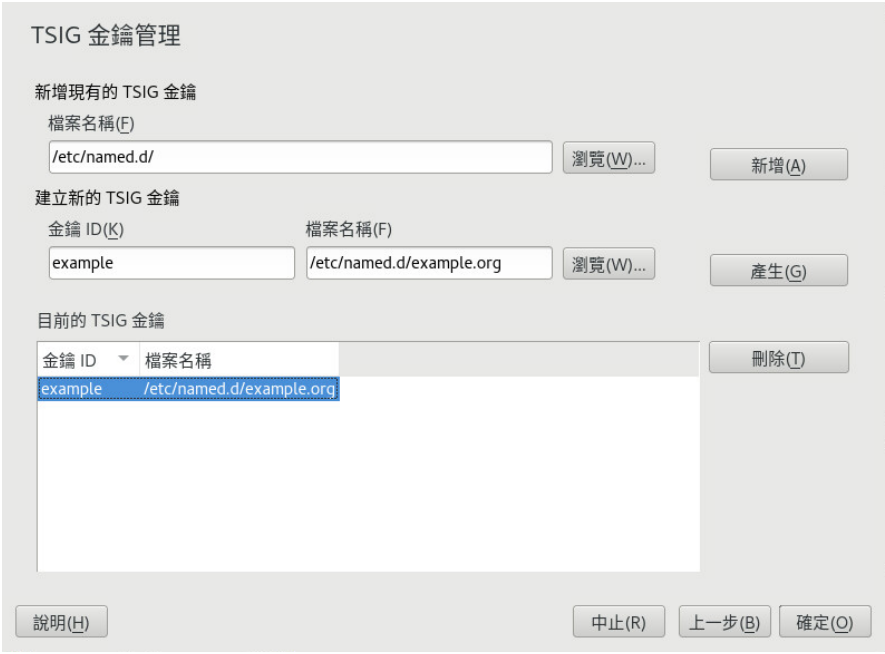
此對話方塊可讓您使用 IP 位址及網路遮罩來指定新的子網路。在對話方塊的中間部分，使用新增、編輯和刪除修改所選子網路的 DHCP 伺服器啓動選項。若要設定子網路的動態 DNS，選取 動態 DNS。



圖形 26.8 DHCP 伺服器：設定子網路

TSIG 金鑰管理

如果在上一個對話方塊中選擇設定動態 DNS，現在可以針對安全區域傳輸設定金鑰管理。選取確定 會帶您到另一個對話方塊，讓您設定動態 DNS 的介面（請參閱 圖形 26.10 「DHCP 伺服器：動態 DNS 的介面組態」）。



圖形 26.9 DHCP 伺服器：TSIG 組態

動態 DNS：介面組態

選取啓用此子網路的動態 DNS，即可啓用子網路的動態 DNS。完成此操作後，使用下拉式方塊啓用正向與反向區域的 TSIG 金鑰（確定這些金鑰同時也是 DNS 和 DHCP 伺服器使用的金鑰）。使用更新全域動態 DNS 設定，來依據動態 DNS 環境自動更新及調整全域 DHCP 伺服器設定。最後，定義每個動態 DNS 的哪些轉遞和反向區域需要更新，為兩個區域都指定主要名稱伺服器的名稱。選取確定返回子網路組態對話方塊（請參閱圖形 26.8 「DHCP 伺服器：設定子網路」）。再次選取確定返回最初的進階組態對話方塊。

圖形 26.10 DHCP 伺服器：動態 DNS 的介面組態

網路介面組態

若要定義 DHCP 伺服器應該監聽的介面以及調整防火牆組態，請從進階組態對話方塊中選取進階 › 介面組態。從顯示的介面清單中，選取一或多個要由 DHCP 伺服器處理的介面。如果所有子網路中的用戶端都必須能與伺服器通訊，並且伺服器主機還要執行防火牆，請對防火牆進行相應的調整。若要這樣做，請選取調整防火牆設定。接著，YaST 會根據新的條件調整 SuSEFirewa112 的規則（請參閱圖形 26.11 「DHCP 伺服器：網路介面和防火牆」），之後您即可選取確定返回最初的對話方塊。

圖形 26.11 DHCP 伺服器：網路介面和防火牆

完成所有組態步驟後，按一下確定關閉對話方塊。伺服器現在會以新的組態啓動。

26.2 DHCP 軟體套件

SUSE Linux Enterprise Server 可以使用 DHCP 伺服器，也可以使用 DHCP 用戶端。可用的 DHCP 伺服器是 `dhcpd`（由 Internet Systems Consortium 發佈）。用戶端提供了 `dhcp-client`（ISC 也有提供）及 `wicked` 套件附帶的工具。

依預設，`wicked` 工具會連同 `wickedd-dhcp4` 和 `wickedd-dhcp6` 服務一起安裝。系統每次開機時，會自動啓動它們，以監視 DHCP 伺服器。它們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。如果情況較為複雜，請使用 ISC `dhcp-client`，此程式透過組態檔 `/etc/dhclient.conf` 和 `/etc/dhclient6.conf` 來控制。

26.3 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，租用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 26.1 「組態檔 /etc/dhcpd.conf」 中的 `/etc/dhcpd.conf` 基本範例檔案。

範例 26.1 組態檔 /ETC/DHCPD.CONF

```
default-lease-time 600;         # 10 minutes
max-lease-time 7200;            # 2  hours

option domain-name "example.com";
option domain-name-servers 192.168.1.116;
option broadcast-address 192.168.2.255;
option routers 192.168.2.1;
option subnet-mask 255.255.255.0;

subnet 192.168.2.0 netmask 255.255.255.0
 {
  range 192.168.2.10 192.168.2.20;
  range 192.168.2.100 192.168.2.200;
 }
```

這個簡單的組態檔應足以讓 DHCP 伺服器在網路中指派 IP 位址。請確定每行結尾都插入分號，否則不會啓動 dhcpd。

範例檔可以分為三個部分。第一個部分定義要求用戶端預設可租用 IP 位址的秒數（`default-lease-time`），此時間過後將需要申請續約。其中還包含機器可以保留 DHCP 伺服器指派之 IP 位址、不申請續約的最長期限的陳述式（`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網域。
- 利用 `option domain-name-servers` 項目，最多可指定三個 DNS 伺服器值，用來將 IP 位址解析為主機名稱，或將主機名稱解析為 IP 位址。最好在設定 DHCP 前先設定機器上或網路上其他位置的名稱伺服器。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 第 25 章 「網域名稱系統」。
- option broadcast-address **這一行定義要求用戶端應使用的廣播位址。**
- 利用 `option routers`，可設定伺服器要將無法傳送至區域網路上之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）。通常情況下，特別是在較小的網路中，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定義網路，包括子網路遮罩。若要完成設定，請指定 DHCP 精靈用來指派 IP 位址給相關用戶端的位址範圍。在範例 26.1 「組態檔 /etc/dhcpd.conf」中，用戶端的指定位址可介於 `192.168.2.10` 與 `192.168.2.20` 之間或 `192.168.2.100` 與 `192.168.2.200` 之間。

編輯這幾行後，便可以使用指令 `systemctl start dhcpd` 來啓動 DHCP 精靈。該精靈可以立即使用。使用指令 `rcdhcpd check-syntax` 執行簡短的語法檢查。如果遇到任何組態方面的非預期問題（例如伺服器因出現錯誤而中止或在啓動時沒有傳回 `done`），則可以使用指令 `journalctl` 查詢主系統記錄中的資訊，找出問題所在（如需更多資訊，請參閱第 15 章 「journalctl：查詢 systemd 日誌」）。

在預設的 SUSE Linux Enterprise Server 系統上，出於安全性考量，DHCP 精靈會在 chroot 環境中啓動。組態檔必須複製到 chroot 環境，如此精靈才可以找到這些檔案。通常情況下不需要擔心發生這種情形，因為指令 `systemctl start dhcpd` 會自動複製檔案。

26.3.1 使用固定 IP 位址的用戶端

DHCP 也可以用來指派預先定義的靜態位址給特定用戶端。明確指派的位址永遠比集區的動態位址優先。當可用位址不足，伺服器必須在用戶端之間重新分配位址時（舉例而言），動態位址便會過期，靜態位址則不同，它永遠不會過期。

為了識別使用靜態位址設定的用戶端，dhcpd 會使用硬體位址（硬體位址是全球唯一的固定數字代碼，由六對八位元組組成），用來識別所有網路裝置（例如，`00:30:6E:08:EC:80`）。如果相對的行（如 範例 26.2 「組態檔的增加部分」 中所示）新增到 範例 26.1 「組態檔 /etc/dhcpd.conf」 的組態檔，DHCP 精靈始終會指定相同的資料集給對應的用戶端。

範例 26.2 組態檔的增加部分

```
host jupiter {
hardware ethernet 00:30:6E:08:EC:80;
fixed-address 192.168.2.100;
}
```

在第一行中輸入相應用戶端的名稱（`host HOSTNAME`，在本例中為 `jupiter`），在第二行中輸入 MAC 位址。在 Linux 主機上，使用指令 `ip link show` 並在後面接上網路裝置（例如，`eth0`）可尋找 MAC 位址。輸出應該包含如下內容

```
link/ether 00:30:6E:08:EC:80
```

在上述範例中，系統會自動給具有網路卡且 MAC 位址為 `00:30:6E:08:EC:80` 的用戶端指定 IP 位址 `192.168.2.100` 及主機名稱 `jupiter`。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。

26.3.2 SUSE Linux Enterprise Server 版本

為了提高安全性，SUSE Linux Enterprise Server 版 ISC DHCP 伺服器在出貨時即套用了 Ari Edelkind 的 non-root/chroot 修補程式。這樣可讓 dhcpd 利用使用者 ID `nobody` 執行，而且也能在 chroot 環境下（`/var/lib/dhcp`）執行 dhcpd。若要這樣做，組態檔 `dhcpd.conf` 必須位於 `/var/lib/dhcp/etc`。init 程序檔啓動時會自動複製檔案到此目錄。

透過檔案 `/etc/sysconfig/dhcpd` 中的項目，可控制伺服器與此功能相關的行為。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的組態檔：

- `/etc/localtime`
- `/etc/host.conf`

- /etc/hosts
- /etc/resolv.conf

啓動 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了這些檔案，則進行所需變更時，也要考量到這些副本。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。

如果組態包含要複製到 chroot 環境的其他檔案，請在檔案 /etc/sysconfig/dhcpd 中的變數 DHCPD_CONF_INCLUDE_FILES 下設定這些檔案。為了確定重新啓動 syslog 精靈後，DHCP 記錄功能仍起作用，/etc/sysconfig/syslog 檔案中還要有另一個項目 SYSLOGD_ADDITIONAL_SOCKET_DHCP。

26.4 更多資訊

如需有關 DHCP 的詳細資訊，請參閱 Internet Systems Consortium 的網站 http://www.isc.org/products/DHCP/。在 dhcpd、dhcpd.conf、dhcpd.leases 和 dhcp-options man 頁面中也可以找到資訊。

27 使用 NFS 共享檔案系統

網路檔案系統（NFS）是允許存取伺服器上檔案的通訊協定，存取方式與存取本地檔案非常相似。

27.1 綜覽

網路檔案系統（NFS）是經過充分證明且受到廣泛支援的標準化網路通訊協定，它允許在單獨的主機之間共用檔案。

網路資訊服務（NIS）可用於在網路中進行集中式使用者管理。將 NFS 和 NIS 結合使用，可透過檔案和目錄權限在網路中進行存取控制。NFS 與 NIS 攜手能讓使用者對網路有清楚的瞭解。

在預設組態中，NFS 完全信任網路，因此會信任連接到可信網路的任何機器。在可透過實體方式存取 NFS 伺服器完全信任的任何網路的任何電腦上，任何具有管理員特權的使用者都可以存取該伺服器提供的所有檔案。

在許多情況下，此安全性層級非常適合以下情形：所信任的網路是真正的私人網路，通常局限於單個機櫃或機房，並且無法進行未經授權的存取。將整個子網路做為一個整體信任的其他情形限制較多，需要更精密的信任機制。為了符合這些情形的需要，NFS 使用 Kerberos 基礎架構來支援各種安全性層級。Kerberos 需要 NFSv4（預設使用該通訊協定）。如需詳細資料，請參閱《Security Guide》，第 6 章「Network Authentication with Kerberos」。

YaST 模組中使用了下列詞彙。

輸出
: 由 NFS 伺服器輸出的目錄，用戶端可將該目錄整合到其系統中。

NFS 用戶端
: NFS 用戶端是指透過「網路檔案系統」通訊協定使用 NFS 伺服器提供之 NFS 服務的系統。TCP/IP 通訊協定已整合到 Linux 核心中；不需要安裝任何其他軟體。

NFS 伺服器

NFS 伺服器向用戶端提供 NFS 服務。執行中的伺服器依賴於以下精靈運作：`nfsd`（worker）、`idmapd`（用於 NFSv4 的 ID 到名稱映射，僅在某些情境下需要）、`statd`（檔案鎖定）和 `mountd`（掛接要求）。

NFSv3

NFSv3 是第 3 版實作，即支援用戶端驗證的「舊」無狀態 NFS。

NFSv4

NFSv4 是新版（第 4 版）實作，支援透過 kerberos 的安全使用者驗證。NFSv4 只需要一個連接埠，因此比 NFSv3 更適合在防火牆後的環境中使用。
協定的定義如 http://tools.ietf.org/html/rfc3530 所示。

pNFS

平行 NFS，NFSv4 的通訊協定延伸。所有 pNFS 用戶端都可以直接存取 NFS 伺服器上的資料。

重要：DNS 所需

原則上，可以只使用 IP 位址來進行所有的輸出。為了避免逾時，您需要有正常運作的 DNS 系統。即使為了記錄目的，DNS 也必不可少，因為掛接的精靈執行的是反向查詢。

27.2 安裝 NFS 伺服器

預設不會安裝 NFS 伺服器。若要使用 YaST 安裝 NFS 伺服器，請依次選擇軟體 › 軟體管理、模式，然後啓用伺服器功能區段的檔案伺服器選項。按接受以安裝所需的套件。

NFS 與 NIS 一樣，都是主從式系統。但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），也可以從其他主機掛接檔案系統（輸入）。

注意：在輸出伺服器上本地掛接 NFS 磁區

SUSE Linux Enterprise Server 上不支援在輸出伺服器本地掛接 NFS 磁碟區。

27.3 設定 NFS 伺服器

NFS 伺服器可透過 YaST 來設定，也可以手動設定。若要進行驗證，還可以將 NFS 與 Kerberos 結合使用。

27.3.1 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，此類伺服器可將目錄和檔案輸出到擁有其存取權的所有主機或一個群組下的全體成員。因此，無需在每個主機上本地安裝應用程式，伺服器也可以提供應用程式。

若要設定這樣的伺服器，請執行下列步驟：

程序 27.1 設定 NFS 伺服器

1. 啓動 YaST 並選取網路服務 › NFS 伺服器；請參閱圖形 27.1 「NFS 伺服器組態工具」。系統可能會提示您安裝其他軟體。

圖形 27.1 NFS 伺服器組態工具

2. 啓用開始選項圓鈕。
3. 如果系統上的防火牆（SuSEfirewall2）處於使用中狀態，請核取在防火牆中開啓連接埠。YaST 會啓用 `nfs` 服務來調整其組態以使其適用於 NFS 伺服器。

4. 檢查是否要啓用 NFSv4。如果停用 NFSv4，YaST 將只支援 NFSv3。如需啓用 NFSv2 的相關資訊，請參閱注意：NFSv2。

    - 如果選取 NFSv4，請另行輸入適當的 NFSv4 網域名稱。`idmapd` 精靈會使用此參數。Kerberos 設定需要該精靈，當用戶端無法處理數字使用者名稱時，也需要使用該精靈。如果您不執行 `idmapd` 或無任何特殊要求，請將它保留為 `localdomain`（預設值）。如需 `idmapd` 精靈的詳細資訊，請參閱 `/etc/idmapd.conf`。

5. 若您需要安全存取伺服器，請按一下啓用 GSS 安全性。先決條件是您的網域中安裝有 Kerberos，且伺服器和用戶端都可進行 Kerberos 驗證。按下一步繼續執行下一個組態對話方塊。

6. 按一下對話方塊上半部分中的新增目錄以輸出目錄。

7. 如果您尚未設定允許的主機，系統會自動彈出另一個對話方塊，可讓您輸入用戶端資訊及選項。輸入主機萬用字元（通常您可以保留預設設定不變）。
   有四種主機萬用字元類型可讓您針對各主機進行設定：單一主機（名稱或 IP 位址）、網路群組、萬用字元（例如 `*` 摽是所有機器都可存取伺服器）以及 IP 網路。
   如需這些選項的詳細資訊，請參閱**輸出**手冊頁。

8. 按一下完成以完成組態。

27.3.2 手動輸出檔案系統

NFS 輸出服務的組態檔案為 `/etc/exports` 和 `/etc/sysconfig/nfs`。如果 NFSv4 伺服器組態包含經過 Kerberos 驗證的 NFS，或者用戶端不能使用數字使用者名稱，則除了這些檔案外，還需要 `/etc/idmapd.conf`。

若要啓動或重新啓動服務，請執行 `systemctl restart nfsserver` 指令。此指令還會將 NFS 伺服器必需的 RPC portmapper 重新啓動。

為確保 NFS 伺服器永遠都會在開機時啓動，請執行 `sudo systemctl enable nfsserver`。

注意：NFSv4

NFSv4 是 SUSE Linux Enterprise Server 上可用的最新版 NFS 通訊協定。現在，針對 NFSv4 輸出的目錄設定方法與 NFSv3 相同。

在 SUSE Linux Enterprise Server 11 上，必須在 `/etc/exports` 中指定結合掛接。現在仍支援此方式，但已有了取代方案。

`/etc/exports`

`/etc/exports` 檔案包含一份項目清單。每一個項目都指出一個共享的目錄，並記錄它的共享方式。`/etc/exports` 中的典型項目會包含：

```
/SHARED/DIRECTORY   HOST(OPTION_LIST)
```

例如：

```
/export/data   192.168.1.2(rw,sync)
```

這裡使用了 IP 位址 `192.168.1.2`，以識別允許的用戶端。您也可以使用主機的名稱以及指向一組主機（`*.abc.com`、`*` 等）或網路群組（`@my-hosts`）的萬用字元。

如需所有選項及其意義的詳細說明，請參閱 man 頁面 `exports`（`man exports`）。

如果您在 NFS 伺服器執行時修改了 `/etc/exports`，則需使用 `sudo systemctl restart nfsserver` 指令重新啓動 NFS 伺服器，以使變更生效。

`/etc/sysconfig/nfs`

`/etc/sysconfig/nfs` 檔案包含一些決定 NFSv4 伺服器精靈行為的參數。參數 `NFS4_SUPPORT` 必須設定為 `yes`（預設值）。`NFS4_SUPPORT` 決定 NFS 伺服器是否支援 NFSv4 輸出和用戶端。

如果您在 NFS 伺服器執行時修改了 `/etc/sysconfig/nfs`，則需使用 `sudo systemctl restart nfsserver` 指令重新啓動 NFS 伺服器，以使變更生效。

提示：裝載選項

在 SUSE Linux Enterprise Server 11 上，必須在 `/etc/exports` 中指定 `--bind` 掛接。現在仍支援此方式，但已有了取代方案。現在，針對 NFSv4 輸出的目錄設定方法與 NFSv3 相同。

注意：NFSv2

如果 NFS 用戶端仍相依於 NFSv2，可透過以下設定在伺服器的 `/etc/sysconfig/nfs` 中將其啓用：

```
NFSD_OPTIONS="-V2"
MOUNTD_OPTIONS="-V2"
```

重新啓動服務後，執行以下指令檢查版本 2 是否可使用：

```
tux > cat /proc/fs/nfsd/versions
+2 +3 +4 +4.1 -4.2
```

`/etc/idmapd.conf`

從 SLE 12 SP1 開始，僅當使用 Kerberos 驗證或用戶端不能使用數字使用者名稱時，才需要 `idmapd` 精靈。自 Linux 核心 2.6.39 起，Linux 用戶端可以使用數字使用者名稱。`idmapd` 精靈會將傳送到伺服器的 NFSv4 要求進行名稱到 ID 的映射，並回覆用戶端。

如果需要，`idmapd` 需在 NFSv4 伺服器上執行。用戶端上的名稱到 ID 映射將由以下套件提供的 `nfsidmap` 來執行： `nfs-client`。

對於可能使用 NFS 來共用檔案系統的機器，請確定有一個統一的方式來為各機器之間的使用者指定使用者名稱和 ID (uid)。您可以透過 NIS、LDAP 或您網域中的任何統一網域驗證機制來達成這個目的。

在用戶端和伺服器兩端的 `/etc/idmapd.conf` 檔案中，`Domain` 參數必須設定為相同的值。如果您不確定，請讓伺服器和用戶端檔案中的網域保持為 `localdomain`。我們在此提出一個組態檔案的例子，如下所示：

```
[General]
Verbosity = 0
Pipefs-Directory = /var/lib/nfs/rpc_pipefs
Domain = localdomain

[Mapping]
Nobody-User = nobody
Nobody-Group = nobody
```

若要啓動 `idmapd` 精靈，請執行 `systemctl start nfs-idmapd`。如果您在精靈執行時修改了 `/etc/idmapd.conf`，則需使用 `systemctl start nfs-idmapd` 指令重新啓動精靈，以使變更生效。

如需更多資訊，請參閱 `idmapd` 與 `idmapd.conf` 的手冊頁（`man idmapd` 與 `man idmapd.conf`）。

27.3.3 NFS 配合使用 Kerberos

若要讓 NFS 使用 Kerberos 驗證，必須啓用一般安全性服務 (GSS)。在初始 YaST NFS 伺服器對話方塊中選取啓用 GSS 安全性。您必須有一個工作中的 Kerberos 伺服器才能使用此功能。YaST 不會設定伺服器，而只是使用所提供的功能。如果您要使用 Kerberos 驗證，則除了 YaST 組態之外，還必須要完成以下步驟才能執行 NFS 組態：

1. 請確定伺服器和用戶端位於相同的 Kerberos 領域中。它們必須存取相同的 KDC（金鑰配送中心）伺服器，並共享它們的 `krb5.keytab` 檔案（所有機器上的預設位置都是 `/etc/krb5.keytab`）。如需有關 Kerberos 的詳細資訊，請參閱《Security Guide》，第 6 章「Network Authentication with Kerberos」
2. 在用戶端上執行 `systemctl start rpc-gssd.service` 以啓動 gssd 服務。
3. 在伺服器上執行 `systemctl start rpc-svcgssd.service` 以啓動 svcgssd 服務。

若要進行 Kerberos 驗證，也需要在伺服器上執行 `idmapd` 精靈。如需詳細資訊，請參閱 `/etc/idmapd.conf`。

如需有關設定啓用 Kerberos 之 NFS 的詳細資訊，請參閱第 27.5 節 「更多資訊」中的連結所提供的內容。

27.4 設定用戶端

您不需安裝其他軟體就能將您的主機設定為 NFS 用戶端。所有需要的套件預設都會安裝。

27.4.1 以 YaST 輸入檔案系統

授權使用者可以使用 YaST NFS 用戶端模組將NFS 目錄從 NFS 伺服器掛接到本地檔案樹。請執行下列步驟：

程序 27.2 輸入 NFS 目錄

1. 啓動 YaST NFS 用戶端模組。
2. 在NFS 共用索引標籤中按一下新增。輸入 NFS 伺服器的主機名稱、要輸入的目錄和在本地掛接此目錄的掛接點。
3. 使用 NFSv4 時，在NFS 設定標籤中選取 啓用 NFSv4。另外，NFSv4 網域名稱必須包含 NFSv4 伺服器所用的相同值。預設網域為 `localdomain`。
4. 若要為 NFS 使用 Kerberos 驗證，則 GSS 安全性必須啓用。選取啓用 GSS 安全性。
5. 如果您要使用防火牆並希望允許遠端電腦存取服務，請啓用NFS 設定索引標籤中的在防火牆中開啓埠。防火牆的狀態顯示於核取方塊旁。
6. 按一下確定，儲存變更。

組態將會寫入 `/etc/fstab` 中，並會掛接指定的檔案系統。當您稍後啓動 YaST 組態用戶端時，它也會從這個檔案讀取現有組態。

提示：用做根檔案系統的 NFS

在透過網路將根分割區掛接為 NFS 共用的（無磁碟）系統中，設定可用於存取 NFS 共用的網路裝置時需特別小心。

將系統關閉或重新開機時，預設的處理順序是先關閉網路連接，然後卸載根分割區。對於 NFS 根分割區，這種順序會造成問題，因為在尚未與 NFS 共用啓動網路連接的情況下，根分割區無法完全卸載。為防止系統停用相關的網路裝置，請依第 16.4.1.2.5 節 「啓動網路裝置」中所述開啓網路裝置組態索引標籤，然後在裝置啓動窗格中選取在 NFSroot 時。

27.4.2 手動輸入檔案系統

從 NFS 伺服器手動輸入檔案系統的先決條件是有 RPC 埠對應程式正在執行。`nfs` 服務負責正確啓動該程式；因此，請以 `root` 身分輸入 `systemctl start nfs`，以啓動該服務。接著，可以像處理本地分割區一樣，使用 `mount` 指令在檔案系統中掛接遠端檔案系統：

```
tux > sudo mount HOST:REMOTE-PATHLOCAL-PATH
```

例如，若要從 `nfs.example.com` 機器輸入使用者目錄，可以使用：

```
tux > sudo mount nfs.example.com:/home /home
```

27.4.2.1 使用自動裝載服務

autofs 精靈可用於自動掛接遠端檔案系統。請將下列項目加入 `/etc/auto.master` 檔案：

```
/nfsmounts /etc/auto.nfs
```

如果能正確填入 `auto.nfs` 檔案，`/nfsmounts` 目錄此後便會成為用戶端上所有 NFS 掛接作業的根部。選擇 `auto.nfs` 這個名稱是從方便角度考量，您可以自行選擇任何名稱。使用以下指令在 `auto.nfs` 中為所有 NFS 掛接作業新增項目：

```
localdata -fstype=nfs server1:/data
nfs4mount -fstype=nfs4 server2:/
```

以 `root` 身分執行 `systemctl start autofs` 以啟動設定。在此範例中，`server1` 的 `/data` 目錄 `/nfsmounts/localdata` 會掛接 NFS，而 `server2` 的 `/nfsmounts/nfs4mount` 會掛接 NFSv4。

如果在 autofs 服務執行期間有程式編輯了 `/etc/auto.master` 檔案，則必須使用 `systemctl restart autofs` 重新啟動自動掛載器，才能使變更生效。

27.4.2.2 手動編輯 /etc/fstab

`/etc/fstab` 中典型的 NFSv3 掛接項目如下：

```
nfs.example.com:/data /local/path nfs rw,noauto 0 0
```

對於 NFSv4 掛接，請在第三欄中使用 `nfs4` 而不是 `nfs`：

```
nfs.example.com:/data /local/pathv4 nfs4 rw,noauto 0 0
```

`noauto` 選項可防止在啟動時自動掛接檔案系統。如果要手動掛接各檔案系統，可以縮短掛接指令的長度，僅指定掛接點：

```
tux > sudo mount /local/path
```

注意：啓動時掛接

請注意，如果未輸入 `noauto` 選項，系統的 `init` 程序檔會在啓動時處理這些檔案系統的掛接。

27.4.3 平行 NFS（pNFS）

NFS 是最舊的通訊協定之一，開發於八十年代。雖然如此，NFS 對於共用小型檔案還是綽綽有餘的。但是，當您要傳輸大型檔案或有大量的用戶端要存取資料時，NFS 伺服器會成為瓶頸，嚴重影響系統效能。出現這種情況的原因是檔案大小快速變大，而乙太網路的相對速度無法完全跟上這一變化。

當您向一般 NFS 伺服器要求檔案時，伺服器會尋找檔案中繼資料、收集所有資料，並透過網路將資料傳輸到您的用戶端。但是，不論檔案的大小，效能瓶頸都會變得很明顯：

- 對於小型檔案，大部分時間都用在收集中繼資料上.
- 對於大型檔案，大部分時間則用在將資料從伺服器傳輸至用戶端上。

pNFS（或平行 NFS）克服了這個局限性，因為它將檔案系統中繼資料與資料位置分隔開來。因此，pNFS 需要兩種類型的伺服器：

- 中繼資料或控制伺服器，用於處理所有非資料流量
- 一或多個儲存伺服器，用於存放資料

中繼資料與儲存伺服器構成了一個邏輯 NFS 伺服器。當用戶端想要讀取或寫入時，中繼資料伺服器會告知 NFSv4 用戶端使用哪個儲存伺服器來存取檔案區塊。用戶端可以直接存取伺服器上的資料。

SUSE Linux Enterprise Server 僅在用戶端上支援 pNFS。

27.4.3.1 使用 YaST 設定 pNFS 用戶端

依程序 27.2 「輸入 NFS 目錄」 中所示繼續操作，但要按一下pNFS (v4.1)核取方塊並選擇性地按一下NFSv4 共享。YaST 會執行所有必要的步驟，並將所有需要的選項寫入檔案 `/etc/exports`。

27.4.3.2 手動設定 pNFS 用戶端

請參閱第 27.4.2 節 「手動輸入檔案系統」 開始設定。大部分組態均由 NFSv4 伺服器執行。對於 pNFS，唯一的差異是將 `minorversion` 選項和中繼資料伺服器 MDS 伺服器新增至 `mount` 指令：

```
tux > sudo mount -t nfs4 -o minorversion=1 MDS_SERVER MOUNTPOINT
```

為了協助進行除錯，請在 `/proc` 檔案系統中變更該值：

```
tux > sudo echo 32767 > /proc/sys/sunrpc/nfsd_debug
tux > sudo echo 32767 > /proc/sys/sunrpc/nfs_debug
```

27.5 更多資訊

除了 `exports`、`nfs` 和 `mount` 的 man 頁面以外，`/usr/share/doc/packages/nfsidmap/README` 中也提供了有關設定 NFS 伺服器和用戶端的資訊。如需更多線上文件，請參閱下列網站：

- 如需詳細的線上技術文件，請造訪 SourceForge (http://nfs.sourceforge.net/)。
- 如需設定已監督之 NFS 的指示，請參閱 NFS 第 4 版開放原始碼實作參考 (http://www.citi.umich.edu/projects/nfsv4/linux/krb5-setup.html)。
- 如果您有任何關於 NFSv4 的問題，請參閱 Linux NFSv4 常見問答集 (http://www.citi.umich.edu/projects/nfsv4/linux/faq/)。

28 Samba

使用 Samba，就可以將 Unix 機器設定為 macOS、Windows 以及 OS/2 機器的檔案與列印伺服器。Samba 已經是一個開發至完全成熟且相當複雜的產品。使用 YaST 或手動編輯組態檔案來設定 Samba。

28.1 術語

下列為 Samba 文件和 YaST 模組中常用的詞彙。

SMB 通訊協定

Samba 使用基於 NetBIOS 服務的 SMB（伺服器訊息區塊）通訊協定。由於 Microsoft 發行了此通訊協定，因此其他的軟體製造商可以建立連接至 Microsoft 網域網路的連接。使用 Samba，SMB 通訊協定就可以在 TCP/IP 通訊協定上運作，因此 TCP/IP 通訊協定必須安裝在所有的用戶端上。

提示：IBM z Systems：NetBIOS 支援

IBM z Systems 僅支援經由 TCP/IP 的 SMB。在這些系統上不提供 NetBIOS 支援。

CIFS 通訊協定

CIFS（一般網際網路檔案系統）通訊協定是 Samba 所支援的另一種通訊協定。CIFS 定義用於網路上的標準遠端檔案系統存取通訊協定，讓使用者群組可以透過網路分工合作和共享文件。

NetBIOS

NetBIOS 是用來在提供名稱服務的機器之間進行通訊的軟體介面（API）。它允許連接至網路的機器保留自己的名稱。在保留後，就可以使用名稱來定址這些機器。在此沒有檢查名稱的中央程序。在網路上的任何機器都可以保留它所需的任何數量名稱，只要這些名稱尚未使用。可以針對不同的網路結構實作 NetBIOS 介面。NetBEUI 是與網路硬體結合相對密切的一種實作，不過通常稱為 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，為了方便管理，一般建議使用與 DNS 主機名稱相對應的名稱，或者在本地使用 DNS。Samba 預設是使用此對應名稱。

Samba 伺服器
: Samba 伺服器可為用戶端提供 SMB/CIFS 服務和 NetBIOS over IP 命名服務。對於 Linux 系統，Samba 伺服器有三種精靈可用：smbd（用於 SMB/CIFS 服務）、nmbd（用於命名服務）及 winbind（用於驗証）。

Samba 用戶端
: Samba 用戶端是透過 SMB 通訊協定，使用 Samba 伺服器所提供之 Samba 服務的系統。常用作業系統（例如 Windows 和 macOS）都支援 SMB 通訊協定。TCP/IP 通訊協定必須安裝在所有的電腦上。Samba 提供適用於不同 Unix 類別的用戶端。就 Linux 而言，有一個 SMB 的核心模組，允許在 Linux 系統層級上整合 SMB 資源。您不必為 Samba 用戶端執行任何精靈。

共享
: SMB 伺服器透過共享方式向用戶端提供資源。共享是指印表機和位在伺服器上的目錄及其子目錄。它是利用名稱來輸出，並且可藉由其名稱來存取。共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。

DC
: 網域控制器（DC）是處理網域中帳戶的伺服器。進行資料複製時，可在一個網域中使用其他領域控制器。

28.2 安裝 Samba 伺服器

若要安裝 Samba 伺服器，請啓動 YaST 並選取軟體 › 軟體管理。選擇檢視 › 模式，然後選取檔案伺服器。請確認安裝所需套件，完成此安裝程序。

28.3 啓動和停止 Samba

您可以在開機時自動啓動 Samba 伺服器，也可以手動啓動或停止。啓動和停止原則是 YaST Samba 伺服器組態的一部分（如 第 28.4.1 節 「使用 YaST 設定 Samba 伺服器」 所述）。

在指令行中，使用 `systemctl stop smb nmb` 可停止 Samba 所需的服務，使用 `systemctl start nmb smb` 則可啓動這些服務。`smb` 服務會視需要處理 `winbind`。

提示：winbind

`winbind` 是一項獨立服務，同樣也是以單獨的 `samba-winbind` 套件提供。

28.4 設定 Samba 伺服器

SUSE® Linux Enterprise Server 中的 Samba 伺服器可以兩種方式設定：使用 YaST 設定或手動設定。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。

28.4.1 使用 YaST 設定 Samba 伺服器

若要設定 Samba 伺服器，請啓動 YaST 並選取網路服務 › Samba 伺服器。

28.4.1.1 初始的 Samba 組態

第一次啓動模組時，系統會啓動Samba 安裝對話方塊，提示您對伺服器管理進行一些基本設定。組態設定結束時，系統會提示您輸入 Samba 管理員密碼（Samba root 使用者密碼）。之後再啓動該模組時，會顯示Samba 組態對話方塊。

Samba 安裝對話方塊包含兩個步驟與一些選擇性的詳細設定：

工作群組或網域名稱

在 工作群組或網域名稱 中選取現有的名稱，或輸入新的名稱，並按一下 下一步。

Samba 伺服器類型

在下一步中，指定伺服器是應該做為主要網域控制器（PDC）、備份網域控制器（BDC）還是不做為任何網域控制器。按下一步繼續。

如果不想繼續設定伺服器的詳細組態，請按一下確定確認。然後，在最後一個快顯方塊中設定Samba root 使用者密碼。

之後可在Samba 組態對話方塊的啓動、共享、身分、信任的網域與LDAP 設定索引標籤中變更所有設定。

28.4.1.2 進階 Samba 組態

第一次啓動 Samba 伺服器模組時，執行第 28.4.1.1 節 「初始的 Samba 組態」 中所述的兩個初始步驟後，Samba 組態對話方塊會直接顯示。用此調整您的 Samba 伺服器組態。

編輯組態後，按一下確定儲存設定。

28.4.1.2.1 啓動伺服器

在啓動索引標籤中，設定 Samba 伺服器的啓動。若每次系統開機時都要啓動服務，請選取開機時。若要啓用手動啓動，請選擇手動。如需有關啓動 Samba 伺服器的詳細資訊，請參閱第 28.3 節 「啓動和停止 Samba」 。

在此索引標籤中，您也可以開啓您的防火牆中的連接埠。若要執行此動作，請選取在防火牆中開啓埠。如果您有多個網路介面，請按一下防火牆詳細資訊，選取介面，並按一下確定來選取 Samba 服務的網路介面。

28.4.1.2.2 共享

在共享索引標籤中，決定要啓用的 Samba 共享。標籤中有一些預先定義的共同，如 home 和 printer。使用切換狀態以切換作用中與非作用中。按一下新增可新增新的共享，按一下刪除可刪除選取的共享。

允許使用者共享自己的目錄讓許可的群組中的群組成員能與其他使用者共享自己的目錄。例如：users 針對本地範圍，DOMAIN\Users 針對網域範圍。使用者還必須確定檔案系統的許可權允許存取。請使用最多共享數限制可建立的共享總數。若要允許在沒有驗證的情況下存取使用者共享，請啓用允許訪客存取。

28.4.1.2.3 識別

在身分索引標籤中，您可以決定主機關聯的網域（基本設定），以及是否要在網路中使用替代的主機名稱（NetBIOS 主機名稱）。也可以使用 Microsoft Windows 網際網路名稱服務（WINS）進行名稱解析。在這種情況下，請啓用使用 WINS 解析主機名稱，並決定是否透過 DHCP 取回 WINS 伺服器。若要設定進階全域設定或使用者驗證來源（例如 LDAP 而非 TDB 資料庫），請按一下進階設定。

28.4.1.2.4 信任的網域

若要讓其他網域的使用者存取您的網域，請在信任的網域索引標籤中進行適當的設定。若要新增網域，請按一下新增。若要移除所選網域，請按一下移除。

28.4.1.2.5 LDAP 設定

在索引標籤LDAP 設定中，您可決定 LDAP 伺服器是否使用驗證。若要測試 LDAP 伺服器的連接，請按一下測試連接。若要查看進階 LDAP 設定或使用者預設值，請按一下進階設定。

如需 LDAP 組態的詳細資訊，請參閱《Security Guide》，第 5 章「LDAP—A Directory Service」。

28.4.2 手動設定伺服器

如果您想要使用 Samba 做為伺服器，請安裝 `samba`。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。以下預設區段包含個別檔案與印表機共享：

- [homes]
- [profiles]
- [users]
- [groups]
- [Printers]
- [print$]

透過此方法，您可以設定不同的共用選項，或在 `[global]` 區段設定全域共用選項，這使得組態檔案更容易理解。

28.4.2.1 global 區段

應該修改 `[global]` 區段的以下參數，以符合網路設定的要求，從而讓其他機器能在 Windows 環境中透過 SMB 存取 Samba 伺服器。

`workgroup = WORKGROUP`

這一行是將 Samba 伺服器指定給工作群組。請以網路環境中適當的工作群組取代 `WORKGROUP`。Samba 伺服器會以其 DNS 名稱顯示，除非此名稱已指定給網路中的其他機器。如果沒有可用的 DNS 名稱，請使用 `netbiosname=MYNAME` 設定伺服器名稱。如需更多有關此參數的詳細資料，請參閱 `smb.conf` man 頁面。

`os level = 20`

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）。為了避免現有 Windows 網路因 Samba 伺服器設定不當而中斷，應選擇非常低的值，如 `2`。如需此主題的詳細資訊，可參閱《Samba 3 Howto》的「Network Browsing」（網路瀏覽）一章；如需《Samba 3 Howto》的詳細資訊，請參閱第 28.9 節 「更多資訊」。

如果網路中沒有其他的 SMB 伺服器（例如，Windows 2000 伺服器），而且您希望 Samba 伺服器保留本地環境中存在的所有系統的清單，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，請啓用 `wins server` 選項，並將其值設為該 WINS 伺服器的 IP 位址。

如果您的各 Windows 機器連接到不同的子網路，而它們又需要看到彼此，您必須設定一部 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 伺服器啓用了這個設定值。`wins server` 與 `wins support` 選項絕不能在 `smb.conf` 檔案中同時啓用。

28.4.2.2 共享

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 光碟機開放成共享，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 共享 CD-ROM 光碟機。

範例 28.1 CD-ROM 共用

```
[cdrom]
       comment = Linux CD-ROM
       path = /media/cdrom
       locking = No
```

`[cdrom]` 與 `comment`

`[cdrom]` 區段項目是網路上所有 SMB 用戶端都可以看到之共享的名稱。可以另外再加入一個 `comment`，以進一步描述共享。

`path = /media/cdrom`

`path` 會輸出 `/media/cdrom` 目錄。

利用限制非常嚴格的預設組態，就可以將這種共享只開放給出現在此系統上的使用者共享。如果這個共享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]`

`[homes]` 共享在這裡特別重要。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，就可以連接到主目錄。

範例 28.2 [HOMES] 共享

```
[homes]
        comment = Home Directories
        valid users = %S
        browseable = No
        read only = No
        inherit acls = Yes
```

[homes]

只要沒有其他的共享，使用共享的使用者名稱連接至 SMB 伺服器，就會使用 `[homes]` 共享指示來動態產生共享。產生的共用名稱是使用者名稱。

`valid users = %S`

成功建立連接後，就會以共享的具體名稱取代 `%S`。對於 `[homes]` 共用，永遠為使用者名稱。因此，對使用者共享的存取權限僅限於該使用者。

`browseable = No`

這個設定值讓共享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享的權限。若要開放共享為可寫入的，請設定 `read only = No` 的值，這與 `writable = Yes` 同義。

`create mask = 0640`

那些不是以 MS Windows NT 為基礎的系統無法理解 Unix 權限的概念，因此它們在建立檔案時無法指定權限。`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享。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。

警告：不要與 Samba 共用 NFS 掛接

與 Samba 共用 `NFS` 掛接可能導致資料遺失，並且不支援這樣做。請直接在檔案伺服器上安裝 Samba，或者考慮使用替代方式，例如 `iSCSI`。

28.4.2.3 安全性層級

為了提高安全性，每個共享存取權都以密碼保護。SMB 提供下列幾種權限檢查方式：

使用者層級安全性（`安全性 = 使用者`）

此變體在 SMB 中引入了使用者概念。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。

ADS 層級安全性（`安全性 = ADS`）

在這種模式下，Samba 在 Active Directory 環境中以網域成員的身分執行。要以此模式作業，執行 Samba 的機器需要安裝與設定 Kerberos。您必須讓使用 Samba 的機器加入 ADS 領域。可使用 YaST 的 Windows 網域成員模組完成此操作。

網域層級安全性（`安全性 = 網域`）

僅在機器已加入到 Windows NT 網域時，此模式才會正常工作。Samba 將嘗試驗證使用者名稱與密碼，方法是將其傳送至 Windows NT 主網域或備份網域控制器。這與 Windows NT 伺服器用來驗證的方法相同。需要將加密密碼參數設定為 `yes`。

共享、使用者、伺服器或網域層級安全性的選項會套用至整部伺服器。因為無法針對伺服器組態的個別共享提供共享層級的安全性，並針對其他的共享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 3 HOWTO》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。

28.5 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。

28.5.1 使用 YaST 設定 Samba 用戶端

設定 Samba 用戶端以存取 Samba 或 Windows 伺服器上的資源（檔案或印表機）。在網路服務 › Windows 網域成員對話方塊中輸入 NT 或 Active Directory 網域或工作群組。如果您啓用了Linux 驗證也使用 SMB 資訊，則使用者驗證將會在 Samba、NT 或 Kerberos 伺服器上執行。

按一下進階設定可以指定進階組態選項。例如，使用安裝伺服器目錄表格可設定在驗證時自動掛接伺服器主目錄。這樣，使用者便可以存取其位於 CIFS 上的主目錄。如需詳細資訊，請參閱 `pam_mount` 的 man 頁面。

完成所有設定之後，在對話方塊中進行確認以完成組態設定。

28.6 做為登入伺服器的 Samba

在以 Windows 用戶端為主的網路中，通常會建議使用者只註冊一個有效的帳戶與密碼。在以 Windows 為基礎的網路中，這個任務是由主要網域控制器 (PDC) 來處理。您可以使用已設定為 PDC 的 Windows NT 伺服器，但也可借助 Samba 伺服器完成此任務。在 `smb.conf` 的 `[global]` 區段中必須編輯的項目如 範例 28.3 「在 smb.conf 中的全域區段」 所示。

範例 28.3 在 SMB.CONF 中的全域區段

```
[global]
    workgroup = WORKGROUP
    domain logons = Yes
    domain master = Yes
```

需要準備符合 Winodws 加密格式的使用者帳戶與密碼。請使用 `smbpasswd -a name` 指令來執行此動作。使用下列指令為電腦建立網域帳戶 (Windows 網域概念所需)：

```
useradd hostname\$
smbpasswd -a -m hostname
```

使用 `useradd` 指令，就會加上貨幣符號。當使用 `-m` 參數時，`smbpasswd` 指令就會自動插入這個符號。加備註的組態範例 (`/usr/share/doc/packages/Samba/examples/smb.conf.SuSE`) 包含一些設定，可讓此任務自動執行。

```
add machine script = /usr/sbin/useradd -g nogroup -c "NT Machine Account" \
-s /bin/false %m\$
```

為確保 Samba 可正確執行此程序檔，請選擇擁有所需之管理員權限的 Samba 使用者，並將其新增至 `ntadmin` 群組。這樣就可以透過下列指令將 `Domain Admin` 狀態指定給此 Linux 群組內的所有使用者：

```
net groupmap add ntgroup="Domain Admins" unixgroup=ntadmin
```

28.7 含 Active Directory 的網路中之 Samba 伺服器

若您同時執行 Linux 伺服器與 Windows 伺服器，您可建立兩個獨立的驗證系統與網路，或將兩部伺服器透過一個中央驗證系統連接到一個網路。由於 Samba 可與 Active Directory 網域共同運作，因此您可將 SUSE Linux Enterprise Server 加入 Active Directory (AD) 中。

若要加入 AD 網域，請執行下列步驟：

1. 以 `root` 身份登入並啓動 YaST。
2. 啓動網路服務 › Windows 網域成員。
3. 在Windows 領域成員畫面的領域或工作群組中輸入要加入的領域。

圖形 28.1 決定 WINDOWS 網域成員

4. 核取同時使用 SMB 資訊進行 Linux 驗證，以在伺服器上使用 SMB 來源進行 Linux 驗證。
5. 按一下確定，並在出現提示時確認要加入網域。
6. 為 Windows 管理員提供 AD 伺服器上的密碼，並按一下確定。

您的伺服器現在已可使用 Active Directory 網域控制器上的所有驗證資料。

提示：身分對應

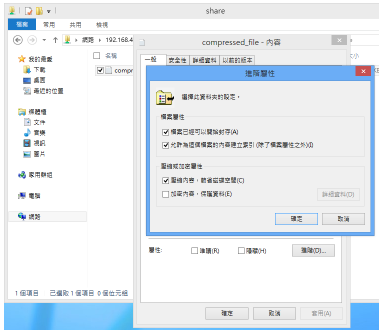
如果環境中含有多個 Samba 伺服器，則建立的 UID 和 GID 不一致。分配給使用者的 UID 與使用者首次登入的順序有關，這會導致 UID 在不同伺服器之間出現衝突。若要解決此問題，您需要使用身分對應。請參閱 https://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/idmapper.html，以取得詳細資料。

28.8 進階主題

本部分介紹用於管理 Samba 套裝軟體中用戶端部分與伺服器部分的進階方法。

28.8.1 Btrfs 上的透明檔案壓縮

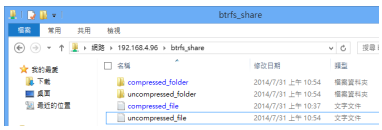
Samba 允許用戶端針對 Btrfs 檔案系統中的共用遠端操作檔案與目錄壓縮旗標。Windows 檔案總管可讓使用者透過檔案 › 內容 › 進階對話方塊來標識要進行透明壓縮的檔案/目錄：



圖形 28.2 WINDOWS 檔案總管進階屬性對話方塊

帶有壓縮旗標的檔案將以透明方式進行壓縮，當使用者存取或修改這些檔案時，基礎檔案系統會將其解壓縮。這通常可以節省儲存容量，不過，在存取檔案時會造成額外的 CPU 負擔。除非新檔案和目錄是使用 FILE_NO_COMPRESSION 選項建立的，否則，它們會繼承父目錄的壓縮旗標。

Windows 檔案總管以不同的顯示區分壓縮檔案/目錄和未壓縮檔案/目錄：



圖形 28.3 列有壓縮檔案的 WINDOWS 檔案總管目錄

啓用 Samba 共用壓縮的方法有兩種，一種是手動將以下內容

```
vfs objects = btrfs
```

新增至 `/etc/samba/smb.conf` 中的共用組態，另一種是使用 YaST：網路服務 › Samba 伺服器 › 新增，然後核取使用 Btrfs 功能。

如需 Btrfs 上壓縮功能的一般綜覽，請參閱《儲存管理指南》，第 1 章「Linux 中檔案系統的綜覽」，第 1.2.2.1 節「掛接壓縮的 Btrfs 檔案系統」。

28.8.2 快照

快照也稱為陰影副本，是指某個檔案系統子磁碟區在某個特定時間點的狀態副本。在 Linux 中，使用 Snapper 工具來管理這些快照。Btrfs 檔案系統或簡易佈建的 LVM 磁碟區支援快照。Samba 套裝軟體支援透過伺服器端和用戶端的 FSRVP 通訊協定管理遠端快照。

28.8.2.1 先前版本

Samba 伺服器上的快照可以做為檔案或目錄的先前版本公開給遠端 Windows 用戶端。若要在 Samba 伺服器上啓用快照，必須符合以下條件：

- SMB 網路共用存放在 Btrfs 子磁碟區上。
- SMB 網路共用路徑中包含相關的 snapper 組態檔案。可以使用以下指令建立 snapper 檔案

  ```
  snapper -c <cfg_name> create-config /path/to/share
  ```

如需有關 snapper 的詳細資訊，請參閱第 7 章 「使用 Snapper 進行系統復原和快照管理」。

- 必須允許相關使用者存取快照目錄樹。如需詳細資訊，請參閱 vfs_snapper 手冊頁（`man 8 vfs_snapper`）的PERMISSIONS（許可權）部分。

若要支援遠端快照，需要修改 `/etc/samba/smb.conf` 檔案。若要完成此操作，您可以選取YaST › 網路服務 › Samba 伺服器，或者使用以下指令手動增強相關的共用區段

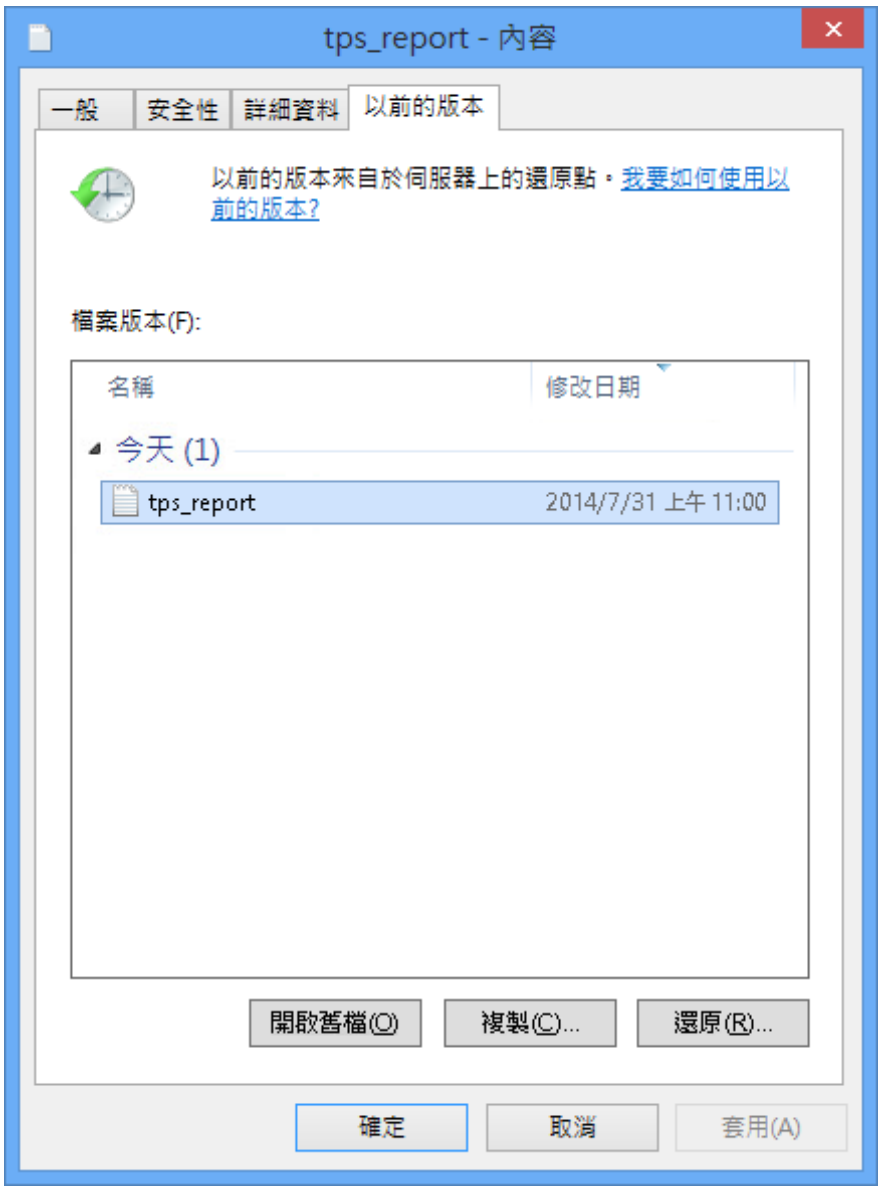
```
vfs objects = snapper
```

請注意，您需要重新啓動 Samba 服務後，對 `smb.conf` 所做的手動變更才能生效：

```
systemctl restart nmb smb
```

圖形 28.4 在啓用快照的情況下新增 SAMBA 共用

經過設定後，可以透過 Windows 檔案總管中某個檔案或目錄之以前的版本索引標籤存取由 snapper 針對 Samba 共用路徑建立的快照。



圖形 28.5 WINDOWS 檔案總管中的以前的版本索引標籤

28.8.2.2 遠端共用快照

依預設，只能在 Samba 伺服器本地透過 snapper 指令行公用程式或者使用 snapper 時間線功能來建立和刪除快照。

可將 Samba 設定為使用檔案伺服器遠端 VSS 通訊協定（File Server Remote VSS Protocol，FSRVP）來處理來自遠端主機的共用快照建立和刪除要求。

除了第 28.8.2.1 節 「先前版本」中所述的組態和先決條件以外，還需要在 `/etc/samba/smb.conf` 中設定以下全域組態：

```
[global]
rpc_daemon:fssd = fork
registry shares = yes
include = registry
```

然後，FSRVP 用戶端（包括 Samba 的 `rpcclient` 以及 Windows Server 2012 的 `DiskShadow.exe`）便可以指示 Samba 為指定的共用建立或刪除快照，並將該快照公開為新共用。

28.8.2.3 使用 `rpcclient` 從 Linux 遠端管理快照

`samba-client` 套件中有一個 FSRVP 用戶端，它可以遠端要求 Windows/Samba 伺服器建立並公開指定共用的快照。然後，您可以使用 SUSE Linux Enterprise Server 中的現有工具掛接公開的共用，並備份其檔案。向伺服器發出的要求將使用 `rpcclient` 二進位檔案傳送。

範例 28.4 使用 `rpcclient` 要求 WINDOWS SERVER 2012 共用快照

以 `EXAMPLE` 網域中管理員的身分連接到 `win-server.example.com` 伺服器：

```
# rpcclient -U 'EXAMPLE\Administrator' ncacn_np:win-server.example.com[ndr64,sign]
Enter EXAMPLE/Administrator's password:
```

檢查 SMB 共用是否對 `rpcclient` 可見：

```
rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)
```

檢查 SMB 共用是否支援建立快照：

```
rpcclient $> fss_is_path_sup windows_server_2012_share \
UNC \\WIN-SERVER\windows_server_2012_share\ supports shadow copy requests
```

要求建立共用快照：

```
rpcclient $> fss_create_expose backup ro windows_server_2012_share
```

```
13fe880e-e232-493d-87e9-402f21019fb6: shadow-copy set created
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777): \
\\WIN-SERVER\windows_server_2012_share\ shadow-copy added to set
13fe880e-e232-493d-87e9-402f21019fb6: prepare completed in 0 secs
13fe880e-e232-493d-87e9-402f21019fb6: commit completed in 1 secs
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777): \
share windows_server_2012_share@{1C26544E-8251-445F-BE89-D1E0A3938777} \
exposed as a snapshot of \\WIN-SERVER\windows_server_2012_share\
```

確認伺服器是否已公開快照共用：

```
rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)

netname: windows_server_2012_share@{1C26544E-8251-445F-BE89-D1E0A3938777}
remark: (null)
path:   \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy{F6E6507E-F537-11E3-9404-
B8AC6F927453}\Shares\windows_server_2012_share\
password:       (null)
```

嘗試刪除快照共用：

```
rpcclient $> fss_delete windows_server_2012_share \
13fe880e-e232-493d-87e9-402f21019fb6 1c26544e-8251-445f-be89-d1e0a3938777
13fe880e-e232-493d-87e9-402f21019fb6(1c26544e-8251-445f-be89-d1e0a3938777): \
\\WIN-SERVER\windows_server_2012_share\ shadow-copy deleted
```

確認伺服器是否已移除快照共用：

```
rpcclient $> netshareenum
netname: windows_server_2012_share
remark:
path:   C:\Shares\windows_server_2012_share
password:       (null)
```

28.8.2.4 使用 `DiskShadow.exe` 從 Windows 遠端管理快照

您也可以在充當用戶端的 Windows 環境中管理 Linux Samba 伺服器上 SMB 共用的快照。Windows Server 2012 提供了 `DiskShadow.exe` 公用程式，該公用程式可以如第 28.8.2.3 節 「使用 `rpcclient` 從 Linux 遠端管理快照」中所述的 `rpcclient` 那樣管理遠端共用。請注意，首先您需要妥善設定 Samba 伺服器。

以下範例程序說明了如何設定 Samba 伺服器，使 Windows Server 用戶端能夠管理其共用的快照。請注意，`EXAMPLE` 是在測試環境中使用的 Active Directory 網域，`fsrvp-server.example.com` 是 Samba 伺服器的主機名稱，`/srv/smb` 是 SMB 共享的路徑。

程序 28.1 SAMBA 伺服器組態設定詳細說明

1. 透過 YaST 加入到 Active Directory 網域。 相關資訊，請參閱第 28.7 節 「含 Active Directory 的網路中之 Samba 伺服器」。

2. 確定「使用中網域 DNS」項目正確無誤：

```
fsrvp-server:~ # net -U 'Administrator' ads dns register \
fsrvp-server.example.com <IP address>
Successfully registered hostname with DNS
```

3. 在 `/srv/smb` 位置建立 Btrfs 子磁碟區

```
fsrvp-server:~ # btrfs subvolume create /srv/smb
```

4. 為路徑 `/srv/smb` 建立 snapper 組態檔案

```
fsrvp-server:~ # snapper -c <snapper_config> create-config /srv/smb
```

5. 建立路徑為 `/srv/smb` 的新共用，並啓用 YaST 的公開快照核取方塊。確定將以下片段新增到 `/etc/samba/smb.conf` 中的 global 部分，如第 28.8.2.2 節 「遠端共用快照」中所述：

```
[global]
 rpc_daemon:fssd = fork
 registry shares = yes
 include = registry
```

6. 使用 `systemctl restart nmb smb` 重新啓動 Samba

7. 設定 snapper 許可權：

```
fsrvp-server:~ # snapper -c <snapper_config> set-config \
ALLOW_USERS="EXAMPLE\\\\Administrator EXAMPLE\\\\win-client$"
```

確定也允許任何 ALLOW_USERS 瀏覽 `.snapshots` 子目錄。

```
fsrvp-server:~ # snapper -c <snapper_config> set-config SYNC_ACL=yes
```

重要：路徑逸出

請注意「\」逸出！請逸出兩次，以確定 `/etc/snapper/configs/<snapper_config>` 中儲存的值逸出一次。

"EXAMPLE\win-client$" 對應於 Windows 用戶端電腦帳戶。對此帳戶進行驗證後，Windows 將發出初始 FSRVP 要求。

8. 授予 Windows 用戶端帳戶必要的權限：

```
fsrvp-server:~ # net -U 'Administrator' rpc rights grant \
"EXAMPLE\\win-client$" SeBackupPrivilege
Successfully granted rights.
```

不需要對 "EXAMPLE\Administrator" 使用者執行上一條指令，因為已授予該使用者權限。

程序 28.2 執行 WINDOWS 用戶端設定和 `DiskShadow.exe`

1. 開機 Windows Server 2012（範例主機名稱為 WIN-CLIENT）。
2. 就像在 SUSE Linux Enterprise Server 上一般，加入到同一個 Active Directory 網域 EXAMPLE。
3. 重新開機。
4. 開啓 Powershell。
5. 啓動 `DiskShadow.exe`，然後開始執行備份程序：

```
PS C:\Users\Administrator.EXAMPLE> diskshadow.exe
Microsoft DiskShadow version 1.0
Copyright (C) 2012 Microsoft Corporation
On computer:  WIN-CLIENT,  6/17/2014 3:53:54 PM

DISKSHADOW> begin backup
```

6. 指定每次程式離開、重設或重新開機時要保留的陰影副本：

```
DISKSHADOW> set context PERSISTENT
```

7. 檢查指定的共用是否支援快照，然後建立一個快照：

```
DISKSHADOW> add volume \\fsrvp-server\sles_snapper

DISKSHADOW> create
Alias VSS_SHADOW_1 for shadow ID {de4ddca4-4978-4805-8776-cdf82d190a4a} set as \
 environment variable.
Alias VSS_SHADOW_SET for shadow set ID {c58e1452-c554-400e-a266-d11d5c837cb1} \
 set as environment variable.

Querying all shadow copies with the shadow copy set ID \
 {c58e1452-c554-400e-a266-d11d5c837cb1}

 * Shadow copy ID = {de4ddca4-4978-4805-8776-cdf82d190a4a}     %VSS_SHADOW_1%
    - Shadow copy set: {c58e1452-c554-400e-a266-d11d5c837cb1}  %VSS_SHADOW_SET%
    - Original count of shadow copies = 1
    - Original volume name: \\FSRVP-SERVER\SLES_SNAPPER\ \
      [volume not on this machine]
    - Creation time: 6/17/2014 3:54:43 PM
    - Shadow copy device name:
      \\FSRVP-SERVER\SLES_SNAPPER@{31afd84a-44a7-41be-b9b0-751898756faa}
    - Originating machine: FSRVP-SERVER
    - Service machine: win-client.example.com
    - Not exposed
    - Provider ID: {89300202-3cec-4981-9171-19f59559e0f2}
    - Attributes:  No_Auto_Release Persistent FileShare

Number of shadow copies listed: 1
```

8. 完成備份程序：

```
DISKSHADOW> end backup
```

9. 建立快照後，嘗試將它刪除，並驗證刪除結果：

```
DISKSHADOW> delete shadows volume \\FSRVP-SERVER\SLES_SNAPPER\
Deleting shadow copy {de4ddca4-4978-4805-8776-cdf82d190a4a} on volume \
 \\FSRVP-SERVER\SLES_SNAPPER\ from provider \
{89300202-3cec-4981-9171-19f59559e0f2} [Attributes: 0x04000009]...

Number of shadow copies deleted: 1

DISKSHADOW> list shadows all

Querying all shadow copies on the computer ...
No shadow copies found in system.
```

28.9 更多資訊

Samba 文件包含在 `samba-doc` 套件中，預設不會安裝該套件。您可以使用 `zypper install samba-doc` 來安裝。在指令行中輸入 `apropos samba` 可顯示一些手冊頁，或者流覽 `/usr/share/doc/packages/samba` 目錄可取得更多的線上文件与範例。`examples` 子目錄中提供了一個帶有備註的範例組態（`smb.conf.SUSE`）。另一個可以查看 Samba 相關資訊的檔案是 `/usr/share/doc/packages/samba/README.SUSE`。

由 Samba 團隊提供的《Samba HOWTO》（請參閱 https://wiki.samba.org）包含了疑難排解一節。除此之外，文件的第五部份提供檢查組態的逐步指南。

29 使用 Autofs 按需掛接

autofs 是一個程式，可以根據需要掛接指定的目錄。它以核心模組為基礎，效率很高，並且可以管理本地目錄和網路共用。這些自動掛接點僅在存取時掛接，一段時間不使用後即會卸載。這種按需行為可節省頻寬，在效能上優於 /etc/fstab 管理的靜態掛接。雖然 autofs 是一個控制程序檔，但是 automount 才是執行實際自動掛接的指令（精靈）。

29.1 安裝

SUSE Linux Enterprise Server 上預設未安裝 autofs。若要使用它的自動掛接功能，請先使用以下指令進行安裝

```
sudo zypper install autofs
```

29.2 組態

您需要使用 vim 之類的文字編輯器編輯 autofs 的組態檔案，來手動設定該工具。設定 autofs 有兩個基本步驟 — master 映射檔案與特定映射檔案。

29.2.1 Master 映射檔案

autofs 的預設 master 映射檔案是 /etc/auto.master。若要變更其位置，可以在 /etc/sysconfig/autofs 檔案中變更 DEFAULT_MASTER_MAP_NAME 選項的值。以下是 SUSE Linux Enterprise Server 中預設 master 映射檔案的內容：

```
#
# Sample auto.master file
# This is an automounter map and it has the following format
# key [ -mount-options-separated-by-comma ] location
# For details of the format look at autofs(5).1
#
#/misc  /etc/auto.misc2
```

```
#/net -hosts
#
# Include /etc/auto.master.d/*.autofs❸
#
#+dir:/etc/auto.master.d
#
# Include central master map if it can be found using
# nsswitch sources.
#
# Note that if there are entries for /net or /misc (as
# above) in the included master map any keys that are the
# same will not be seen as the first read key seen takes
# precedence.
#
+auto.master❹
```

❶ `autofs` 手冊頁（`man 5 autofs`）提供了許多有關自動掛載器映射的重要資訊。

❷ 雖然依預設透過 # 列為注釋，但是這是簡單的自動掛載器映射語法的範例。

❸ 如果您需要將 master 映射拆分為幾個檔案，請將該行取消注釋標記，並將映射（字尾為 `.autofs`）置於 `/etc/auto.master.d/` 目錄中。

❹ `+auto.master` 可確保 NIS（請參閱《Security Guide》，第 3 章「Using NIS」，第 3.1 節「Configuring NIS Servers」瞭解 NIS 詳細資訊）的使用者仍可找到其 master 映射。

`auto.master` 中的項目有三個欄位，語法如下：

```
mount point      map name      options
```

掛接點
: 用於掛接 `autofs` 檔案系統的基本位置，例如 `/home`。

映射名稱
: 要用於掛接之映射來源的名稱。如需映射檔案的語法，請參閱第 29.2.2 節 「映射檔案」。

選項
: 這些選項（若指定）依預設將套用至給定映射中的所有項目。

提示：更多資訊

如需選用 **映射類型**、**格式** 和 **選項** 之特定值的詳細資訊，請參閱 auto.master 手冊頁（`man 5 auto.master`）。

`auto.master` 中的下列項目可指示 `autofs` 在 `/etc/auto.smb` 中搜尋，並在 `/smb` 目錄中建立掛接點。

```
/smb    /etc/auto.smb
```

29.2.1.1 直接掛接

直接掛接可在相關映射檔案內指定的路徑中建立掛接點。它並不在 `auto.master` 中指定掛接點，而是以 `/-` 取代掛接點欄位。例如，下面一行指示 `autofs` 在 `auto.smb` 中所指定的位置上建立掛接點：

```
/-        /etc/auto.smb
```

提示：不含完整路徑的映射

如果未以映射檔案的完整本地或網路路徑指定映射檔案，則會使用名稱服務開關（NSS）組態尋找映射檔案。

```
/-        auto.smb
```

29.2.2 映射檔案

重要：其他映射類型

雖然檔案是使用 `autofs` 自動掛接之映射的最常見類型，但是還有其他一些類型。映射指定可以是指令的輸出，也可以是 LDAP 或資料庫中查詢的結果。如需映射類型的詳細資訊，請參閱手冊頁 `man 5 auto.master`。

映射檔案指定（本地或網路）來源位置，掛接點則指出在本地將來源掛接在何處。映射的一般格式與 master 映射相似。區別在於選項出現在掛接點與位置之間，而不是項目的末尾：

```
mount point      options      location
```

確定對應檔案未標示為可執行檔。可透過執行 `chmod -x MAP_FILE` 移除可執行檔位元。

mount point
: 指定將來源掛接在何處。這可以是要新增至 `auto.master` 中所指定基礎掛接點的單個目錄名稱（亦稱為**間接**掛接），也可以是掛接點的完整路徑（直接掛接，請參閱第 29.2.1.1 節 「直接掛接」）。

options
: 為相關項目指定選用的掛接點清單，內容以逗號分隔。如果 `auto.master` 還包含此映射檔案的選項，則會附加這些選項。

location
: 指定要掛接的檔案系統的來源，通常是 NFS 或 SMB 磁區，一般表示為**主機名稱:路徑名稱**。如果要掛接的檔案系統以「/」開頭（例如本地 `/dev` 項目或 smbfs 共用），需要前置冒號符號「:」，例如 `:/dev/sda1`。

29.3 操作與除錯

本節介紹如何控制 `autofs` 服務操作，以及如何在調整自動掛載器操作時檢視更多除錯資訊。

29.3.1 控制 autofs 服務

`autofs` 服務的操作由 `systemd` 控制。對 `autofs` 而言，`systemctl` 指令的一般語法是

```
sudo systemctl SUB_COMMAND autofs
```

其中，`SUB_COMMAND` 是下列項目之一：

enable
: 在開機時啓動自動掛載器精靈。

start
: 啓動自動掛載器精靈。

stop
: 停止自動掛載器精靈。自動掛接點不可存取。

status
: 列印 `autofs` 服務的目前狀態以及部分相關記錄檔。

restart
: 停止然後啓動自動掛載器，以便終止所有執行中的精靈，然後再啓動新的精靈。

reload
: 檢查目前的 `auto.master` 映射，重新啓動項目已變更的精靈，然後啓動新項目的新精靈。

29.3.2 自動掛載器問題除錯

如果您在使用 `autofs` 掛接目錄時遇到問題，實用的方法是手動執行 `automount` 精靈並觀看其輸出訊息：

1. 停止 `autofs`。

   ```
   sudo systemctl stop autofs
   ```

2. 從一個終端機在前景手動執行 `automount`，以產生詳細輸出。

   ```
   sudo automount -f -v
   ```

3. 從另一個終端機嘗試透過存取掛接點（例如，透過 `cd` 或 `ls`）掛接自動掛接的檔案系統。

4. 從第一個終端機檢查 `automount` 的輸出，以瞭解有關掛接為何失敗或者甚至為何未嘗試掛接的更多資訊。

29.4 自動掛接 NFS 共用

下面的程序說明如何設定 `autofs` 以自動掛接網路上可用的 NFS 共用。該程序要用到上文中的資訊，並假設您熟悉 NFS 輸出。如需 NFS 詳細資訊，請參閱第 27 章 「使用 NFS 共享檔案系統」。

1. 編輯映射檔案 `/etc/auto.master`：

```
sudo vim /etc/auto.master
```

在 `/etc/auto.master` 末尾為新的 NFS 掛接新增項目：

```
/nfs      /etc/auto.nfs      --timeout=10
```

此指令指示 `autofs` 基本掛接點是 `/nfs`，NFS 共用在 `/etc/auto.nfs` 映射中指定，並且此映射中的所有共用如果在 10 秒鐘內未曾使用，則自動卸載。

2. 為 NFS 共用建立新的映射檔案：

```
sudo vim /etc/auto.nfs
```

在 `/etc/auto.nfs` 中，通常每個 NFS 共用對應單獨一行內容，第 29.2.2 節「映射檔案」中包含其格式的詳細描述。新增一行，說明掛接點及 NFS 共用網路位址：

```
export      jupiter.com:/home/geeko/doc/export
```

上面一行表示系統會應要求將 `jupiter.com` 主機上的 `/home/geeko/doc/export` 目錄自動掛接到本地主機上的 `/nfs/export` 目錄（`/nfs` 取自 `auto.master` 映射）。`/nfs/export` 目錄將由 `autofs` 自動建立。

3. 如果您先前靜態掛接了相同的 NFS 共用，則可以選擇性地將 `/etc/fstab` 中相關的行列為注釋。該行應類似於：

```
#jupiter.com:/home/geeko/doc/export /nfs/export nfs defaults 0 0
```

4. 重新載入 `autofs` 並檢查它是否正常運作：

```
sudo systemctl restart autofs
```

```
# ls -l /nfs/export
total 20
drwxr-xr-x  6 1001 users 4096 Oct 25 08:56 ./
drwxr-xr-x  3 root root     0 Apr  1 09:47 ../
drwxr-xr-x  5 1001 users 4096 Jan 14  2013 .images/
drwxr-xr-x 10 1001 users 4096 Aug 16  2013 .profiled/
drwxr-xr-x  3 1001 users 4096 Aug 30  2013 .tmp/
drwxr-xr-x  4 1001 users 4096 Oct 25 08:56 SLE-12-manual/
```

如果您能看到遠端共用上的檔案清單，則表示 `autofs` 正常工作。

29.5 進階主題

本節描述了 `autofs` 基本介紹以外的主題 — 自動掛接網路上可用的 NFS 共用，在映射檔案中使用萬用字元以及有關 CIFS 檔案系統特定的資訊。

29.5.1 `/net` 掛接點

如果您使用許多 NFS 共用，則這個掛接點非常有用。`/net` 會根據需要自動掛接本地網路上的所有 NFS 共用。該項目已存在於 `auto.master` 檔案中，因此，您只需取消其注釋標記並重新啓動 `autofs` 即可：

```
/net      -hosts
```

```
systemctl restart autofs
```

例如，如果您有個名為 `jupiter` 的伺服器以及名為 `/export` 的 NFS 共用，您可以在指令行上鍵入下列指令進行掛接：

```
# cd /net/jupiter/export
```

29.5.2 使用萬用字元自動掛接子目錄

如果您的某個目錄含有多個子目錄，並且您需要將這些子目錄個別自動掛接（一般情況下，該目錄是包含各個使用者主目錄的 `/home` 目錄），`autofs` 提供了便捷的解決方案。

對於主目錄，請在 `auto.master` 中新增下面一行：

```
/home      /etc/auto.home
```

現在，您需要將正確的映射新增至 `/etc/auto.home` 檔案，以便自動掛接使用者的主目錄。有個解決方案是為每個目錄建立獨立的項目：

```
wilber      jupiter.com:/home/wilber
penguin      jupiter.com:/home/penguin
tux      jupiter.com:/home/tux
[...]
```

這種方法非常麻煩，因為您需要在 `auto.home` 中管理使用者清單。您可以使用星號「*」而不是掛接點，使用和號「&」而不是要掛接的目錄。

```
*      jupiter:/home/&
```

29.5.3 自動掛接 CIFS 檔案系統

如果您要自動掛接 SMB/CIFS 共用（有關 SMB/CIFS 協定的詳細資訊，請參閱第 28 章「Samba」），則需要修改映射檔案的語法。在選項欄位中新增 `-fstype=cifs`，並在共用位置前面加上冒號「:」。

```
mount point      -fstype=cifs      ://jupiter.com/export
```

30 SLP

要想設定網路用戶端，需要深入瞭解透過網路提供的服務（例如列印或 LDAP）。為了簡化在網路用戶端上對此類服務的設定，「服務位置通訊協定」（SLP）應運而生。SLP 可讓區域網路中的所有用戶端都知道選定服務的可用性和組態資料。支援 SLP 的應用程式可以使用此資訊，以便自動設定。

SUSE® Linux Enterprise Server 支援使用 SLP 所提供的安裝來源進行安裝，並包含許多具有 SLP 整合支援的系統服務。您可以使用 SLP 以提供主要的功能給網路上的用戶端，例如系統上的安裝伺服器、檔案伺服器或是列印伺服器。提供 SLP 支援的服務包括 cupsd、login、ntp、openldap2、postfix、rpasswd、rsyncd、saned、sshd（透過 fish）、vnc 和 ypserv。

依預設，系統會安裝在網路用戶端上使用 SLP 服務所需的所有套件。但是，如果要透過 SLP 提供服務，請檢查是否已安裝 `openslp-server` 套件。

30.1 SLP 前端 `slptool`

`slptool` 是一個指令行工具，用於查詢和註冊 SLP 服務。查詢功能對診斷非常有用。下面列出了最重要的 `slptool` 子指令。`slptool --help` 會列出所有可用的選項與功能。

findsrvtypes
: 列出網路上可用的所有服務類型。

```
tux >  slptool findsrvtypes
service:install.suse:nfs
service:install.suse:ftp
service:install.suse:http
service:install.suse:smb
service:ssh
service:fish
service:YaST.installation.suse:vnc
service:smtp
service:domain
service:management-software.IBM:hardware-management-console
service:rsync
service:ntp
service:ypserv
```

findsrvs SERVICE_TYPE

列出提供 SERVICE_TYPE 的所有伺服器

```
tux >  slptool findsrvs service:ntp
service:ntp://ntp.example.com:123,57810
service:ntp://ntp2.example.com:123,57810
```

findattrs SERVICE_TYPE//HOST

列出 HOST 上 SERVICE_TYPE 的屬性

```
tux >  slptool findattrs service:ntp://ntp.example.com
(owner=tux),(email=tux@example.com)
```

register SERVICE type//HOST:PORT "(ATTRIBUTE=VALUE),(ATTRIBUTE=VALUE)"

在 HOST 上使用選擇性屬性清單註冊 SERVICE_TYPE

```
slptool register service:ntp://ntp.example.com:57810 \
"(owner=tux),(email=tux@example.com)"
```

deregister SERVICE_TYPE//host

在 HOST 上取消註冊 SERVICE_TYPE

```
slptool deregister service:ntp://ntp.example.com
```

如需更多資訊，請執行 `slptool --help`。

30.2 透過 SLP 提供服務

若要提供 SLP 服務，SLP 精靈（`slpd`）必須正在執行。就像 SUSE Linux Enterprise Server 中的大多數系統服務一樣，`slpd` 透過單獨的啓動程序檔來控制。安裝完成後，依預設此精靈將處於非啓動狀態。若要為目前的工作階段啓用該精靈，請執行 `sudo systemctl start slpd`。如果 `slpd` 應在系統啓動時啓動，請執行 `sudo systemctl enable slpd`。

SUSE Linux Enterprise Server 中的許多應用程式透過 `libslp` 程式庫已經具有整合的 SLP 支援。如果尚未使用 SLP 支援來編譯服務，請使用下列方式之一來透過 SLP 進行編譯：

使用 `/etc/slp.reg.d` 的靜態註冊

針對每個新的服務建立個別的註冊檔。下面的範例會註冊一個掃描器服務：

```
## Register a saned service on this system
## en means english language
## 65535 disables the timeout, so the service registration does
## not need refreshes
service:scanner.sane://$HOSTNAME:6566,en,65535
watch-port-tcp=6566
description=SANE scanner daemon
```

在此檔案中最重要的一行為 service URL，它是以 `service:` 開始。這包含服務類型（`scanner.sane`）以及位址（可在其中找到伺服器可用的服務）。`$HOSTNAME` 會以完整的主機名稱自動取代。接著可以找到相關服務的 TCP 埠名稱，它們之間是以冒號分隔。然後輸入服務應該出現的語言以及註冊期間的秒數。這些都應該使用逗號與服務 URL 分隔。在 `0` 與 `65535` 之間設定註冊期間的值。`0` 會防止註冊。`65535` 會移除所有的限制。

註冊檔另外還包含 `watch-port-tcp` 與 `description` 兩個變數。`watch-port-tcp` 將 SLP 服務宣告連結至由 `slpd` 檢查到的服務狀態，即相關服務是否啓用。第二個變數是對顯示在適當瀏覽器中的服務，提供更為精確的描述。

提示：YaST 與 SLP

當您在模組對話方塊中啓動 SLP 時，一些由 YaST 所仲介的服務（例如安裝伺服器或 YOU 伺服器）會自動執行此註冊。接著 YaST 就會為這些服務建立註冊檔。

使用 `/etc/slp.reg` 的靜態註冊

此方法與使用 `/etc/slp.reg.d` 的程序的唯一區別在於所有服務都會在中心檔案內分組。

使用 `slptool` 的動態註冊

如果要動態註冊服務而不使用組態檔案，請使用 slptool 指令行公用程式。這一公用程式還可用於取消註冊現有服務，而無需重新啓動 `slpd`。如需詳細資料，請參閱第 30.1 節 「SLP 前端 `slptool`」。

30.2.1 設定 SLP 安裝伺服器

在網路中透過 SLP 宣告安裝資料可簡化網路安裝，因為透過 SLP 查詢會自動要求提供伺服器 IP 位址或安裝媒體路徑等安裝資料。如需指示，請參閱《部署指南》，第 8 章「安裝保存安裝來源的伺服器」。

30.3 更多資訊

RFC 2608、2609、2610

RFC 2608 一般會處理 SLP 的定義。RFC 2609 會處理更加詳細的服務 URL 語法，而 RFC 2610 則會透過 SLP 處理 DHCP。

http://www.openslp.org

OpenSLP 計劃的首頁。

`/usr/share/doc/packages/openslp`

此目錄包含 `openslp-server` 套件隨附的 SLP 文件，包括 `README.SuSE`（含有 SUSE Linux Enterprise Server 詳細資料）、RFC 以及兩個介紹性的 HTML 文件。要使用 SLP 功能的程式設計師可參閱 SUSE 軟體開發套件隨附的 `openslp-devel` 套件中的《Programmers Guide》（程式設計指南），以瞭解詳細資訊。

31 Apache HTTP 伺服器

根據 http://www.netcraft.com/ 的調查表明，Apache HTTP 伺服器 (Apache) 是世界上使用最廣泛的 Web 伺服器。它由 Apache 軟體基金會 (http://www.apache.org/) 研發，可在大部分作業系統上使用。SUSE® Linux Enterprise Server 隨附 Apache 2.4 版本。本章將介紹如何安裝、組態設定與設定 Web 伺服器，如何使用 SSL、CGI 與其他模組，以及如何排解 Apache 疑難。

31.1 快速入門

本節中的說明可協助您快速設定和啓動 Apache。您必須登入為 `root` 身分，才能安裝和設定 Apache。

31.1.1 要求

在設定 Apache Web 伺服器之前，請先確定您已符合下列要求：

1. 此機器的網路已正確設定。如需有關這個主題的詳細資訊，請參閱 第 16 章 「基本網路功能」。
2. 此機器的實際系統時間已透過時間伺服器進行同步維護。這是必要動作，因為 HTTP 通訊協定的部分內容會依據正確時間來運作。如需有關這個主題的詳細資訊，請參閱第 24 章 「使用 NTP 進行時間同步化」。
3. 已安裝最新的安全性更新。如果不清楚是否已安裝，請執行「YaST 線上更新」。
4. 防火牆中已開啓預設的 Web 伺服器連接埠 (`80`)。針對這點，請將 SUSEFirewall2 設定成允許在外部區域執行 HTTP 伺服器服務。您可以使用 YaST 完成此設定。如需詳細資料，請參閱 《Security Guide》，第 15 章 「Masquerading and Firewalls」，第 15.4.1 節 「Configuring the Firewall with YaST」。

31.1.2 安裝

SUSE Linux Enterprise Server 中的 Apache 預設不會安裝到系統。若要以可「直接」執行的預先定義標準組態進行安裝，請按照以下步驟操作：

程序 31.1 以預設組態安裝 APACHE

1. 啓動 YaST，然後選取軟體 › 軟體管理。
2. 選擇 檢視 › 模式 ，然後選取 Web 與 LAMP 伺服器。
3. 請確認安裝個別套件，以完成此安裝程序。

31.1.3 開始

您可讓 Apache 在開機時自動啓動或者手動將其啓動。

為確定 Apache 會在開機期間於目標 `multi-user.target` 與 `graphical.target` 中自動啓動，請執行以下指令：

```
root # systemctl enable apache2
```

如需有關 SUSE Linux Enterprise Server 中 systemd 目標的詳細資訊，以及 YaST 服務管理員的說明，請參閱第 13.4 節 「使用 YaST 管理 服務」。

若要使用外圍程序手動啓動 Apache，請執行 `systemctl start apache2`。

程序 31.2 檢查 APACHE 是否正在執行

如果您在啓動 Apache 時未接收到錯誤訊息，這通常表示 Web 伺服器正在執行。若要對此進行測試：

1. 啓動瀏覽器，並開啓 http://localhost/。
   如果 Apache 已啓動且正在執行，系統會顯示「正常運作！」的測試頁面。
2. 如果此頁面沒有出現，請參閱第 31.9 節 「疑難排解」。

現在網頁伺服器已經開始執行，您可以加入自己的文件、根據個人需求調整組態，或是安裝模組來新增功能。

31.2 設定 Apache

SUSE Linux Enterprise Server 提供了兩個組態選項：

- 手動設定 Apache
- 使用 YaST 設定 Apache

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：在組態變更後重新載入或重新啓動 Apache

大多數組態變更需要重新載入（有些還需要重新啓動）Apache 才能生效。請使用 `systemctl reload apache2` 以手動方式重新載入 Apache，或使用第 31.3 節「啓動和停止 Apache」中所述的其中一個重新啓動選項。

如果使用 YaST 來設定 Apache，依照第 31.2.3.2 節 「HTTP 伺服器組態」中所述將HTTP 服務設定為已啓用，即可讓上述操作自動完成。

31.2.1 Apache 組態檔案

本節提供 Apache 組態檔案的綜覽。如果使用 YaST 設定組態，則無需變更這些檔案。不過，如果您要在以後改為以手動方式設定組態，該資訊可能會對您有用。

您可以在下列兩個不同位置找到 Apache 組態檔案：

- `/etc/sysconfig/apache2`
- `/etc/apache2/`

31.2.1.1 `/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、伺服器應該啓動的旗標，以及應該新增至指令行的旗標。此檔案對每個組態選項都進行了詳細說明，因此本文不予以介紹。針對一般用途的網頁伺服器，在 `/etc/sysconfig/apache2` 中的設定應該可以符合任何組態需求。

31.2.1.2 /etc/apache2/

`/etc/apache2/` 代管了 Apache 的所有組態檔案。以下各節將說明每個檔案的用途。每個檔案都包括數個組態選項（也稱指令）。在這些檔案中的每個組態選項都會詳加說明，因此本文將不予以介紹。

Apache 組態檔案的組織方式如下：

```
/etc/apache2/
     |
     |- charset.conv
     |- conf.d/
     |   |
     |   |- *.conf
     |
     |- default-server.conf
     |- errors.conf
     |- httpd.conf
     |- listen.conf
     |- magic
     |- mime.types
     |- mod_*.conf
     |- server-tuning.conf
     |- ssl.*
     |- ssl-global.conf
     |- sysconfig.d
     |   |
     |   |- global.conf
     |   |- include.conf
     |   |- loadmodule.conf . .
     |
     |- uid.conf
     |- vhosts.d
     |   |- *.conf
```

在 /ETC/APACHE2/ 中的 APACHE 組態檔案

`charset.conv`

指定不同語言所要使用的字元集。請不要編輯此檔案。

`conf.d/*.conf`

組態檔案由其他模組新增。這些組態檔案可以依實際需要包含至虛擬主機組態。如需範例，請參閱 `vhosts.d/vhost.template`。若要執行這個動作，您可以為不同的虛擬主機提供不同的模組組合。

`default-server.conf`

使用合理預設值設定所有虛擬主機的全域組態。這時不是變更組態值，而是採用虛擬主機組態覆寫組態值。

`errors.conf`

定義 Apache 處理錯誤的方式。若要自訂這些傳送給所有虛擬主機的訊息，請編輯此檔案。另外一種方法是覆寫虛擬主機組態中的這些指示詞。

`httpd.conf`

主要的 Apache 伺服器組態檔案。請勿變更此檔案。此檔案主要包含 Include 陳述式和全域設定。為此處列出的相關組態檔案覆寫全域設定。變更虛擬主機組態的主機特定設定（例如文件根目錄）。

`listen.conf`

繫結 Apache 至特定的 IP 位址與連接埠在此檔案中也可以設定以名稱為基礎的虛擬主機。如需詳細資料，請參閱第 31.2.2.1.1 節 「以名稱為基礎的虛擬主機」。

`magic`

mime_magic 模組的資料，此模組可協助 Apache 自動判斷不明檔案的 MIME 類型。請不要變更此檔案。

`mime.types`

系統已知的 MIME 類型（實際上是 `/etc/mime.types` 的連結）。請不要編輯此檔案。如果您需要新增這裡未列出的 MIME 類型，請將它們新增到 `mod_mime-defaults.conf`。

`mod_*.conf`

預設已安裝之模組的組態檔案。如需詳細資訊，請參閱第 31.4 節 「安裝、啓用和設定模組」。請注意，選用模組的組態檔案會存放在 `conf.d` 目錄。

`server-tuning.conf`

包含不同 MPM 的組態指令（請參閱第 31.4.4 節 「多重處理模組」）和可控制 Apache 效能的一般組態選項。請在變更此檔案之後為網頁伺服器進行適當測試。

`ssl-global.conf` 和 `ssl.*`

全域 SSL 組態和 SSL 證書資料。如需詳細資訊，請參閱第 31.6 節 「設定提供 SSL 的安全網頁伺服器」。

`sysconfig.d/*.conf`

自動從 `/etc/sysconfig/apache2` 產生的組態檔案。請勿改變其中任何檔案 -- 而是編輯 `/etc/sysconfig/apache2`。請不要將其他組態檔案置於此目錄中。

`uid.conf`

指定要在哪個使用者和群組 ID 之下執行 Apache。請不要變更此檔案。

`vhosts.d/*.conf`

您的虛擬主機組態應存放於此處。該目錄包含採用或不採用 SSL 之虛擬主機的樣板檔案。此目錄中以 `.conf` 結尾的每個檔案都會自動包含在 Apache 組態中。如需詳細資訊，請參閱第 31.2.2.1 節 「虛擬主機組態」。

31.2.2 手動設定 Apache

以手動方式設定 Apache 是指以 `root` 使用者身分來編輯純文字組態檔案。

31.2.2.1 虛擬主機組態

「虛擬主機」一詞，是形容 Apache 從同一部實體機器提供多個通用資源識別字串 (URI，Universal Resource Identifier) 的能力。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) 和硬體開銷 (不需要將各個網域安裝在專屬伺服器上)。虛擬主機可以使用名稱、IP 或是連接埠作為基礎。

若要列出所有現有的虛擬主機，請使用指令 `apache2ctl -S`。該指令將輸出一份清單，顯示預設伺服器和所有虛擬主機，以及它們的 IP 位址和監聽埠。此外，該清單還顯示每個虛擬主機在組態檔案中的位置。

虛擬主機可依照第 31.2.3.1.4 節 「虛擬主機」中所述透過 YaST 設定，或是手動編輯組態檔案來設定。依預設，系統會根據 /etc/apache2/vhosts.d/ 中每部虛擬主機一個組態檔案的設定，為在 SUSE Linux Enterprise Server 中執行的 Apache 做好準備。此目錄中副檔名為 `.conf` 的所有檔案，都會自動包含至組態中。這個目錄會提供虛擬主機的基本樣板 (`vhost.template`，或是適用於提供 SSL 支援之虛擬主機的 `vhost-ssl.template`)。

提示：永遠要建立虛擬主機組態

建議您務必要建立虛擬主機組態檔案，即使網頁伺服器只代管一個網域。如此，您不但可以將網域專屬組態存放在一個檔案中，還可以隨時回復至運作正常的基本組態，只需移動、刪除或重新命名虛擬主機的組態檔案即可。同樣地，您應該也要分別為每個虛擬主機建立組態。

使用名稱型虛擬主機時，建議設定預設組態，以便在網域名稱與虛擬主機組態不相符時使用。系統會首先載入預設虛擬主機的組態。由於組態檔案的順序由檔案名稱決定，因此請在預設虛擬主機組態檔案名稱的開頭使用底線字元（_），以確定讓該檔案最先載入（例如：`_default_vhost.conf`）。

`<VirtualHost> </VirtualHost>` 區塊包含要套用到特定網域的資訊。當 Apache 接收到來自訂義的虛擬主機的用戶端要求時，就會使用本節所包含的指示詞。幾乎所有指示詞都可以用於虛擬主機網路位置。如需更多有關 Apache 組態指示詞的詳細資訊，請參閱http://httpd.apache.org/docs/2.4/mod/quickreference.html。

31.2.2.1.1 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 位址可以為數個網站提供服務。Apache 會使用用戶端所傳送之 HTTP 標頭中的主機欄位，將要求連接到與其中一個虛擬主機宣告相符的 `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虛擬主機。

第一步是為要提供服務的每一個不同的名稱型主機建立 `<VirtualHost>` 區段。在每個 `<VirtualHost>` 區塊內部，您至少需要一個 `ServerName` 指令來指定要為哪個主機提供服務，並需要一個 `DocumentRoot` 指令來顯示該主機的內容位於檔案系統中的哪個位置。

範例 31.1 名稱型 `VirtualHost` 項目的基本範例

```
<VirtualHost *:80>
# This first-listed virtual host is also the default for *:80
ServerName www.example.com
ServerAlias example.com
DocumentRoot /srv/www/htdocs/domain
</VirtualHost>

<VirtualHost *:80>
ServerName other.example.com
DocumentRoot /srv/www/htdocs/otherdomain
```

```
</VirtualHost>
```

開啓的 `VirtualHost` 標籤會將 IP 位址（或完全合格的網域名稱）當作名稱型虛擬主機組態的引數。連結埠號碼指令是選填項目。

允許使用萬用字元 * 做為 IP 位址的替代符號。如果是使用 IPv6 位址，該位址就必須用方括號包住。

範例 31.2 以名稱為基礎的 VirtualHost 指示詞

```
<VirtualHost 192.168.3.100:80>
  ...
</VirtualHost>

<VirtualHost 192.168.3.100>
  ...
</VirtualHost>

<VirtualHost *:80>
  ...
</VirtualHost>

<VirtualHost *>
  ...
</VirtualHost>

<VirtualHost [2002:c0a8:364::]>
  ...
</VirtualHost>
```

31.2.2.1.2 以 IP 為基礎的虛擬主機

此虛擬主機組態替代方法需要為一台機器設定多個 IP。一個 Apache 例項可裝載多個網域，每個網域都會指定不同的 IP。

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）。

下列範例將示範，Apache 正執行於 IP `192.168.3.100` 的電腦上，並負責代管 IP `192.168.3.101` 與 `192.168.3.102` 兩個領域。每部虛擬伺服器都必須具備個別的 `VirtualHost` 區塊。

範例 31.3 以 IP 為基礎的 VirtualHost 指示詞

```
<VirtualHost 192.168.3.101>
  ...
</VirtualHost>
```

```
<VirtualHost 192.168.3.102>
  ...
</VirtualHost>
```

這裡出淚的 `VirtualHost` 指示詞，只能指定給 `192.168.3.100` 以外的其他介面。如果 `192.168.3.100` 也有設定 `Listen` 指示詞，這時就必須建立另一個以 IP 為基礎的虛擬主機，來回應 HTTP 要求給該介面 -- 另一種做法是套用預設伺服器組態（`/etc/apache2/default-server.conf`）所顯示的指示詞。

31.2.2.1.3 基本虛擬主機組態

每個虛擬主機組態中至少要有下列指令，才能設定虛擬主機。如需瞭解更多選項的詳細資訊，請參閱 `/etc/apache2/vhosts.d/vhost.template`。

`ServerName`
: 完全合格網域名稱，其下是應該要建立位址的主機。

`DocumentRoot`
: 目錄路徑，Apache 會從此路徑為此主機提供檔案。基於安全性考量，存取整個檔案系統是預設禁止的動作，所以您必須明確解除鎖定這個位在 `Directory` 容器中的目錄。

`ServerAdmin`
: 伺服器管理員的電子郵件地址．這個地址可顯示在 Apache 建立的錯誤頁面（舉例說明）。

`ErrorLog`
: 此虛擬主機的錯誤記錄檔案。雖然沒必要為每個虛擬主機分別建立錯誤記錄檔案，但是多數人會這樣做，以便除錯更加容易。`/var/log/apache2/` 是 Apache 記錄檔案的預設目錄。

`CustomLog`
: 此虛擬主機的存取記錄檔案。雖然沒必要為每個虛擬主機分別建立存取記錄檔案，但是多數人會這樣做，以便分別為每部主機分析存取統計資料。`/var/log/apache2/` 是 Apache 記錄檔案的預設目錄。

正如前面所述，存取整個檔案系統已因安全性考量而預設為禁止動作。因此，請將 Apache 要處理之檔案所在的目錄明確解除鎖定 — 例如 `DocumentRoot`。

```
<Directory "/srv/www/www.example.com/htdocs">
  Require all granted
</Directory>
```

注意：Require all granted

在先前版本的 Apache 中，陳述式 `Require all granted` 表達為：

```
Order allow,deny
Allow from all
```

`mod_access_compat` 模組仍然支援這種舊式語法。

此完整組態看起來如下：

範例 31.4 基本 VirtualHost 組態

```
<VirtualHost 192.168.3.100>
  ServerName www.example.com
  DocumentRoot /srv/www/www.example.com/htdocs
  ServerAdmin webmaster@example.com
  ErrorLog /var/log/apache2/www.example.com_log
  CustomLog /var/log/apache2/www.example.com-access_log common
  <Directory "/srv/www/www.example.com/htdocs">
  Require all granted
  </Directory>
</VirtualHost>
```

31.2.3 使用 YaST 設定 Apache

若要使用 YaST 來設定您的 Web 伺服器，請啓動 YaST 並選取網路服務 › HTTP 伺服器。第一次啓動模組時，HTTP 伺服器精靈會啓動，提示您對伺服器管理進行一些基本設定。完成精靈之後，每當您呼叫HTTP 伺服器模組時，HTTP 伺服器組態對話方塊就會啓動。如需詳細資訊，請參閱第 31.2.3.2 節 「HTTP 伺服器組態」。

31.2.3.1 HTTP 伺服器精靈

HTTP 伺服器精靈包含有五個步驟。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執行更具體的設定。

31.2.3.1.1 網路裝置選擇

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：已知埠、註冊埠和動態或私人埠）不供其他服務使用，則皆可供您使用。預設設定為在連接埠 80 上監聽所有網路介面（IP 位址）。

核取在防火牆中開啓埠選項，可在防火牆中開啓 Web 伺服器監聽的連接埠。若要使網頁伺服器在網路（包括 LAN、WAN 或公用網際網路）上為可用狀態，請核取此選項。只有在測試時，並且此時不須由外部網路存取 Web 伺服器，才可以關閉該連接埠。如果您有多個網路介面，請按一下防火牆細節以指定應在哪些介面上開啓連接埠。

按下一步繼續設定組態。

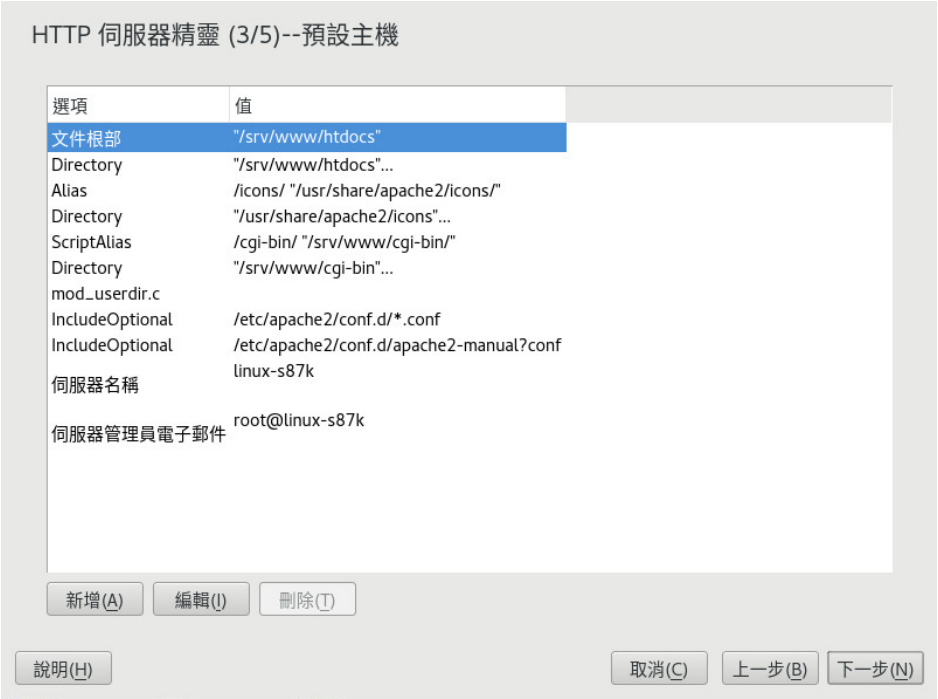
31.2.3.1.2 模組

使用模組組態選項，可以啓用或停用 Web 伺服器應支援的程序檔語言。如需有關啓用或停用其他模組的詳細資訊，請參閱第 31.2.3.2.2 節 「伺服器模組」。按一下下一步，繼續進行下一個對話方塊。

31.2.3.1.3 預設主機

此選項與預設網頁伺服器相關。正如第 31.2.2.1 節 「虛擬主機組態」內容所述，Apache 可以在一部實體機器上提供多個虛擬主機。組態檔案中第一個宣告的虛擬主機通常稱為預設主機。每部虛擬主機都會繼承預設主機的組態。

若要編輯主機設定（又稱為指令），請在表格中選取相應項目，然後按一下編輯。若要新增指示詞，請按一下新增。若要刪除指示詞，請選取該指示詞，然後按一下刪除。



圖形 31.1 HTTP 伺服器精靈：預設主機

這是伺服器預設值的清單：

`Document Root`

目錄路徑，Apache 會從此路徑為此主機提供檔案。`/srv/www/htdocs` 是預設位置。

`Alias`

使用 `Alias` 指令可以將 URL 對應到實體檔案系統位置。這表示某個路徑即使不在檔案系統的 `Document Root` 中，仍可藉由該路徑別名的 URL 進行存取。預設的 SUSE Linux Enterprise Server `Alias` `/icons` 指向 `/usr/share/apache2/icons`，做為目錄索引檢視中顯示的 Apache 圖示。

`ScriptAlias`

功能相似於 `Alias` 指示詞，`ScriptAlias` 指示詞可以將 URL 映射到檔案系統位置。兩者差別在於 `ScriptAlias` 可以將目標目錄指定作為 CGI 位置，表示該 CGI 程序檔必須在該位置執行。

`Directory`

使用 `Directory` 設定時，您可以指定一組組態選項，只將其中的選項套用到特定目錄。

在此可以設定 `/srv/www/htdocs`、`/usr/share/apache2/icons` 和 `/srv/www/cgi-bin` 目錄的存取和顯示選項。其中預設值應該不需要進行改變。

`Include`

使用 Include 可以指定其他的組態檔案。預先設定的 `Include` 指示詞有兩個：`/etc/apache2/conf.d/` 為外部模組隨附之組態檔所在的目錄。使用此指示詞時，此目錄中所有以 `.conf` 結尾的檔案都會包含在內。使用第二個指示詞時，`/etc/apache2/conf.d/apache2-manual.conf`，即 `apache2-manual` 組態檔將包含在內。

`Server Name`

這個項目可以指定用戶端用來聯絡網頁伺服器的預設 URL。請使用完全合格的網域名稱（FQDN）來連接 `http://FQDN/` 的網頁伺服器或其 IP 位址。您不能在此選擇任意名稱 -- 該伺服器必須是「已知」採用這個名稱。

`Server Administrator E-Mail`

伺服器管理員的電子郵件地址．這個地址可顯示在 Apache 建立的錯誤頁面（舉例說明）。

完成設定預設主機步驟後，請按一下下一步，繼續下一個組態步驟。

31.2.3.1.4 虛擬主機

在此步驟中，精靈會顯示已完成設定之虛擬主機的清單（請參閱第 31.2.2.1 節 「虛擬主機組態」）。如果在啓動 YaST HTTP 精靈之前尚未進行手動變更，則不會顯示虛擬主機。

若要新增主機，請按一下新增開啓對應的對話方塊，並在其中輸入主機的基本資訊，例如伺服器名稱、伺服器內容根目錄（`DocumentRoot`）和管理員電子郵件。伺服器解析可用來決定主機的識別方式（以名稱為基礎或是以 IP 為基礎）。透過變更虛擬主機 ID指定名稱或 IP 位址

按一下下一步，繼續進入虛擬主機組態對話方塊的第二部分。

在虛擬主機組態對話方塊的第二部分中，您可以指定是否要啓用 CGI 程序檔、以及這些程序檔要使用哪個目錄。您也可以在此啓用 SSL。如果執行了這個動作，您就必須同時指定證書的路徑。如需有關 SSL 和證書的詳細資訊，請參閱第 31.6.2 節 「設定提供 SSL 的 Apache」。使用目錄索引選項，可以指定當用戶端要求目錄時要顯示哪個

檔案（預設為 `index.html`）。新增一或多個檔案名稱（以空格分隔）可變更此設定。使用啓用公用 HTML，便可在伺服器的 `http://www.example.com/~USER` 下存取使用者公用目錄（`~USER/public_html/`）的內容。

重要：建立虛擬主機

您不能在此隨意新增虛擬主機。如果使用以名稱為基礎的虛擬主機，就必須在網路上解析每部主機名稱。如果是使用以 IP 為基礎的虛擬主機，每個可用 IP 位址就只能指派一部主機。

31.2.3.1.5 摘要

這是精靈的最後一個步驟。您可以在此處決定 Apache 伺服器啓動的方式和時間：開機時啓動或手動啓動。同時可檢視目前已完成之組態的簡短摘要。如果您接受目前設定，請按一下完成以完成組態設定。若要進行變更，請按一下上一步，直至所需的對話方塊顯示。按一下 HTTP 伺服器進階組態便可開啓第 31.2.3.2 節 「HTTP 伺服器組態」所介紹的對話方塊。

圖形 31.2 HTTP 伺服器精靈：摘要

31.2.3.2 HTTP 伺服器組態

HTTP 伺服器組態對話方塊可提供比精靈更多的組態調整（精靈 只會在第一次設定網頁伺服器時執行）。其中包含下列要介紹的四個索引標籤。在此變更的任何選項都無法立即生效 -- 您必須先按一下完成進行確認之後，它們才會生效。按一下中止，系統將不變更組態模組，並會捨棄您的變更。

31.2.3.2.1 監聽連接埠和位址

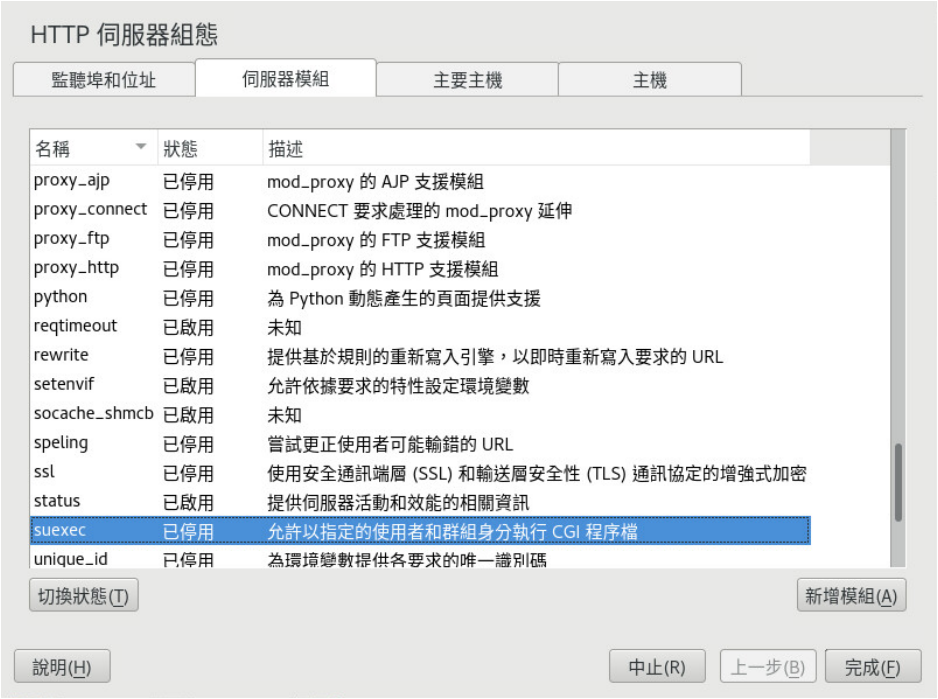
在HTTP 服務中，選取執行（啓用）或停止（停用）Apache。在監聽埠中，新增、編輯或刪除可透過其使用伺服器的位址和連接埠。預設會在連接埠 80 上監聽所有介面。任何情況下都必須核取在防火牆中開啓埠，否則將無法從外部連接 Web 伺服器。只有在測試時，並且此時不須由外部網路存取 Web 伺服器，才可以關閉該連接埠。如果您有多個網路介面，請按一下防火牆細節以指定應在哪些介面上開啓連接埠。

使用記錄檔案時，可檢視存取記錄檔案或錯誤記錄檔案。測試組態時此選項非常有用。記錄檔案會在另一個視窗中開啓，您也可以在此處重新啓動或重新載入 Web 伺服器。如需詳細資料，請參閱第 31.3 節 「啓動和停止 Apache」。這些指令會立即生效，其記錄訊息也會即刻顯示。

圖形 31.3 HTTP 伺服器組態：監聽連接埠和位址

31.2.3.2.2 伺服器模組

您可以按一下切換狀態，變更 Apache2 模組的狀態（啓用或停用）。按一下新增模組，可新增已經安裝但是未列出的新模組。若要更進一步認識模組，請參閱第 31.4 節 「安裝、啓用和設定模組」。



圖形 31.4 HTTP 伺服器組態：伺服器模組

31.2.3.2.3 主要主機

這些對話方塊相同於前面已介紹過的對話方塊。請參閱第 31.2.3.1.3 節 「預設主機」和第 31.2.3.1.4 節 「虛擬主機」。

31.3 啓動和停止 Apache

如果已如第 31.2.3 節 「使用 YaST 設定 Apache」中所述使用 YaST 設定，Apache 會在開機時間在 `multi-user.target` 與 `graphical.target` 中啓動。若要變更此行為，可以使用 YaST 的服務管理員或使用 `systemctl` 指令行工具（`systemctl enable` 或 `systemctl disable`）。

若要在執行中的系統上啓動、停止或操作 Apache，請依照下方所述使用 `systemctl` 或 `apachectl` 指令。

如需 `systemctl` 指令的一般資訊，請參閱第 13.2.1 節 「管理正在執行的系統中的服務」。

`systemctl status apache2`
: 檢查 Apache 是否已啓動。

`systemctl start apache2`
: 啓動尚未執行的 Apache。

`systemctl stop apache2`
: 透過終止父處理程序來停止 Apache。

`systemctl restart apache2`
: 停止 Apache，然後重新啓動。啓動之前並未在執行中的網頁伺服器。

`systemctl try-restart apache2`
: 若 Apache 已在執行中，則將其停止並重新啓動。

`systemctl reload apache2`
: 通知所有 Apache 衍生處理程序在關機之前先完成各自的要求，以停止網頁伺服器。每個程序結束後都會被一個新啓動的程序所取代，最終會將 Apache 完全「重新啓動」。

提示：在線上環境中重新啓動 Apache

使用此指令，您無需中斷連接，即可啓用 Apache 組態中的變更。

`systemctl stop apache2`
: 於 `GracefulShutdownTimeout` 所設定的指定期間後停止 Web 伺服器，以確保系統可以完成現有的要求。

`apachectl configtest`
: 在不影響執行中之網頁伺服器的情況下，檢查組態檔案的語法。因為這項檢查會在每次伺服器啓動、重新載入或重新啓動時强制進行，所以通常並不需要明確執行該測試（如果這時有找到組態錯誤，網頁伺服器就不會完成啓動、重新載入或是重新啓動）。

`apachectl status` 和 `apachectl fullstatus`

分別傾印簡要或完整的狀態畫面。需要啓用 `mod_status` 模組，並安裝文字型瀏覽器（例如 `links` 或 `w3m`）。除此之外，`status` 必須加入 `/etc/sysconfig/apache2` 檔案的 `APACHE_SERVER_FLAGS`。

提示：其他旗標

如果您為指令指定其他旗標，這些旗標就會傳遞至 Web 伺服器。

31.4 安裝、啓用和設定模組

Apache 軟體採用了模組化設計：除了部分核心任務，其餘所有功能皆由模組處理。這方面的發展很快，甚至連 HTTP 都是由模組（http_core）處理。

Apache 模組可以在建構時編譯成 Apache 二進位檔案，或在執行時期動態載入。如需瞭解如何動態載入模組的詳細資訊，請參閱第 31.4.2 節 「啓用和停用」。

Apache 模組可以分成四種不同類別：

基本模組

基本模組會依預設編譯到 Apache。SUSE Linux Enterprise Server 的 Apache 中僅編譯了 `mod_so`（載入其他模組要用到）與 `http_core`。所有其他模組均以共享物件的方式提供，即在執行期間加入，而不包含在伺服器二進位檔案中。

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 Enterprise Server 中，這類模組以共用物件方式提供，並可在執行時期載入到 Apache。

外部模組

標示為外部的模組不會包含於 Apache 正式發行版本中。不過，SUSE Linux Enterprise Server 提供了其中的幾個模組。

多重處理模組（MPM）

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。

31.4.1 模組安裝

如果您已依照第 31.1.2 節 「安裝」 中所述完成預設安裝，則下列模組此時都已安裝：所有基礎模組與延伸模組、多重處理模組 Prefork MPM 以及外部模組 `mod_python`。

您可以啓動 YaST，然後選擇軟體 › 軟體管理，來安裝其他的外部模組。現在，請選擇檢視 › 搜尋，然後搜尋 apache。結果清單中除其他套件之外，還會包含所有可用的外部 Apache 模組。

31.4.2 啓用和停用

以手動方式或透過 YaST 啓用或停用特定模組。在 YaST 中，若要啓用或停用程序檔語言模組（PHP5、Perl 和 Python），需要使用第 31.2.3.1 節 「HTTP 伺服器精靈」中所述的模組組態。所有其他模組都可以依據第 31.2.3.2.2 節 「伺服器模組」說明步驟來啓用或停用。

如果您想手動啓用或停用這些模組，請分別使用指令 `a2enmod MODULE` 或 `a2dismod MODULE`。`a2enmod -l` 會輸出目前所有使用中的模組清單。

重要：包含外部模組的組態檔案

如果您已經手動啓用外部模組，請確定將其組態檔案載入至所有的虛擬主機組態。外部模組的組態檔案會存放在 `/etc/apache2/conf.d/` 之下，而且預設會載入 `/etc/apache2/default-server.conf`。如需更精細的控制，可以將 `/etc/apache2/default-server.conf` 中的內容設定為備註，並將其僅新增至特定的虛擬主機。如需範例，請參閱 `/etc/apache2/vhosts.d/vhost.template`。

31.4.3 基本和延伸模組

Apache 說明文件中詳細介紹了所有的基本模組和延伸模組。本文件只提供最重要模組的概要說明。如需關於每個模組的詳細資訊，請參閱 http://httpd.apache.org/docs/2.4/mod/。

mod_actions

提供在需要特定 MIME 類型（例如 application/pdf）、具有特定副檔名的檔案（例如 .rpm）或特定要求方法（例如 GET）時執行程序檔的方法。這是預設啓用的模組。

mod_alias

提供 Alias 和 Redirect 指示詞，供您用來將 URL 映射至特定目錄（Alias）或將所要求的 URL 重新導向至其他位置。這是預設啓用的模組。

mod_auth*

驗證模組提供了幾種不同的驗證方式：使用 mod_auth_basic 進行基本驗證，或使用 mod_auth_digest 進行摘要驗證。

mod_auth_basic 和 mod_auth_digest 必須與驗證提供者模組 mod_authn_*（例如適用於以文字檔案為基礎之驗證的 mod_authn_file），以及驗證模組 mod_authz_*（例如適用於使用者驗證的 mod_authz_user）結合使用。

關於此主題的詳細資訊，請參閱 http://httpd.apache.org/docs/2.4/howto/auth.html 上的驗證 HOWTO。

mod_autoindex

Autoindex 會在沒有任何索引檔案（例如，index.html）出現時產生目錄清單。這些索引的外觀可加以設定。這是預設啓用的模組。然而，目錄清單已預設為停用，經由 Options 指示詞來覆寫虛擬主機組態的這項設定。這個模組的預設組態檔案會存放在 /etc/apache2/mod_autoindex-defaults.conf。

mod_cgi

執行 CGI 程序檔時必須使用 mod_cgi。這是預設啓用的模組。

mod_deflate

使用這個模組時，Apache 可以設定成即時壓縮成指定檔案類型之後，再進行傳送。

mod_dir

mod_dir 可提供 DirectoryIndex 指示詞，供您用來設定當要求目錄（預設是 index.html）時要自動傳遞哪類檔案。它還提供另一項功能：當目錄要求沒有包含末尾斜線時，就會自動重新導向到正確 URL。這是預設啓用的模組。

`mod_env`

控制傳遞給 CGI 程序檔或 SSI 頁面的環境。可以在呼叫 httpd 程序的外圍程序中設定、取消設定或傳遞環境變數。這是預設啓用的模組。

`mod_expires`

使用 `mod_expires` 時，您可以透過傳送 `Expires` 標頭，來控制代理和瀏覽器快取重新整理文件的頻率。這是預設啓用的模組。

`mod_http2`

Apache 可以使用 `mod_http2` 取得對 HTTP/2 通訊協定的支援。在 `VirtualHost` 中指定 `Protocols h2 http/1.1` 即可實現支援。

`mod_include`

`mod_include` 可讓您使用 Server Side Include (SSI)，這項工具會提供動態產生 HTML 頁面的基本功能。這是預設啓用的模組。

`mod_info`

可透過 http://localhost/server-info/ 提供伺服器組態的綜合綜覽。基於安全性考量，您應該永遠限制這個 URL 的存取權限。預設只允許 `localhost` 存取這個 URL。`mod_info` 是在 `/etc/apache2/mod_info.conf` 中設定。

`mod_log_config`

使用此模組時，您可以設定 Apache 記錄檔案的外觀。這是預設啓用的模組。

`mod_mime`

Mime 模組會根據所傳送檔案的副檔名（例如，HTML 文件的副檔名為 `text/html`）來確定檔案是否包含正確的 MIME 標頭。這是預設啓用的模組。

`mod_negotiation`

內容協商（Content Negotiation）所需的模組。如需更多詳細資訊，請參閱 http://httpd.apache.org/docs/2.4/content-negotiation.html。這是預設啓用的模組。

`mod_rewrite`

可提供 `mod_alias` 的功能，但具備更多功能和更大的靈活性。使用 `mod_rewrite` 時，您可以依據多個規則、要求標頭和其他條件來重新導向 URL。

`mod_setenvif`

根據用戶端的要求設定環境變數，如用戶端傳送的瀏覽器字串或用戶端的 IP 位址。這是預設啓用的模組。

`mod_spelling`

`mod_spelling` 會嘗試自動修正 URL 中出現的打字錯誤，例如大小寫錯誤。

`mod_ssl`

啓用網頁伺服器和用戶端之間的加密連接。如需詳細資料，請參閱 第 31.6 節「設定提供 SSL 的安全網頁伺服器」。這是預設啓用的模組。

`mod_status`

可透過 `http://localhost/server-status/` 提供有關伺服器活動及效能的資訊。基於安全性考量，您應該永遠限制這個 URL 的存取權限。預設只允許 `localhost` 存取這個 URL。`mod_status` 是在 `/etc/apache2/mod_status.conf` 中設定。

`mod_suexec`

`mod_suexec` 可讓您以不同使用者和群組身分來執行 CGI 程序檔。這是預設啓用的模組。

`mod_userdir`

啓用 `~USER/` 下使用者特定的目錄。在組態中必須指定 `UserDir` 指示詞。這是預設啓用的模組。

31.4.4 多重處理模組

SUSE Linux Enterprise Server 提供了兩種不同的多重處理模組 (MPM) 來搭配 Apache 使用。

- Prefork MPM
- Worker MPM

31.4.4.1 Prefork MPM

Prefork MPM 會實作未產生執行緒、正在進行 Prefork 的 Web 伺服器。這個模組會讓 Web 伺服器以類似 Apache 1.x 版的行為作業。在此版本中，Apache 會透過衍生出獨立的子程序，將各個要求分開處理。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。

雖然透過這種以處理程序為主的方法可以提供穩定性，但是比起 Worker MPM，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。

31.4.4.2 Worker MPM

Worker MPM 會提供多執行緒 Web 伺服器。執行緒是「輕量級」的處理程序。執行緒和處理程序相比的優點是，它消耗的資源較少。Worker MPM 不只會衍生子處理程序，它還可使用執行緒和伺服器處理程序，來為要求提供服務。預衍生的子程序具有多重執行緒。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。

一個主要缺點就是 Worker MPM 的穩定性：當某執行緒毀損時，處理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 上使用通用閘道介面 (CGI) 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 模組都能安全地使用執行緒，這樣就無法配合 worker MPM 使用。

警告：搭配 MPM 使用 PHP 模組

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`。

31.4.5 外部模組

此處提供了 SUSE Linux Enterprise Server 隨附的所有外部模組的清單。在列出目錄中找出模組的說明文件。

`mod_apparmor`

為 Apache 提供額外支援，對由 `mod_php5` 和 `mod_perl` 這類模組處理的個別 CGI 程序檔設定 AppArmor 限制。

套件名稱：`apache2-mod_apparmor`
詳細資訊：《Security Guide》

`mod_perl`

`mod_perl` 可讓您使用內嵌解譯器來執行 Perl 程序檔。內嵌在伺服器的常駐解譯器可以避免因啓動外部解譯器造成的負荷，以及在 Perl 啓動階段時降低速度。

套件名稱：`apache2-mod_perl`
組態檔案：`/etc/apache2/conf.d/mod_perl.conf`
詳細資訊：`/usr/share/doc/packages/apache2-mod_perl`

`mod_php5`

PHP 是一種伺服器端、跨平台式的 HTML 內嵌程序檔語言。

套件名稱：`apache2-mod_php5`
組態檔案：`/etc/apache2/conf.d/php5.conf`
詳細資訊：`/usr/share/doc/packages/apache2-mod_php5`

`mod_python`

`mod_python` 允許在 Apache HTTP 伺服器中內嵌 Python，以便大幅提高效能和增加網頁應用程式的設計彈性。

套件名稱：`apache2-mod_python`
詳細資訊：`/usr/share/doc/packages/apache2-mod_python`

`mod_security`

`mod_security` 提供 Web 應用程式防火牆，用於保護 Web 應用程式免受各種攻擊。此外，它還支援 HTTP 流量監控和即時分析。

套件名稱：`apache2-mod_security2`

組態檔案：`/etc/apache2/conf.d/mod_security2.conf`
詳細資訊：`/usr/share/doc/packages/apache2-mod_security2`
文件：http://modsecurity.org/documentation/

31.4.6 編譯

Apache 允許進階使用者編寫自訂模組進行延伸。若要開發 Apache 模組或編譯協力廠商模組，除了相對應開發工具外，還需要套件 `apache2-devel`。`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` 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 的動態共享物件 (DSO)。

`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/lib64/apache2`。
- `/usr/sbin/apxs2-prefork` — 適合用於 prefork MPM 模組。安裝位置是 `/usr/lib64/apache2-prefork`。
- `/usr/sbin/apxs2-worker` — 適合用於 worker MPM 模組。安裝位置是 `/usr/lib64/apache2-worker`。

請使用下列指令透過原始碼安裝並啓用模組：

```
cd /path/to/module/source
apxs2 -cia MODULE.c
```

其中，`-c` 用於編譯模組，`-i` 用於安裝模組，`-a` 用於啓用模組。如需有關 `apxs2` 的其他選項資訊，請參閱 `apxs2(1)` man 頁面。

31.5 啓用 CGI 程序檔

Apache 的通用閘道介面 (CGI) 可讓您使用程式或程序檔（通常稱 CGI 程序檔）建立動態內容。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 PHP 等程式檔設計語言。

若要讓 Apache 傳送 CGI 程序檔建立的內容，就必須啓用 `mod_cgi` 模組。同時還需要用到 `mod_alias`。這兩種都是預設啓用的模組。如需啓用模組的詳細資訊，請參閱第 31.4.2 節 「啓用和停用」。

警告：CGI 安全性

允許伺服器執行 CGI 程序檔會產生潛在的安全性弱點。請參考第 31.8 節 「避免安全性問題」，以取得其他資訊。

31.5.1 Apache 組態

在 SUSE Linux Enterprise Server 中，CGI 程序檔只能在 `/srv/www/cgi-bin/` 目錄中執行。這個位置已設定用來執行 CGI 程序檔。如果您已經建立虛擬主機組態（請參閱第 31.2.2.1 節 「虛擬主機組態」）並想要將程序檔放置到主機特定的目錄，則必須解除鎖定和設定此目錄。

範例 31.5 VIRTUALHOST CGI 組態

```
ScriptAlias /cgi-bin/ "/srv/www/www.example.com/cgi-bin/" ❶

<Directory "/srv/www/www.example.com/cgi-bin/">
 Options +ExecCGI ❷
 AddHandler cgi-script .cgi .pl ❸
 Require all granted ❹
</Directory>
```

❶ 告知 Apache 依照 CGI 程序檔方式來處理位在這個目錄中的所有檔案。

❷ 啓用 CGI 程序檔執行

❸ 告知伺服器依照 CGI 程序檔方式來處理包含 `.pl` 和 `.cgi` 副檔名的檔案。依據個人需要來加以調整。

❹ `Require` 指令控制預設存取狀態。在此範例中，授予指定目錄的存取權並無限制。如需驗證和授權的詳細資訊，請參閱 http://httpd.apache.org/docs/2.4/howto/auth.html。

31.5.2 執行程序檔範例

CGI 程式設計不同於「一般」程式設計；因為 CGI 程式和程序檔的最前面必須是 MIME-Type 標頭，例如 `Content-type: text/html`。這個標頭會傳送到用戶端，使其瞭解所接收內容的類型。其次，程序檔的輸出必須是用戶端（通常指網頁瀏覽器）可了解的內容，例如 HTML（一般情況）、純文字或影像。

Apache 套件會在 `/usr/share/doc/packages/apache2/test-cgi` 提供簡單的測試程序檔。這個程序檔會以純文字方式輸出部分環境變數的內容。請將這段程序檔複製到 `/srv/www/cgi-bin/` 或虛擬主機的程序檔目錄（`/srv/www/www.example.com/cgi-bin/`），並將其命名為 `test.cgi`。編輯檔案，讓 `#!/bin/sh` 位於第一行。

可由 Web 伺服器存取的檔案應該屬於 `root` 使用者所有。如需其他資訊，請參閱第 31.8 節 「避免安全性問題」。因為網頁伺服器可由不同使用者身分執行，所以 CGI 程序檔必須具備可供全球執行和可供全球讀取等特性。變更 CGI 目錄和使用 `chmod 755 test.cgi` 指令，便可套用適當的許可權。

現在，請呼叫 `http://localhost/cgi-bin/test.cgi` 或 `http://www.example.com/cgi-bin/test.cgi`。這時應該會顯示 「CGI/1.0 測試程序檔報告」。

31.5.3 CGI 疑難排解

如果這時沒有顯示測試程式的輸出結果，而是出現錯誤訊息，請檢查下列項目：

CGI 疑難排解

- 您是否有在變更組態之後重新載入伺服器？如果沒有，請使用 `systemctl reload apache2` 重新載入
- 您是否已正確設定自訂的 CGI 目錄（若有的話）？如果您不確定，請在預設的 CGI 目錄 `/srv/www/cgi-bin/` 中測試此程序檔，並使用 `http://localhost/cgi-bin/test.cgi` 進行呼叫。
- 檔案許可權是否正確？請切換至 CGI 目錄並執行 `ls -l test.cgi`。輸出將以下面的字串開頭

```
-rwxr-xr-x  1 root root
```

- 請確定程序檔沒有包含任何程式設計錯誤。如果您未變更過 `test.cgi`，此情況應該就不會發生，但是如果您是使用自己的程式，請務必確定這些程式中沒有任何程式設計錯誤。

31.6 設定提供 SSL 的安全網頁伺服器

如果 Web 伺服器和用戶端之間會傳輸信用卡資訊等敏感性資料，最好使用需經過驗證的安全加密連接。`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。使用 SSL/TLS 時，Web 伺服器和用戶端之間會建立私人連接。如此便可確保資料完整性，使用戶端與伺服器可以彼此進行驗證。

為了完成這個目的，伺服器會在回覆任何 URL 要求之前，先傳送可證明伺服器有效身分之相關資訊的 SSL 證書。如此即可確保該伺服器是此通訊的唯一正確端點。此外，該證書會在用戶端和伺服器端建立加密連接，以便在沒有洩漏敏感、純文字內容的風險情況下傳輸資訊。

`mod_ssl` 本身不會實作 SSL/TLS 通訊協定，而是做為 Apache 和 SSL 程式庫之間的介面。在 SUSE Linux Enterprise Server 中，使用的是 OpenSSL 程式庫。OpenSSL 會自動隨 Apache 完成安裝。

使用 `mod_ssl` 搭配 Apache 的最明顯特徵，就是 URL 的字首都會加上 `https://`，而不是 `http://`。

31.6.1 建立 SSL 憑證

您必須建立 SSL 證書，才能在 Web 伺服器上使用 SSL/TLS 功能。網頁伺服器和用戶端在彼此驗證時要用到這項證書，以便讓任一方可以清楚識別對方。為了確保證書的完整性，其必須由每位使用者信任的一方加以簽章。

您可以建立下列三種類型的證書：僅供測試使用的「虛擬」證書、供已定義信任圈使用者使用的自我簽發證書，以及由獨立、公開的證書授權機構（CA）簽發的證書。

證書的建立可分為兩個步驟。首先產生證書授權機構的私密金鑰，接著再使用該金鑰簽發伺服器證書。

提示：更多資訊

若要進一步瞭解 SSL/TLS 的概念和定義，請參閱 http://httpd.apache.org/docs/2.4/ssl/ssl_intro.html。

31.6.1.1 建立「虛擬」證書

若要產生虛構證書，請呼叫程序檔 `/usr/bin/gensslcert`。此操作會建立或覆寫下列檔案。使用 `gensslcert` 的可選參數可以微調證書。如需詳細資訊，可呼叫 `/usr/bin/gensslcert -h`。

- `/etc/apache2/ssl.crt/ca.crt`
- `/etc/apache2/ssl.crt/server.crt`
- `/etc/apache2/ssl.key/server.key`
- `/etc/apache2/ssl.csr/server.csr`

`ca.crt` 的複製本也會放在 `/srv/www/htdocs/CA.crt` 提供下載。

重要：僅供測試之用

虛構證書絕對不可用於生產環境系統。這類證書只能用於測試目的。

31.6.1.2 建立自我簽發證書

如果您要設定一個安全 Web 伺服器供內部網路或已定義的一群使用者圈使用，則透過您自己的證書管理中心（CA）簽發證書就可能足以有效符合此時的證書需求。請注意，此類網站的造訪者將看到類似「此網站不可信」的警告，因為網頁瀏覽器不能識別自行簽署的證書。

重要：自行簽署的證書

僅在供認識您、且信任您為證書管理中心之使用者存取的網頁伺服器上，方可使用自我簽署證書。我們不建議您在公開商店等場所使用此類證書。

首先您需要產生證書登記申請（CSR）。然後使用 `openssl`，並使用 `PEM` 作為證書格式。在此步驟中，系統會要求您輸入密碼片語並回答若干問題。請記住該密碼片語，日後還將使用。

```
sudo openssl req -new > new.cert.csr
Generating a 1024 bit RSA private key
..++++++
.........++++++
writing new private key to 'privkey.pem'
Enter PEM pass phrase: ❶
Verifying - Enter PEM pass phrase: ❷
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]: ❸
State or Province Name (full name) [Some-State]: ❹
Locality Name (eg, city) []: ❺
Organization Name (eg, company) [Internet Widgits Pty Ltd]: ❻
Organizational Unit Name (eg, section) []: ❼
Common Name (for example server FQDN, or YOUR name) []: ❽
Email Address []: ❾

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: ❿
An optional company name []: ⓫
```

❶ 填寫密碼片語。

❷ ...再次填寫密碼片語（並記住它）。

❸ 填寫以 2 個字母表示的國家/地區代碼，例如 `GB` 或 `CZ`。

❹ 填寫您所在的州/省。

❺ 填寫城市名稱，例如 `Prague`。

❻ 填寫您在職的組織機構名稱。

❼ 填寫您的組織單位，沒有則保留為空白。

❽ 填寫伺服器的網域名稱，或者您的名字和姓氏。

❾ 填寫您的辦公電子郵件地址。

10 將挑戰密碼保留空白，否則您每次重新啓動 Apache Web 伺服器時都需要輸入該密碼。

11 填寫選填的公司名稱，或保留為空白。

現在，您可以產生證書。您將再次使用 `openssl`，並且證書的格式是預設 `PEM`。

程序 31.3 產生證書

1. 將金鑰的私密部分輸出至 `new.cert.key`。系統將提示您輸入在建立證書登記申請（CSR）時所輸入的密碼片語。

```
sudo openssl rsa -in privkey.pem -out new.cert.key
```

2. 根據您在登記申請中填寫的資訊產生證書的公開部分。`-days` 選項指定證書到期之前的時間長度。您可以撤銷證書，或在證書到期之前更換證書。

```
sudo openssl x509 -in new.cert.csr -out new.cert.cert -req \
-signkey new.cert.key -days 365
```

3. 將證書檔案複製到相關的目錄，以便 Apache 伺服器可以讀取這些檔案。請確定私密金鑰 `/etc/apache2/ssl.key/server.key` 無法辨識，而公開 PEM 證書 `/etc/apache2/ssl.crt/server.crt` 則可辨識。

```
sudo cp new.cert.cert /etc/apache2/ssl.crt/server.crt
sudo cp new.cert.key /etc/apache2/ssl.key/server.key
```

提示：公用證書位置

最後一個步驟是將公用證書檔案從 `/etc/apache2/ssl.crt/server.crt` 複製到使用者可存取的位置，以便使用者將該檔案納入其網頁瀏覽器已知和信任的 CA 清單中。否則，瀏覽器會報告該證書是由不明授權機構所簽發。

31.6.1.3 取得官方簽發證書

目前有多家可簽署證書的官方證書管理中心。這類證書是由值得信任的協力廠商所簽署，因此可以完全信任。對外運作的安全網頁伺服器通常已取得官方簽發證書。最常用的證書管理中心（CA）一覽表可參閱 https://en.wikipedia.org/wiki/Certificate_authority#Providers↗。

在要求官方簽署的證書時，您並不需要向 CA 傳送證書，而只需傳送證書簽署要求 (Certificate Signing Request，CSR)。若要建立 CSR，請執行以下指令：

```
openssl req -new -newkey rsa:2048 -nodes -keyout newkey.pem -out newreq.pem
```

系統會要求您輸入可辨識名稱。這時系統會要求您回答一些問題，例如國家/地區名或組織名稱。請輸入有效資料 -- 您在此時輸入的每項資料將來都會顯示在證書中並用於檢查。您不需要回答每個問題。如果有不適用的問題或是您希望保留空白，請使用「.」。一般名稱是指 CA 本身的名稱 -- 請選擇明顯的名稱，例如 `My company` CA。最後，必須輸入挑戰密碼和替用的公司名稱。

從您呼叫程序檔的目錄中找出此 CSR。這個檔案名稱是 `newreq.pem`。

31.6.2 設定提供 SSL 的 Apache

在網頁伺服器端上，SSL 和 TLS 要求的預設連接埠是 443。同時有傾聽連接埠 80 的「一般」Apache 和傾聽連接埠 443 之已啓用 SSL/TLS 的 Apache，並不會產生衝突。事實上，HTTP 和 HTTPS 可以執行相同的 Apache 例項。通常這時會使用不同的虛擬主機，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：防火牆組態

請不要忘記為連接埠 443 上已啓用 SSL 的 Apache 開啓防火牆。您可以依據《Security Guide》，第 15 章「Masquerading and Firewalls」，第 15.4.1 節「Configuring the Firewall with YaST」所述方式，透過 YaST 完成這個動作。

在全域伺服器組態中，預設會啓用 SSL 模組。如果主機上停用了該模組，請使用 `a2enmod ssl` 指令將其啓用。若最終要啓用 SSL，需使用旗標 「SSL」 啓動伺服器。為此，請呼叫 `a2enflag SSL`（區分大小寫！）。如果選擇使用密碼來加密伺服器證書，您還需要提高 `/etc/sysconfig/apache2` 中 `APACHE_TIMEOUT` 的設定值，以便您在啓動 Apache 時有足夠的時間輸入該密碼片語。請重新啓動伺服器來確保這些變更生效。只是重新載入並無法保證變更生效。

虛擬主機組態目錄包含了樣板 `/etc/apache2/vhosts.d/vhost-ssl.template` 和 SSL 特定指示詞（將提供詳細文件說明）。如需一般虛擬主機組態的詳細資訊，請參閱第 31.2.2.1 節 「虛擬主機組態」。

若要開始設定組態，請將範本複製到 `/etc/apache2/vhosts.d/mySSL-host.conf`，並對其進行編輯。充分調整下列指示詞的值：

- `DocumentRoot`
- `ServerName`
- `ServerAdmin`
- `ErrorLog`
- `TransferLog`

31.6.2.1 以名稱為基礎的虛擬主機和 SSL

依預設，在只有一個 IP 位址的伺服器上，無法同時執行多個啓用了 SSL 的虛擬主機。以名稱為基礎的虛擬主機需要 Apache 瞭解所要求的伺服器名稱。使用 SSL 連接的問題在於，此類要求只能在使用預設虛擬主機建立了 SSL 連接後才能讀取。如此一來，使用者將會收到證書與伺服器名稱不符的警告訊息。

SUSE Linux Enterprise Server 提供了一項 SSL 通訊協定的延伸：伺服器名稱指示（Server Name Indication，SNI）。該項延伸會在 SSL 協議中傳送虛擬網域的名稱，以此解決這個問題。這可讓伺服器早些「切換」至正確的虛擬網域，並在瀏覽器中顯示正確的證書。

SUSE Linux Enterprise Server 中預設會啓用 SNI。若要針對 SSL 啓用以名稱為基礎的虛擬主機，請依照第 31.2.2.1.1 節 「以名稱為基礎的虛擬主機」中所述設定伺服器（請注意，您需要對 SSL 使用連接埠 `443` 而非 `80`）。

重要：SNI 瀏覽器支援

用戶端上也必須支援 SNI。不過，只有大部分瀏覽器支援 SNI，某些較舊的瀏覽器不支援。如需詳細資訊，請參閱https://en.wikipedia.org/wiki/Server_Name_Indication#Support。

若要設定對不支援 SNI 的瀏覽器的處理方式，請使用指令 `SSLStrictSNIVHostCheck`。若在伺服器組態中設定為 `on`，則不支援 SNI 的瀏覽器對所有虛擬主機的存取都會遭到拒絕。如果 `VirtualHost` 指令中設定為 `on`，將拒絕存取此特定主機。

若在伺服器組態中設定為 `off`，則伺服器將表現為不支援 SNI。將由定義的第一個虛擬主機（連接埠 443）來處理 SSL 要求。

31.7 在同一部伺服器上執行多個 Apache 例項

從 SUSE® Linux Enterprise Server 12 SP1 開始，您可以在同一部伺服器上執行多個 Apache 例項。與執行多個虛擬主機相比，這可以帶來諸多的優勢（請參閱第 31.2.2.1 節 「虛擬主機組態」）：

- 如果需要將虛擬主機停用一段時間，您需要變更 Web 伺服器組態並將其重新啓動才能使變更生效。
- 如果一個虛擬主機出現問題，您需要重新啓動所有的虛擬主機。

您可以照常執行預設的 Apache 例項：

```
sytemctl start apache2
```

它將會讀取預設的 `/etc/sysconfig/apache2` 檔案。如果該檔案不存在，或者存在但未設定 `APACHE_HTTPD_CONF` 變數，則該例項將會讀取 `/etc/apache2/httpd.conf`。

若要啓動另一個 Apache 例項，請執行：

```
systemctl start apache2@INSTANCE_NAME
```

例如：

```
systemctl start apache2@example_web.org
```

依預設，例項會使用 `/etc/apache2@example_web.org/httpd.conf` 做為主要組態檔案，您可以透過設定 `/etc/sysconfig/apache2@example_web.org` 中的 `APACHE_HTTPD_CONF` 予以覆寫。

下面顯示了一個設定更多 Apache 例項的範例。請注意，您需要以 `root` 身分執行所有指令。

程序 31.4 設定其他 APACHE 例項

1. 依據 `/etc/sysconfig/apache2` 建立一個新的組態檔案，例如 `/etc/sysconfig/apache2@example_web.org`：

```
cp /etc/sysconfig/apache2 /etc/sysconfig/apache2@example_web.org
```

2. 編輯檔案 `/etc/sysconfig/apache2@example_web.org`，將包含以下內容的行

```
APACHE_HTTPD_CONF
```

變更為

```
APACHE_HTTPD_CONF="/etc/apache2/httpd@example_web.org.conf"
```

3. 依據 `/etc/apache2/httpd.conf` 建立檔案 `/etc/apache2/httpd@example_web.org.conf`。

```
cp /etc/apache2/httpd.conf /etc/apache2/httpd@example_web.org.conf
```

4. 編輯 `/etc/apache2/httpd@example_web.org.conf`，將

```
Include /etc/apache2/listen.conf
```

變更為

```
Include /etc/apache2/listen@example_web.org.conf
```

檢閱所有指令，並視需要予以變更。您可能需要變更

```
Include /etc/apache2/global.conf
```

並為每個例項建立新的 `global@example_web.org.conf`。建議將

```
ErrorLog /var/log/apache2/error_log
```

變更為

```
ErrorLog /var/log/apache2/error@example_web.org_log
```

以便每個例項都有個別的記錄。

5. 依據 `/etc/apache2/listen.conf` 建立 `/etc/apache2/listen@example_web.org.conf`。

```
cp /etc/apache2/listen.conf /etc/apache2/listen@example_web.org.conf
```

6. 編輯 `/etc/apache2/listen@example_web.org.conf`，將

```
Listen 80
```

變更為要執行新例項的埠號，例如 82：

```
Listen 82
```

若要透過安全通訊協定（請參閱第 31.6 節 「設定提供 SSL 的安全網頁伺服器」）執行新的 Apache 例項，還需將下面一行

```
Listen 443
```

（範例）變更為

```
Listen 445
```

7. 啓動新的 Apache 例項：

```
systemctl start apache2@example_web.org
```

8. 在網頁瀏覽器中開啓 `http://server_name:82`，檢查伺服器是否正在執行中。如果先前變更了新例項的錯誤記錄檔案名稱，您可以檢查這項變更：

```
tail -f /var/log/apache2/error@example_web.org_log
```

下面是在同一部伺服器上設定多個 Apache 例項時要注意的幾點：

- `/etc/sysconfig/apache2@INSTANCE_NAME` 檔案可以包含與 `/etc/sysconfig/apache2` 相同的變數，包括模組載入和 MPM 設定。
- 當有其他例項正在執行時，不需要執行預設的 Apache 例項。
- 如果未使用 `HTTPD_INSTANCE` 環境變數另行指定，Apache 輔助程式公用程式 `a2enmod`、`a2dismod` 和 `apachectl` 將在預設的 Apache 例項上執行。以下範例

```
export HTTPD_INSTANCE=example_web.org
a2enmod access_compat
a2enmod status
apachectl start
```

會將 `access_compat` 和 `status` 模組新增到 `/etc/sysconfig/apache2@example_web.org` 的 `APACHE_MODULES` 變數，然後啓動 `example_web.org` 例項。

31.8 避免安全性問題

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。軟體和意外的錯誤設定不可避免地會產生安全性問題。下面是可用來處理這些問題的幾項秘訣。

31.8.1 更新軟體

SUSE 會在發現 Apache 軟體弱點時，發出安全性建議事項。其中包含弱點修正指示，應盡可能套用。請由下列位置取得 SUSE 安全性公告：

- 網頁： http://www.suse.com/support/security/
- 郵寄清單歸檔： http://lists.opensuse.org/opensuse-security-announce/
- 安全性聲明清單： http://www.suse.com/support/update/

31.8.2 DocumentRoot 許可權

依預設，在 SUSE Linux Enterprise Server 中，`DocumentRoot` 目錄 `/srv/www/htdocs` 與 CGI 目錄 `/srv/www/cgi-bin` 的所有權屬於 `root` 使用者和群組。這些許可權不可變更。如果目錄對所有人開放寫入權限，則任何使用者都可以將檔案放入其中。然後，這些檔案可能會由具有 `wwwrun` 許可權的 Apache 執行，而這種情況可能會造成使用者取得非預期的檔案系統資源存取權限。使用 `/srv/www` 子目錄來存放虛擬主機的 `DocumentRoot` 和 CGI 目錄，並確定這些目錄所有權屬於 `root` 使用者和群組。

31.8.3 檔案系統存取

依預設，`/etc/apache2/httpd.conf` 已設定成拒絕存取整個檔案系統。切勿覆寫這些指示詞，不過您可以特別啓用 Apache 應當能夠讀取之所有目錄的存取權限。如需詳細資料，請參閱第 31.2.2.1.3 節 「基本虛擬主機組態」。如果要執行這個動作，請確保沒有任何重要檔案（例如密碼或系統組態檔案）可由外界進行讀取。

31.8.4 CGI 程序檔

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，因此會產生常見的安全性問題。將從伺服器執行的程序檔，只能由伺服器管理員信任的來源進行安裝 -- 通常最好不要讓使用者執行自己的程序檔。同時建議您為所有程序檔進行安全性稽核。

為了盡可能簡化程序檔的管理工作，通常建議您限制 CGI 程序檔在特定目錄中執行，而不是全域性開放執行。您可以使用 `ScriptAlias` 和 `Option ExecCGI` 指示詞來進行組態設定。SUSE Linux Enterprise Server 的預設組態不允許隨處執行 CGI 程序檔。

所有 CGI 程序檔都是以相同使用者身分執行，所以不同的程序檔彼此之間可能會產生衝突。module suEXEC 可讓您以不同使用者和群組身分來執行 CGI 程序檔。

31.8.5 使用者目錄

在啓用使用者目錄（使用 `mod_userdir` 或 `mod_rewrite`）時，您應該審慎考慮不要允許覆寫 `.htaccess` 檔案，因為這會允許使用者覆寫安全性設定。至少您應該使用 `AllowOverRide` 指示詞來限制使用者的應用範圍。在 SUSE Linux Enterprise Server 中，`.htaccess` 檔案預設處於啓用狀態，但使用者在使用 `mod_userdir`（請參閱 `/etc/apache2/mod_userdir.conf` 組態檔案）時不允許覆寫任何 `Option` 指令。

31.9 疑難排解

如果 Apache 未啓動，網頁就無法存取，或者使用者無法連接網頁伺服器，因此找出問題的根源是很重要的工作。下面是您可在其中尋找錯誤原因的幾個常見位置以及需要檢查的重點：

`apache2.service` 子指令的輸出：

不要使用 `/usr/sbin/apache2ctl` 二進位檔案來啓動和停止 Web 伺服器，而應使用 `systemctl` 指令（如第 31.3 節 「啓動和停止 Apache」中所述）。`systemctl status apache2` 詳細描述了錯誤，甚至還提供了修復組態錯誤的提示。

記錄檔案與詳細層級

無論發生了嚴重錯誤還是非嚴重錯誤，都可以檢查 Apache 記錄檔案尋找原因，主要檢查預設位於 `/var/log/apache2/error_log` 的錯誤記錄檔案。此外，如果需要檢視記錄檔案中更多的詳細資訊，還可以透過 `LogLevel` 指示詞來控制記錄訊息的詳細程度。

提示：簡單測試

使用 `tail -F /var/log/apache2/MY_ERROR_LOG` 指令檢視 Apache 記錄訊息。然後執行 `systemctl restart apache2`。現在，請嘗試連接到瀏覽器，並檢查輸出結果。

防火牆與連接埠

一個常見的錯誤是，沒有在伺服器的防火牆組態中開啓 Apache 的連接埠。如果是使用 YaST 來設定 Apache，就要透過其他選項來檢查這個特定問題（請參閱第 31.2.3 節 「使用 YaST 設定 Apache」）。如果您要手動設定 Apache，請透過 YaST 防火牆模組來跗

如果無法透過這些功能來查出錯誤原因，則請查閱 http://httpd.apache.org/bug_report.html 中的線上 Apache 錯誤資料庫。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。

31.10 更多資訊

`apache2-doc` 套件在許多位置包含了完整的 Apache 手冊，用於本地安裝及作為參考文件。這個套件不是預設安裝選項 — 安裝此套件最快的方式就是使用 `zypper in apache2-doc` 指令。完成安裝之後，http://localhost/manual/ 中將會有 Apach 手冊可供使用。您也可以從 http://httpd.apache.org/docs-2.4/ 網站位置來存取這份手冊。`/usr/share/doc/packages/apache2/README.*` 目錄會提供 SUSE 特定組態秘訣資訊。

31.10.1 Apache 2.4

如需 Apache 2.4 最新功能的清單，請參閱 http://httpd.apache.org/docs/2.4/new_features_2_4.html。如需從 2.2 升級至 2.4 版的資訊，請參閱下列網址資訊：http://httpd.apache.org/docs-2.4/upgrading.html。

31.10.2 Apache 模組

有關第 31.4.5 節 「外部模組」 中簡要介紹的外部 Apache 模組的詳細資訊，可在以下位置找到：

`mod_apparmor`
: http://en.opensuse.org/SDB:AppArmor

`mod_auth_kerb`
: http://modauthkerb.sourceforge.net/

`mod_perl`
: http://perl.apache.org/

`mod_php5`
: http://www.php.net/manual/en/install.unix.apache2.php

`mod_python`
: http://www.modpython.org/

`mod_security`
: http://modsecurity.org/

31.10.3 開發

如需更多有關開發 Apache 模組或是參與 Apache 網頁伺服器計畫的詳細資訊，請參閱下列主題內容：

Apache 開發人員資訊
: http://httpd.apache.org/dev/

Apache 開發人員說明文件

http://httpd.apache.org/docs/2.4/developer/

31.10.4 其他資源

如果您在 SUSE Linux Enterprise Server 中遇到與 Apache 有關的問題，請查閱「技術資訊搜尋」，網址為：http://www.suse.com/support。關於 Apache 的歷程，請參閱 http://httpd.apache.org/ABOUT_APACHE.html。此頁面也說明稱伺服器為 Apache 的原因。

32 使用 YaST 設定 FTP 伺服器

使用 YaST 的FTP 伺服器模組，可以將機器設定為 FTP（檔案傳輸通訊協定）伺服器。匿名和/或驗證使用者可以連接至您的機器並使用 FTP 通訊協定下載檔案，還可以將檔案上載至 FTP 伺服器（視組態而定）。YaST 使用 vsftpd（非常安全的 FTP 精靈）。

如果 YaST FTP 伺服器模組在您的系統中不可用，請安裝 `yast2-ftp-server` 套件。

若要使用 YaST 設定 FTP 伺服器，請執行以下步驟：

1. 開啓 YaST 控制中心並選擇網路服務 › FTP 伺服器，或以 `root` 身分執行 `yast2 ftp-server` 指令。
2. 如果系統中未安裝任何 FTP 伺服器，系統將在「YaST FTP 伺服器」模組啓動時詢問您要安裝哪個伺服器。選擇 vsftpd 伺服器，然後確認對話方塊。
3. 在啓動對話方塊中，設定啓動 FTP 伺服器的相關選項。如需詳細資訊，請參閱第 32.1 節 「啓動 FTP 伺服器」。
   在一般對話方塊中，設定 FTP 目錄、歡迎訊息、檔案建立遮罩和其他參數。如需詳細資訊，請參閱第 32.2 節 「FTP 一般設定」。
   在效能對話方塊中，設定會影響 FTP 伺服器負載的參數。如需詳細資訊，請參閱第 32.3 節 「FTP 效能設定」。
   在驗證對話方塊中，設定匿名和/或已驗證的使用者是否可使用 FTP 伺服器。如需詳細資訊，請參閱第 32.4 節 「驗證」。
   在進階設定對話方塊中，設定 FTP 伺服器的操作模式、SSL 連接與防火牆設定。如需詳細資訊，請參閱第 32.5 節 「進階設定」。
4. 按完成以儲存這些組態。

32.1 啓動 FTP 伺服器

在FTP 啓動對話方塊的服務啓動框架中，可以設定 FTP 伺服器的啓動方式。您可以選擇在系統開機時自動啓動伺服器，或是手動啓動伺服器。如果 FTP 伺服器只能在提出 FTP 連接要求之後啓動，請選擇透過 xinetd。

FTP 伺服器的目前狀態將顯示在FTP 啓動對話方塊的開啓與關閉框架中。按一下立即啓動 FTP可以啓動 FTP 伺服器。若要停止伺服器，可按一下立即停止 FTP。變更伺服器的設定後，請按一下儲存設定並立即重新啓動 FTP。如果按一下完成離開組態模組，組態將儲存。

圖形 32.1 FTP 伺服器組態 — 啓動

32.2 FTP 一般設定

在FTP 一般設定對話方塊的一般設定框架中，您可以設定連接至 FTP 伺服器後將會顯示的歡迎訊息。

如果核取Chroot 每一個選項，則所有本地使用者登入後都將進入其主目錄中的 Chroot Jail。此選項含安全性要求，特別是當使用者具有上載權限或外圍程序存取權限時，因此啓用此選項時請務必小心。

如果核取詳細記錄選項，則將記錄所有 FTP 請求與回應。

您可以透過 Umask 對由匿名和/或已驗證的使用者建立之檔案的權限加以限制。請在匿名使用者 Umask中為匿名使用者設定檔案建立遮罩，並在已驗證使用者 Umask中為已驗證的使用者設定檔案建立遮罩。應以八進位數字（前面加零）輸入遮罩。如需有關 umask 的詳細資訊，請參閱 umask 的 man 頁面（`man 1p umask`）。

在FTP 目錄框架中設定用於匿名使用者與已驗證使用者的目錄。按一下瀏覽可從本地檔案系統選取要使用的目錄。對於匿名使用者，預設 FTP 目錄為 `/srv/ftp`。請注意，vsftpd 不允許所有使用者都可以寫入此目錄，而會為匿名使用者建立具有寫入權限的子目錄 `upload`。

32.3 FTP 效能設定

在效能對話方塊中，設定會影響 FTP 伺服器負載的參數。最大閒置時間表示遠端用戶端在兩個 FTP 指令之間等待的最長時間（以分鐘計）。如果發生了更長時間的閒置，將中斷遠端用戶端的連接。同一個 IP 最大用戶端數決定可從單一 IP 位址連接的最大用戶端數量。最大用戶端數決定可以連接的最大用戶端數量。任何額外用戶端都將收到一則錯誤訊息。

可在最大本地速率和最大匿名速率中分別為本地已驗證的使用者及匿名用戶端設定最大資料傳輸速率。速率設定的預設值為 `0`，表示資料傳輸速率無限制。

32.4 驗證

在驗證對話方塊的啓用/停用匿名使用者和本地使用者框架中，您可以設定允許存取 FTP 伺服器的使用者。您還可以從以下選項中進行選擇：僅授予匿名使用者存取權限、僅授予驗證使用者（擁有系統帳戶）存取權限，或者同時授予這兩類使用者存取權限。

若要允許使用者將檔案上傳到 FTP 伺服器，請核取驗證對話方塊的上傳框架中的啓用上傳。在此處核取對應的方塊，您就可以允許包括匿名使用者在內的使用者上傳或建立目錄。

注意：vsftp — 允許匿名使用者上傳檔案

如果使用的是 vsftpd 伺服器而您希望匿名使用者能夠上載檔案或建立目錄，則需要在匿名 FTP 目錄中為所有使用者建立具有寫入權限的子目錄。

32.5 進階設定

FTP 伺服器既可以在主動模式中執行，也可以在被動模式中執行。依預設伺服器在被動模式中執行。若要切換至主動模式，請取消核取進階設定對話方塊中的啓用被動模式選項。您還可以調整被動模式的最小連接埠與被動模式的最大連接埠選項來變更伺服器上用於資料流的連接埠範圍。

若要在用戶端與伺服器之間使用加密通訊，可以選取啓用 SSL。檢查要支援之協定的版本，並指定用於 SSL 加密連接的 DSA 證書。

如果系統受防火牆保護，請核取在防火牆中開啓埠以啓用到 FTP 伺服器的連接。

32.6 更多資訊

如需 FTP 伺服器的詳細資訊，請參閱 `vsftpd` 與 `vsftpd.conf` 的手冊頁。

33 代理伺服器 Squid

Squid 是廣泛用於 Linux 與 UNIX 平台的代理快取。這表示它會將要求的網際網路物件（例如網頁伺服器或 FTP 伺服器上的資料），儲存在比伺服器更接近要求工作站的機器上。您可設定多階層，以確保即使在對終端使用者而言透明的模式下，也能達到最佳的回應速度和較低的頻寬使用量。您可使用其他諸如 squidGuard 的軟體來過濾 Web 內容。

Squid 可做為代理快取記憶體。它會將物件要求從用戶端（在此例中是從網頁瀏覽器）重新導向至伺服器。當從伺服器而來的要求物件到達時，它會將物件傳送到用戶端，並在硬碟快取記憶體中保留物件的副本。快取的其中一個優點是，當有數個用戶端要求相同的物件時，可以從硬碟快取來提供該物件。這可讓用戶端比從網際網路更快地擷取資料。此程序還可以減少網路流量。

除實際的快取外，Squid 還提供眾多其他功能：

- 在代理伺服器的互通階層之間分配負載
- 針對所有存取代理的用戶端定義嚴密的存取控制清單
- 允許或拒絕使用其他應用程式存取特定網頁
- 針對頻繁造訪的網頁產生統計資料，用於評估網頁瀏覽習慣

Squid 不是一般的代理。一般而言，它只會代理 HTTP 連接。它支援 FTP、Gopher、SSL 以及 WAIS 通訊協定，但不支援其他網際網路通訊協定，例如針對新聞或視訊會議的通訊協定。因為 Squid 只支援將 UDP 通訊協定用於在不同快取之間提供通訊功能，許多多媒體程式並不受支援。

33.1 關於代理快取的說明

當 Squid 做為代理快取記憶體時，使用方法有多種。若與防火牆合併，它有助於提高安全性。多個代理可一起使用。它也可以判斷應該快取物件類型和持續的時間長短。

33.1.1 Squid 以及安全性

Squid 可與防火牆配合使用，以便使用代理快取記憶體來保護內部網路不受外部的存取。防火牆將會拒絕所有的用戶端存取 Squid 以外的外部服務。所有的網路連接都必須由代理來建立。藉由這種組態方式，Squid 可完全控制網頁存取。

如果防火牆組態中包含 DMZ，代理應該在此區域內操作。第 33.6 節 「設定操作順暢的代理」介紹如何實作透明代理。這簡化了用戶端的組態，因為在此情況下，它們不需要有關代理的任何資訊。

33.1.2 多個快取

經過設定之後，可在多個 Squid 例項之間交換物件。這可減少系統的總負荷，並可增加在區域網路中取回物件的機會。您也可以設定快取階層，使快取可以將物件要求轉送至旁支快取或上層快取，如此其可從區域網路中的另一個快取或直接從來源要求物件。

為快取記憶體階層選擇適當的拓樸是非常重要的，因為這樣它就不會增加網路的整體流量。就大型的網路而言，理想的做法是：為每個子網路設定代理伺服器，並將它們與上層代理連接，再連線至 ISP 的代理快取。

這些通訊都是由在 UDP 通訊協定最上層執行的 ICP（網際網路快取通訊協定）所處理。在快取之間的資料傳輸是使用以 TCP 為基礎的 HTTP（超文字傳輸通訊協定，Hypertext Transmission Protocol）來處理。

為了找到最適合向其要求物件的伺服器，一個快取會將 ICP 要求傳送到所有旁支代理。旁支代理會透過 ICP 回應答覆這些要求。若偵測到物件，它們會使用代碼 `HIT`，若未偵測到，則使用 `MISS`。

如果發現多個 `HIT` 回應，代理伺服器會依據哪個快取傳送回覆的速度最快或哪部伺服器較近等因素來決定要從哪部伺服器下載。如果沒有收到滿意的回應，則會將要求傳送到上層快取。

注意：Squid 如何避免物件重複

為了避免網路上不同的快取記憶體中出現物件重複，系統會使用其他 ICP 通訊協定。例如 CARP（快取陣列路由通訊協定）或 HTCP（超文字快取通訊協定）。在網路中維護愈多的物件，則找到所需物件的機率也就愈大。

33.1.3 快取網際網路物件

網路中提供的很多物件都不是靜態的，例如動態產生的頁面和 TLS/SSL 加密內容。此類物件是不會被快取的，因為每次存取這些物件時，它們都會改變。

為確定物件在快取中應保留的時間，系統會為物件指定其中一種狀態。網頁以及代理伺服器會藉由新增標頭至這些物件來找出物件的狀態，例如「上一次修改」或「到期」以及對應的日期。也可使用指定不得快取的物件的其他標題。

快取中的物件通常會因缺少可用磁碟空間而透過 LRU（最久未用）之類的演算法進行替換。這表示代理會刪除那些最久沒有被要求的物件。

33.2 系統要求

系統要求主要取決於該系統必須承擔的最大網路負載。因此請檢查負載尖峰值，因為在這些時段，負載可能會達到日平均值的四倍以上。若不能確定，請稍稍高估系統要求。若 Squid 的工作負荷臨近其處理能力上限，可能會嚴重影響服務品質。接下來的幾節將依重要程度依次指出系統因素：

1. RAM 大小
2. CPU 速度/實體 CPU 核心
3. 磁碟快取的大小
4. 硬碟/SSD 及其架構

33.2.1 RAM

Squid 所需的記憶體（RAM）容量與快取中的物件數量有直接的關係。隨機存取記憶體的速度比硬碟/SSD 快很多。因此，請務必讓 Squid 程序擁有充足的記憶體，因為若使用交換磁碟，系統效能會大幅降低。

Squid 也會將快取記憶體物件的參照以及常要求的物件儲存在主記憶體中以加速此資料的擷取速度。除此之外，Squid 需要在記憶體中保留其他的資料，例如所有已處理 IP 位址的表格、精確的網域名稱快取、最常要求的物件、存取控制清單、緩衝區等等。

33.2.2 CPU

Squid 已經過優化，在處理器核心數量較少（4 - 8 個實體核心）的情況下工作狀態最好，這樣每個核心都能提供出色的效能。像超執行緒這類提供虛擬核心的技術會影響效能。

要充分利用多個 CPU 核心，必須設定多個寫入不同快取裝置的工作執行緒。預設情況下，多核心支援通常都處於停用狀態。

33.2.3 磁碟快取的大小

在小的快取中，`HIT`（發現要求的物件已在該處）的機率比較小，因為快取很容易就會填滿，所以較少要求的物件會替換為新的物件。例如，如果快取的可用空間為 1 GB，而使用者一天僅瀏覽 10 MB，則需要 100 多天才會將快取填滿。

確定所需快取大小的最簡易方法是考量連接的最大傳輸速率。1 Mbit/s 連接的最大傳輸速率為 128 KB/s。如果所有流量都流向快取，一個小時內這些量會達到 460 MB。假設所有流量都是在 8 小時工作時間產生，則一天內會達到 3.6 GB。由於連接一般都不會用到其容量上限，因此可以假設快取記憶體所處理的總資料容量大約為 2 GB。因此在本例中，Squid 需要 2 GB 磁碟空間來存放一天內快取的資料瀏覽量。

33.2.4 硬碟/SSD 架構

速度在快取處理過程中扮演很重要的角色，因此應該特別注意這個因素。對於硬碟/SSD，此參數稱為隨機搜尋時間或隨機讀取效能，按毫秒計算。因為 Squid 從硬碟/SSD 讀取或寫入其中的資料區塊通常都很小，所以硬碟/SSD 的搜尋時間/讀取效能比其資料輸送量更重要。

若要用做代理，高轉速硬碟或 SSD 是最好的選擇。使用硬碟時，採用多個較小硬碟的效果可能更佳，因為每個硬碟都有單獨的快取目錄，可避免讀取次數過多。

如果采用 RAID 系統，可以犧牲速度來提升可靠性。但是基於效能考量，請避免使用（軟體）RAID5 及類似設定。

檔案系統的選擇通常無關緊要。但是，使用掛接選項 `noatime` 可提升效能 - Squid 提供自己的時戳，因此無需檔案系統追蹤存取時間。

33.3 Squid 基本用法

如果尚未安裝套件 `squid` ，請予以安裝。預設情況下，`SUSE® Linux Enterprise Server` 上不會安裝 squid 套件。

Squid 在 SUSE® Linux Enterprise Server 中已經過預先設定，可在安裝後直接啓動。為了確保啓動更平順，應該將網路設定為至少使用一部名稱伺服器，而且可連接網際網路。如果撥號連接是使用動態 DNS 組態，就有可能產生問題。在此情況下，至少應指定名稱伺服器，因為如果 Squid 在 `/etc/resolv.conf` 中未偵測到 DNS 伺服器，就不會啓動。

33.3.1 啓動 Squid

若要啓動 Squid，請使用：

```
tux > sudo systemctl start squid
```

若要讓 Squid 隨系統一起啓動，請使用 `systemctl enable squid` 啓用該服務。

33.3.2 檢查 Squid 是否正在運作

若要檢查 Squid 是否正在執行，請選擇下列其中一種方式：

- 使用 `systemctl`：

  ```
  tux > systemctl status squid
  ```

  此指令的輸出應指出 Squid **已載入** 且 **在使用中（執行中）**。

- 使用 Squid 自身：

  ```
  tux > sudo squid -k check | echo $?
  ```

  此指令的輸出應當為 `0`，但也可能包含其他警告或訊息。

若要測試 Squid 在本地系統上的功能，請選擇下列其中一種方式：

- 若要進行測試，您可以使用指令行工具 `squidclient`，它可向 Web 要求輸出回應，類似於 `wget` 或 `curl`。

與這些工具不同的是，`squidclient` 會自動連接至 Squid 的預設代理設定 `localhost:3128`。不過，如果您變更過 Squid 的組態，則需要透過指令行選項將 `squidclient` 設定為使用其他設定。如需詳細資訊，請參閱 `squidclient --help`。

範例 33.1 使用 `squidclient` 提交的要求

```
tux > squidclient http://www.example.org
HTTP/1.1 200 OK
Cache-Control: max-age=604800
Content-Type: text/html
Date: Fri, 22 Jun 2016 12:00:00 GMT
Expires: Fri, 29 Jun 2016 12:00:00 GMT
Last-Modified: Fri, 09 Aug 2013 23:54:35 GMT
Server: ECS (iad/182A)
Vary: Accept-Encoding
X-Cache: HIT
x-ec-custom-error: 1
Content-Length: 1270
X-Cache: MISS from moon❶
X-Cache-Lookup: MISS from moon:3128
Via: 1.1 moon (squid/3.5.16)❷
Connection: close

<!doctype html>
<html>
<head>
    <title>Example Domain</title>
[...]
</body>
</html>
```

範例 33.1 「使用 `squidclient` 提交的要求」中顯示的輸出可以分成兩個部分：

1. 回應的通訊協定標題：空白行前面的幾行。
2. 回應的實際內容：空白行後面的幾行。

若要驗證是否使用了 Squid，請參閱所選的標題行：

❶ 標題 `X-Cache` 的值指明所要求的文件不在電腦 `moon` 的 Squid 快取中（`MISS`）。
上面的範例包含兩行 `X-Cache`。您可以忽略第一個 `X-Cache` 標題，因為它是由來源 Web 伺服器的內部快取軟體產生。

❷ 標題 `Via` 的值指明 HTTP 版本、電腦名稱和正在使用的 Squid 的版本。

- 使用瀏覽器：將代理設定為 `localhost`，連接埠設定為 `3128`。然後，您可以載入一個頁面，並在瀏覽器的審查器或開發人員工具的網路面板中檢查回應標題。該標題應該會以範例 33.1 「使用 `squidclient` 提交的要求」中所述的類似方式再次產生。

若要允許本地系統和其他系統的使用者存取 Squid 和網際網路，請將 `/etc/squid/squid.conf` 組態檔中的 `http_access deny all` 項目變更為 `http_access allow all`。然而，當您這麼做時，請考量到此動作將使 Squid 可供任何人完全存取。因此，需定義用於控制對代理的存取權限的 ACL（存取控制清單）。修改組態檔案後，必須重新載入或重新啓動 Squid。如需 ACL 的詳細資訊，請參閱第 33.5.2 節 「存取控制的選項」。

如果 Squid 在成功啓動後不久就結束，請檢查名稱伺服器項目是否有誤，或者是否缺少 `/etc/resolv.conf` 檔案。Squid 會在 `/var/log/squid/cache.log` 檔案中記錄啓動失敗的原因。

33.3.3 停止、重新載入和重新啓動 Squid

若要重新載入 Squid，請選擇以下其中一種方法：

- 使用 `systemctl`：

```
root # systemctl reload squid
```

或

```
root # systemctl restart squid
```

- 使用 YaST：
在 Squid 模組中，按一下儲存設定並立即重新啓動 Squid。按鈕.

若要停止 Squid，請選擇以下其中一種方法：

- 使用 `systemctl`：

```
root # systemctl stop squid
```

- 使用 YaST

  在 Squid 模組中，按一下立即停止 Squid。按鈕.

關閉 Squid 可能需要一些時間，因為 Squid 會等待最長半分鐘時間，來中斷與用戶端的連接並將其資料寫入磁碟（請參閱 `/etc/squid/squid.conf` 中的 `shutdown_lifetime` 選項）。

警告：終止 Squid

使用 `kill` 或 `killall` 來終止 Squid 有可能會損毀快取。若要能夠重新啓動 Squid，必須刪除損毀的快取。

33.3.4 移除 Squid

從系統中移除 Squid 並不會移除快取階層與記錄檔案。若要移除這些階層，請手動刪除 `/var/cache/squid` 目錄。

33.3.5 本地 DNS 伺服器

即使本地 DNS 伺服器不管理自己的網域，也可以設定本地 DNS 伺服器。它可以做為僅供快取的名稱伺服器，也可以透過根 名稱伺服器來解析 DNS 要求，而不需任何特殊的組態（請參閱第 25.4 節 「啓動 BIND 名稱伺服器」）。如何達成此目的，端視您在設定網際網路連線的組態時，是否選擇動態的 DNS 而定。

動態 DNS

一般而言，使用動態 DNS 時，會由提供者在建立網際網路連接期間設定 DNS 伺服器，且 `/etc/resolv.conf` 本地檔案會自動進行調整。可以使用以下 sysconfig 變數在檔案 `/etc/sysconfig/network/config` 中控制此行為：`NETCONFIG_DNS_POLICY`。設定 `NETCONFIG_DNS_POLICY` 至 `""`（使用 YaST sysconfig 編輯器）。

然後在 `/etc/resolv.conf` 檔案中新增本地 DNS 伺服器，並將 `localhost` 的 IP 位址設定為 `127.0.0.1`。這樣，Squid 每次啓動時就能找到本地名稱伺服器。

為了使提供者的名稱伺服器可供存取，請在 `/etc/named.conf` 組態檔案中的 `forwarders` 下指定該名稱伺服器及其 IP 位址。若使用動態 DNS，則可在建立連接時自動完成上述操作，方法是將 sysconfig 變數 `NETCONFIG_DNS_POLICY` 設定為 `auto`。

靜態 DNS
: 若使用靜態 DNS，建立連接時將不會執行任何自動的 DNS 調整，所以不必變更任何 sysconfig 變數。但是您必須依照動態 DNS中所述在 `/etc/resolv.conf` 檔案中指定本地 DNS 伺服器。除此之外，還必須在 `/etc/named.conf` 檔案中的 `forwarders` 下手動指定提供者的靜態名稱伺服器及其 IP 位址。

提示：DNS 與防火牆

如果您有執行防火牆，請確定 DNS 要求可以通過防火牆。

33.4 YaST Squid 模組

YaST Squid 模組包含以下索引標籤：

啓動
: 指定啓動 Squid 的方式，以及在哪些介面上開啓哪個防火牆連接埠。

HTTP 連接埠
: 定義 Squid 將用來監聽用戶端 HTTP 要求的所有連接埠。

重新整理模式
: 定義 Squid 如何處理快取中的物件。

快取記憶體設定
: 定義有關快取記憶體、最大和最小物件大小等的設定。

快取目錄
: 定義 Squid 用來儲存所有快取交換檔案的頂層目錄。

存取控制
: 透過 ACL 群組控制對 Squid 伺服器的存取。

記錄和逾時

定義用於存取、快取和快取儲存記錄檔案的路徑，以及連接逾時和用戶端存留期。

雜項

設定管理員的語言和電子郵件地址。

33.5 Squid 組態檔案

所有的 Squid 代理伺服器的設定值都是在 `/etc/squid/squid.conf` 檔案中設定。在第一次啓動 Squid 時，此檔案不需做任何變更，但是外部用戶端一開始為拒絕存取。代理可供 `localhost` 使用。預設的連接埠是 `3128`。預先安裝的 `/etc/squid/squid.conf` 組態檔可提供關於選項及許多範例的詳細資訊。

許多項目標有備註，因此以備註字元 `#` 開頭。相關規格請見行尾。給定值一般與預設值相關聯，因此僅移除備註符號而不變更任何參數通常毫無效果。若有可能，請保留原始的備註行，在該行的下方插入選項以及修改過的值。如此一來，就可以輕易復原預設值，並與變更做比較。

提示：在更新後調整組態檔案

如果您是從較早的 Squid 版本更新，建議您編輯新的 `/etc/squid/squid.conf`，並且只套用在舊檔案中所做的變更。

有時候，該檔案中會新增、移除或修改 Squid 選項。因此，如果您嘗試使用舊的 `squid.conf`，Squid 可能會無法正常運作。

33.5.1 一般組態選項

下面列出了 Squid 的一些組態選項，但並不詳盡。`/etc/squid/squid.conf.documented` 中列出了 Squid 套件的完整選項清單，其中僅做了簡單記錄。

`http_port` 連接埠

這是 Squid 監聽用戶端要求所用的連接埠。預設的連接埠是 `3128`，但是 `8080` 也是常用的連接埠。

`cache_peer 主機名稱 類型 代理連接埠 ICP 連接埠`

此選項允許建立協同工作的快取網路。快取對等是一台同樣代管網路快取且與您自己的電腦有某種關係的電腦。類型指定關係的類型。類型可以是 `parent` 或 `sibling`。

對於主機名稱，請指定要使用的代理的名稱或 IP 位址。對於代理連接埠，請指定瀏覽器中要使用的連接埠號碼（通常為 `8080`）。將 ICP 連接埠設定為 `7`，如果上層的 ICP 連接埠未知且其用途與提供者無關，則設定為 `0`。

若要讓 Squid 以網頁瀏覽器而非代理的方式工作，請禁止使用 ICP 通訊協定。您可以附加選項 `default` 和 `no-query` 來實現此目的。

`cache_mem 大小`

此選項定義 Squid 可用於常用回覆的記憶體容量。預設值為 `8 MB`。這不指定 Squid 的記憶體使用量，而且可以超過。

`cache_dir 儲存類型 快取目錄 快取大小 層級 1 目錄 層級 2 目錄`

選項 `cache_dir` 定義磁碟快取的目錄。在 SUSE Linux Enterprise Server 上的預設組態中，Squid 不會建立磁碟快取。

預留位置儲存類型可以是下列其中一種：

- 目錄式儲存類型：`ufs`、`aufs`（預設類型）、`diskd`。這三類都是 `ufs` 儲存形式的變體。不過，雖然 `ufs` 是做為核心 Squid 執行緒的一部分執行，但 `aufs` 是在單獨的執行緒中執行，而 `diskd` 則使用單獨的程序。這表示後兩種類型可避免因磁碟 I/O 而封鎖 Squid。

- 資料庫式儲存系統：`rock`。此儲存形式依賴於單一資料庫檔案，在此檔案中，每個物件佔用固定大小的一或多個記憶體單位（「插槽」）。

下文將只介紹基於 `ufs` 的儲存類型的參數。`rock` 的參數有些不同。

快取目錄是磁碟快取的目錄，預設為 `/var/cache/squid`。快取大小是該目錄的最大大小（以 MB 為單位），預設設定為 100 MB。請將其設定為介於可用磁碟空間的 50% 和最大 80% 之間的值。

最後兩個值層級 1 目錄和層級 2 目錄指定快取目錄中建立的子目錄數。預設情況下，在快取目錄下的第一層級會建立 16 個子目錄，其中每個子目錄下又有 256 個子目錄。提高這些值時請務必謹慎，因為建立太多目錄會導致效能問題。

如果您有數個共用一個快取的磁碟，請指定數行 `cache_dir`。

`cache_access_log` 記錄檔案 ，

`cache_log` 記錄檔案 ，

`cache_store_log` 記錄檔案

這三個選項指定 Squid 記錄其所有動作的路徑。一般情況下，無需變更此處的任何設定。如果 Squid 負荷過重，則可能需要將快取與記錄檔案分散到數個磁碟上。

`client_netmask` 網路遮罩

此選項允許透過套用子網路遮罩在記錄檔案中遮罩用戶端的 IP 位址。例如，若要將 IP 位址的最後一位設定為 `0`，請指定 `255.255.255.0`。

`ftp_user` 電子郵件

此選項允許設定 Squid 應該用於匿名 FTP 登入的密碼。請在此處指定有效的電子郵件地址，因為一些 FTP 伺服器會檢查這些資料的有效性。

`cache_mgr` 電子郵件

如果 Squid 意外當機，將會向此電子郵件地址傳送一封郵件。預設值為網站管理員。

`logfile_rotate` 值

如果您執行 `squid -k rotate`，則 `Squid` 會輪替記錄檔案。在此程序中會計算檔案的數量，而且在到達指定的值後，就會覆寫最舊的檔案。預設值為 `10`，表示輪替編號為 0 到 9 的記錄檔案。

但是，在 SUSE Linux Enterprise Server 上，記錄檔案的輪替是透過使用 `logrotate` 和組態檔案 `/etc/logrotate.d/squid` 自動執行的。

`append_domain` 網域

使用 `append_domain` 可指定當未指定網域時自動附加的網域。通常可在此處指定您自己的網域，因此在瀏覽器中指定 `www` 將存取您自己的 Web 伺服器。

`forwarded_for` 狀態

如果此選項設定為 `on`，則會將如下所示的一行新增至標題：

```
X-Forwarded-For: 192.168.0.1
```

如果您將選項設定為 `off`，則 Squid 會從 HTTP 要求中移除用戶端的 IP 位址及系統名稱。

`negative_ttl 時間` ，

`negative_dns_ttl 時間`

如果設定了這些選項，Squid 將快取某些類型的失敗，例如 `404` 回應。之後，它將拒絕發出新要求，即使當時資源可供使用。

預設情況下，`negative_ttl` 設定為 `0`，`negative_dns_ttl` 設定為 `1 minute`。這表示預設情況下不會快取對 Web 要求的負面回應，但會將 DNS 要求的負面回應快取 1 分鐘。

`never_direct allow ACL 名稱`

為了防止 Squid 接受直接來自網際網路的要求，請使用選項 `never_direct` 以強制連接連到另一個代理。事先必須已在 `cache_peer` 中指定該代理。如果將 `ACL 名稱` 指定為 `all`，則所有要求將直接轉送至 `parent`。有時這可能是必要的，例如，您所使用的提供者規定了其代理的使用方式或拒絕其防火牆直接存取網際網路時。

33.5.2 存取控制的選項

Squid 會提供一個詳細系統來控制代理存取。這些存取控制清單（ACL）都是包含依順序處理的規則的清單。在使用 ACL 前必須先進行定義。某些預設的 ACL，例如 `all` 與 `localhost` 已經存在。然而，僅定義 ACL 並不代表實際上會套用。僅當存在對應的 `http_access` 規則時才會套用。

選項 `acl` 的語法如下所示：

```
acl ACL_NAME TYPE DATA
```

此語法中的預留位置含義如下所示：

- 名稱 `ACL_NAME` 可以隨意選擇。
- 對於 `TYPE`，`/etc/squid/squid.conf` 檔案的 `ACCESS CONTROLS` 區段中提供了多個不同的選項，您可以從中選取。
- `DATA` 的規格視個別的 ACL 類型而定，也可以從檔案讀取。例如，「透過」主機名稱、IP 位址或 URL。

若要在 YaST Squid 模組中新增規則，請開啓該模組，然後按一下存取控制索引標籤。在「ACL 群組」清單下按一下新增，然後輸入規則的名稱、類型及其參數。

如需 ACL 規則類型的詳細資訊，請參閱 http://www.squid-cache.org/Versions/v3/3.5/cfgman/acl.html 上的 Squid 文件。

範例 33.2 定義 ACL 規則

```
acl mysurfers srcdomain .example.com ❶
acl teachers src 192.168.1.0/255.255.255.0 ❷
acl students src 192.168.7.0-192.168.9.0/255.255.255.0 ❸
acl lunch time MTWHF 12:00-15:00 ❹
```

❶ 此 ACL 將 `mysurfers` 定義為來自 `.example.com` 中的所有使用者（由 IP 的反向查閱確定）。

❷ 此 ACL 將 `teachers` 定義為 IP 位址以 `192.1 8.1.` 開頭的電腦的使用者。

❸ 此 ACL 將 `students` 定義為 IP 位址以 `192.168.7.`、`192.168.8.` 或 `192.168.9.` 開頭的電腦的使用者。

❹ 此 ACL 將 `lunch` 定義為星期一至星期五的中午到下午 3 點之間的某個時間。

`http_access allow` ACL 名稱

`http_access` 定義哪些人可使用代理，以及哪些人可存取網際網路的哪些內容。為此必須定義 ACL。上文中已經定義了 `localhost` 和 `all`，您可以透過 `deny` 或 `allow` 拒絕或允許對它們的存取。您可以建立包含任何數量 `http_access` 項目的清單，並依從上到下的順序處理。系統會依據出現的先後順序允許或拒絕存取相應的 URL。最後一個項目應該永遠為 `http_access deny all`。在下面的範例中，`localhost` 可隨意存取任何內容，而所有其他的主機則完全被拒絕存取：

```
http_access allow localhost
http_access deny all
```

在另一個使用這些規則的範例中，`teachers` 群組永遠具有網際網路的存取權。群組 `students` 僅在星期一至星期五的午餐時間才有存取權：

```
http_access deny localhost
http_access allow teachers
http_access allow students lunch time
http_access deny all
```

基於可讀性考量，請在組態檔案 `/etc/squid/squid.conf` 中將所有 `http_access` 選項指定在一個區塊中。

`url_rewrite_program 路徑`

此選項可用於指定 URL 重寫器。例如，它可以是 squidGuard（`/usr/sbin/squidGuard`），以便允許封鎖不需要的 URL。藉由此選項，可以透過代理驗證和適當的 ACL 針對各類使用者群組單獨控制網際網路存取權限。

如需 squidGuard 的詳細資訊，請參閱第 33.8 節 「squidGuard」。

`auth_param basic program 路徑`

如果必須在代理上對使用者進行驗證，請設定對應的程式，例如 `/usr/sbin/pam_auth`。當第一次存取 `pam_auth` 時，使用者會看到需要其指定使用者名稱與密碼的登入視窗。此外，您還需要一個 ACL，以便僅讓具備有效登入的用戶端使用網際網路：

```
acl password proxy_auth REQUIRED

http_access allow password
http_access deny all
```

在 `acl proxy_auth` 選項中，使用 `REQUIRED` 表示接受所有有效的使用者名稱。`REQUIRED` 還可以替換為允許的使用者名稱清單。

`ident_lookup_access allow ACL 名稱`

藉由此選項，可以執行 ident 要求，以便為類型為 `src` 的 ACL 定義的所有用戶端確定每個使用者的身分。或者，可以針對所有用戶端使用此選項，對於 `ACL 名稱`，則套用預先定義的 ACL `all`。

`ident_lookup_access` 涵蓋的所有用戶端都必須執行 ident 精靈。在 Linux 上，您可以將 `pidentd`（`pidentd` 套件）用做 ident 精靈。對於其他作業系統，通常有免費的軟體可供使用。若要確保只有 ident 查閱成功的用戶端才有權存取，請定義對應的 ACL：

```
acl identhosts ident REQUIRED

http_access allow identhosts
http_access deny all
```

在 `acl identhosts ident` 選項中，使用 `REQUIRED` 表示接受所有有效的使用者名稱。`REQUIRED` 還可以替換為允許的使用者名稱清單。

使用 `ident` 會延緩存取時間，因為對每個要求都要重複 ident 查閱。

33.6 設定操作順暢的代理

使用代理伺服器的一般方式如下：網頁瀏覽器向代理伺服器的某個連接埠傳送要求，代理永遠都會提供要求的這些物件，而不論它們是否在其快取中。不過，在某些情況下，Squid 的透明代理模式很有效：

- 若出於安全原因，建議所有的用戶端都使用代理瀏覽網際網路。
- 若所有的用戶端都必須使用代理，不論它們是否清楚這一點。
- 若網路上的代理已轉移，但現有的用戶端需要保留其原有的組態。

透明代理會攔截並回應網頁瀏覽器的要求，因此網頁瀏覽器可收到所要求的頁面，但並不知道它們來自何處。顧名思義，整個過程對於使用者而言是透明的。

程序 33.1 SQUID 充當透明代理（指令行）

1. 在 `/etc/squid/squid.conf` 的 `http_port` 選項行中新增 `transparent` 參數：

   ```
   http_port 3128 transparent
   ```

2. 重新啓動 Squid：

   ```
   tux > sudo systemctl restart squid
   ```

3. 設定 SuSEFirewall2 以將 HTTP 流量重新導向至 `http_proxy` 中指定的連接埠（在上例中為 3128）。若要執行此動作，請編輯組態檔案 `/etc/sysconfig/SuSEfirewall2`。
   此範例假設您使用的是以下裝置：

   - 指向網際網路的裝置：`FW_DEV_EXT="eth1"`
   - 指向網路的裝置：`FW_DEV_INT="eth0"`

   在防火牆上定義供不受信任的（外部）網路（如網際網路）存取的連接埠與服務（請參閱 `/etc/services`）。在此範例中，僅提供對外的 Web 服務：

   ```
   FW_SERVICES_EXT_TCP="www"
   ```

   在從安全（內部）網路存取的防火牆上，定義埠或服務（請參閱 `/etc/services`），兩者都是透過 TCP 與 UDP 服務：

```
FW_SERVICES_INT_TCP="domain www 3128"
FW_SERVICES_INT_UDP="domain"
```

如此可允許存取 Web 服務與 Squid（預設埠為 `3128`）。「domain」 服務代表 DNS（網域名稱服務）。這個服務使用非常普遍。否則，只需從上述項目中移除 `domain` 並將以下選項設定為 `no`：

```
FW_SERVICE_DNS="yes"
```

`FW_REDIRECT` 這一選項十分重要，因為它的作用是將 HTTP 流量實際重新導向至特定的連接埠。組態檔案會在選項上面的備註中對語法進行解釋：

```
# Format:
# list of <source network>[,<destination network>,<protocol>[,dport[:lport]]
# Where protocol is either tcp or udp. dport is the original
# destination port and lport the port on the local machine to
# redirect the traffic to
#
# An exclamation mark in front of source or destination network
# means everything EXCEPT the specified network
```

即：

1. 指定存取代理防火牆的內部網路的 IP 位址與網路遮罩。
2. 指定這些用戶端的要求傳送至的 IP 位址與網路遮罩。如果是網頁瀏覽器，請將網路指定為 `0/0`，萬用字元表示「可到任何位置」
3. 指定這些要求最初傳送至的連接埠。
4. 指定要將所有這些要求重新導向至的連接埠。在下面的範例中，只有 Web 服務（連接埠 `80`）會重新導向至代理連接埠（連接埠 `3128`）。如果要新增更多網路或服務，請在相應項目中使用空格進行分隔。
   由於 Squid 支援除 HTTP 以外的通訊協定，您還可以將要求從其他連接埠重新導向至代理。例如，您還可以重新導向連接埠 21（FTP）和 443（HTTPS 或 SSL）。

因此，對於 Squid 組態，您可以使用：

```
FW_REDIRECT="192.168.0.0/16,0/0,tcp,80,3128"
```

4. 在組態檔案 `/etc/sysconfig/SuSEfirewall2` 中，請確定 `START_FW` 這一項設定為 `"yes"`。

5. 重新啓動 SuSEFirewall2：

```
tux > sudo systemctl restart SuSEfirewall2
```

6. 若要確認一切正常，請查看 `/var/log/squid/access.log` 中的 Squid 記錄檔案。若要確認是否已正確設定所有的連接埠，請從您的網路外的任何電腦對本機器上的連接埠進行掃描。只應開啓 Web 服務（埠 80）。若要使用 `nmap` 掃描連接埠，請使用以下指令：

```
nmap -O IP_ADDRESS
```

程序 33.2 SQUID 充當透明代理（YAST）

1. 啓動 YaST Squid 模組：

   a. 在*啓動*索引標籤中，啓用*在防火牆中開啓連接埠*。按一下*防火牆詳細資料*，以選取要在其上開啓連接埠的介面。僅當已啓用防火牆時，此選項才可用。

   b. 在 *HTTP 連接埠*索引標籤中，選取包含連接埠 `3128` 的第一行。

   c. 按一下*編輯*按鈕。此時會顯示一個小視窗，您可在其中編輯目前的 HTTP 連接埠。選取*透明*。

   d. 結束時按一下*確定*。

2. 依據*程序 33.1 「Squid 充當透明代理（指令行）」*的*步驟 3* 中所述設定防火牆設定。

33.7 使用 Squid 快取管理員 CGI 介面（cachemgr.cgi）

Squid 快取管理員 CGI 介面（`cachemgr.cgi`）是一種 CGI 公用程式，用來顯示執行中 Squid 程序的記憶體使用量統計資料。您也可以用它來方便地管理快取和檢視統計資料，因為不需要登入伺服器。

程序 33.3 設定 cachemgr.cgi

1. 確定 Apache Web 伺服器正在系統上執行。依第 31 章 「Apache HTTP 伺服器」中所述方式設定 Apache。請著重參閱第 31.5 節 「啓用 CGI 程序檔」。若要檢查 Apache 是否已在執行中，請使用：

```
tux > sudo systemctl status apache2
```

若顯示 `inactive`，您可以使用 SUSE Linux Enterprise Server 預設設定啓動 Apache：

```
tux > sudo systemctl start apache2
```

2. 現在，在 Apache 中啓用 `cachemgr.cgi`。為此，請為 `ScriptAlias` 建立一份組態檔案。
在目錄 `/etc/apache2/conf.d` 中建立該檔案，並將其命名為 `cachemgr.conf`。在該檔案中新增下列內容：

```
ScriptAlias /squid/cgi-bin/ /usr/lib64/squid/

<Directory "/usr/lib64/squid/">
Options +ExecCGI
AddHandler cgi-script .cgi
Require host HOST_NAME
</Directory>
```

將 `HOST_NAME` 替換為您要用於存取 `cachemgr.cgi` 的電腦的主機名稱。這樣可限制只有您的電腦方可存取 `cachemgr.cgi`。若要允許從任何位置存取該檔案，請改用 `Require all granted`。

3.
   - 如果 Squid 與您的 Apache Web 伺服器在同一台電腦上執行，則無需對 `/etc/squid/squid.conf` 進行任何變更，但需驗證 `/etc/squid/squid.conf` 是否包含以下幾行：

```
http_access allow manager localhost
http_access deny manager
```

這幾行允許您從自己的電腦（`localhost`）存取管理員介面，但不允許從其他位置存取。

- 如果 Squid 與您的 Apache Web 伺服器不在同一台電腦上執行，您需要新增其他規則，以便允許從 Squid 的 CGI 程序檔存取。為您的伺服器定義 ACL（將 `WEB_SERVER_IP` 替換為您的 Web 伺服器的 IP 位址）：

```
acl webserver src WEB_SERVER_IP/255.255.255.255
```

確定下列規則都包含在組態檔案中。與預設組態相比，只有中間的規則是新增的。不過，這些規則的前後順序非常重要。

```
http_access allow manager localhost
http_access allow manager webserver
http_access deny manager
```

4. 或者，您也可以為 `cachemgr.cgi` 設定一或多個密碼。這樣還可允許執行更多動作，例如在遠端關閉快取，或者檢視有關快取的詳細資訊。為此，請使用管理員的一或多個密碼以及允許的動作清單設定選項 `cache_mgr` 與 `cachemgr_passwd`。例如，若要明確允許在未經驗證的情況下檢視索引頁面、功能表、60 分鐘計數器的平均值，允許使用密碼 `secretpassword` 切換離線模式，以及完全停用任何其他功能，請使用以下組態：

```
cache_mgr user
cachemgr_passwd none index menu 60min
cachemgr_passwd secretpassword offline_toggle
cachemgr_passwd disable all
```

`cache_mgr` 定義使用者名稱。`cache_mgr` 定義使用哪個密碼允許哪些動作。`none` 和 `disable` 是特殊關鍵字：`none` 表示無需使用密碼，`disable` 會徹底停用功能。

登入 `cachemgr.cgi` 之後，便可全面查看完整的動作清單。若要瞭解在組態檔案中需如何參考操作，請查看動作頁面 URL 中 `&operation=` 後面的字串。`all` 是特殊關鍵字，表示所有動作。

5. 在組態檔案變更後重新載入 Squid 和 Apache：

```
tux > sudo systemctl reload squid
```

6. 若要檢視統計資料，請移至先前設定的 `cachemgr.cgi` 頁面。例如 `http://webserver.example.org/squid/cgi-bin/cachemgr.cgi`。
選擇適當的伺服器，若已設定，請指定使用者名稱和密碼。然後按一下繼續並瀏覽不同的統計資料。

33.8 squidGuard

本節目的不是說明 squidGuard 的廣泛組態，而是簡單介紹並給予使用上的一些建議。如需更深入的組態問題，請參閱 squidGuard 網站，網址為 http://www.squidguard.org。

squidGuard 屬於自由軟體 (GPL)，是一個靈活而快速的過濾器，也是重新導向器以及 Squid 的存取控制器外掛程式。它可讓您在 Squid 快取上，針對不同的使用者群組，使用不同的限制來定義多重存取規則。squidGuard 使用 Squid 的標準重新導向器介面。squidGuard 可以執行下列動作：

- 將某些使用者的 Web 存取權限制為一系列已接受或已知的 Web 伺服器或 URL。
- 針對某些使用者，封鎖某些列示或列為黑名單的網頁伺服器或 URL 的存取權。
- 針對某些使用者，封鎖符合一般運算式或文字清單的 URL。
- 將封鎖的 URL，重新導向至「智慧型」的 CGI 資訊頁面。
- 將未註冊的使用者重新導向至註冊表單。
- 將橫幅重新導向至空白的 GIF。
- 根據時間、星期、日期等，使用不同的存取規則。
- 針對不同的使用者群組，使用不同的規則。

squidGuard 與 Squid 無法用於：

- 編輯、過濾或審查文件內的文字。
- 編輯、過濾或審查 HTML 內嵌的程序檔，例如 JavaScript。

程序 33.4 設定 SQUIDGUARD

1. 使用前，請先安裝 `squidGuard`。

2. 以 `/etc/squidguard.conf` 提供最小的組態檔。組態範例請見 http://www.squidguard.org/Doc/examples.html。稍後請使用較複雜的組態設定值來測試。

3. 然後，建立一個「存取遭拒」HTML 頁面或 CGI 頁面，以便在用戶端要求存取黑名單中的網站時，Squid 可以重新導向到該頁面。強烈建議使用 Apache。

4. 現在，請將 Squid 設定為使用 squidGuard。在 `/etc/squid/squid.conf` 檔案中使用下列項目：

```
redirect_program /usr/bin/squidGuard
```

5. 另一個稱為 `redirect_children` 的選項會設定在機器上執行之「重新導向」(在此範例中為 squidGuard) 程序的數量。您設定的程序越多，所需的 RAM 就越多。先嘗試較小的數字，例如 `4`：

```
redirect_children 4
```

6. 最後，執行 `systemctl reload squid`，讓 Squid 載入新的組態。現在，請使用瀏覽器測試您的設定值。

33.9 使用 Calamaris 產生快取報告

Calamaris 是一種 Perl 程序檔，用來產生 ASCII 或 HTML 格式的快取記憶體活動報告。它使用原生的 Squid 存取記錄檔。Calamaris 的首頁網址為 http://cord.de/calamaris-english。此工具不屬於 SUSE Linux Enterprise Server 預設安裝範圍，若想使用它，可以安裝 `calamaris` 套件。

以 `root` 身分登入，然後輸入：

```
cat access1.log [access2.log access3.log] | calamaris OPTIONS > reportfile
```

如果使用的記錄檔案不止一個，請確定它們依時間順序排列，時間越早的檔案越靠前。為此，您可以如上例一般逐個列出檔案，也可以使用 `access{1..3}.log`。

`calamaris` 可使用下列選項：

`-a`
: 輸出所有可用的報告

`-w`
: 以 HTML 報告輸出

`-l`
: 在報告標題中包含訊息或標誌

在程式的手冊頁中，使用 `man calamaris` 可以找到各種選項的詳細資訊。

以下是典型的範例：

```
cat access.log.{10..1} access.log | calamaris -a -w \
> /usr/local/httpd/htdocs/Squid/squidreport.html
```

這會將報告放在網頁伺服器的目錄中。需要有 Apache 才能檢視報告。

33.10 更多資訊

請瀏覽 Squid 的首頁，網址為 http://www.squid-cache.org/。此處可以找到「Squid 使用者指南」(Squid User Guide) 以及有關 Squid 常見問題集 (FAQ) 的豐富資訊。

除此之外，還可以在 http://www.squid-cache.org/Support/mailing-lists.html 中找到 Squid 的郵件清單。

34 透過 SFCB 實作的網路企業管理

34.1 簡介與基本概念

SUSE® Linux Enterprise Server (SLES) 提供一系列基於開放式標準的工具，用於統一管理不同的計算系統和環境。我們的企業解決方案執行 Distributed Management Task Force 所建議的標準。以下段落將介紹其基本元件。

Distributed Management Task Force, Inc (DMTF) 是業內一個引領企業和網際網路環境管理標準開發的組織。該組織的目標是統一管理標準與計劃，並制定整合性更強、更為經濟有效且更具互通性的管理解決方案。DMTF 標準為實施控制和通訊提供了通用的系統管理元件。它們的解決方案與平台和技術無關。網路企業管理和 Common Information Model 是其重要技術之一。

網路企業管理 (WBEM) 是一套管理和網際網路標準技術。WBEM 的開發目的是為了統一管理企業計算環境。它能夠讓整個行業使用 Web 技術提供一系列整合性極強的管理工具。WBEM 包含以下標準：

- 資料模型：標準的通用資訊模型 (CIM)
- 編碼規格：CIM-XML 編碼規格
- 輸送機制：透過 HTTP 的 CIM 操作

Common Information Model 是一個概念性的資訊模型，對系統管理進行了描述。該模型並非限於某種實作，而且可以實現管理系統、網路、服務與應用程式之間管理資訊的交換。CIM 由兩個部分組成 — CIM 規格與 CIM 綱要。

- CIM 規格描述語言、命名和中繼綱要。中繼綱要是正式的模型定義。它定義了用來表達此模型的詞彙，以及這些詞彙的使用和語意。中繼綱要包含類別 (class)、內容 (property) 和方法 (method) 這幾個元素。它還允許將指示與關聯做為類別的類型，將參考做為內容的類型。
- CIM 綱要提供實際的模型描述。它提供一組類別，這些類別包含的屬性和關聯可提供完整解讀的概念框架，使用者可運用此框架來組織關於受管理環境的可用資訊。

共用資訊模型物件管理員（Common Information Model Object Manager，CIMOM）就是 CIM 物件管理員，更詳細說來，就是指根據 CIM 標準管理物件的應用程式。CIMOM 會管理 CIMOM 提供者與 CIM 用戶端之間的通訊，而管理員則會管理系統。

CIMOM 提供者是指透過 CIMOM 來執行用戶端應用程式所要求之特定工作的軟體。每個提供者都會執行 CIMOM 綱要的一項或多項範疇工作。這些提供者會直接與硬體互動。

Standards Based Linux Instrumentation for Manageability（SBLIM）是為實現網路企業管理（WBEM）而設計的一系列工具。SUSE® Linux Enterprise Server 使用 SBLIM 專案 Small Footprint CIM Broker 的開放原始碼 CIMOM（或 CIM 伺服器）。

Small Footprint CIM Broker 是一款適用於資源受限或內嵌式環境的 CIM 伺服器，可同時實現模組化與輕量化。它基於開放式標準，並支援 CMPI 提供者、CIM-XML 編碼和受管理物件格式（MOF）。這款伺服器不僅設定自由度高，而且在提供者當機時仍能穩定運作。此外，它還支援各種傳輸通訊協定（例如 HTTP、HTTPS、Unix Domain Socket、服務位置通訊協定（SLP）和 Java 資料庫連接（JDBC）），因此存取方便。

34.2 設定 SFCB

若要設定 Small Footprint CIM Broker（SFCB）環境，請務必在 SUSE Linux Enterprise Server 安裝期間選取 YaST 中的網路企業管理模式。或者，也可以選擇將其做為一個元件，安裝於目前正在執行的伺服器上。請確定系統上已安裝下列套件：

cim-schema，Common Information Model（CIM）綱要
: 包含 Common Information Model（CIM）。CIM 是描述網路或企業環境內所有管理資訊的模型。CIM 由規格和綱要所組成。其中的規格定義了與其他管理模型整合的詳細資訊。而綱要則會提供實際的模型描述。

cmpi-bindings-pywbem
: 包含可在 Python 中寫入並執行 CMPI 類型 CIM 提供者的介面卡。

cmpi-pywbem-base
: 包含基礎系統 CIM 提供者。

cmpi-pywbem-power-management
: 包含基於 DSP1027 的電源管理提供者。

python-pywbem

包含 Python 模組，用於透過 WBEM 通訊協定進行 CIM 操作呼叫，以查詢及更新受管理物件。

cmpi-provider-register，與 CIMOM 無關的提供者註冊公用程式

包含一個公用程式，無論 CIMOM 在系統中執行何種操作，都可讓 CMPI 提供者套件進行註冊。

sblim-sfcb，Small Footprint CIM Broker

包含 Small Footprint CIM Broker。這是一個 CIM 伺服器，執行透過 HTTP 通訊協定的 CIM 操作。這款伺服器功能全，資源佔用率低，因此特別適合內嵌式環境以及資源受限的環境。SFCB 允許透過通用管理程式介面（CMPI）寫入提供者。

sblim-sfcc

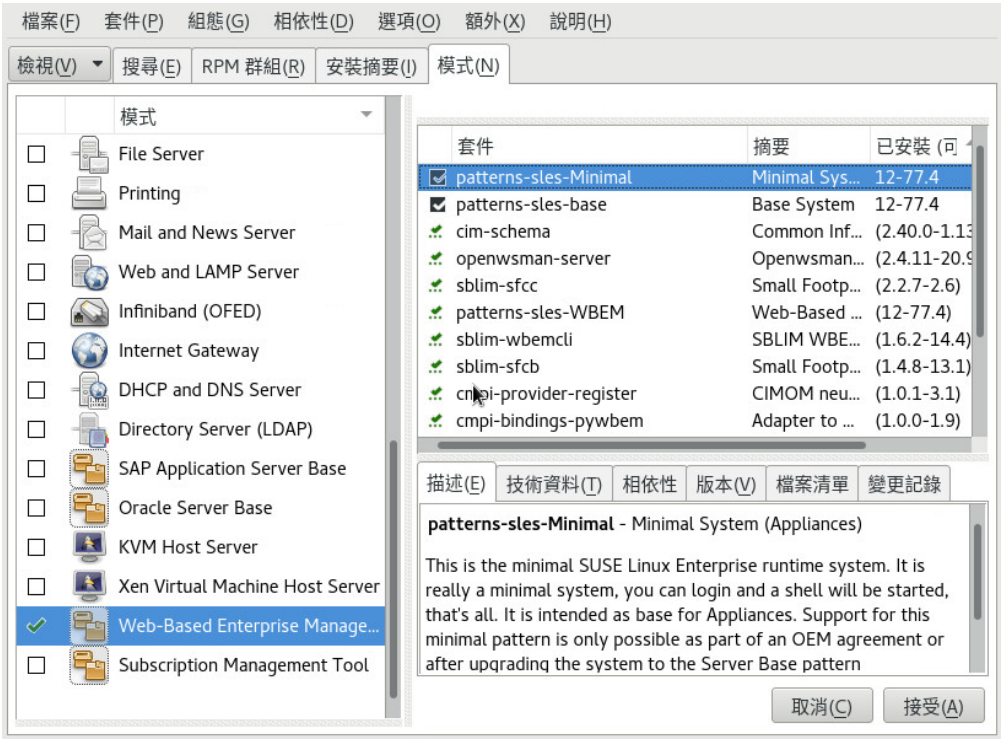
包含 Small Footprint CIM Client Library 執行時期程式庫。

sblim-wbemcli

包含 WBEM 指令行介面。它是一個獨立的指令行 WBEM 用戶端，特別適合基本系統管理任務。

smis-providers

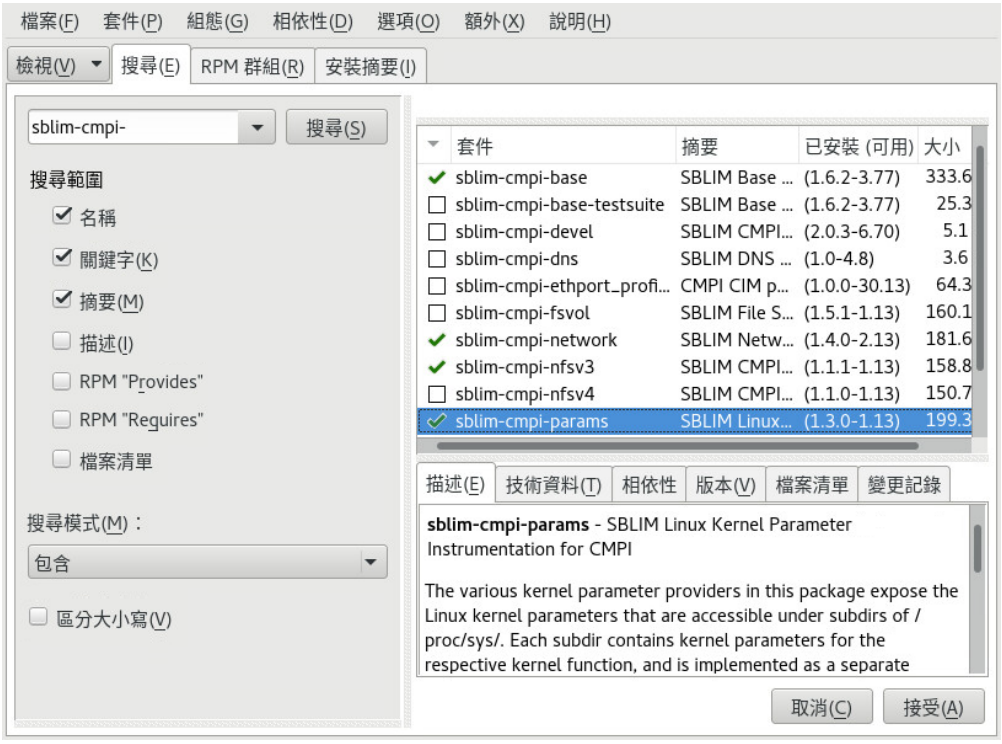
包含多個提供者，適用於 Linux 檔案系統上的磁碟區和快照。它們均基於 SNIA 的 SMI 磁碟區管理設定檔與複製服務設定檔。



圖形 34.1 網路企業管理模式的套件選擇

34.2.1 安裝其他提供者

SUSE® Linux Enterprise Server 軟體儲存庫包含網路企業管理安裝模式中沒有的其他 CIM 提供者。您可以透過在 YaST 軟體安裝模組中搜尋模式 `sblim-cmpi-`，輕鬆獲得這些提供者的清單與安裝狀態。這些提供者能夠完成各種系統管理任務，例如 DHCP、NFS 或核心參數設定。安裝要與 SFCB 配合使用的提供者很有必要。



圖形 34.2 其他 CIM 提供者的套件選擇

34.2.2 啓動、停止和檢查 SFCB 的狀態

CIM 伺服器 sfcbd 精靈隨網路企業管理軟體一起安裝，並且預設會在系統啓動時啓動。下表將說明如何啓動、停止和檢查 sfcbd 的狀態。

表格 34.1 管理 SFCBD 的指令

任務	Linux 指令
啓動 sfcbd	在指令行中以 `root` 身分輸入 `systemctl start sfcb`。
停止 sfcbd	在指令行中以 `root` 身分輸入 `systemctl stop sfcb`。
檢查 sfcbd 狀態	在指令行中以 `root` 身分輸入 `systemctl status sfcb`。

34.2.3 確保安全的存取

SFCB 的預設設定相當安全。不過，仍須檢查 SFCB 元件存取的安全級別是否符合您組織的要求。

34.2.3.1 證書

安全通訊端層（SSL）傳輸必須使用證書，才能執行安全的通訊服務。安裝 SFCB 時，會產生自行簽署的證書。

您可以透過變更 `/etc/sfcb/sfcb.cfg` 中的 `sslCertificateFilePath: PATH_FILENAME` 設定，以商用證書或自行簽署的證書的路徑取代預設證書的路徑。該檔案必須為 PEM 格式。

預設產生的伺服器證書會存放在以下位置：

`/etc/sfcb/server.pem`

注意：SSL 證書的路徑

預設產生的證書檔案 `servercert.pem` 與 `serverkey.pem` 會儲存於 `/etc/ssl/servercerts` 目錄。檔案 `/etc/sfcb/client.pem`、`/etc/sfcb/file.pem` 和 `/etc/sfcb/server.pem` 為上述檔案的符號連結。

若要產生新證書，請以 `root` 身分在指令行中輸入以下指令：

```
tux > sh /usr/share/sfcb/genSslCert.sh
Generating SSL certificates in .
Generating a 2048 bit RSA private key
.....................................................................+++
.+++
writing new private key to '/var/tmp/sfcb.0Bjt69/key.pem'
-----
```

預設情況下，程序檔會在目前的工作目錄下產生證書 `client.pem`、`file.pem` 和 `server.pem`。若要讓程序檔在 `/etc/sfcb` 目錄中產生證書，則需要在指令中附加該路徑。如果這些檔案已存在，系統只會顯示警告訊息，不會覆寫舊證書。

```
tux > sh /usr/share/sfcb/genSslCert.sh /etc/sfcb
Generating SSL certificates in .
```

```
WARNING: server.pem SSL Certificate file already exists.
         old file will be kept intact.
WARNING: client.pem SSL Certificate trust store already exists.
         old file will be kept intact.
```

必須從檔案系統中移除舊證書，然後再次執行指令。

若要變更 SFCB 使用證書的方式，請參閱第 34.2.3.3 節 「驗證」。

34.2.3.2 埠

預設情況下，SFCB 設定為接受所有透過安全連接埠 5989 傳輸的通訊。以下段落將介紹通訊連接埠設定以及建議的組態。

連接埠 5989（安全）

SFCB 通訊使用的安全連接埠，提供 HTTPS 服務。這是預設值。如果使用這項設定，當透過網際網路在伺服器和工作站之間傳送時，所有 CIMOM 和用戶端應用程式之間的通訊都會進行加密。使用者必須使用用戶端應用程式進行驗證才能連接 SFCB 伺服器。建議您保留此設定。如果用戶端應用程式和受監控的節點之間存在路由器和防火牆，則必須在這些路由器和防火牆上開啓此連接埠，SFCB CIMOM 才能與必要的應用程式進行通訊。

連接埠 5988（不安全）

SFCB 通訊使用的非安全連接埠，提供 HTTP 服務。這項設定已預設為停用。使用這項設定時，所有 CIMOM 和用戶端應用程式之間的通訊，在透過網際網路在伺服器和工作站之間傳送時都會開放，任何人無須經過任何驗證即可檢視。建議您只在嘗試為 CIMOM 相關問題除錯時使用這項設定。在解決問題後，請再次停用非安全連接埠選項。如果 SFCB CIMOM 要與需要進行不安全存取的必要應用程式進行通訊，則必須在用戶端應用程式與受監控節點之間的路由器和防火牆上開啓此連接埠。

如果您要變更預設的埠指定，請參閱第 34.2.3.2 節 「埠」。

34.2.3.3 驗證

SFCB 支援 HTTP 基本驗證和基於用戶端證書的驗證（HTTP over SSL 連接）。在 SFCB 組態檔案（預設為 `/etc/sfcb/sfcb.cfg`）中指定 `doBasicAuth =true`，可啓用基本 HTTP 驗證。SFCB 的 SUSE® Linux Enterprise Server 安裝支援可插入驗證模組（PAM）方法，因此本地 root 使用者可以使用本地 root 使用者身分證明向 SFCB CIMOM 進行驗證。

如果 `sslClientCertificate` 組態內容設為 `accept` 或 `require`，則 SFCB HTTP 介面卡會在用戶端透過 HTTP over SSL（HTTPS）連接時要求其提供證書。如果指定 `require`，則用戶端必須提供有效的證書（依據透過 `sslClientTrustStore` 指定的用戶端信任儲存區）。如果用戶端無法提供有效證書，CIM 伺服器會拒絕連接。

`sslClientCertificate =accept` 設定產生的作用可能並不明顯。不過，當要同時允許基本驗證與用戶端證書驗證時，此設定非常有用。如果用戶端能夠提供有效的證書，HTTPS 連接便會建立，並且不再執行基本驗證程序。如果此功能無法驗證該證書，則會執行 HTTP 基本驗證。

34.3 SFCB CIMOM 組態

SFCB 是 CIM 伺服器的輕量化實作，但設定的自由度很高。有多個選項可控制其行為。控制 SFCB 伺服器的方法有三種：

- 設定相應的環境變數
- 使用指令行選項
- 變更其組態檔案

34.3.1 環境變數

有多個環境變數會直接影響 SFCB 的行為。您需要透過 `systemctl restart sfcb` 重新啓動 SFCB 精靈，才能讓上述變更生效。

`PATH`
: 指定 `sfcbd` 精靈及公用程式的路徑。

`LD_LIBRARY_PATH`

指定 sfcb 執行時期程式庫的路徑。或者，您也可以將此路徑新增至系統層級的動態載入程式組態檔案 `/etc/ld.so.conf`。

`SFCB_PAUSE_PROVIDER`

指定提供者名稱。SFCB 伺服器會在首次載入提供者後暫停。此時，您可以將執行時期除錯程式附加至提供者的程序，以進行除錯。

`SFCB_PAUSE_CODEC`

指定 SFCB 轉碼器的名稱（目前僅支援 `http`）。SFCB 伺服器會在首次載入轉碼器後暫停。此時，您可以將執行時期除錯程式附加至程序。

`SFCB_TRACE`

為 SFCB 指定除錯訊息的等級。有效值為 0（無除錯訊息），或 1（重要除錯訊息）至 4（所有除錯訊息）。預設值為 1。

`SFCB_TRACE_FILE`

SFCB 預設會將其除錯訊息輸出至標準錯誤輸出 (STDERR)。若要將除錯訊息改寫到指定的檔案，可以設定此變數。

`SBLIM_TRACE`

為 SBLIM 提供者指定除錯訊息的等級。有效值為 0（無除錯訊息），或 1（重要除錯訊息）至 4（所有除錯訊息）。

`SBLIM_TRACE_FILE`

SBLIM 提供者預設會將其追蹤訊息輸出至 STDERR。若要將追蹤訊息改寫到指定的檔案，可以設定此變數。

34.3.2 指令行選項

SFCB 精靈 `sfcbd` 有多個指令行選項，可開啓或關閉特定的執行時期功能。請在 SFCB 精靈啓動時輸入這些選項。

`-c、--config-file=檔案`

SFCB 精靈啓動時，預設會從 `/etc/sfcb/sfcb.cfg` 中讀取組態。借助此選項，可以指定替代的組態檔案。

`-d、--daemon`

強制 sfcbd 及其子程序在背景執行。

`-s、--collect-stats`

開啓執行時期的統計資料收集。各種 sfcbd 執行時期統計資料將被寫入到目前工作目錄下的 `sfcbStat` 檔案。預設情況下，不會收集任何統計資料。

`-l、--syslog-level=記錄層級`

指定系統記錄機能的詳細度。`記錄層級` 可以是 LOG_INFO、LOG_DEBUG 或 LOG_ERR，預設為 LOG_ERR。

`-k、--color-trace=記錄層級`

使用不同的色彩列印各程序的追蹤輸出，以方便您除錯。

`-t、--trace-components=數字`

啓動元件層級的訊息追蹤，其中 `數字` 為使用 OR 指定的位元遮罩整數，用於定義要追蹤的元件。如果指定 `-t ?`，會列出所有元件及其關聯的整數位元遮罩：

```
tux > sfcbd -t ?
---   Traceable Components:     Int       Hex
---             providerMgr:       1  0x0000001
---             providerDrv:       2  0x0000002
---              cimxmlProc:       4  0x0000004
---              httpDaemon:       8  0x0000008
---                 upCalls:      16  0x0000010
---                encCalls:      32  0x0000020
---         ProviderInstMgr:      64  0x0000040
---        providerAssocMgr:     128  0x0000080
---                providers:     256  0x0000100
---              indProvider:     512  0x0000200
---        internalProvider:    1024  0x0000400
---              objectImpl:    2048  0x0000800
---                   xmlIn:    4096  0x0001000
---                  xmlOut:    8192  0x0002000
---                 sockets:   16384  0x0004000
---               memoryMgr:   32768  0x0008000
---                msgQueue:   65536  0x0010000
---              xmlParsing:  131072  0x0020000
---          responseTiming:  262144  0x0040000
---               dbpdaemon:  524288  0x0080000
---                     slp: 1048576  0x0100000
```

`-t 2019` 會顯示 sfcbd 的內部函數，但不會產生過多訊息，因此很有用。

34.3.3 SFCB 組態檔案

SFCB 會在啓動後從組態檔案 `/etc/sfcb/sfcb.cfg` 讀取其執行時期組態。若想覆寫啓動時的這一行為，可以使用 `-c` 選項。

組態檔案包含多組 `option : VALUE`，每行一組。變更此檔案時，可以使用以所用環境原生格式儲存檔案的任何文字編輯器。

如果設定中包含以數字符號（#）設為備註的選項，則該設定會使用預設設定。

下面的選項清單可能不完整。如需完整的清單，請參閱 `/etc/sfcb/sfcb.cfg` 和 `/usr/share/doc/packages/sblim-sfcb/README` 中的相關內容。

34.3.3.1 httpPort

用途

指定 sfcbd 監聽來自 CIM 用戶端之 HTTP（不安全）請求的本地連接埠值。預設值為 `5988`。

語法

`httpPort: PORT_NUMBER`

34.3.3.2 enableHttp

用途

指定 SFCB 是否應接受 HTTP 用戶端連接。預設值為 `false`。

語法

`enableHttp: OPTION`

選項	描述
true	啓用 HTTP 連接。
false	停用 HTTP 連接。

34.3.3.3 httpProcs

用途

指定可同時建立之 HTTP 用戶端連接的最大數量。達到該數量後，所有新的內送 HTTP 請求都將遭到阻擋。預設值為 `8`。

語法

```
httpProcs: MAX_NUMBER_OF_CONNECTIONS
```

34.3.3.4 httpUserSFCB、httpUser

用途

這些選項可控制 HTTP 伺服器將以何使用者的身分執行。如果 `httpUserSFCB` 為 `true`，HTTP 將以 SFCB 主程序所用的相同使用者身分執行。如果為 `false`，將使用為 `httpUser` 指定的使用者名稱。此設定用於 HTTP 和 HTTPS 兩種伺服器。若 `httpUserSFCB` 設定為 false，就必須指定 `httpUser`。預設值為 `true`。

語法

```
httpUserSFCB: true
```

34.3.3.5 httpLocalOnly

用途

指定是否將 HTTP 要求局限於本地主機。預設值為 `false`。

語法

```
httpLocalOnly: false
```

34.3.3.6 httpsPort

用途

指定 sfcbd 監聽來自 CIM 用戶端之 HTTPS 請求的本地連接埠值。預設值為 `5989`。

語法

```
httpsPort: port_number
```

34.3.3.7 enableHttps

用途

指定 SFCB 是否接受 HTTPS 用戶端連接。預設值為 `true`。

語法

```
enableHttps: option
```

選項	描述
true	啓用 HTTPS 連接。
false	停用 HTTPS 連接。

34.3.3.8 httpsProcs

用途

指定可同時建立之 HTTPS 用戶端連接的最大數量。達到該數量後，所有新的內送 HTTPS 請求都將遭到阻擋。預設值為 `8`。

語法

`httpsProcs: MAX_NUMBER_OF_CONNECTIONS`

34.3.3.9 enableInterOp

用途

指定 SFCB 是否提供 `interop` 名稱空間，用於指示支援。預設值為 `true`。

語法

`enableInterOp: OPTION`

選項	描述
true	啓用 interop 名稱空間。

選項	描述
false	停用 interop 名稱空間。

34.3.3.10 provProcs

用途

指定可同時執行之提供者程序的最大數量。達到該數量之後，如果有新的內送請求要求載入新提供者，則會先自動卸載某個現有的提供者。預設值為 32。

語法

provProcs: MAX_NUMBER_OF_PROCS

34.3.3.11 doBasicAuth

用途

接受請求之前，依據用戶端使用者識別碼開啓或關閉基本驗證。預設值為 true，表示系統會執行基本用戶端驗證。

語法

doBasicAuth: OPTION

選項	描述
true	啓用基本驗證。

選項	描述
false	停用基本驗證。

34.3.3.12 basicAuthLib

用途

指定本地程式庫名稱。SFCB 伺服器會載入程式庫，以驗證用戶端使用者識別碼。預設值為 `sfcBasicPAMAuthentication`。

語法

`provProcs: MAX_NUMBER_OF_PROCS`

34.3.3.13 useChunking

用途

此選項可啓用或停用 HTTP/HTTPS「區塊化」功能。如果啓用，伺服器將透過多個較小的「區塊」將大量回應資料傳回用戶端，而不是緩衝資料並集中在一個區塊中全部送回。預設值為 `true`。

語法

`useChunking: OPTION`

選項	描述
true	啓用 HTTP/HTTPS 資料區塊化功能。
false	停用 HTTP/HTTPS 資料區塊化功能。

34.3.3.14 keepaliveTimeout

用途

指定 SFCB HTTP 程序在同一個連接上兩次請求間的最長等待時間（秒）。達到此時間後，程序將終止。如果將其設定為 `0`，則會停用 HTTP 保持連線功能。預設值為 `0`。

語法

```
keepaliveTimeout: SECS
```

34.3.3.15 keepaliveMaxRequest

用途

指定一個連接上連續請求的最大數量。如果將其設定為 `0`，則會停用 HTTP 保持連線功能。預設值為 `10`。

語法

```
keepaliveMaxRequest: NUMBER_OF_CONNECTIONS
```

34.3.3.16 registrationDir

用途

指定註冊目錄，其中包含提供者註冊資料、階段區域和靜態儲存庫。預設值為 `/var/lib/sfcb/registration`。

語法

`registrationDir: DIR`

34.3.3.17 providerDirs

用途

指定 SFCB 從中搜尋提供者程式庫的目錄清單，以空格分隔。預設值為 `/usr/lib64 /usr/lib64 /usr/lib64/cmpi`。

語法

`providerDirs: DIR`

34.3.3.18 providerSampleInterval

用途

指定提供者管理員檢查閒置提供者的間隔（秒）。預設值為 `30`。

語法

`providerSampleInterval: SECS`

34.3.3.19 providerTimeoutInterval

用途

指定提供者在閒置多少時間（秒）後由提供者管理員卸載。預設值為 `60`。

語法

`providerTimeoutInterval: SECS`

34.3.3.20 providerAutoGroup

用途

如果提供者註冊檔案未指定其他任何群組，並且該選項設定為 `true`，則同一個共享程式庫中的所有提供者都會在同一個程序中執行。

語法

`providerAutoGroup: OPTION`

選項	描述
true	啓用提供者分組。
false	停用提供者分組。

34.3.3.21 sslCertificateFilePath

用途

指定包含伺服器證書的檔案名稱。檔案必須為 PEM（隱私增强郵件，採用 RFC 1421 與 RFC 1424 標準）格式。不過，此檔案僅在 `enableHttps` 設定為 `true` 時才要求使用。預設值為 `/etc/sfcb/server.pem`。

語法

`sslCertificateFilePath: PATH`

34.3.3.22 sslKeyFilePath

用途

指定包含伺服器證書之私密金鑰的檔案名稱。該檔案必須為 PEM 格式，可以不含密碼片語保護。不過，此檔案僅在 `enableHttps` 設定為 `true` 時才要求使用。預設值為 `/etc/sfcb/file.pem`。

語法

`sslKeyFilePath: PATH`

34.3.3.23 sslClientTrustStore

用途

指定包含 CA 或自行簽署之客戶端證書的檔案名稱。此檔必須為 PEM 格式，不過僅在 `sslClientCertificate` 設定為 `accept` 或 `require` 時才要求使用。預設值為 `/etc/sfcb/client.pem`。

語法

`sslClientTrustStore: PATH`

34.3.3.24 sslClientCertificate

用途

指定 SFCB 處理用戶端證書驗證的方式。如果設定為 `ignore`，則不會要求用戶端提供證書。如果設定為 `accept`，則會要求用戶端提供證書，不過即使用戶端不提供，操作也不會失敗。如果設定為 `require`，則會在客戶端不提供證書時拒絕客戶端連接。預設值為 `ignore`。

語法

`sslClientCertificate: OPTION`

選項	描述
ignore	停用用戶端證書的請求。
accept	停用用戶端證書的請求。 即使不提供證書，也不會失敗。
require	如果沒有有效的證書，便拒絕用戶端連接。

34.3.3.25 certificateAuthLib

用途

指定本地程式庫的名稱，以便請求基於用戶端證書的使用者驗證。僅當 `sslClientCertificate` 未設定為 `ignore` 時才需要指定。預設值為 `sfcCertificateAuthentication`。

語法

`certificateAuthLib: FILE`

34.3.3.26 traceLevel

用途

指定 SFCB 的追蹤層級。您可以透過設定環境變數 `SFCB_TRACE_LEVEL` 的方式對其進行覆寫。預設值為 `0`。

語法

`traceLevel: NUM_LEVEL`

34.3.3.27 traceMask

用途

指定 SFCB 的追蹤遮罩。您可以透過指令行選項 `--trace-components` 對其進行覆寫。預設值為 `0`。

語法

`traceMask: MASK`

34.3.3.28 traceFile

用途

指定 SFCB 的追蹤檔案。您可以透過設定環境變數 `SFCB_TRACE_FILE` 的方式對其進行覆寫。預設值為 `stderr`（標準錯誤輸出）。

語法

`traceFile: OUTPUT`

34.4 進階 SFCB 任務

本章介紹有關 SFCB 使用的更多進階主題。為了便於理解，您需要先掌握 Linux 檔案系統的基本知識，並具備 Linux 指令行的操作經驗。本章包括以下任務：

- 安裝 CMPI 提供者
- 測試 SFCB
- 使用 `wbemcli` CIM 用戶端

34.4.1 安裝 CMPI 提供者

若要安裝 CMPI 提供者，需要確定其共享程式庫複製到 `providerDirs` 組態選項指定的其中一個目錄，請參閱第 34.3.3.17 節 「providerDirs」。還必須使用 `sfcbstage` 與 `sfcbrepos` 指令正確註冊提供者。

系統通常會為 SFCB 準備提供者套件，以便安裝程序能夠正確進行註冊。此外，還為 SFCB 準備了大多數 SBLIM 提供者。

34.4.1.1 類別儲存庫

類別儲存庫是 SFCB 儲存 CIM 類別之相關資訊的位置。它通常由包含名稱空間元件的目錄樹組成。常見的 CIM 名稱空間為 `root/cimv2` 或 `root/interop`，它們會分別轉譯為檔案系統上的類別儲存庫目錄路徑

`/var/lib/sfcb/registration/repository/root/cimv2`

和

`/var/lib/sfcb/registration/repository/root/interop`

每個名稱空間目錄都包含檔案 `classSchemas`。該檔案內有所有在該名稱空間下註冊之 CIM 類別的編譯二進位表示。同時還包含有關其 CIM 超類別的必要資訊。

此外，每個名稱空間目錄可能會包含檔案 `qualifiers`，內有該命名空間的所有修飾詞。Sfcbd 重新啓動時，類別提供者會掃描目錄 `/var/lib/sfcb/registration/repository/` 及其所有子目錄，以確定註冊的名稱空間。然後解碼 `classSchemas` 檔案，並建立每個名稱空間的類別階層。

34.4.1.2 新增類別

SFCB 無法執行線上 CIM 類別操作。您需要在離線模式下新增、變更或移除類別，然後使用 `systemctl restart sfcb` 重新啓動 SFCB 服務以註冊變更。

儲存提供者類別和註冊資訊時，SFCB 會使用一個稱為階段區域的位置。在 SUSE® Linux Enterprise Server 系統中，即 `/var/lib/sfcb/stage/` 下的目錄結構。

若要新增提供者，需要：

- 將提供者類別定義檔案複製到階段區域目錄（`/var/lib/sfcb/stage/mofs`）下的 `./mofs` 子目錄。
- 將包含類別名稱或提供者類型，以及可執行程式庫檔案名稱的註冊檔案複製到 `./regs` 子目錄。

階段目錄中有兩個預設「mof」（類別定義）檔案：`indication.mof` 與 `interop.mof`。執行 `sfcbrepos` 指令後，根階段目錄 `/var/lib/sfcb/stage/mofs` 下的 MOF 檔案將複製到每個名稱空間中。`interop.mof` 只會編譯至 `interop` 名稱空間。

目錄配置可能如下例所示：

```
tux > ls /var/lib/sfcb/stage
```

```
default.reg  mofs  regs
```

```
tux > ls /var/lib/sfcb/stage/mofs
indication.mof  root
```

```
tux > ls /var/lib/sfcb/stage/mofs/root
cimv2  interop  suse  virt
```

```
tux > ls -1 /var/lib/sfcb/stage/mofs/root/cimv2 | less
Linux_ABIParameter.mof
Linux_BaseIndication.mof
Linux_Base.mof
Linux_DHCPElementConformsToProfile.mof
Linux_DHCPEntity.mof
[..]
OMC_StorageSettingWithHints.mof
OMC_StorageVolumeDevice.mof
OMC_StorageVolume.mof
OMC_StorageVolumeStorageSynchronized.mof
OMC_SystemStorageCapabilities.mof
```

```
tux > ls -1 /var/lib/sfcb/stage/mofs/root/interop
ComputerSystem.mof
ElementConformsToProfile.mof
HostSystem.mof
interop.mof
Linux_DHCPElementConformsToProfile.mof
[..]
OMC_SMIElementSoftwareIdentity.mof
OMC_SMISubProfileRequiresProfile.mof
OMC_SMIVolumeManagementSoftware.mof
ReferencedProfile.mof
RegisteredProfile.mof
```

```
tux > ls -1 /var/lib/sfcb/stage/regs
AllocationCapabilities.reg
Linux_ABIParameter.reg
Linux_BaseIndication.reg
Linux_DHCPGlobal.reg
Linux_DHCPRegisteredProfile.reg
[..]
OMC_Base.sfcb.reg
OMC_CopyServices.sfcb.reg
OMC_PowerManagement.sfcb.reg
OMC_Server.sfcb.reg
RegisteredProfile.reg
```

```
tux > cat /var/lib/sfcb/stage/regs/Linux_DHCPRegisteredProfile.reg
[Linux_DHCPRegisteredProfile]
   provider: Linux_DHCPRegisteredProfileProvider
```

```
   location: cmpiLinux_DHCPRegisteredProfile
   type: instance
   namespace: root/interop
#
[Linux_DHCPElementConformsToProfile]
   provider: Linux_DHCPElementConformsToProfileProvider
   location: cmpiLinux_DHCPElementConformsToProfile
   type: instance association
   namespace: root/cimv2
#
[Linux_DHCPElementConformsToProfile]
   provider: Linux_DHCPElementConformsToProfileProvider
   location: cmpiLinux_DHCPElementConformsToProfile
   type: instance association
   namespace: root/interop
```

SFCB 會對每個提供者使用一個自訂的提供者註冊檔案。

注意：SBLIM 提供者註冊檔案

SBLIM 網站上的所有 SBLIM 提供者都已包含用於產生 SFCB 之 .reg 檔案的註冊檔案。

SFCB 註冊檔案的格式如下：

```
[<class-name>]
   provider: <provide-name>
   location: <library-name>
   type: [instance] [association] [method] [indication]
   group: <group-name>
   unload: never
   namespace: <namespace-for-class> ...
```

其中，

`<class-name>`

CIM 類別名稱（必要）

`<provider-name>`

CMPI 提供者名稱（必要）

`<location-name>`

提供者程式庫名稱（必要）

`type`

提供者類型（必要）可以是 `instance`、`association`、`method` 或 `indication` 的任意組合。

`<group-name>`

可以將多個提供者組合在一起，在單一程序下執行，以便進一步減少對執行時期資源的佔用。在相同 <group-name> 下註冊的所有提供者會在同一程序下執行。預設情況下，每個提供者會做為獨立的程序執行。

`unload`

指定提供者的卸載規則。目前唯一支援的選項為 `never`，即不監控提供者的閒置時間，也不卸載提供者。預設情況下，當提供者的閒置時間超過組態檔案中指定的值時，就會被卸載。

`namespace`

可以執行此提供者之名稱空間的清單。這是必要選項，雖然對大多數提供者而言，此值均為 `root/cimv2`。

在階段區域中儲存所有類別定義和提供者註冊檔案後，需要使用指令 `sfcbrepos -f` 重建 SFCB 類別儲存庫。

您可以使用此方式新增、變更或移除類別。重建類別儲存庫後，使用指令 `systemctl restart sfcb` 重新啓動 SFCB。

此外，SFCB 套件還包含一個公用程式，會將提供者類別 mof 檔案和註冊檔案複製到階段區域中的正確位置。

`sfcbstage -r [provider.reg] [class1.mof] [class2.mof] ...`

執行此指令行後，仍需重建類別儲存庫並重新啓動 SFCB 服務。

34.4.2 測試 SFCB

SFCB 套件包含兩個測試程序檔：`wbemcat` 和 `xmltest`。

`wbemcat` 會透過 HTTP 通訊協定將原始 CIM-XML 資料傳送至在連接埠 5988 上監聽的指定 SFCB 主機（預設為 localhost）。然後顯示傳回的結果。以下檔案包含標準 EnumerateClasses 請求的 CIM-XML 表示：

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
  <MESSAGE ID="4711" PROTOCOLVERSION="1.0">
    <SIMPLEREQ>
      <IMETHODCALL NAME="EnumerateClasses">
        <LOCALNAMESPACEPATH>
          <NAMESPACE NAME="root"/>
          <NAMESPACE NAME="cimv2"/>
        </LOCALNAMESPACEPATH>
        <IPARAMVALUE NAME="ClassName">
          <CLASSNAME NAME=""/>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="DeepInheritance">
          <VALUE>TRUE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="LocalOnly">
          <VALUE>FALSE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="IncludeQualifiers">
          <VALUE>FALSE</VALUE>
        </IPARAMVALUE>
        <IPARAMVALUE NAME="IncludeClassOrigin">
          <VALUE>TRUE</VALUE>
        </IPARAMVALUE>
      </IMETHODCALL>
    </SIMPLEREQ>
  </MESSAGE>
</CIM>
```

將此請求傳送至 SFCB CIMOM 會傳回有註冊提供者之所有支援類別的清單。假設將檔案儲存為 `cim_xml_test.xml` 。

```
tux > wbemcat cim_xml_test.xml | less
HTTP/1.1 200 OK
Content-Type: application/xml; charset="utf-8"
Content-Length: 337565
Cache-Control: no-cache
CIMOperation: MethodResponse

<?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateClasses">
[..]
<CLASS NAME="Linux_DHCPParamsForEntity" SUPERCLASS="CIM_Component">
<PROPERTY.REFERENCE NAME="GroupComponent" REFERENCECLASS="Linux_DHCPEntity">
</PROPERTY.REFERENCE>
<PROPERTY.REFERENCE NAME="PartComponent" REFERENCECLASS="Linux_DHCPParams">
</PROPERTY.REFERENCE>
</CLASS>
</IRETURNVALUE>
</IMETHODRESPONSE>
```

```
</SIMPLERSP>
</MESSAGE>
</CIM>
```

列出的類別會視系統上安裝的提供者而有所不同。

第二個程序檔 `xmltest` 也用於將原始 CIM-XML 測試檔案傳送至 SFCB CIMOM。傳送之後，它會將傳回的結果與之前儲存的「OK」結果檔案進行比較。如果不存在對應的「OK」檔案，則會予以建立，供後續使用。

```
tux > xmltest cim_xml_test.xml
Running test cim_xml_test.xml ... OK
        Saving response as cim_xml_test.OK
root # xmltest cim_xml_test.xml
Running test cim_xml_test.xml ... Passed
```

34.4.3 指令行 CIM 用戶端：wbemcli

除了 `wbemcat` 與 `xmltest` 外，SBLIM 專案還包含更進階的指令行 CIM 用戶端 `wbemcli`。該用戶端用於向 SFCB 伺服器傳送 CIM 請求，並顯示傳回的結果。它不依賴 CIMOM 程式庫，並可用於所有與 WBEM 相容的實作。

例如，如果需要列出由註冊到您 SFCB 的 SBLIM 提供者執行的所有類別，可向 SFCB 傳送 「EnumerateClasses」 (ec) 要求：

```
tux > wbemcli -dx ec http://localhost/root/cimv2
To server: <?xml version="1.0" encoding="utf-8" ?>
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0"><SIMPLEREQ><IMETHODCALL \
    NAME="EnumerateClasses"><LOCALNAMESPACEPATH><NAMESPACE NAME="root"> \
    </NAMESPACE><NAMESPACE NAME="cimv2"></NAMESPACE> \
    </LOCALNAMESPACEPATH>
<IPARAMVALUE NAME="DeepInheritance"><VALUE>TRUE</VALUE> \
    </IPARAMVALUE>
<IPARAMVALUE NAME="LocalOnly"><VALUE>FALSE</VALUE></IPARAMVALUE>
<IPARAMVALUE NAME="IncludeQualifiers"><VALUE>FALSE</VALUE> \
    </IPARAMVALUE>
<IPARAMVALUE NAME="IncludeClassOrigin"><VALUE>TRUE</VALUE> \
    </IPARAMVALUE>
</IMETHODCALL></SIMPLEREQ>
</MESSAGE></CIM>
From server: Content-Type: application/xml; charset="utf-8"
From server: Content-Length: 337565
From server: Cache-Control: no-cache
From server: CIMOperation: MethodResponse
From server: <?xml version="1.0" encoding="utf-8" ?>
```

```
<CIM CIMVERSION="2.0" DTDVERSION="2.0">
<MESSAGE ID="4711" PROTOCOLVERSION="1.0">
<SIMPLERSP>
<IMETHODRESPONSE NAME="EnumerateClasses">
<IRETURNVALUE>
<CLASS NAME="CIM_ResourcePool" SUPERCLASS="CIM_LogicalElement">
<PROPERTY NAME="Generation" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="ElementName" TYPE="string">
</PROPERTY>
<PROPERTY NAME="Description" TYPE="string">
</PROPERTY>
<PROPERTY NAME="Caption" TYPE="string">
</PROPERTY>
<PROPERTY NAME="InstallDate" TYPE="datetime">
</PROPERTY>
[..]
<CLASS NAME="Linux_ReiserFileSystem" SUPERCLASS="CIM_UnixLocalFileSystem">
<PROPERTY NAME="FSReservedCapacity" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="TotalInodes" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="FreeInodes" TYPE="uint64">
</PROPERTY>
<PROPERTY NAME="ResizeIncrement" TYPE="uint64">
<VALUE>0</VALUE>
</PROPERTY>
<PROPERTY NAME="IsFixedSize" TYPE="uint16">
<VALUE>0</VALUE>
</PROPERTY>
[..]
```

`-dx` 選項會顯示由 `wbemcli` 傳送至 SFCB 的實際 XML，以及實際收到的 XML。在上例中，傳回的第一個類別是 `CIM_ResourcePool`，然後是 `Linux_ReiserFileSystem`。所有其他註冊類別也會顯示類似的項目。

如果省略 `-dx` 選項，則 `wbemcli` 只會顯示傳回資料的精簡表示：

```
tux > wbemcli ec http://localhost/root/cimv2
localhost:5988/root/cimv2:CIM_ResourcePool Generation=,ElementName=, \
    Description=,Caption=,InstallDate=,Name=,OperationalStatus=, \
    StatusDescriptions=,Status=,HealthState=,PrimaryStatus=, \
    DetailedStatus=,OperatingStatus=,CommunicationStatus=,InstanceID=, \
    PoolID=,Primordial=,Capacity=,Reserved=,ResourceType=, \
    OtherResourceType=,ResourceSubType=, \AllocationUnits=
localhost:5988/root/cimv2:Linux_ReiserFileSystem FSReservedCapacity=, \
    TotalInodes=,FreeInodes=,ResizeIncrement=,IsFixedSize=,NumberOfFiles=, \
    OtherPersistenceType=,PersistenceType=,FileSystemType=,ClusterSize=, \
    MaxFileNameLength=,CodeSet=,CasePreserved=,CaseSensitive=, \
    CompressionMethod=,EncryptionMethod=,ReadOnly=,AvailableSpace=, \
    FileSystemSize=,BlockSize=,Root=,Name=,CreationClassName=,CSName=, \
```

```
    CSCreationClassName=,Generation=,ElementName=,Description=,Caption=, \
    InstanceID=,InstallDate=,OperationalStatus=,StatusDescriptions=, \
    Status=,HealthState=,PrimaryStatus=,DetailedStatus=,OperatingStatus= \
    ,CommunicationStatus=,EnabledState=,OtherEnabledState=,RequestedState= \
    ,EnabledDefault=,TimeOfLastStateChange=,AvailableRequestedStates=, \
    TransitioningToState=,PercentageSpaceUse=
    [..]
```

34.5 更多資訊

如需有關 WBEM 和 SFCB 的詳細資料，請參閱以下來源：

http://www.dmtf.org
: Distributed Management Task Force 網站

http://www.dmtf.org/standards/wbem/
: 網路企業管理 (WBEM) 網站

http://www.dmtf.org/standards/cim/
: Common Information Model (CIM) 網站

http://sblim.wiki.sourceforge.net/
: Standards Based Linux Instrumentation (SBLIM) 網站

http://sblim.sourceforge.net/wiki/index.php/Sfcb
: Small Footprint CIM Broker (SFCB) 網站

http://sblim.sourceforge.net/wiki/index.php/Providers
: SBLIM 提供者套件

V 行動電腦

35 Linux 的行動計算功能

人們多半會將行動計算與筆記型電腦、PDA、行動電話以及它們彼此間的資料交換聯想在一起。行動硬體元件（如外接硬碟、隨身碟或數位相機）可以連接到筆記型電腦或桌上電腦系統。行動計算環境需要許多軟體元件，而且有些應用程式是專為行動用途量身訂製的。

35.1 筆記型電腦

筆記型電腦的硬體與一般桌上電腦系統不同。因為可交換性、空間要求和耗電量之類的準則必須考慮在內。行動硬體的製造廠商開發了一些標準介面，例如 PCMCIA（國際個人電腦記憶卡協會）、Mini PCI 和 Mini PCIe，可以使用它們來擴充筆記型電腦的硬體。這些標準涵蓋了記憶卡、網路介面卡和外接硬碟。

35.1.1 省電

製造筆記型電腦時採用了電能優化系統元件，因此不必靠電力系統就可以使用該類電腦。它們在省電方面的功效不亞於作業系統。SUSE® Linux Enterprise Server 支援各種控制筆記型電腦電源消耗的方法，在使用電池電源時，這些方法對電腦運作時間的影響各不相同。下表按照省電作用從高到低排列：

- 調節 CPU 速度。
- 暫停時關閉顯示器亮度。
- 手動調整顯示器亮度。
- 將不使用的熱插拔配件（USB CD-ROM、外接滑鼠、不使用的 PCMCIA 卡、Wi-Fi 等）斷開連接。
- 閒置時降低硬碟轉速。

關於 SUSE Linux Enterprise Server 中電源管理的詳細背景資訊，請參閱第 37 章 「電源管理」。

35.1.2 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。許多服務與環境息息相關，因而基礎用戶端必須經過重新設定。SUSE Linux Enterprise Server 會為您處理此工作。

圖形 35.1 將行動電腦整合到現有環境中

筆記型電腦在小型家用網路及公司網路之間來回交替使用的情形，影響的服務包括：

網路

包括 IP 位址指定、名稱解析、網際網路連接、以及與其他網路的連接。

列印

視網路而定，必須出現可用印表機及可用列印伺服器目前的資料庫。

電子郵件和代理

列印時，對應伺服器的清單必須是最新的。

X（圖形環境）

如果您的筆記型電腦暫時性地連接到投影機或外接顯示器，則必須要有不同的顯示器組態。

SUSE Linux Enterprise Server 提供了數種將筆記型電腦整合到現有作業環境的方法：

NetworkManager

它能讓您在多種網路環境或不同類型的網路（例如 GPRS、EDGE 或 3G 等行動寬頻、無線 LAN 和乙太網路）之間輕鬆地自動切換。NetworkManager 支援無線 LAN 中的 WEP 與 WPA-PSK 加密，還支援撥號連接。GNOME 桌面包含 NetworkManager 的前端。如需詳細資訊，請參閱第 36.3 節 「設定網路連線」。

表格 35.1 NETWORKMANAGER 的使用案例

我的電腦.....	使用 NetworkManager
是筆記型電腦	是
有時候會連接到不同網路	是
提供網路服務（例如 DNS 或 DHCP）	否
只使用靜態 IP 位址	否

在不應由 NetworkManager 處理網路組態時，請使用 YaST 工具來設定網路連線。

提示：DNS 組態和各種類型的網路連接

如果您常常攜帶筆記型電腦出差，並且需要變更不同類型的網路連接，則只要使用 DHCP 正確指定所有 DNS 位址，NetworkManager 便可正常運作。如果某些連接使用靜態 DNS 位址，請將該位址新增至 `/etc/sysconfig/network/config` 中的 `NETCONFIG_DNS_STATIC_SERVERS` 選項。

SLP

服務位置通訊協定（SLP）簡化了筆記型電腦到現有網路的連接。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 還可用來安裝系統，縮短對適用安裝來源的搜尋時間。如需有關 SLP 的詳細資訊，請參閱第 30 章 「SLP」。

35.1.3 軟體選項

在行動使用場合有各種任務領域，它們由專屬軟體實現：系統監控（特別是電池充電）、資料同步化，以及與週邊和網際網路的無線通訊。以下幾節涵蓋了 SUSE Linux Enterprise Server 為每項任務提供的最重要的應用程式。

35.1.3.1 系統監控

SUSE Linux Enterprise Server 提供了兩種系統監控工具：

電源管理

電源管理是一款可讓您調整與 GNOME 桌面行為關聯之節能的應用程式。通常，您可以透過電腦 › 控制中心 › 系統 › 電源管理來存取該應用程式。

系統監視器

系統監視器會將度量系統參數收集到一個監控環境中。依預設，它會在三個索引標籤中顯示輸出資訊。程序提供目前執行之程序的詳細資訊，例如 CPU 負載、記憶體使用率，或程序 ID 編號及優先程度。您可以自訂收集之資料的顯示和過濾方式。若要新增新類型的程序資訊，請按一下程序表的標題，然後選擇要隱藏或

新增至檢視窗的欄。還可以在各資料頁監控不同的系統參數，或透過網路同時收集各種機器的資料。資源索引標籤顯示 CPU、記憶體和網路歷程的圖表，檔案系統索引標籤列出所有分割區及其使用情況。

35.1.3.2 同步化資料

在沒有連接網路的行動機器與公司中連接網路的工作站之間切換時，必須讓所有個體間已處理的資料保持同步。這些資料可能包括出差時以及在公司內部需要用到的電子郵件資料夾、目錄以及個別檔案。這兩種情況中的解決方案如下：

同步化電子郵件
: 使用 IMAP 帳戶在公司網路中儲存您的電子郵件。然後可以從工作站使用任何中斷連接但支援 IMAP 的電子郵件用戶端（如 Mozilla Thunderbird 或 Evolution）存取這些電子郵件，如《GNOME 使用者指南》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。使用郵件用戶端中實作的用於傳送郵件的 SMTP 伺服器，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。使用最廣的是一項名為 `rsync` 的指令行工具。如需詳細資訊，請參閱其手冊頁（`man 1 rsync`）.

35.1.3.3 無線通訊：Wi-Fi

Wi-Fi 在這幾種無線技術中覆蓋範圍最廣，是唯一一種適用於大型網路（有時甚至是在空間上分離的網路）的操作技術。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。稱為存取點的裝置，是做為啓用 Wi-Fi 裝置的基礎工作站，而且充當著存取網際網路的中介角色。行動使用者可以在存取點之間切換，端視所在位置及哪個存取點提供最佳連接而定。與行動電話的通訊方式類似，供 Wi-Fi 使用者使用的大型網路，不用將它們結合到特定位置就能存取。

Wi-Fi 卡使用由 IEEE 組織提出的 802.11 標準進行通訊。此標準最初用於最大傳輸率 2 MBit/s。其間已增加許多新的標準來提高資料傳輸率。這些補充項目定義調變、傳輸輸出及傳輸率等詳細資訊（請參閱表格 35.2 「Wi-Fi 標準綜覽」）。此外，很多公司都實作具有專利權或草擬功能的硬體。

表格 35.2 WI-FI 標準綜覽

名稱 (802.11)	頻率 (GHz)	最大傳輸率 (MBit/s)	備註
a	5	54	較少干涉
b	2.4	11	較不普遍
g	2.4	54	廣泛採用，與 11b 向後相容
n	2.4 與/或 5	300	通用
ac	5	最高約為 865	預計在 2015 年會較為普遍
ad	60	最高約為 7000	2012 年發行，目前較不普遍；不受 SUSE Linux Enterprise Server 支援

SUSE® Linux Enterprise Server 不支援 802.11 舊式網路卡。支援使用 802.11 a/b/g/n 的大多數網路卡。新卡通常符合 802.11n 標準，但也有使用 802.11g 的卡。

35.1.3.3.1 運作模式

使用無線網路時，您可以用各種不同的技術和組態來確保快速、高品質的安全連接。Wi-Fi 卡通常在受管理模式下工作。但是，不同的作業類型需要不同的設定。無線網路可分為四種網路模式：

受管理模式（基礎架構模式），透過存取點（預設模式）

管理網路中有一個管理元件，即存取點。在此模式（又稱為基礎架構或預設模式）下，網路中 Wi-Fi 工作站的所有連接都會通過存取點，存取點也充當乙太網路的連接點。此模式會使用各種不同的驗證機制（WPA 等），以確保只有獲得授權的工作站才可以連接。這也是能耗最低的主要模式。

臨機操作模式（對等網路）

臨機操作網路中沒有存取點。由於工作站之間可直接進行通訊，因此臨機操作網路通常比受管理網路要慢。不過，臨機操作網路中的傳輸範圍及參與工作站的數量相當有限。也不支援 WPA 驗證。此外，並非所有卡都能可靠地支援臨機操作模式。

主要模式

在主要模式下，Wi-Fi 卡將用做存取點（假設您的卡支援此模式）。如需 Wi-Fi 卡的詳細資訊，請造訪 http://linux-wless.passys.nl 。

網狀模式

無線網狀網路是透過網狀拓撲組織的。無線網狀網路的連線遍佈所有無線網狀節點。屬於此網路的每個節點會連接到其他節點以共用該連線，這種連線的覆蓋區域可能很大。(在 SLE12 中不受支援)。

35.1.3.3.2 驗證

無線網路比有線網路更容易受到攔截和危害，因此各項標準均包含驗證和加密方式。

早期的 Wi-Fi 卡僅支援 WEP（有線等效加密）。然而，WEP 已證實不夠安全。為此，Wi-Fi 業者定義了一項稱為 WPA 的擴充標準，用於克服 WEP 的弱點。WPA 有時與 WPA2 同義，應做為預設的驗證方法。

一般情況下，使用者無法選擇驗證方法。例如，當網路卡在受管理模式下工作時，驗證由存取點設定。NetworkManager 會顯示驗證方法。

35.1.3.3.3 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，後來還可以為 104 位元。視 24 位元的啓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，與不加密網路相比，使用 WEP 的效果更佳。

某些廠商已實作了非標準「動態 WEP」。其功能與 WEP 完全相同並具有同樣的弱點，唯一的區別在於，金鑰會透過金鑰管理服務定期進行變更。

TKIP（定義於 WPA/IEEE 802.11i 中）

此金鑰管理協定定義於 WPA 標準中，使用與 WEP 相同的加密演算法，其弱點則均已消除。因為每個資料封包都有一個新的金鑰，所以攻擊這些金鑰等於白費力氣。TKIP 與 WPA-PSK 必須配合使用。

CCMP（定義於 IEEE 802.11i 中）

CCMP 說明金鑰管理。通常與 WPA-EAP 配合使用，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。

35.1.3.4 無線通訊：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。也可以用來連接覆蓋範圍內的各台電腦。連接鍵盤或滑鼠等無線系統元件時，也可以使用藍芽。不過，此技術的範圍還不足以連接遠端系統與網路。此時就可選擇使用 Wi-Fi 技術來穿透實體障礙物（如牆）進行通訊。

35.1.3.5 無線通訊：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。向收件人遠距離傳輸檔案的任務由行動網路來處理。IrDA 的另一種應用是在公司內以無線方式傳輸列印工作。

35.1.4 資料安全性

理想而言，您應使用多種方式保護筆記型電腦上的資料，不受未授權的存取。可以從下列三大面向來採取適當的安全性措施：

防止竊取
: 不論在什麼時候，都要避免機器遭到竊取。各種保全工具（例如鎖鏈）都可以在零售店中買到。

增強式驗證
: 除了透過登入和密碼進行標準驗證之外，還使用生物驗證。SUSE Linux Enterprise Server 支援指紋驗證。

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。中介紹了如何使用 SUSE Linux Enterprise Server《Security Guide》，第 11 章「Encrypting Partitions and Files」 建立加密分割區。另一種方案是在使用 YaST 新增使用者時建立加密的主目錄。

重要：資料安全性和暫停寫入到磁碟

暫停寫入到磁碟時，不會卸載加密分割區。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
: 任何資料傳輸，不論其傳輸方式為何，我們都應保護其安全。有關 Linux 及網路的一般安全性議題，請參閱《Security Guide》，第 1 章「Security and Confidentiality」。

35.2 行動硬體

SUSE Linux Enterprise Server 能夠自動偵測 FireWire (IEEE 1394) 或 USB 上的行動儲存裝置。詞彙行動儲存裝置表示任何種類的 FireWire 或 USB 硬碟、快閃式磁碟或數位相機。當這些裝置透過對應介面與系統連接時，系統會自動予以偵測並設

定。GNOME 的檔案管理員會提供靈活的行動硬體項目處理方式。若要安全卸載所有這些媒體，請使用檔案管理員的卸載磁碟區(GNOME) 功能。如需詳細資訊，請參閱《GNOME 使用者指南》。

外接磁碟 (USB 和 FireWire)
: 系統正確辨識外部硬碟後，其圖示即會出現在檔案管理員中。按一下圖示，會顯示裝置的內容。在此可以建立、編輯或刪除目錄及檔案。若要重新命名某個硬碟，請從右鍵快顯功能表中選取相應的功能表項目。此名稱的變更，僅限顯示於檔案管理員中。依據其將裝置掛接到 `/media` 中的描述子不會受影響。

USB 隨身碟
: 這些裝置如外接硬碟一樣由系統處理。同樣可以在檔案管理員中重新命名項目。

35.3 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第 35.1.3.3 節 「無線通訊：Wi-Fi」中提及。行動電話通訊協定的組態，於手冊中均有說明。

35.4 更多資訊

有關行動裝置與 Linux 所有問題的參考重點，請參閱 http://tuxmobil.org/。該網站的各個區段，對應筆記型電腦、PDA、行動電話及其他行動硬體中軟體和硬體的各個方面。

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裝置的資訊。

關於筆記型電腦相關議題，SUSE 提供了專屬的郵件清單（使用德文）。請參閱http://lists.opensuse.org/opensuse-mobile-de/。在此清單中，使用者與開發人員討論了 SUSE Linux Enterprise Server 行動計算的所有觀念。以英文張貼的文章會收到回應，但是大部分的歸檔資訊僅有德文版本。使用 http://lists.opensuse.org/opensuse-mobile/ 可獲取英文張貼的文章。

36 使用 NetworkManager

NetworkManager 是筆記型電腦與其他可攜式電腦的理想解決方案。它允許對網路連接使用一流的加密類型和標準，包括連至 802.1X 保護網路的連接。802.1X 是「區域網路和都會區網路的 IEEE 標準 — 基於連接埠的網路存取控制」。有了 NetworkManager，您在外出時，就不必顧慮網路介面的組態設定，也不必考慮如何在有線網路或無線網路之間進行切換。NetworkManager 可自動連接已知的無線網路，或是同時管理多個網路連接，然後按預設使用速度最快的連接。此外，您還可以手動在可用網路之間切換，並使用系統匣中的 Applet 管理網路連接。

同一時間有多個連接處於使用中狀態，而不僅僅是一個。因此，您可以在將筆記型電腦從乙太網路中斷開後，繼續透過無線網路保持連接狀態。

36.1 NetworkManager 的使用案例

NetworkManager 提供了精巧且直觀的使用者介面，可以讓使用者輕鬆切換網路環境。但是 NetworkManager 解決方案不適用於下列情況：

- 您的電腦會為網路的其他電腦提供網路服務，例如，它是 DHCP 或 DNS 伺服器。
- 您的電腦為 Xen 伺服器，或您的系統是 Xen 之中的虛擬系統。

36.2 啓用或停用 NetworkManager

對於筆記型電腦，NetworkManager 預設處於啓用狀態。但是您也可以在 YaST 網路設定模組中隨時將其啓用或停用。

1. 執行 YaST，然後前往系統 › 網路設定。
2. 此時將開啓網路設定對話方塊。前往全域選項索引標籤。
3. 若要使用 NetworkManager 設定和管理網路連接：
    a. 在網路設定方法欄位中，選取使用 NetworkManager 進行使用者控制。

b. 按一下確定，然後關閉 YaST。

c. 依第 36.3 節 「設定網路連線」中所述使用 NetworkManager 設定您的網路連線。

4. 停用 NetworkManager 並使用您自己的組態控制網路

a. 在網路設定方法欄位中，選擇由 wicked 控制。

b. 按一下確定。

c. 透過 YaST 使用自動組態（透過 DHCP 或靜態 IP 位址）設定您的網路卡。如需使用 YaST 設定網路組態的詳細資訊，請參閱第 16.4 節 「使用 YaST 手動設定網路連接」。

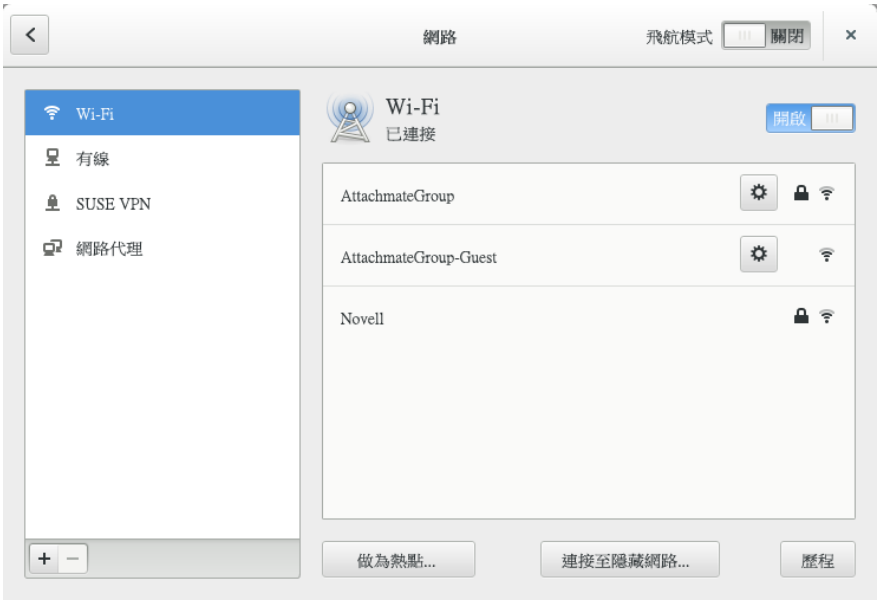
36.3 設定網路連線

在 YaST 中啓用 NetworkManager 之後，使用 GNOME 中提供的 NetworkManager 前端設定網路連線。它會顯示所有類型網路連線的索引標籤，如有線、無線、行動寬頻、DSL 及 VPN 連線。

若要在 GNOME 中開啓網路組態對話方塊，請透過狀態功能表開啓設定功能表，然後按一下網路。

注意：選項的可用性

在某些系統設定下，您可能無法設定連接。在安全環境中，某些選項可能會被鎖定，或者需要 root 許可權。請向您的系統管理員洽詢詳細資料。



圖形 36.1 GNOME 網路連線對話方塊

程序 36.1 新增與編輯連線

1. 開啓 NetworkManager 組態對話方塊。
2. 新增連線：
   a. 按一下左下角的+圖示。
   b. 選取偏好的連線類型並依照指示進行操作。
   c. 完成後，按一下新增。
   d. 確認變更後，開啓「狀態功能表」，所顯示的可用網路清單中便會出現新設定的網路連線。
3. 編輯連線：
   a. 選取要編輯的項目。
   b. 按一下齒輪圖示以開啓連接設定對話方塊。
   c. 插入您的變更，然後按一下套用以儲存變更。
   d. 若要讓您的連線可用做系統連線，請移至身分索引標籤，並選中使其可供其他使用者使用核取方塊。如需使用者與系統連線的詳細資訊，請參閱第 36.4.1 節 「使用者和系統連接」。

36.3.1 管理有線網路連線

如果您的電腦連接的是有線網路，請使用 NetworkManager Applet 管理連線。

1. 開啓「狀態功能表」，然後按一下有線以變更連接的詳細資料或將其關閉。
2. 若要變更設定，請按一下有線設定，然後按一下齒輪圖示。
3. 若要關閉所有網路連線，請啓動飛航模式設定。

36.3.2 管理無線網路連線

可見無線網路均列在無線網路下的 GNOME NetworkManager Applet 功能表中。每個網路的訊號強度也會顯示在功能表中。加密的無線網路以盾牌圖示標明。

程序 36.2 連接至可見無線網路

1. 若要連接至可見無線網路，請開啓「狀態功能表」，然後按一下Wi-Fi。
2. 按一下開啓以啓用。
3. 按一下選取網路，選取您的 Wi-Fi 網路，然後按一下連線。
4. 如果該網路已加密，則會開啓一個組態對話方塊。其中會顯示網路使用的加密類型以及用於輸入登入身分證明的文字方塊。

程序 36.3 連接至不可見無線網路

1. 若要連接至因未廣播其服務設定識別碼（SSID 或 ESSID）而無法自動偵測到的網路，請開啓「狀態功能表」，然後按一下 Wi-Fi。
2. 按一下Wi-Fi 設定以開啓詳細設定功能表。
3. 確定您的 Wi-Fi 處於啓用狀態，然後按一下連接至隱藏網路。
4. 在開啓的對話方塊中，於網路名稱內輸入 SSID 或 ESSID，並根據需要設定加密參數。

明確指定的無線網路會儘可能長久地保持連接狀態。如果在此期間插入網路線，則會連接已設定為儘可能保持連接的連接，而無線連接仍將延續。

36.3.3 將 Wi-Fi/藍芽卡設定成存取點

如果您的 Wi-Fi/藍芽卡支援存取點模式，您可以使用 NetworkManager 來設定組態。

1. 開啓「狀態功能表」，然後按一下Wi-Fi。
2. 按一下Wi-Fi 設定以開啓詳細設定功能表。
3. 按一下做為熱點，並依照指示進行操作。
4. 使用隨後出現的對話方塊中顯示的身分證明連接至遠端機器上的熱點。

36.3.4 NetworkManager 和 VPN

NetworkManager 支援多項虛擬私人網路 (VPN) 技術。對於每種技術，SUSE Linux Enterprise Server 都附帶了可為 NetworkManager 提供一般支援的基礎套件。除此之外，您還需要為 Applet 安裝相應的桌面專屬套件。

OpenVPN
: 若要使用此 VPN 技術，請安裝：

  - NetworkManager-openvpn
  - NetworkManager-openvpn-gnome

vpnc (Cisco AnyConnect)
: 若要使用此 VPN 技術，請安裝：

  - NetworkManager-vpnc
  - NetworkManager-vpnc-gnome

PPTP (點對點通道通訊協定)
: 若要使用此 VPN 技術，請安裝：

  - NetworkManager-pptp
  - NetworkManager-pptp-gnome

以下程序介紹如何使用 NetworkManager 將電腦設定為 OpenVPN 用戶端。其他類型的 VPN 的設定方法與此類似。

開始之前，確定套件 `NetworkManager-openvpn-gnome` 已安裝且所有相依項均已解析。

程序 36.4 使用 NETWORKMANAGER 設定 OPENVPN

1. 按一下面板右側的狀態圖示，然後按一下扳手和螺絲刀圖示，開啓應用程式設定。在視窗全部設定中選擇網路。
2. 按一下 + 圖示。
3. 依次選取 VPN 和 OpenVPN。
4. 選擇驗證類型。依據 OpenVPN 伺服器的設定，選擇證書（TLS）或藉由證書的密碼（TLS）。
5. 將必要的值插入相應的文字方塊。範例組態如下所示：

閘道	VPN 伺服器的遠端端點
使用者名稱	使用者（僅當選取藉由證書的密碼（TLS）時可用）
密碼	使用者的密碼（僅當選取藉由證書的密碼（TLS）時可用）
使用者驗證	`/etc/openvpn/client1.crt`
CA 證書	`/etc/openvpn/ca.crt`
私密金鑰	`/etc/openvpn/client1.key`

6. 按一下新增完成組態。
7. 若要啓用連接，請在設定應用程式的網路面板中，按一下切換按鈕。或者，按一下面板右側的狀態圖示，然後依次按一下 VPN 的名稱和連接。

36.4 NetworkManager 和安全性

NetworkManager 將無線連接分為受信任和不受信任兩種。受信任的連接是過去您明確選取過的任何網路，除此以外的連線都屬於不受信任。受信任的連接以存取點的名稱和 MAC 位址來識別。使用 MAC 位址可確保別的存取點不能使用受信任連接的名稱。

NetworkManager 會定期掃描是否有可用的無線網路。如果找到多個受信任的網路，便自動選取最近使用的那個網路。如果所有網路都不受信任，NetworkManager 會等待您做出選擇。

如果加密設定變更，但名稱和 MAC 位址未變，NetworkManager 會嘗試進行連接，但它會先要求您確認新的加密設定並提供所有更新，例如新的金鑰。

如果從使用無線連接切換成離線模式，NetworkManager 會將 SSID 或 ESSID 設為空白。以確保該網路卡斷開連接。

36.4.1 使用者和系統連接

NetworkManager 可識別兩種類型的連接：**使用者**連接與**系統**連接。使用者連接是第一位使用者登入時 NetworkManager 可使用的連接，要求使用者提供所有必要的身分證明。使用者登出後，連接即會斷開，並從 NetworkManager 中移除。定義為系統連接的連接可由所有使用者共用，且在使用者登入之前，只要 NetworkManager 啓動後即可使用。對於系統連接，必須在連接建立時提供所有身分證明。此類系統連接可用於自動連接要求驗證的網路。如需使用 NetworkManager 設定使用者連線或系統連線的相關資訊，請參閱第 36.3 節 「設定網路連線」。

36.4.2 儲存密碼與身分證明

如果您不希望每次連接至加密網路時都要重新輸入身分證明，則可以使用 GNOME 鑰匙圈管理員將加密的身分證明儲存在磁碟中，並以主密碼加以保護。

NetworkManager 也可從證書儲存區取回安全連線（例如，加密的有線、無線或 VPN 連線）的證書。若需更多資訊，請參閱《Security Guide》，第 12 章「Certificate Store」。

36.5 常見問答集

以下提供了有關使用 NetworkManager 設定特殊網路選項的常見問題。

36.5.1 如何將連接關聯到特定裝置?

依預設，NetworkManager 中的連線是特定於裝置類型的：它們適用於相同類型的所有實體裝置。如果一種連線類型對應多個實體裝置（例如，您的機器配有兩個乙太網路卡），可以將連線關聯到特定的裝置。

若要在 GNOME 中執行此操作，請先查詢裝置的 MAC 位址（使用 Applet 中的連接資訊，或者使用 `nm-tool` 或 `wicked show all` 等指令行工具的輸出）。然後開啓用於設定網路連接的對話方塊，並選擇要修改的連接。在有線或無線索引標籤中輸入裝置的MAC 位址，並確認您的變更。

36.5.2 在偵測到多個存取點具有相同 ESSID 的的情況下，如何指定特定的存取點?

如果有具有不同無線頻段（a/b/g/n）的多個存取點可用，系統預設會自動選擇訊號最強的存取點。要置換此存取點，請在設定無線連接時使用BSSID欄位。

基本服務組識別碼（BSSID）可唯一識別每個基本服務組。在基礎結構基本服務組中，BSSID 是無線存取點的 MAC 位址。在獨立的（臨機操作）基本服務組中，BSSID 是由 46 位元隨機數字產生的本地管理的 MAC 位址。

依照第 36.3 節 「設定網路連線」 中所述啓動用於設定網路連接的對話方塊。選擇要修改的無線連接，然後按一下編輯。在無線索引標籤中，輸入 BSSID。

36.5.3 如何與其他電腦共享網路連接?

主要裝置（連接至網際網路的裝置）不需要任何特殊組態。不過，您需要以如下方式設定連接至本地 Hub 或機器的裝置：

1. 依照第 36.3 節 「設定網路連線」 中所述啓動用於設定網路連接的對話方塊。選擇要修改的連接，然後按一下編輯。切換到 IPv4 設定索引標籤，然後從方法下拉式方塊中，啓用分享給其他電腦。這樣可讓 IP 流量轉遞並會執行裝置上的 DHCP 伺服器。確認您在 NetworkManager 中所做的變更。

2. 由於 DCHP 伺服器使用連接埠 `67`，請確定防火牆未封鎖此連接埠：在共用連接的機器上啓動 YaST，然後選取安全性與使用者 › 防火牆。切換到允許的服務類別。如果DCHP 伺服器尚未顯示為允許的服務，則從要允許的服務中選取DCHP 伺服器，然後按一下新增。確認您在 YaST 中所做的變更。

36.5.4 如何使用動態位址（DHCP、PPP 與 VPN）提供靜態 DNS 資訊？

如果 DHCP 伺服器提供的 DNS 資訊（與/或路由）無效，則可將其置換。依照第 36.3 節 「設定網路連線」 中所述啓動用於設定網路連接的對話方塊。選擇要修改的連接，然後按一下編輯。切換到 IPv4 設定索引標籤，然後從方法下拉式方塊中，啓用僅限自動（DHCP）位址。在DNS 伺服器與搜尋網域欄位中輸入 DNS 資訊。若要忽略自動取得的路由設定，請按一下路由，然後啓用相應的核取方塊。確認您的變更。

36.5.5 如何在使用者登入之前將 NetworkManager 連接到受密碼保護的網路？

定義用於此目的的**系統連接**。如需詳細資訊，請參閱第 36.4.1 節 「使用者和系統連接」。

36.6 疑難排解

可能發生連接問題。與 NetworkManager 相關的一些常見問題有：Applet 未啓動或缺少 VPN 選項。解析和預防這些問題的方法會視使用的工具而定。

NetworkManager 桌面 Applet 不啓動

如果設定了 NetworkManager 控制的網路，Applet 會自動啓動。如果 Applet 未啓動，請依第 36.2 節 「啓用或停用 NetworkManager」中所述，檢查 YaST 中是否啓用 NetworkManager。然後，請確定 NetworkManager-gnome 套件亦已安裝。

如果桌面 Applet 已安裝，但因為某些原因而不執行，您可以用手動方式啓動。如果安裝了桌面 Applet，但其由於某些原因並未執行，請使用 `nm-applet` 指令手動將其啓動。

NetworkManager Applet 不包含 VPN 選項

針對 NetworkManager 的支援、Applet 與適用於 NetworkManager 的 VPN 分佈於獨立套件中。如果您的 NetworkManager Applet 不包含 VPN 選項，請檢查是否安裝了包含 NetworkManager 支援的 VPN 技術套件。如需詳細資訊，請參閱第 36.3.4 節 「NetworkManager 和 VPN」。

無可用的網路連接

如果已正確設定網路連接，且網路連接的所有其他元件（路由器等）也均已設定並正在執行中，有時需要重新啓動電腦的網路介面。若要執行此操作，請以 `root` 身分登入指令行並執行 `systemctl restart wicked`。

36.7 更多資訊

您可以在下列網站和目錄中找到有關 NetworkManager 的詳細資訊：

NetworkManager 專案頁面

http://projects.gnome.org/NetworkManager/

套件文件

另請參閱下列目錄中有關 NetworkManager 和 GNOME Applet 的最新資訊：

- `/usr/share/doc/packages/NetworkManager/`，
- `/usr/share/doc/packages/NetworkManager-gnome/`。

37 電源管理

System z IBM z Systems 上不提供本章所述的功能和硬體，因此本章內容與這些平台不相關。

電源管理對筆記型電腦十分重要，對其他系統也很有用。ACPI（Advanced Configuration and Power Interface，進階組態和電源介面）可以在所有現代電腦（筆記型電腦、桌上型電腦和伺服器）上使用。電源管理技術需要配備合適的硬體與 BIOS 常式。大多數筆記型電腦和許多新式的桌上型電腦及伺服器都符合這些需求。此技術還可以控制 CPU 頻率比例，這有助於省電及降低噪音。

37.1 省電功能

省電功能不僅對於筆記型電腦的行動用途很重要，對於桌上型系統也很重要。主要功能以及在 ACPI 中的用途包括：

待命
: 不支援。

暫停（於記憶體）
: 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，而不用開機或重新啓動應用程式。此功能等同於 ACPI 狀態 S3。

睡眠（暫停磁碟）
: 在此操作模式，會將整個系統狀態寫入硬碟，然後關閉系統。至少要有與 RAM 一樣大的交換分割區，才能寫入所有作用中資料。要從此狀況重新啓用需耗時 30 到 90 秒。還原時會回到暫停前的狀態。有些製造商會為此模式提供有用的混合功能，例如 IBM Thinkpad 中的 RediSafe。對應的 ACPI 狀態為 S4。在 Linux 中，暫停寫入到磁碟是由獨立於 ACPI 之外的核心常式來執行。

注意：透過 `mkswap` 進行格式化時變更了交換分割區的 UUID

如果可能，請勿使用 `mkswap` 重新格式化現有的交換分割區。否則，使用 `mkswap` 重新格式化將變更交換分割區的 UUID 值。請改為透過 YaST 重新格式化（將更新 `/etc/fstab`），或者手動調整 `/etc/fstab`。

電池監視器

ACPI 會檢查電池充電狀態並提供相關資訊。此外，ACPI 會在電力到達某個關鍵狀態時，協調要執行的動作。

自動關閉電源

關機後，電腦會關閉電源。此功能很重要，尤其是在電池用盡前所執行的自動關機。

處理器速度控制

與 CPU 連結時有三種方式可節省電源：頻率和電壓比例（也稱為 PowerNow! 或 Speedstep）、調節，以及讓處理器進入睡眠狀態（C 狀態）。依據電腦的操作模式，也可以合併這些操作方法。

37.2 進階組態與電源介面（ACPI）

ACPI 主要用於讓作業系統設定和控制個別的硬體元件。ACPI 取代了「電源管理隨插即用（PnP）」與「進階電源管理（APM）」。它能提供一些資訊，包括電池、交流電轉接器、溫度、風扇以及「關閉蓋子」或「電池電力不足」等系統事件。

BIOS 會提供一些表格，內含關於個別元件與硬體的存取方法等資訊。作業系統會使用這些資訊來執行任務，像是指定中斷或啓用和停用元件。因為作業系統會執行儲存於 BIOS 中的指令，所以 BIOS 實行會決定其功能。journald 中報告了 ACPI 能偵測和載入的表格。如需有關檢視這些日誌記錄訊息的詳細資訊，請參閱第 15 章 「`journalctl`：查詢 `systemd` 日誌」。請參閱第 37.2.2 節 「疑難排解」，以取得更多有關 ACPI 問題疑難排解的資訊。

37.2.1 控制 CPU 效能

CPU 可以使用三種方式省電：

- 頻率和電壓比例
- 調節時鐘頻率（T 狀態）
- 使處理器進入睡眠狀態（C 狀態）

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。

頻率比例及調節只在處理器忙碌時使用，因為在處理器閒置時，必定會套用最經濟的 C 狀態。如果 CPU 正忙碌，頻率比例是建議的省電方法。通常處理器僅有部份的工作負載。在此情況中，可以使用較低的頻率。通常，最佳方法是使用依核心需求調節器來控制動態頻率比例。

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。

如需更多資訊，請參閱《System Analysis and Tuning Guide》，第 11 章「Power Management」。

37.2.2 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。問題通常是因 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出現 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。若電腦未開機，以下其中一個開機參數可能有用：

`pci=noacpi`
: 不使用 ACPI 來設定 PCI 裝置。

`acpi=ht`
: 僅執行一個簡單的資源組態。不將 ACPI 用於其他目的。

`acpi=off`
: 關閉 ACPI。

警告：未使用 ACPI 的開機問題

有些較新的機器（尤其是 SMP 系統及 AMD64 系統）需透過 ACPI 以正確設定硬體。關閉這些機器的 ACPI 會發生隨之而來的問題。

有時，透過 USB 或 FireWire 連接的硬體會另機器混淆。如果機器拒絕開機，則拔除所有不需要的硬體插頭，並再試一次。

開機後，可使用 `dmesg -T | 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 以外的因素所導致的問題）。如果在分析 ACPI 表格時發生問題，可以將最重要的表格 DSDT（Differentiated System Description Table，區分系統描述表）替換為改良版本。在此情況中，會忽略 BIOS 的錯誤 DSDT。程序在第 37.4 節 「疑難排解」 中描述。

在核心組態中，有個啓用 ACPI 除錯訊息的切換。如果具有 ACPI 除錯功能的核心已編譯並安裝，則會發出詳細資訊。

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。

37.2.2.1 更多資訊

- http://tldp.org/HOWTO/ACPI-HOWTO/ (詳細的 ACPI HOWTO，內含 DSDT 修補程式)
- http://www.acpi.info (進階組態與電源介面規格)
- http://acpi.sourceforge.net/dsdt/index.php (Bruno Ducrot 的 DSDT 修補程式)

37.3 硬碟的休眠

在 Linux 中，可在不需使用硬碟時，讓硬碟完全進入睡眠狀態，或是讓硬碟以更省電、更安靜的方式來運作。在目前的筆記型電腦中，您不用手動關閉硬碟，因為它們會在不用的時候自動進入省電操作模式。不過，如果您想最大限度地省電，可以使用 `hdparm` 指令嘗試下面的幾種方法。

該指令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` 會讓硬碟進入睡眠狀態。`hdparm -S X` 會讓硬碟閒置一段時間後減慢執行速度。使用以下值取代 `X`：`0` 表示停用此機制，會使硬碟持續執行。`1` 到 `240` 的值將乘以 5 秒。`241` 到 `251` 的值則是以 30 分鐘為一個單位，依序從 30 分鐘的閒置到 11 倍的 330 分鐘的閒置。

可以使用 `-B` 選項來控制硬碟內部的省電選項。可從 `0` 到 `255` 中選取一個值，以最大化省電效果或最大化電力輸出。其結果視硬碟用途而定，難以評估。如果要減少硬碟噪音，請使用 `-M` 選項。從 `128` 到 `254` 中選取一個值，以決定要安靜或快速。

通常，要讓硬碟進入睡眠不是件容易的事。在 Linux 中，會有多個程序寫入硬碟中，因而重複喚醒硬碟。因此，有必要去瞭解 Linux 如何處理那些要寫入硬碟的資料 首先，會將所有資料在 RAM 中做緩衝處理。`pdflush` 精靈會監控此緩衝區。當資料到達特定的時間限制，或當緩衝區已填滿至某一程度時，會將緩衝區的內容注入硬碟。緩衝區的大小則動態地由記憶體地的大小及系統負載來決定。依預設，pdflush 會設成較短的間隔，以最大化資料的完整性。pdflush 每 5 秒檢查一次緩衝區，並將資料寫入硬碟。以下變數較為重要：

`/proc/sys/vm/dirty_writeback_centisecs`

包含 pdflush 執行緒喚醒前的延遲時間（以百分之一秒為單位）。

`/proc/sys/vm/dirty_expire_centisecs`

定義最遲應在其後將改動分頁寫出的時間框架。預設值為 `3000`，即 30 秒。

`/proc/sys/vm/dirty_background_ratio`

pdflush 開始寫入改動分頁之前改動分頁的最大百分比。預設值為 `5`%。

`/proc/sys/vm/dirty_ratio`

改動分頁超過此總記憶體的百分比時，系統會強制執行程序以在其時間片段內將改動寫入緩衝區，而不是繼續寫入改動。

警告：損害資料完整性

變更 `pdflush` 精靈的設定會損害資料的完整性。

除了這些程序之外，`Btrfs`、`Ext3`、`Ext4` 記錄檔案系統及其他檔案系統不經由 `pdflush` 便將中繼資料寫入硬碟，也會使得硬碟無法減速。

另一個重要因素在於啓動程式的行為方式。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。

在這一點上，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。

37.4 疑難排解

所有錯誤訊息和警示均記錄在可以使用 `journalctl` 指令查詢的系統日誌中（如需詳細資訊，請參閱第 15 章 「`journalctl`：查詢 `systemd` 日誌」）。下列幾節涵蓋了一些最常見的問題。

37.4.1 CPU 頻率沒有作用

請參閱核心來源以瞭解您的處理器是否受支援。您需要特殊核心模組或模組選項以啓用 CPU 頻率控制。如果 `kernel-source` 套件已安裝，則此資訊可在 `/usr/src/linux/Documentation/cpu-freq/*` 中找到。

37.5 更多資訊

- http://en.opensuse.org/SDB:Suspend_to_RAM — 如何使「暫停寫入到 RAM」正常工作
- http://old-en.opensuse.org/Pm-utils — 如何修改一般暫停架構

VI 疑難排解

38 說明和文件

SUSE® Linux Enterprise Server 隨附了各種來源的資訊和文件，其中有許多已整合到安裝的系統中。

`/usr/share/doc` 中的文件
: 這是傳統的說明目錄，包含系統的各種文件檔案及版本說明。子目錄 `packages` 中還提供了所安裝套件的相關資訊。如需詳細資訊，請參閱第 38.1 節 「文件目錄」。

外圍程序指令的 man 頁面與資訊頁
: 使用外圍程序時，不需要記住指令的選項。一般而言，外圍程序會以 man 頁面與資訊頁的方式提供整合式說明。如需詳細資訊，請參閱第 38.2 節 「線上文件」與第 38.3 節 「Info 頁面」。

桌面說明中心
: GNOME 桌面的說明中心（說明）以可搜尋的方式提供了對系統上最重要的文件資源的集中存取途徑。這些資源包括已安裝之應用程式的線上說明、man 頁面、資訊頁以及產品隨附的 SUSE 手冊。

某些應用程式的獨立說明套件
: 使用 YaST 安裝新軟體時，通常會自動安裝軟體文件，而且會顯示在桌面的說明中心內。不過，有些應用程式（例如 GIMP）可能具有不同的線上說明套件，它們可以使用 YaST 獨立安裝，但不會整合到說明中心內。

38.1 文件目錄

安裝 Linux 系統後，可以在 `/usr/share/doc` 這個傳統目錄中找到文件。該目錄通常包含系統上已安裝之套件的相關資訊，以及版本說明和手冊等。

注意：內容取決於已安裝的套件

在 Linux 系統中，有許多手冊及其他類型的文件都可以像軟體一樣以套件的形式獲取。`/usr/share/docs` 中包含的資訊量和類型也取決於所安裝的（文件）套件。如果找不到此處提及的子目錄，請檢查您的系統中是否已安裝相應的套件，並根據需要使用 YaST 予以新增。

38.1.1 SUSE 手冊

我們提供了不同語言的手冊，有 HTML 和 PDF 兩種版本。在 `manual` 子目錄下，可以找到您產品適用的大多數 HTML 版 SUSE 手冊。如需您產品適用的所有文件的綜覽，請參閱手冊的序。

如果安裝了多種語言，`/usr/share/doc/manual` 可能會包含不同語言版本的手冊。HTML 版本的 SUSE 手冊也可以在兩種桌面系統的說明中心內找到。若想瞭解 PDF 和 HTML 版手冊在安裝媒體上的位置，請參閱 SUSE Linux Enterprise Server 的版本說明。這些手冊位於所安裝系統的 `/usr/share/doc/release-notes/` 目錄內，也可以在 http://www.suse.com/releasenotes/ 中產品特定的網頁內找到其線上版本。

38.1.2 套件文件

在 `packages` 下面，可找到系統上已安裝之軟體套件中所包含的文件。對每個套件都會建立子目錄 `/usr/share/doc/packages/PACKAGENAME`。其中通常包含套件的讀我檔案，有時還包含範例、組態檔案或其他程序檔。以下清單介紹了 `/usr/share/doc/packages` 下包含的一般檔案。以下每個項目不一定都存在，許多套件可能只包含其中的一部分。

`AUTHORS`
: 主要開發者清單。

`BUGS`
: 已知錯誤或異常狀況。可能還包含 Bugzilla 網頁的連結，透過該網頁可以搜尋所有錯誤。

`CHANGES` ，
`ChangeLog`
: 版本之間的變更摘要。因為它非常詳盡，所以對於開發人員而言通常很有幫助。

`COPYING` ，
`LICENSE`
: 授權資訊。

`FAQ`
: 自郵寄清單或新聞群組所收集的問題與解答。

`INSTALL`

在系統上安裝此套件的方法。由於在您讀取此檔案時已安裝套件，因此可以安心地忽略此檔案的內容。

`README` 、 `README.*`

有關軟體的一般資訊。例如用途和使用方法。

`TODO`

尚未執行但可能會在未來執行的項目。

`MANIFEST`

具有簡短摘要的檔案清單。

`NEWS`

說明此版本的新功能。

38.2 線上文件

man 頁面 Linux 系統的重要部分。它們提供指令用法以及所有可用選項與參數的說明。man 頁面可以使用 `man` 後接指令名稱的方式進行存取，例如 `man ls`。

man 頁面會直接在外圍程序中顯示。若要導覽這些頁面，請使用 Page ↑ 和 Page ↓ 上移和下移。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。如 表格 38.1 「Man 頁面—類別與說明」（取自 man 本身的 man 頁面）所示，man 頁面會依類別儲存。

表格 38.1 MAN 頁面—類別與說明

數字	描述
1	執行程式或外圍程序指令
2	系統呼叫（核心提供的函數）
3	程式庫呼叫（程式庫中的函數）
4	特殊檔案（通常位於 `/dev` 中）
5	檔案格式和慣例（`/etc/fstab`）

數字	描述
6	遊戲
7	其他（包括巨集套件與慣例），例如，man(7)、groff(7)
8	系統管理指令（通常只適用於 root）
9	核心常式（非標準）

每一個線上文件由標籤為 NAME、SYNOPSIS、DESCRIPTION、SEE ALSO、LICENSING 以及 AUTHOR 等的許多部份組成。視指令的類型，可能還包括其他可用區段。

38.3 Info 頁面

Info 頁面是您系統上另一個重要的資訊來源。它們所提供的資訊通常比 man 頁面更為詳盡。這些頁面中的內容不止指令行選項，有時還包含完整的教學課程或參考文件。若要檢視特定指令的資訊頁面，請輸入 `info`，後接指令名稱，例如 `info ls`。您可以直接在外圍程序中使用檢視器瀏覽資訊頁面，並可顯示不同的區段（稱為「節點」）。使用 Space 和 <– 分別可以向前和向後移動。在節點內，您也可以使用 Page ↑ 和 Page ↓ 進行瀏覽，但要前往上一個或下一個節點，只能使用 Space 和 <–。按 Q 可以結束檢視模式。並不是所有指令都隨附資訊頁面，反之亦然。

38.4 線上資源

除了 `/usr/share/doc` 下所安裝之 SUSE 手冊的線上版本外，您還可以在 Web 上存取產品專屬的手冊與文件。如需關於 SUSE Linux Enterprise Server 所有可用文件的綜覽，請查看產品專屬文件網頁 http://www.suse.com/doc/。

如果您要搜尋其他產品相關的資訊，也可以造訪下列網站：

SUSE 技術支援

如果您有疑問或需要一些解決方案來解決技術問題，可在 http://www.suse.com/support/ 中找到 SUSE 技術支援。

SUSE 論壇

Novell 提供了多個論壇，您可以進入其中套論有關 SUSE 產品的話題。請參閱 http://forums.suse.com/ 以取得論壇清單。

SUSE 交流園地

一個線上社群，提供可下載的文章、提示、問答集以及免費工具，網址為：http://www.suse.com/communities/conversations/

GNOME 文件

http://library.gnome.org/ 上提供了適用於 GNOME 使用者、管理員以及開發人員的文件。

The Linux Documentation Project

The Linux Documentation Project (TLDP) 由志願者組成的團體運營，該團體負責撰寫 Linux 相關文件（請參閱 http://www.tldp.org）。TLDP 或許是能夠提供最全面的 Linux 相關文件的資源了。這組文件包含入門者的教學課程，但主要是針對有經驗的使用者和專業系統管理員。TLDP 已免費公開發行 HOWTO、常見問題集以及指南（手冊）。SUSE Linux Enterprise Server 中也隨附了一部分來自 TLDP 的文件

您還可以嘗試通用搜尋引擎。例如，如果您在燒錄 CD 或進行 LibreOffice 檔案轉換時遇到問題，則可以使用搜尋片語 `Linux CD-RW 說明` 或 `OpenOffice 檔案轉換問題`。

39 收集系統資訊以供支援所用

為了讓使用者快速綜覽機器的所有相關系統資訊，SUSE Linux Enterprise Server 提供了 `hostinfo` 套件。該套件還可以協助系統管理員檢查污染的（不受支援的）核心，或者機器上安裝的任何協力廠商套件。

出現問題時，可以使用 `supportconfig` 指令行工具或 YaST 支援模組建立詳細的系統報告。這兩種方法都會收集系統的相關資訊，包括目前的核心版本、硬體、已安裝套件、分割區設定及其他資訊。最後會產生一個包含多個檔案的 TAR 歸檔。在建立服務要求（SR）後，您可以將該 TAR 歸檔上傳至全球技術支援。該歸檔有助於找出您所報告的問題，並可以協助您解決問題。

此外，您可以分析 `supportconfig` 輸出來發現已知問題，以協助快速解決問題。為此，SUSE Linux Enterprise Server 提供了一個裝置和一個指令行工具用於進行 `Supportconfig 分析`（SCA）。

39.1 顯示目前系統資訊

在登入伺服器時，要想快速方便地綜覽所有相關系統資訊，請使用套件 `hostinfo`。機器上安裝該套件後，控制台將會向登入此機器的任何 `root` 使用者顯示以下資訊：

範例 39.1 以 `root` 身分登入時的 `hostinfo` 輸出

```
Hostname:                 earth
Current As Of:            Wed 12 Mar 2014 03:57:05 PM CET
Distribution:             SUSE Linux Enterprise Server 12
 -Service Pack:           0
Architecture:             x86_64
Kernel Version:           3.12.12-3-default
 -Installed:              Mon 10 Mar 2014 03:15:05 PM CET
 -Status:                 Not Tainted
Last Updated Package:     Wed 12 Mar 2014 03:56:43 PM CET
 -Patches Needed:         0
 -Security:               0
 -3rd Party Packages:     0
IPv4 Address:             ens3 192.168.1.1
Total/Free/+Cache Memory: 983/95/383 MB (38% Free)
Hard Disk:                /dev/sda 10 GB
```

如果輸出顯示 `tainted` 核心狀態，請參閱第 39.6 節 「核心模組支援」以瞭解更多詳細資料。

39.2 使用 Supportconfig 收集系統資訊

若要建立包含詳細系統資訊的 TAR 歸檔以送交全球技術支援，請直接使用 `supportconfig` 指令行工具，或者使用 YaST 支援模組。該指令行工具由預設安裝的套件 `supportutils` 提供。YaST 支援模組也以該指令行工具為基礎。

39.2.1 建立服務要求號碼

系統隨時都可以產生 Supportconfig 歸檔。但是，要將 supportconfig 資料送交全球技術支援，首先需要產生一個服務要求號碼。上傳歸檔以獲得支援時，您需要使用此號碼。

若要建立服務要求，請造訪 https://scc.suse.com/support/requests 並依照螢幕上的指示執行操作。記下您的 12 位數服務要求號碼。

注意：隱私權聲明

SUSE 和 Micro Focus 將系統報告視為機密資料。關於我們在隱私方面所做承諾的詳細資訊，請參閱 https://www.suse.com/company/policies/privacy/。

39.2.2 上傳目標

在建立服務要求號碼後，可以依照程序 39.1 「使用 YaST 向支援人員提交資訊」或程序 39.2 「從指令行向支援人員提交資訊」中所述將 supportconfig 歸檔上傳到全球技術支援。使用下列上傳目標之一：

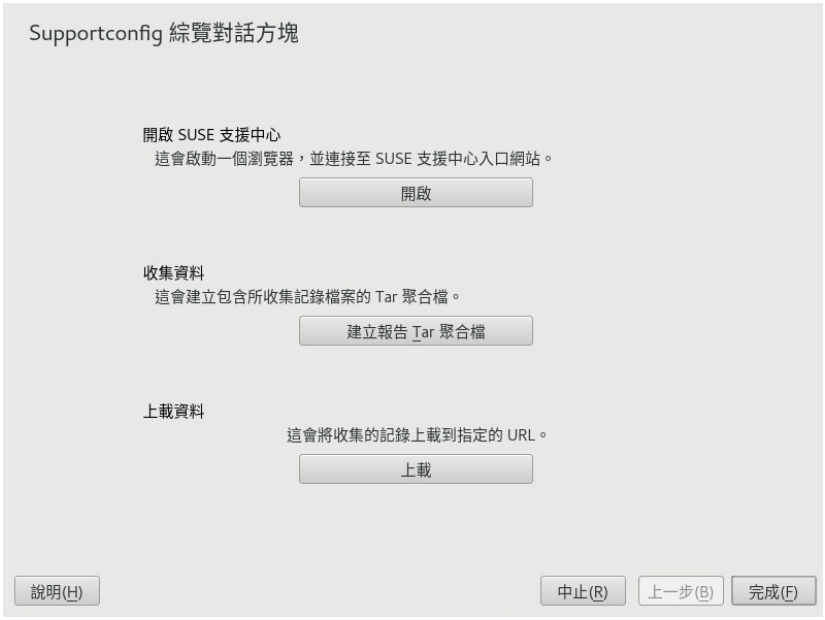
- 美國客戶：ftp://ftp.novell.com/incoming
- EMEA（歐洲、中東和非洲）：ftp://support-ftp.suse.com/in

或者，可以使用以下服務要求 URL 手動將該 TAR 歸檔附加到您的服務要求：https://scc.suse.com/support/requests。

39.2.3 使用 YaST 建立 Supportconfig 歸檔

若要使用 YaST 收集系統資訊，請執行下列步驟：

1. 啓動 YaST 並開啓支援模組。



2. 按一下建立報告 Tar 聚合檔。

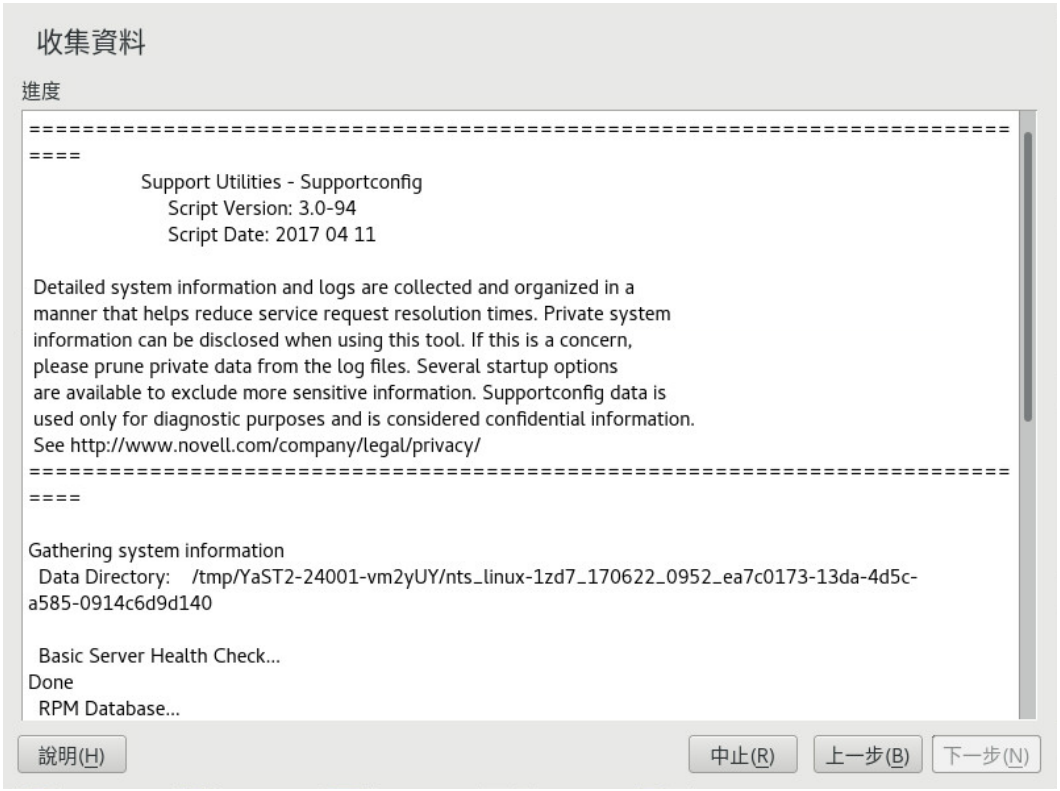
3. 在隨後出現的視窗中，從選項圓鈕清單中選取一個 supportconfig 選項。依預設，系統會預先選取使用自訂（進階）設定。如果要先測試報告功能，請使用僅收集最少量的資訊。關於其他選項的某些背景資訊，請參閱 `supportconfig` 的 man 頁面。
   按下一步繼續。

4. 輸入您的聯絡人資訊。該資訊將寫入名為 `basic-environment.txt` 的檔案，並包含在要建立的歸檔中。

5. 如果要在結束資訊收集程序時將歸檔提交至全球技術支援，則需要指定上傳資訊。YaST 會自動推薦一個上傳伺服器。如果要修改該伺服器，請參閱第 39.2.2 節 「上傳目標」，以詳細瞭解可以使用哪些上傳伺服器。
   如果希望稍後提交歸檔，則可以暫時將上傳資訊保留空白。

6. 按下一步繼續。

7. 系統即開始收集資訊。



該程序完成後，按下一步繼續。

8. 檢查資料收集：選取記錄檔案的檔案名稱可以在 YaST 中檢視其內容。在將 TAR 歸檔提交至支援人員之前，若要移除您不希望包含在該歸檔中的檔案，請使用從資料移除。按下一步繼續。

9. 儲存該 TAR 歸檔。如果您以 `root` 使用者身分啓動了 YaST 模組，則 YaST 預設會建議將該歸檔儲存到 `/var/log`（否則將儲存到您的主目錄）。檔案名稱格式為 `nts_主機_日期_時間.tbz`。

10. 如果要直接將該歸檔上傳給支援人員，請確定啓用了將記錄檔案 Tar 聚合檔上傳到 URL。這裡顯示的上傳目標是步驟 5 中 YaST 建議的上傳目標。如果要修改上傳目標，請在第 39.2.2 節 「上傳目標」中查看關於哪些上傳伺服器可用的詳細資訊。

11. 如果要跳過上傳步驟，請停用將記錄檔案 Tar 聚合檔上傳到 URL。

12. 確認變更以關閉 YaST 模組。

39.2.4 從指令行建立 Supportconfig 歸檔

以下程序顯示如何建立 supportconfig 歸檔但不將它直接提交給支援人員。要上傳該歸檔，需要依照程序 39.2 「從指令行向支援人員提交資訊」中所述，結合某些選項執行指令。

1. 開啓外圍程序，切換為 `root` 身分。
2. 執行 `supportconfig`，且不使用任何選項。此操作會收集預設的系統資訊。
3. 等待工具完成操作。
4. 預設的歸檔位置為 `/var/log`，檔案名稱格式為 `nts_主機_日期_時間.tbz`

39.2.5 Supportconfig 通用選項

呼叫 `supportconfig` 公用程式時通常不會顯示任何選項。請使用 `supportconfig -h` 顯示所有選項的清單，或參閱 man 頁面。下面的清單簡要概述了一些常見的使用案例：

減少所收集資訊的大小

使用最少量選項（`-m`）：

```
supportconfig -m
```

將資訊限制為特定的主題

如果您已使用預設的 `supportconfig` 輸出找到問題所在，並發現該問題只與特定的區域或功能集相關，則您在下一次執行 `supportconfig` 時，應將收集的資訊限制為特定的區域。例如，如果您偵測到 LVM 出現問題，並想要測試最近對 LVM 組態所做的變更，則合適的做法是僅收集關於 LVM 的最少量 supportconfig 資訊：

```
supportconfig -i LVM
```

要查看可用來將收集之資訊限制為特定區域的功能關鍵字的完整清單，請執行

```
supportconfig -F
```

在輸出中包含其他聯絡資訊：

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

（在一行中輸入所有指令）

收集已輪替的記錄檔案

```
supportconfig -l
```

這對記錄量較大的環境，或在重新開機後 syslog 輪替記錄檔案時核心發生當機的情況特別實用。

39.3 將資訊提交至全球技術支援

可以使用 YaST 支援模組或 `supportconfig` 指令行公用程式向全球技術支援提交系統資訊。如果您遇到伺服器問題，想要獲得支援人員的協助，則首先需要建立一個服務要求。如需詳細資料，請參閱第 39.2.1 節 「建立服務要求號碼」。

以下範例使用 `12345678901` 做為服務要求號碼的預留位置。請以您在第 39.2.1 節 「建立服務要求號碼」中建立的服務要求號碼取代 `12345678901`。

程序 39.1 使用 YAST 向支援人員提交資訊

以下程序假設您已建立一個 supportconfig 歸檔，但尚未上傳。請確定已按第 39.2.3 節 「使用 YaST 建立 Supportconfig 歸檔」中步驟 4所述，在歸檔中包含了您的聯絡資訊。關於如何透過一個步驟產生並提交 supportconfig 歸檔的指示，請參閱第 39.2.3 節 「使用 YaST 建立 Supportconfig 歸檔」。

1. 啓動 YaST 並開啓支援模組。
2. 按一下上傳。
3. 在含記錄檔案的套件中，指定現有 supportconfig 歸檔的路徑，或者按一下瀏覽找到該歸檔。
4. YaST 會自動推薦一個上傳伺服器。如果要修改該伺服器，請參閱第 39.2.2 節 「上傳目標」，以詳細瞭解可以使用哪些上傳伺服器。

按下一步繼續。

5. 按一下完成。

程序 39.2 從指令行向支援人員提交資訊

以下程序假設您已建立一個 supportconfig 歸檔，但尚未上傳。關於如何透過一個步驟產生並提交 supportconfig 歸檔的指示，請參閱第 39.2.3 節 「使用 YaST 建立 Supportconfig 歸檔」。

1. 伺服器連接至網際網路：

   a. 要使用預設上載目標，請執行：

   ```
   supportconfig -ur 12345678901
   ```

   b. 對於安全上傳目標，請使用以下指令：

   ```
   supportconfig -ar 12345678901
   ```

2. 伺服器未連接至網際網路

   a. 執行以下指令：

   ```
   supportconfig -r 12345678901
   ```

b. 將 `/var/log/nts_SR12345678901*tbz` 歸檔手動上傳到我們的 FTP 伺服器之一。要使用哪個伺服器取決於您所在的位置。如需綜覽，請參閱第 39.2.2 節 「上傳目標」。

3. TAR 歸檔傳輸到我們 FTP 伺服器的內送目錄後，會自動附加到您的服務要求中。

39.4 分析系統資訊

您可以分析使用 `supportconfig` 建立的系統報告來發現已知問題，以幫助快速解決問題。為此，SUSE Linux Enterprise Server 提供了一個裝置和一個指令行工具用於進行 `Supportconfig 分析` (SCA)。SCA 裝置是一個非互動式伺服器端工具。SCA 工具（`scatool`）在用戶端的指令行上執行。這兩個工具都能分析來自受影響伺服器的 supportconfig 歸檔。初始伺服器分析在 SCA 裝置或執行 scatool 的工作站上進行。線上伺服器上不會發生任何分析週期。

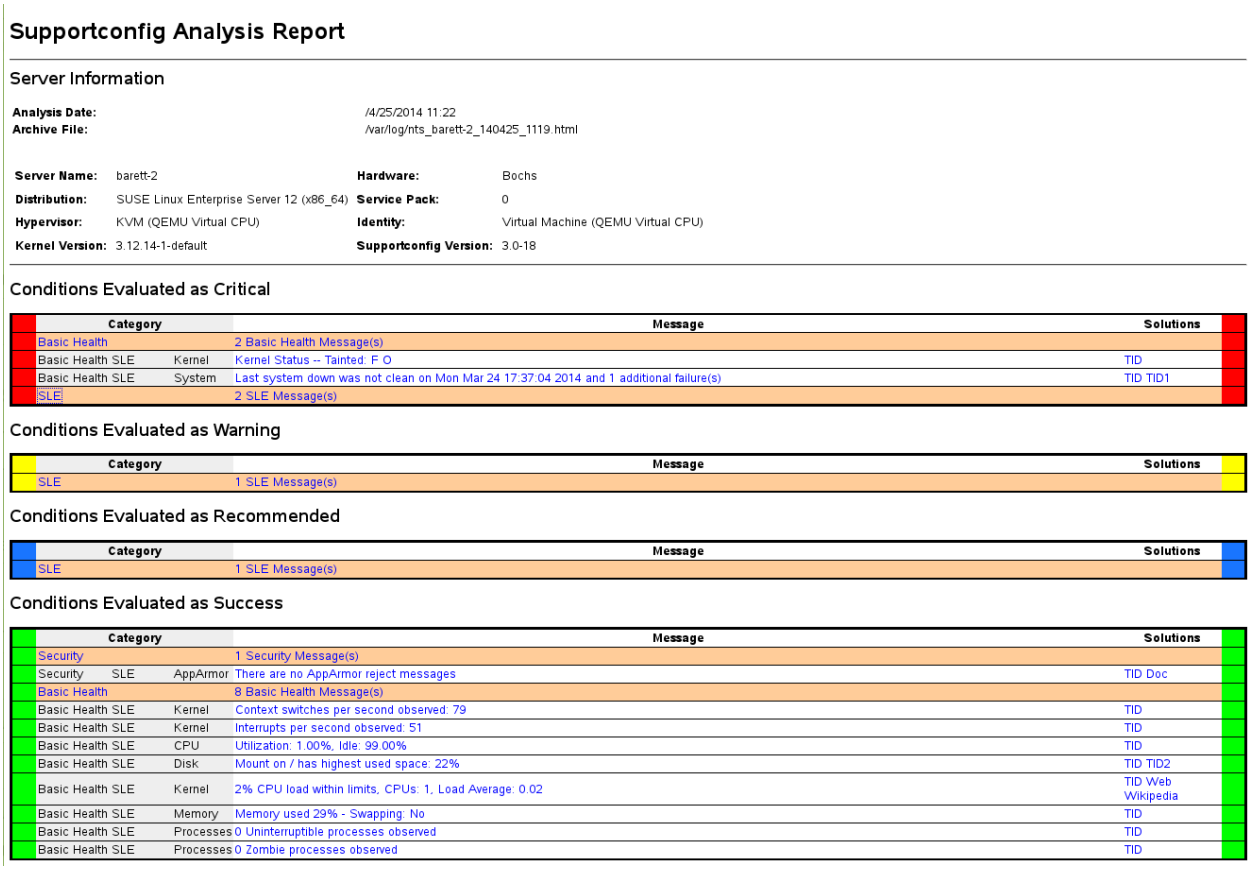
此外，該裝置與指令行工具另外還需要產品特定的模式，這樣它們才能分析關聯產品的 supportconfig 輸出。每種模式都是一個程序檔，用於針對某個已知問題分析和評估 supportconfig 歸檔。模式以 RPM 套件的形式提供。

例如，如果您想要分析 SUSE Linux Enterprise 11 機器上產生的 supportconfig 歸檔，則需要將 `sca-patterns-sle11` 套件隨 SCA 工具一併安裝（或者，在您想要用做 SCA 裝置伺服器的機器上安裝該套件）。要分析 SUSE Linux Enterprise 10 機器上產生的 supportconfig 歸檔，需要安裝 `sca-patterns-sle10` 套件。

您也可以依照第 39.4.3 節 「開發自訂分析模式」中的簡要描述開發自己的模式。

39.4.1 SCA 指令行工具

SCA 指令行工具讓您既可使用 `supportconfig`，又可使用本地機器上安裝的特定產品的分析模式來分析該機器。該工具將建立一份顯示分析結果的 HTML 報告。如需取得範例說明，請參閱圖形 39.1 「SCA 工具產生的 HTML 報告」。



Supportconfig Analysis Report

Server Information

Analysis Date: /4/25/2014 11:22
Archive File: /var/log/nts_barett-2_140425_1119.html

Server Name:	barett-2	**Hardware:**	Bochs
Distribution:	SUSE Linux Enterprise Server 12 (x86_64)	**Service Pack:**	0
Hypervisor:	KVM (QEMU Virtual CPU)	**Identity:**	Virtual Machine (QEMU Virtual CPU)
Kernel Version:	3.12.14-1-default	**Supportconfig Version:**	3.0-18

Conditions Evaluated as Critical

Category			Message	Solutions
Basic Health			2 Basic Health Message(s)	
Basic Health	SLE	Kernel	Kernel Status -- Tainted: F O	TID
Basic Health	SLE	System	Last system down was not clean on Mon Mar 24 17:37:04 2014 and 1 additional failure(s)	TID TID1
SLE			2 SLE Message(s)	

Conditions Evaluated as Warning

Category			Message	Solutions
SLE			1 SLE Message(s)	

Conditions Evaluated as Recommended

Category			Message	Solutions
SLE			1 SLE Message(s)	

Conditions Evaluated as Success

Category			Message	Solutions
Security			1 Security Message(s)	
Security	SLE	AppArmor	There are no AppArmor reject messages	TID Doc
Basic Health			8 Basic Health Message(s)	
Basic Health	SLE	Kernel	Context switches per second observed: 79	TID
Basic Health	SLE	Kernel	Interrupts per second observed: 51	TID
Basic Health	SLE	CPU	Utilization: 1.00%, Idle: 99.00%	TID
Basic Health	SLE	Disk	Mount on / has highest used space: 22%	TID TID2
Basic Health	SLE	Kernel	2% CPU load within limits, CPUs: 1, Load Average: 0.02	TID Web Wikipedia
Basic Health	SLE	Memory	Memory used 29% - Swapping: No	TID
Basic Health	SLE	Processes	0 Uninterruptible processes observed	TID
Basic Health	SLE	Processes	0 Zombie processes observed	TID

圖形 39.1 SCA 工具產生的 HTML 報告

`scatool` 指令由 `sca-server-report` 套件提供。系統上預設不會安裝該套件。此外，您需要 `sca-patterns-base` 套件，以及與您想要在其上執行 `scatool` 指令的機器上所安裝產品相符之任一產品特定的 `sca-patterns-*` 套件。

以 `root` 使用者身分或者結合 `sudo` 執行 `scatool` 指令。在呼叫 SCA 工具時，您可以分析現有的 `supportconfig` TAR 歸檔，也可以讓該工具透過一個步驟產生並分析新的歸檔。該工具還提供了一個互動式控制台（具有 Tab 鍵補齊功能），並允許使用者在外部機器上執行 `supportconfig`，然後在本地機器上執行後續分析。

下面提供了一些範例指令：

`sudo scatool -s`

呼叫 `supportconfig` 並在本地機器上產生新的 supportconfig 歸檔。透過套用與所安裝產品相符的 SCA 分析模式來分析歸檔，以發現已知問題。顯示基於分析結果產生之 HTML 報告的路徑。通常，該報告會寫入 supportconfig 歸檔所在的同一個目錄。

`sudo scatool -s -o /opt/sca/reports/`

與 `sudo scatool -s` 類似，唯一的差別在於，HTML 報告會寫入 `-o` 選項指定的路徑。

`sudo scatool -a PATH_TO_TARBALL_OR_DIR`

分析指定的 supportconfig 歸檔檔案（或者 supportconfig 歸檔解壓縮到的指定目錄）。產生的 HTML 報告儲存在 supportconfig 歸檔或目錄所在的位置。

`sudo scatool -a SLES_SERVER.COMPANY.COM`

與外部伺服器 `SLES_SERVER.COMPANY.COM` 建立 SSH 連接，並在該伺服器上執行 `supportconfig`。supportconfig 歸檔隨後將複製回本地機器，並在該機器上進行分析。產生的 HTML 報告儲存在預設的 `/var/log` 目錄中。（`SLES_SERVER.COMPANY.COM` 上只建立 supportconfig 歸檔）。

`sudo scatool -c`

啓動 `scatool` 的互動式控制台。按 →| 兩次可查看可用指令。

關於其他選項和資訊，請執行 `sudo scatool -h` 或參閱 `scatool` 的 man 頁面。

39.4.2 SCA 裝置

如果您決定使用 SCA 裝置來分析 supportconfig 歸檔，則需要設定一台伺服器（或虛擬機器）做為專用的 SCA 裝置伺服器。然後，便可以使用 SCA 裝置伺服器，在企業中執行 SUSE Linux Enterprise Server 或 SUSE Linux Enterprise Desktop 的所有機器上分析 supportconfig 歸檔。您只需要將 supportconfig 歸檔上傳到該裝置伺服器，等待它進行分析。此程序無需任何互動。在 MariaDB 資料庫中，SCA 裝置將會追蹤已分析的所有 supportconfig 歸檔。您可以直接從裝置 Web 介面閱讀 SCA 報告。或者，可以讓裝置透過電子郵件將 HTML 報告傳送給任何管理使用者。如需詳細資料，請參閱第 39.4.2.5.4 節 「透過電子郵件傳送 SCA 報告」。

39.4.2.1 快速安裝

若要透過指令行快速安裝和設定 SCA 裝置，請依照此處的指示操作。該程序適用於進階使用者，並主要針對純安裝與指令的設定。如需詳細資訊，請參閱第 39.4.2.2 節 「先決條件」到第 39.4.2.3 節 「安裝與基本設定」中的詳細描述。

先決條件

- Web 與 LAMP 模式
- Web 與程序檔模組（您必須註冊機器才能選取此模組）。

注意：需要 root 權限

以下程序中的所有指令必須以 root 身分執行。

程序 39.3 使用匿名 FTP 進行上傳的安裝

設定並執行裝置後，將不再需要人工互動。因此，在使用 cron 工作建立和上傳 supportconfig 歸檔時，非常適合使用這種方法來設定裝置。

1. 在要安裝裝置的機器上，登入控制台並執行以下指令：

```
zypper install sca-appliance-* sca-patterns-* vsftpd
systemctl enable apache2
systemctl start apache2
systemctl enable vsftpd
systemctl start vsftpd
yast ftp-server
```

2. 在 YaST FTP 伺服器中，選取驗證 › 啓用上載 › 匿名使用者可上載 › 完成 › 是，以建立 /srv/ftp/upload。
3. 執行以下指令：

```
systemctl enable mysql
systemctl start mysql
mysql_secure_installation
setup-sca -f
```

mysql_secure_installation 將建立一個 MariaDB root 密碼。

程序 39.4 使用 SCP/TMP 進行上傳的安裝

這種設定裝置的方法需要在輸入 SSH 密碼時進行人工互動。

1. 在要安裝裝置的機器上，登入控制台。
2. 執行以下指令：

```
zypper install sca-appliance-* sca-patterns-*
systemctl enable apache2
systemctl start apache2
sudo systemctl enable mysql
systemctl start mysql
mysql_secure_installation
setup-sca
```

39.4.2.2 先決條件

若要執行 SCA 裝置伺服器，需要滿足以下先決條件：

- 安裝所有 `sca-appliance-*` 套件。
- 安裝 `sca-patterns-base` 套件。此外，需要為您想要使用裝置分析的 supportconfig 歸檔類型安裝產品特定的 `sca-patterns-*`。
- Apache
- PHP
- MariaDB
- 匿名 FTP 伺服器（選擇性）

39.4.2.3 安裝與基本設定

如第 39.4.2.2 節 「先決條件」中所列，SCA 裝置與其他套件存在若干相依性。因此，在安裝和設定 SCA 裝置伺服器之前，需要做一些準備工作：

1. 對於 Apache 和 MariaDB，需安裝 `Web` 和 `LAMP` 安裝模式。
2. 設定 Apache 和 MariaDB，並視需要設定一個匿名 FTP 伺服器。如需詳細資訊，請參閱 第 31 章 「Apache HTTP 伺服器」 與 第 32 章 「使用 YaST 設定 FTP 伺服器」。
3. 將 Apache 和 MariaDB 設定為在開機時啓動：

```
sudo systemctl enable apache2 mysql
```

4. 啓動這兩個服務：

```
sudo systemctl start apache2 mysql
```

現在，您便可以依照程序 39.5 「安裝和設定 SCA 裝置」中所述安裝和設定 SCA 裝置。

程序 39.5 安裝和設定 SCA 裝置

安裝這些套件後，可以使用 `setup-sca` 程序檔來對 SCA 裝置使用的 MariaDB 管理與報告資料庫進行基本設定。

使用該程序檔可以設定以下選項，以便將 supportconfig 歸檔從您的機器上傳到 SCA 裝置：

- `scp`
- 匿名 FTP 伺服器

1. 安裝裝置和 SCA 基本模式程式庫：

```
sudo zypper install sca-appliance-* sca-patterns-base
```

2. 此外，請為您要分析的 supportconfig 歸檔類型安裝模式套件。例如，如果您的環境中安裝了 SUSE Linux Enterprise Server 11 和 SUSE Linux Enterprise Server 12 伺服器，請安裝 `sca-patterns-sle11` 和 `sca-patterns-sle12` 這兩個套件。

   若要安裝所有可用模式：

```
zypper install sca-patterns-*
```

3. 若要對 SCA 裝置進行基本設定，請使用 `setup-sca` 程序檔。呼叫該指令的方式取決於您要以哪種方式將 supportconfig 歸檔上傳到 SCA 裝置伺服器：

   - 如果您設定了使用 `/srv/ftp/upload` 目錄的匿名 FTP 伺服器，請結合 `-f` 選項執行設定程序檔，並依照螢幕上的指示執行操作。

     ```
     setup-sca -f
     ```

注意：使用其他目錄的 FTP 伺服器

如果 FTP 伺服器使用的目錄不是 `/srv/ftp/upload`，請先調整以下組態檔案，使其指向正確的目錄：`/etc/sca/sdagent.conf` 與 `/etc/sca/sdbroker.conf`。

- 如果您要透過 `scp` 將 supportconfig 檔案上傳到 SCA 裝置伺服器的 `/tmp` 目錄，請不要使用任何參數，直接呼叫設定程序檔，然後依照螢幕上的指示執行操作：

```
setup-sca
```

該設定程序檔將會依據它的要求執行一些檢查，並設定所需的元件。它會提示您輸入兩個密碼：您設定之 MariaDB 的 MySQL `root` 密碼，以及用於登入 SCA 裝置 Web 介面的 Web 使用者密碼。

4. 輸入現有的 MariaDB `root` 密碼。SCA 裝置將使用該密碼連接到 MariaDB。

5. 定義 Web 使用者的密碼。該密碼將寫入 `/srv/www/htdocs/sca/web-config.php`，並設定為使用者 `scdiag` 的密碼。以後，您可隨時變更使用者名稱和密碼，請參閱第 39.4.2.5.1 節 「Web 介面的密碼」。

在成功完成安裝和設定後，便可以開始使用 SCA 裝置，請參閱第 39.4.2.4 節 「使用 SCA 裝置」。但是，您應修改部分選項，如變更 Web 介面的密碼、變更 SCA 模式更新來源、啓用歸檔模式，或者設定電子郵件通知。如需相關的詳細資訊，請參閱第 39.4.2.5 節 「自訂 SCA 裝置」。

警告：資料保護

由於 SCA 裝置伺服器上的報告包含已分析其 supportconfig 歸檔之機器的安全相關資訊，因此，請務必保護好 SCA 裝置伺服器上的資料，以防未經授權的人員存取。

39.4.2.4 使用 SCA 裝置

您可以將現有的 supportconfig 歸檔手動上傳到 SCA 裝置，也可以一步即完成建立新 supportconfig 歸檔並將其上傳到 SCA 裝置的操作。可以透過 FTP 或 SCP 來上傳。對於這兩種上傳方式，您需要知道可用來存取 SCA 裝置的 URL。要透過 FTP 上傳，需要為 SCA 裝置設定一台 FTP 伺服器，請參閱程序 39.5 「安裝和設定 SCA 裝置」。

39.4.2.4.1 將 Supportconfig 歸檔上傳到 SCA 裝置

- 若要建立 supportconfig 歸檔並透過（匿名）FTP 上傳：

```
sudo supportconfig -U “ftp://SCA-APPLIANCE.COMPANY.COM/upload”
```

- 若要建立 supportconfig 歸檔並透過 SCP 上傳：

```
sudo supportconfig -U “scp://SCA-APPLIANCE.COMPANY.COM/tmp”
```

  系統將提示您輸入執行 SCA 裝置之伺服器的 `root` 使用者密碼。

- 如果要手動上傳一或多個歸檔，請將現有的歸檔檔案（通常位於 `/var/log/nts_*.tbz`）複製到 SCA 裝置中。對於目標，請使用裝置伺服器的 `/tmp` 目錄或 `/srv/ftp/upload` 目錄（如果為 SCA 裝置伺服器設定了 FTP）。

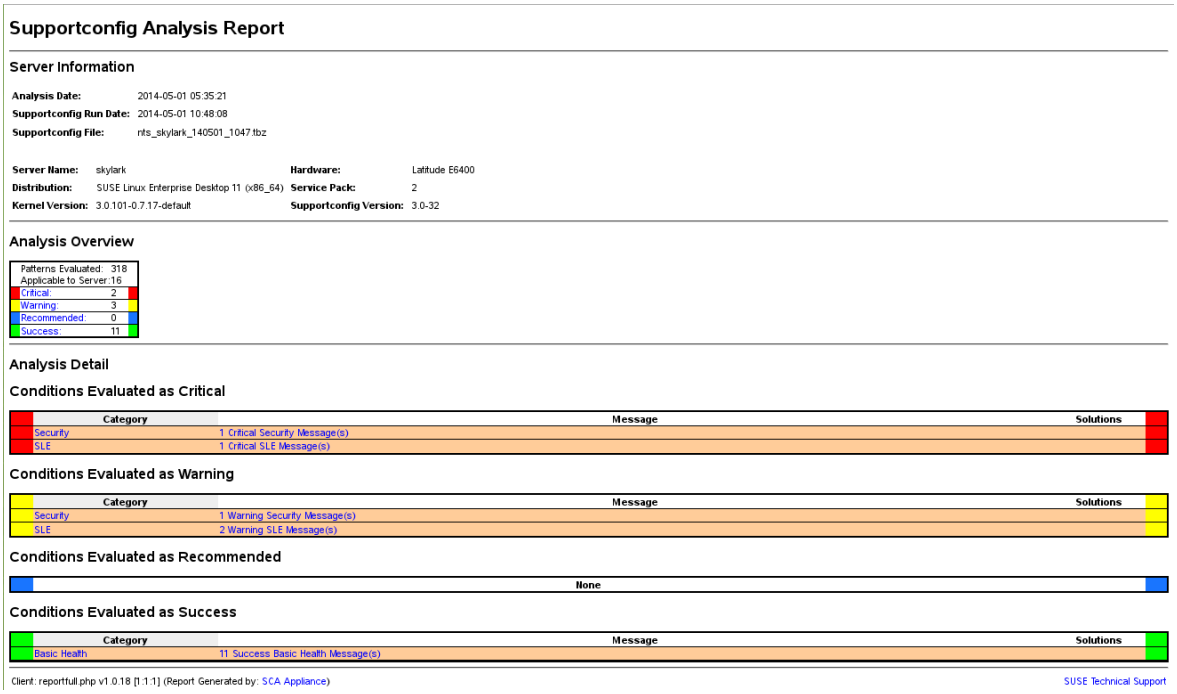
39.4.2.4.2 檢視 SCA 報告

可以在裝有瀏覽器並能存取 SCA 裝置之報告索引頁面的任何機器上檢視 SCA 報告。

1. 啓動網頁瀏覽器並確定 JavaScript 和 Cookie 已啓用。
2. 輸入 SCA 裝置的報告索引頁面做為 URL。

   ```
   https://sca-appliance.company.com/sca
   ```

   如有疑問，請諮詢您的系統管理員。
3. 系統將提示您輸入用於登入的使用者名稱和密碼。



Supportconfig Analysis Report

Server Information

Analysis Date: 2014-05-01 05:35:21
Supportconfig Run Date: 2014-05-01 10:48:08
Supportconfig File: nts_skylark_140501_1047.tbz

Server Name: skylark
Distribution: SUSE Linux Enterprise Desktop 11 (x86_64)
Kernel Version: 3.0.101-0.7.17-default
Hardware: Latitude E6400
Service Pack: 2
Supportconfig Version: 3.0-32

Analysis Overview

Patterns Evaluated:	318
Applicable to Server:	16
Critical:	2
Warning:	3
Recommended:	0
Success:	11

Analysis Detail

Conditions Evaluated as Critical

Category	Message	Solutions
Security	1 Critical Security Message(s)	
SLE	1 Critical SLE Message(s)	

Conditions Evaluated as Warning

Category	Message	Solutions
Security	1 Warning Security Message(s)	
SLE	2 Warning SLE Message(s)	

Conditions Evaluated as Recommended

None

Conditions Evaluated as Success

Category	Message	Solutions
Basic Health	11 Success Basic Health Message(s)	

Client: reportfull.php v1.0.18 [1.1.1] (Report Generated by: SCA Appliance) SUSE Technical Support

圖形 39.2 SCA 裝置產生的 HTML 報告

4. 登入後，按一下您要閱讀之報告的日期。

5. 首先按一下基本狀態類別將其展開。

6. 在訊息欄中按一下個別項目。SUSE 知識庫中的相應文章即會開啓。閱讀建議的解決方案，並遵循所述的指示操作。

7. 如果 Supportconfig 分析報告的解決方案欄顯示了其他項目，請按一下這些項目。閱讀建議的解決方案，並遵循所述的指示操作。

8. 請查看 SUSE 知識庫（http://www.suse.com/support/kb/），以瞭解與 SCA 識別之問題直接相關的結果。設法解決這些問題。

9. 檢查有無可前瞻性處理的結果，以免將來發生問題。

39.4.2.5 自訂 SCA 裝置

以下幾節顯示了如何變更 Web 介面的密碼、變更 SCA 模式更新來源、啓用歸檔模式，以及設定電子郵件通知。

39.4.2.5.1 Web 介面的密碼

SCA 裝置 Web 介面要求提供使用者名稱和密碼才能登入。預設的使用者名稱為 `scdiag`，預設的密碼為 `linux`（如果未做其他指定，請參閱程序 39.5 「安裝和設定 SCA 裝置」）。請儘早將預設密碼變更為一個較為安全的密碼。您也可以修改使用者名稱。

程序 39.6 變更 WEB 介面的使用者名稱或密碼

1. 在 SCA 裝置伺服器的系統控制台上以 `root` 使用者身分登入。
2. 在編輯器中開啓 `/srv/www/htdocs/sca/web-config.php`。
3. 視需要變更 `$username` 和 `$password` 的值。
4. 儲存檔案並離開。

39.4.2.5.2 SCA 模式的更新

依預設，所有 `sca-patterns-*` 套件將由一個 `root` cron 工作來定期更新，該工作將在夜間執行 `sdagent-patterns` 程序檔，而該程序檔又會執行 `zypper update sca-patterns-*`。定期的系統更新將會更新所有 SCA 裝置套件和模式套件。若要手動更新 SCA 裝置和模式，請執行：

```
sudo zypper update sca-*
```

系統預設從 SUSE Linux Enterprise 12 SP5 更新儲存庫安裝更新。如果需要，您可以將更新來源變更為某台 SMT 伺服器。當 `sdagent-patterns` 執行 `zypper update sca-patterns-*` 時，將從目前設定的更新通道中取得更新。如果該通道在 SMT 伺服器上，將從該伺服器提取套件。

程序 39.7 停用 SCA 模式的自動更新

1. 在 SCA 裝置伺服器的系統控制台上以 `root` 使用者身分登入。
2. 在編輯器中開啓 `/etc/sca/sdagent-patterns.conf`。
3. 將項目

   ```
   UPDATE_FROM_PATTERN_REPO=1
   ```

   變更為

```
UPDATE_FROM_PATTERN_REPO=0
```

4. 儲存檔案並離開。機器無需重新啓動就能套用變更。

39.4.2.5.3 歸檔模式

系統在分析了 supportconfig 歸檔並將其結果儲存在 MariaDB 資料庫中後，會從 SCA 裝置中刪除所有這些歸檔。但是，若要進行疑難排解，在機器中保留 supportconfig 歸檔的副本可能會有所幫助。依預設，歸檔模式處於停用狀態。

程序 39.8 在 SCA 裝置中啓用歸檔模式

1. 在 SCA 裝置伺服器的系統控制台上以 `root` 使用者身分登入。
2. 在編輯器中開啓 `/etc/sca/sdagent.conf`。
3. 將項目

   ```
   ARCHIVE_MODE=0
   ```

   變更為

   ```
   ARCHIVE_MODE=1
   ```

4. 儲存檔案並離開。機器無需重新啓動就能套用變更。

啓用歸檔模式後，SCA 裝置會將 supportconfig 檔案儲存至 `/var/log/archives/saved` 目錄，而不會將其刪除。

39.4.2.5.4 透過電子郵件傳送 SCA 報告

SCA 裝置可透過電子郵件傳送所分析之各 supportconfig 的 HTML 報告檔案。預設此功能是停用的。啓用該功能後，您可以定義要將報告傳送到的電子郵件地址清單，並定義會觸發報告傳送動作的狀態訊息層級（`STATUS_NOTIFY_LEVEL`）。

STATUS_NOTIFY_LEVEL 的可能值

$STATUS_OFF
: 停用傳送 HTML 報告功能。

$STATUS_CRITICAL
僅傳送包含「關鍵」狀態的 SCA 報告。

$STATUS_WARNING
僅傳送包含「警告」或「關鍵」狀態的 SCA 報告。

$STATUS_RECOMMEND
僅傳送包含「建議」、「警告」或「關鍵」狀態的 SCA 報告。

$STATUS_SUCCESS
傳送包含「成功」、「建議」、「警告」或「關鍵」狀態的 SCA 報告。

程序 39.9 為 SCA 報告設定電子郵件通知

1. 在 SCA 裝置伺服器的系統控制台上以 `root` 使用者身分登入。
2. 在編輯器中開啓 `/etc/sca/sdagent.conf`。
3. 搜尋 `STATUS_NOTIFY_LEVEL` 項目。該項目預設設定為 `$STATUS_OFF`（停用電子郵件通知）。
4. 若要啓用電子郵件通知，請將 `$STATUS_OFF` 變更為要針對其產生電子郵件報告的狀態訊息層級，例如：

   ```
   STATUS_NOTIFY_LEVEL=$STATUS_SUCCESS
   ```

   如需詳細資料，請參閱STATUS_NOTIFY_LEVEL 的可能值。
5. 若要定義要將報告傳送到的收件人清單：
   a. 搜尋 `EMAIL_REPORT='root'` 項目。
   b. 請使用您要向其傳送 SCA 報告的電子郵件地址清單取代 `root`。各電子郵件地址必須以空格分隔。例如：

      ```
      EMAIL_REPORT='tux@my.company.com wilber@your.company.com'
      ```
6. 儲存檔案並離開。機器無需重新啓動就能套用變更。以後產生的所有 SCA 報告都將透過電子郵件傳送到指定地址。

39.4.2.6 備份和還原資料庫

若要備份和還原儲存 SCA 報告的 MariaDB 資料庫，請依照如下所述使用 `scadb` 指令。

程序 39.10 備份資料庫

1. 在執行 SCA 裝置之伺服器的系統控制台上，以 `root` 使用者身分登入。
2. 執行以下指令以將裝置置於維護模式：

   ```
   scadb maint
   ```

3. 使用以下指令啓動備份程序：

   ```
   scadb backup
   ```

   資料將儲存到 TAR 歸檔 `sca-backup-*sql.gz` 中。
4. 如果您正在使用模式建立資料庫開發自己的模式（參閱第 39.4.3 節 「開發自訂分析模式」），則還要備份以下資料：

   ```
   sdpdb backup
   ```

   資料將儲存到 TAR 歸檔 `sdp-backup-*sql.gz` 中。
5. 將以下資料複製到另一台機器或外部儲存媒體中：

   - `sca-backup-*sql.gz`
   - `sdp-backup-*sql.gz`
   - `/usr/lib/sca/patterns/local`（僅當您已建立自訂模式時才需要複製該資料）
6. 使用以下指令重新啓動 SCA 裝置：

   ```
   scadb reset agents
   ```

程序 39.11 還原資料庫

若要基於您的備份還原資料庫，請按如下所述執行操作：

1. 在執行 SCA 裝置之伺服器的系統控制台上，以 `root` 使用者身分登入。
2. 將最新的 `sca-backup-*sql.gz` 和 `sdp-backup-*sql.gz` TAR 歸檔複製到 SCA 裝置伺服器。

3. 若要解壓縮檔案，請執行：

```
gzip -d *-backup-*sql.gz
```

4. 若要將資料輸入資料庫，請執行：

```
scadb import sca-backup-*sql
```

5. 如果您正在使用模式建立資料庫建立自己的模式，則還要透過以下指令輸入以下資料：

```
sdpdb import sdp-backup-*sql
```

6. 如果您正在使用自訂模式，則還要基於備份資料還原 `/usr/lib/sca/patterns/local`。

7. 使用以下指令重新啓動 SCA 裝置：

```
scadb reset agents
```

8. 使用以下指令更新資料庫中的模式模組：

```
sdagent-patterns -u
```

39.4.3 開發自訂分析模式

SCA 裝置隨附了一個完整的模式開發環境（SCA 模式資料庫），可讓您開發自己的自訂模式。模式可用任何程式設計語言編寫。若要使這些模式可用於 supportconfig 分析程序，需要將其儲存到 `/usr/lib/sca/patterns/local` 並使其可執行。然後，SCA 裝置與 SCA 工具將會針對做為分析報告一部分的新 supportconfig 歸檔執行這些自訂模式。關於如何建立（和測試）自己模式的詳細指示，請參閱http://www.suse.com/communities/conversations/sca-pattern-development/。

39.5 在安裝期間收集資訊

安裝期間無法使用 `supportconfig`。不過，您可以使用 `save_y2logs` 從 YaST 收集記錄檔案。此指令將在 `/tmp` 目錄下建立 `.tar.xz` 歸檔。

如果在安裝之初便出現問題，可以透過 `linuxrc` 建立的記錄檔案收集資訊。`linuxrc` 是在 YaST 啓動之前執行的小指令。此記錄檔案位於 `/var/log/linuxrc.log`。

重要：安裝記錄檔案在安裝後的系統中不可用

安裝期間可用的記錄檔案在安裝後的系統中已不再可用。您可以在安裝程式執行期間妥善儲存安裝記錄檔案。

39.6 核心模組支援

對於任何企業作業系統，一個重要的要求就是您獲得的環境方面的支援層級。核心模組是硬體（「控制器」）與作業系統之間最為相關的連接器。SUSE Linux Enterprise 中的每個核心模組都有一個 `supported` 旗標，該旗標可使用以下三個值：

- 「yes」，相當於 `supported`
- 「external」，相當於 `supported`
- 「」（空白，未設定），相當於 `unsupported`

以下規則適用：

- 依預設，自我重新編譯的核心的所有模組都會標示為 unsupported。
- SUSE 合作夥伴支援的核心模組以及使用 `SUSE SolidDriver 程式` 提供的核心模組會標示為 「external」。
- 如果未設定 `supported` 旗標，載入此模組便會污染該核心。系統不支援污染的核心。不支援的核心模組包含在一個附加的 RPM 套件（`kernel-FLAVOR-extra`）中，該套件只適用於 SUSE Linux Enterprise Desktop 和 SUSE Linux Enterprise 工作站延伸。預設不會載入這些核心（`FLAVOR=default|xen|...`）。此外，安裝程式中將不提供這些不受支援的模組，並且 `kernel-FLAVOR-extra` 套件也不會包含在 SUSE Linux Enterprise 媒體中。
- 不是依據與 Linux 核心授權相容的授權提供的核心模組也會污染核心。如需詳細資訊，請參閱 `/usr/src/linux/Documentation/sysctl/kernel.txt` 及 `/proc/sys/kernel/tainted` 的狀態。

39.6.1 技術背景

- Linux 核心：在 SUSE Linux Enterprise 12 SP5 上，/proc/sys/kernel/unsupported 的值預設設為 2（載入不受支援的模組時，syslog 中不發出警告）。安裝程式以及已安裝的系統中均使用此預設值。如需詳細資訊，請參閱 /usr/src/linux/Documentation/sysctl/kernel.txt。
- modprobe：用於檢查模組相依性及載入模組的 modprobe 公用程式會相應地檢查 supported 旗標的值。如果該值為 「yes」 或 「external」，則會載入該模組，否則不會載入。關於如何覆寫此行為的資訊，請參閱第 39.6.2 節 「使用不受支援的模組」。

注意：支援

SUSE 一般不支援透過 modprobe -r 移除儲存模組。

39.6.2 使用不受支援的模組

儘管廣泛可支援性非常重要，但有時會發生需要載入不受支援之模組的情況（例如，要進行測試或除錯，或者硬體供應商提供了 Hotfix）。

- 若要覆寫預設行為，請編輯 /etc/modprobe.d/10-unsupported-modules.conf，並將變數 allow_unsupported_modules 的值變更為 1。如果 initrd 中需要一個不受支援的模組，則請記得執行 dracut -f 以更新 initrd。
  如果只想嘗試載入模組一次，可將 --allow-unsupported-modules 選項與 modprobe 結合使用。如需詳細資訊，請參閱 modprobe 的 man 頁面。
- 在安裝期間，可透過驅動程式更新磁碟新增不受支援的模組，這樣便會載入這些模組。若要在開機期間以及開機後強制載入不受支援的模組，請使用核心指令行選項 oem-modules。安裝和啓始化 suse-module-tools 套件時，系統將評估核心旗標 TAINT_NO_SUPPORT（/proc/sys/kernel/tainted）。如果核心已污染，將啓用 allow_unsupported_modules。這可以防止不受支援的模組在正在安裝的系統中載入失敗。如果安裝期間沒有任何不受支援的模組，並且未使用其他特殊的核心指令行選項（oem-modules=1），則預設行為仍是禁止不受支援的模組。

請記住，載入和執行不受支援的模組會導致 SUSE 不支援該核心和整個系統。

39.7 更多資訊

- `man supportconfig` — `supportconfig` 的 man 頁面。
- `man supportconfig.conf` — supportconfig 組態檔案的 man 頁面。
- `man scatool` — `scatool` 的 man 頁面。
- `man scadb` — `scadb` 的 man 頁面。
- `man setup-sca` — `setup-sca` 的 man 頁面。
- https://mariadb.com/kb/en/ — MariaDB 文件。
- http://httpd.apache.org/docs/ 和第 31 章 「Apache HTTP 伺服器」 — 關於 Apache Web 伺服器的文件。
- 第 32 章 「使用 YaST 設定 FTP 伺服器」 — 關於如何設定 FTP 伺服器的文件。
- http://www.suse.com/communities/conversations/sca-pattern-development/ — 關於如何建立（和測試）自己的 SCA 模式的指示。
- http://www.suse.com/communities/conversations/basic-server-health-check-supportconfig/ — 使用 Supportconfig 進行的基本伺服器狀態檢查。
- https://www.novell.com/communities/coolsolutions/cool_tools/create-your-own-supportconfig-plugin/ — 建立自己的 Supportconfig 外掛程式。
- http://www.suse.com/communities/conversations/creating-a-central-supportconfig-repository/ — 建立中心 Supportconfig 儲存庫。

40 一般問題和解決方案

本章介紹一些可能會發生的問題及其解決方案。即使這裡沒有與您完全相同的情況，或許可以從類似情況中獲得一些提示，來解決您遇到的問題。

40.1 尋找並收集資訊

Linux 會非常詳細地報告事件。當系統發生問題時，可以從幾個地方查看相關資訊，主要是 Linux 系統的標準記錄檔案，也有與 SUSE Linux Enterprise Server 系統相關的記錄檔案。大部分記錄檔案都可以透過 YaST（其他 › 檢視開機記錄 」）進行檢視。

YaST 可讓您收集支援團隊所需的所有系統資訊。使用其他 › 支援，然後選取問題類別。在收集到所有資訊之後，將此份資訊連結到您的支援要求。

以下是最常查看的記錄檔案清單，以及它們各自的一般用途。包含 ~ 的路徑表示目前使用者的主目錄。

表格 40.1 記錄檔案

Log File	描述
~/.xsession-errors	來自目前執行中桌上應用程式的訊息。
/var/log/apparmor/	來自 AppArmor 的記錄檔案，請參閱《Security Guide》 以獲得詳細資訊。
/var/log/audit/audit.log	來自 Audit 的記錄檔案，可追蹤對檔案、目錄或系統資源的存取，並追蹤系統呼叫。請參閱《Security Guide》以獲得詳細資訊。
/var/log/mail.*	來自郵件系統的訊息。
/var/log/NetworkManager	NetworkManager 中的記錄檔案，用於收集網路連接性的問題

Log File	描述
/var/log/samba/	目錄包含 Samba 伺服器和用戶端記錄訊息。
/var/log/warn	來自核心和系統記錄精靈的所有訊息，均為「警告」或以上等級。
/var/log/wtmp	二進位檔案包含使用者對於目前機器工作階段的登入記錄。請以 `last` 檢視。
/var/log/Xorg.*.log	來自 X Window System 的多種啓動和執行時期記錄檔案。對於 X 啓動失敗的除錯非常實用。
/var/log/YaST2/	目錄包含 YaST 的動作和其結果。
/var/log/zypper.log	Zypper 的記錄檔案。

與記錄檔不同的是，您的機器亦提供您執行中系統的資訊。請參閱表格 40.2: /proc 檔案系統的系統資訊

表格 40.2 /proc 檔案系統的系統資訊

檔案	描述
/proc/cpuinfo	包含處理器資訊，如類型、廠商、型號與效能。
/proc/dma	顯示目前使用的 DMA 頻道。
/proc/interrupts	顯示使用中的岔斷，以及每種岔斷正在使用的數量。
/proc/iomem	顯示 I/O（輸入/輸出）記憶體的狀態。
/proc/ioports	顯示此時正在使用的 I/O 連接埠。
/proc/meminfo	顯示記憶體狀態。

檔案	描述
/proc/modules	顯示個別模組。
/proc/mounts	顯示目前掛接的裝置。
/proc/partitions	顯示所有硬碟的分割區。
/proc/version	顯示目前的 Linux 版本。

除了 /proc 檔案系統，Linux 核心還會輸出 sysfs 模組（記憶體內檔案系統）的相關資訊。此模組代表核心物件及其屬性和關係。如需有關 sysfs 的詳細資訊，請參閱第 21 章 「使用 udev 進行動態核心裝置管理」 中 udev 的相關內容。表格 40.3 提供了 /sys 下最常用目錄的綜覽。

表格 40.3 /sys 檔案系統的系統資訊

檔案	描述
/sys/block	包含系統中探查到之每個區塊裝置的子目錄。一般情況下，大部分裝置都是磁碟類型的裝置。
/sys/bus	包含每種實體匯流排類型的子目錄。
/sys/class	包含組合為功能型裝置（如圖形、網路和印表機等）的多個子目錄
/sys/device	包含全域裝置階層。

Linux 隨附多種工具可進行系統分析和監控。請參閱《System Analysis and Tuning Guide》，第 2 章「System Monitoring Utilities」以取得用於系統診斷最重要的選項。

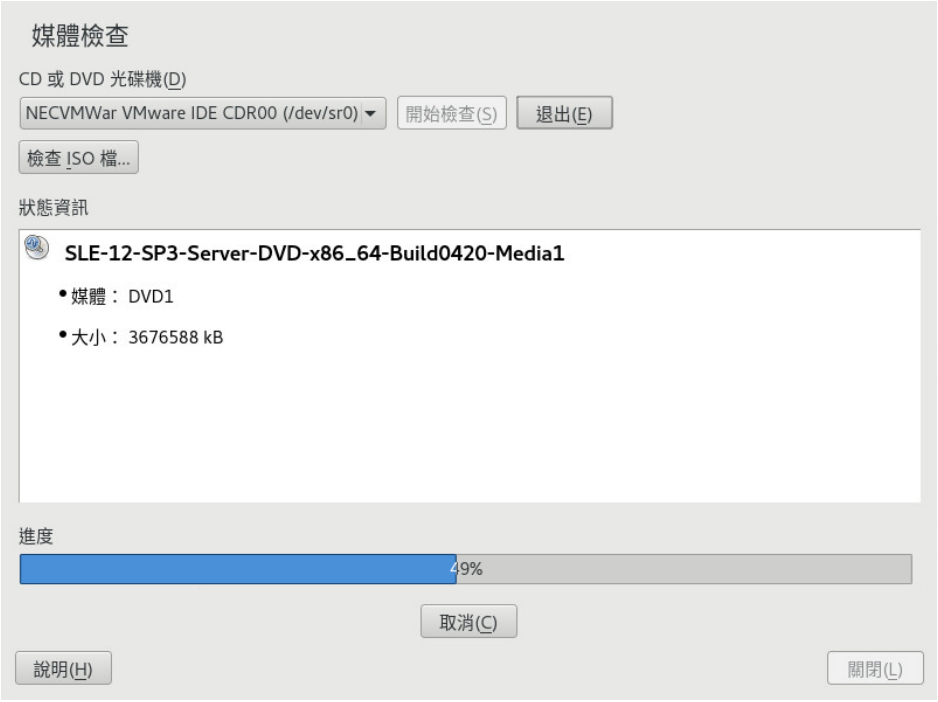
下面的每種情境所採用的編排順序是，先在標題中描述問題，隨後用一兩個段落提供建議的解決方案、詳細解決方案的參考資料，以及對其他相關情境的交叉參照。

40.2 安裝問題

安裝問題是安裝機器時失敗的一個狀況。可能是徹底失敗，也可能是無法啓動圖形安裝程式。本節重點介紹了您可能會遭遇的一般性問題，並對這些狀況提供了可能的解決方案或處理方式。

40.2.1 檢查媒體

若您在使用 SUSE Linux Enterprise Server 安裝媒體時遇到問題，請檢查安裝媒體的完整性。從該媒體開機，然後從開機功能表中選擇檢查安裝媒體。在執行中的系統上，啓動 YaST 並選擇軟體 › 媒體檢查。若要檢查 SUSE Linux Enterprise Server 媒體，請將其插入磁碟機，並按一下 YaST媒體檢查螢幕中的開始檢查。這可能會花費幾分鐘。如果偵測到錯誤，請勿使用此媒體來進行安裝。媒體問題可能是您在自行燒錄媒體時發生的。以較低的速度（4x）燒錄媒體有助於避免出現問題。



圖形 40.1 檢查媒體

40.2.2 無可用的可開機 DVD 磁碟機

如果您的電腦沒有可開機的 DVD-ROM 磁碟機，或是 Linux 不支援您的磁碟機，可以利用幾個選項在不使用內建 DVD 磁碟機的情況下安裝機器：

使用外接開機裝置

如果 BIOS 和安裝核心支援，您可以從外接 DVD 光碟機或 USB 儲存裝置開機。如需如何建立可開機 USB 儲存裝置的指示，請參閱《部署指南》，第 6 章「使用 YaST 安裝」，第 6.2.2 節「PC (AMD64/Intel 64/ARM AArch64)：系統啓動」。

透過 PXE 以網路開機

如果機器沒有 DVD 光碟機，但提供可用的乙太網路連線，則可採用全網路式安裝。請參閱《部署指南》，第 10 章「遠端安裝」，第 10.1.3 節「透過 VNC 執行遠端安裝 — PXE 開機和網路喚醒功能」和《部署指南》，第 10 章「遠端安裝」，第 10.1.6 節「透過 SSH 執行遠端安裝 — PXE 開機和網路喚醒功能」以取得詳細資料。

40.2.2.1 外接開機裝置

Linux 支援大部分現有的 DVD 磁碟機。如果系統沒有 DVD 光碟機，仍然可以使用 USB、FireWire 或 SCSI 來連接外接式 DVD 光碟機，以執行系統開機。這操作主要依賴 BIOS 與所用硬體之間的互動。如果您遭遇到問題，有時更新 BIOS 可能會有幫助。從即時 CD 安裝時，也可以建立用於開機的「即時 隨身碟」。

40.2.3 從安裝媒體開機失敗

機器無法開機安裝媒體的一個原因可能是 BIOS 中的開機順序設定不正確。在 BIOS 開機順序中，必須將 DVD 磁碟機設為開機的第一項。否則機器會嘗試從其他媒體開機，一般會從硬碟開機。如需變更 BIOS 開機順序的指導，可參閱隨主機板隨附的文件，或者參閱下文段落。

BIOS 是提供電腦最基本功能的軟體。主機板供應商會針對自己的硬體提供特製的 BIOS。通常，BIOS 設定只可在特定時間（機器開機時）進行存取。在這個初始化階段，機器會執行一些硬體診斷測試。其中之一是記憶體檢查，由記憶體計數器指示。

當計數器出現時，請尋找指示按下按鍵來存取 BIOS 設定的一行文字，通常在計數器下方或底端某個位置。要按的鍵通常是 Del、F1 或 Esc 中的一個。請按住這個按鍵，直到 BIOS 設定畫面出現為止。

程序 40.1 變更 BIOS 開機順序

1. 使用開機常式所宣告的正確按鍵進入 BIOS，等待 BIOS 畫面出現。
2. 要變更 AWARD BIOS 中的開機順序，請尋找 BIOS FEATURES SETUP（BIOS 功能設定）項目。其他製造商可能使用不同的名稱，例如 ADVANCED CMOS SETUP（進階 CMOS 設定）。當您找到該項目後，請選取並按 Enter 確認。
3. 在隨後開啓的畫面中，找到BOOT SEQUENCE或BOOT ORDER(開機順序) 子項目。按 Page ↑ 或 Page ↓ 鍵來變更設定，直到 DVD 光碟機列在最前面。
4. 請按 Esc 來離開 BIOS 設定畫面。若要儲存變更，請選取SAVE & EXIT SETUP(儲存並結束設定)，或者按 F10。要確認儲存設定，請按 Y。

程序 40.2 在 SCSI BIOS（ADAPTEC 主機介面卡）中變更開機順序

1. 按 Ctrl—A 開啓設定。
2. 選取磁碟公用程式。連接的硬體元件隨即顯示。
   記下您 DVD 磁碟機的 SCSI ID。
3. 使用 Esc 離開功能表。
4. 開啓 Configure Adapter Settings（設定介面卡設定）。在 Additional Options（其他選項）下，請選取 Boot Device Options（開機裝置選項），然後按 Enter。
5. 輸入 DVD 磁碟機的 ID，然後再按 Enter。
6. 按兩下 Esc 回到 SCSI BIOS 的開始畫面。
7. 退出這個畫面，接著按 Yes（是）來啓動電腦。

不管最終安裝將使用何語言和鍵盤配置，大多數 BIOS 組態都使用美國鍵盤配置，如下圖所示：

圖形 40.2 美國鍵盤配置

40.2.4 無法開機

有些硬體類型（多半是極舊或極新的機型）無法安裝。發生此狀況的原因往往是安裝核心不支援此類型的硬體，或是此核心中包含的某些功能（如 ACPI）會對一些硬體造成問題。

若您的系統無法使用標準安裝模式，從第一個安裝開機畫面安裝的話，請嘗試下列方法：

1. 使用尚留在磁碟機中的 DVD，以 Ctrl—Alt—Del 或硬體重設按鈕重新開機。
2. 出現開機畫面時按 F5，並使用鍵盤上的方向鍵導覽至無 ACPI，然後按 Enter 啓動開機與安裝程序。此選項會停用 ACPI 電源管理技術的支援。
3. 如《部署指南》，第 6 章「使用 YaST 安裝」 所述，繼續安裝。

如果失敗，請仍按上述步驟執行，但改選安全設定。此選項會停用 ACPI 和 DMA 支援。大部分硬體將以此選項開機。

若這些選項都失敗的話，請使用開機選項提示，將支援此類硬體所需的其他參數傳送到安裝核心。如需關於可做為開機選項之參數的詳細資訊，請參閱位於 `/usr/src/linux/Documentation/kernel-parameters.txt` 的核心文件。

提示：取得核心文件

安裝 `kernel-source` 套件以檢視核心文件。

在開機以完成安裝之前，可以在開機提示中輸入其他與 ACPI 相關的核心參數：

`acpi=off`
: 此參數會關閉電腦的所有 ACPI 子系統。如果您的電腦無法處理 ACPI 或者您認為電腦的 ACPI 造成問題，此參數會很有幫助。

`acpi=force`
: 永遠啓用 ACPI，即使電腦的 BIOS 出廠日期是在 2000 年以前。若沒有使用 `acpi=off`，設定此參數也會啓用 ACPI。

`acpi=noirq`
: 不將 ACPI 用於 IRQ 路由。

`acpi=ht`
: 只執行足夠啓用超執行緒的 ACPI。

`acpi=strict`
: 降低對不完全與 ACPI 規格相容之平台的容忍度。

`pci=noacpi`
: 停用新 ACPI 系統的 PCI IRQ 路由。

`pnpacpi=off`
: 當您的 BIOS 設定中包含錯誤的岔斷或連接埠時，可透過此選項解決序列或平行問題。

`notsc`
: 停用時戳計數器。可使用此選項解決系統中的計時問題。這是一項新功能。如果您發現機器上出現效能衰退，特別是越到最後，效能越低，甚至完全當機，不妨嘗試此選項。

`nohz=off`
: 停用 nohz 功能。如果您的機器當機，使用此選項可能會有所幫助。其他情況下則毫無用處。

一旦您判斷出正確的參數組合，YaST 就會自動將其寫入開機載入程式組態，以確定系統下次可正確開機。

如果核心載入或者安裝時發生不明錯誤，選取開機功能表的記憶體測試，檢查記憶體。若記憶體測試傳回錯誤，則通常會是硬體錯誤。

40.2.5 無法啓動圖形安裝程式

將媒體插入磁碟機並將系統重新開機後，會出現安裝畫面，但在選取安裝之後，圖形安裝程式並未啓動。

有許多方法可以解決此狀況：

- 嘗試選取安裝對話方塊的其他螢幕解析度。
- 選取文字模式進行安裝。
- 透過 VNC，使用圖形安裝程式進行遠端安裝。

程序 40.3 變更安裝的畫面解析度

1. 開機以進行安裝。
2. 按 F3 開啓功能表，從中選取較低解析度進行安裝。
3. 選取安裝並如 《部署指南》，第 6 章「使用 YaST 安裝」 所述繼續安裝。

程序 40.4 在文字模式下安裝

1. 開機以進行安裝。
2. 按下 F3 並選取文字模式。
3. 選取安裝並如 《部署指南》，第 6 章「使用 YaST 安裝」 所述繼續安裝。

程序 40.5 安裝 VNC

1. 開機以進行安裝。
2. 在開機選項提示中輸入下列文字：

   ```
   vnc=1 vncpassword=SOME_PASSWORD
   ```

   以 VNC 安裝所用的密碼取代 SOME_PASSWORD。

3. 選取 安裝，然後按 Enter 啓動安裝。

   這樣不會直接啓動圖形安裝常式，系統會繼續以文字模式執行，然後終止，顯示含有 IP 位址與連接埠號碼的訊息，根據此訊息便可以透過瀏覽器介面或 VNC 檢視器應用程式找到安裝程式。

4. 如果使用瀏覽器來存取安裝程式，則啓動瀏覽器並輸入 SUSE Linux Enterprise Server 機器安裝常式所提供的位址資訊，並按 Enter：

   ```
   http://IP_ADDRESS_OF_MACHINE:5801
   ```

   瀏覽器視窗中會開啓一個對話方塊，提示您輸入 VNC 密碼。輸入密碼，並如《部署指南》，第 6 章「使用 YaST 安裝」 所述繼續安裝。

   重要：跨平台支援

   若要在任何作業系統下、使用任何瀏覽器、透過 VNC 工作，首先必須啓用 Java 支援。

   出現提示時，提供 VNC 檢視器的 IP 位址和密碼。會開啓一個視窗，顯示安裝對話。請依一般方式繼續安裝。

40.2.6 只有極簡開機畫面被啓動

將媒體插入磁碟機後，BIOS 常式結束，但系統未啓動圖形開機畫面。而是啓動一個非常簡化的文字介面。若機器無法提供足夠的圖形記憶體以轉譯圖形開機畫面，就可能發生此狀況。

雖然文字開機畫面看起來簡化，但其提供的功能幾乎與圖形介面一樣：

開機選項
: 與圖形介面不同的是，這裡無法以鍵盤游標選取開機選項。文字模式開機畫面的開機功能表，會在開機提示時提供一些可輸入的關鍵字。這些關鍵字對映到圖形版本所提供的選項。輸入您的選擇並按 Enter 以啓動開機程序。

自訂開機選項
: 選取開機選項之後，在開機提示中輸入適當的關鍵字，或如第 40.2.4 節 「無法開機」所述輸入自訂開機選項。若要啓動安裝程序，請按下 Enter。

螢幕解析度

使用功能鍵（F1 ... F12）確定安裝的螢幕解析度。若您需要以文字模式開機，請選擇 F3。

40.2.7 記錄檔案

如需安裝期間建立之記錄檔案的詳細資訊，請參閱第 39.5 節 「在安裝期間收集資訊」。

40.3 開機問題

開機問題指的是您的系統無法正常開機的狀況（無法開機進入預期的目標和登入螢幕）。

40.3.1 GRUB 2 開機載入程式無法載入

若硬體運作正常，則可能是開機載入程式已損毀，Linux 無法在機器上啓動。若是這樣的話，必須修復開機載入程式。為此，您需要依第 40.6.2 節 「使用救援系統」中所述啓動救援系統，然後遵照第 40.6.2.4 節 「修改和重新安裝開機載入程式」中的指示。

此外，您可以依照以下方式使用救援系統來修復開機載入程式。從安裝媒體將機器開機。在開機螢幕中，選擇更多 › 將 Linux 系統開機。使用預設核心選項選取包含所安裝系統和核心的磁碟。

系統開機後，啓動 YaST 並切換到系統 › 開機載入程式。確認啓用了將一般開機碼寫入 MRB 選項，然後按確定。如此會透過重寫來修復損毀的開機載入程式，或者安裝缺失的開機載入程式。

另一個機器無法開機的原因可能跟 BIOS 有關：

BIOS 設定

檢查與您硬碟相關的 BIOS 設定。若在目前的 BIOS 設定下找不到硬碟本身，則可能只是未啓動 GRUB 2。

BIOS 開機順序

檢查您系統的開機順序是否包含硬碟。若未啓用硬碟選項，您的系統可能已正確安裝，但在需要存取硬碟時無法開機。

40.3.2 沒有圖形登入

如果機器能啓動，但無法開機進入圖形登入管理員，則問題可能出在預設的 systemd 目標選項或 X Window System 的組態上。若要檢查目前的 systemd 預設目標，請執行指令 `sudo systemctl get-default`。如果傳回的值為 not `graphical.target`，請執行指令 `sudo systemctl isolate graphical.target`。如果圖形登入螢幕已啓動，請登入並啓動YaST › 系統 › 服務管理員，然後將預設系統目標設定為圖形介面。此後，系統應該能夠開機進入圖形登入螢幕。

如果即使已開機或者切換到圖形目標，圖形登入螢幕也不啓動，則原因可能是您的桌面或 X Window 軟體設定錯誤或者已毀損。請檢查 `/var/log/Xorg.*.log` 中的記錄檔，以瞭解 X 伺服器嘗試啓動時的訊息。如果啓動期間桌面發生錯誤，可能會在系統日誌中記錄錯誤訊息，您可以使用指令 `journalctl` 查詢該日誌（如需詳細資訊，請參閱第 15 章 「`journalctl`：查詢 `systemd` 日誌」）。若這些錯誤訊息指出 X 伺服器中有組態問題，請嘗試修復這些問題。若仍然未出現圖形系統，請考慮重新安裝圖形桌面。

40.3.3 無法掛接 Btrfs 根分割區

如果 `btrfs` 根分割區已毀損，請嘗試以下選項：

- 使用 `-o recovery` 選項掛接該分割區。
- 如果不起作用，請在您的根分割區上執行 `btrfs-zero-log`。

40.3.4 強制檢查根分割區

如果根分割區已損毀，請在開機提示字元中使用參數 `forcefsck`。如此即可將選項 `-f`（force）傳遞給 `fsck` 指令。

40.4 登入問題

登入問題是指機器雖然開機到預期的歡迎畫面或登入提示畫面，卻拒絕接受使用者名稱和密碼，或者雖然接受了使用者名稱和密碼，但是行為異常（無法啓動圖形桌面、發生錯誤或轉到了指令行等）。

40.4.1 有效的使用者名稱和密碼組合失敗

這種情形常發生於系統設定為使用網路驗證或目錄服務時，且基於某些原因，會無法從其所設定的伺服器取得結果。身為唯一的本地使用者，根使用者是唯一仍可登入這些機器的使用者。下面是機器看似運作良好卻無法正確執行登入的一些常見原因：

- 網路未作用。如須對此情況的進一步指示，請參閱第 40.5 節 「網路問題」。
- DNS 此時未運作（這樣會阻礙 GNOME 運作，也會妨礙系統驗證安全伺服器的要求）。若機器花費過久的時間回應任何動作的話，表示可能是這種情況。如需此主題的詳細資訊，請參閱第 40.5 節 「網路問題」。
- 若系統設定為使用 Kerberos，則系統的本地時間有可能超過了 Kerberos 伺服器時間所容許的時間差（一般為 300 秒）。若 NTP（網路時間協定）未正確運作，或本地 NTP 伺服器未運作，則 Kerberos 驗證會停止作用，因為它必須仰賴網路上同步的共同時脈才可運作。
- 系統的驗證組態設定錯誤。請檢查 PAM 組態檔案是否有錯字或指示詞順序錯誤。如需關於 PAM 和所包含組態檔案語法的其他背景資料，請參閱《Security Guide》，第 2 章「Authentication with PAM」。
- 主分割區已加密。如需此主題的詳細資訊，請參閱第 40.4.3 節 「無法登入加密的主分割區」。

對於所有非外部網路造成的問題，解決方案就是重新開機進入單一使用者模式，並修復組態後再次開機進入操作模式，以嘗試重新登入。若要開機進入單一使用者模式：

1. 重新啓動系統。會出現開機畫面及提示。
2. 按 Esc 離開開頭顯示畫面，並前往 GRUB 2 文字式功能表。
3. 按 B 進入 GRUB 2 編輯器。

4. 將以下參數新增到包含核心參數的行中：

```
systemd.unit=rescue.target
```

5. 按「F10」。

6. 輸入 `root` 的使用者名稱與密碼。

7. 進行必要的所有變更。

8. 在指令行中輸入 `systemctl isolate graphical.target`，開機進入完整多重使用者及網路模式。

40.4.2 有效的使用者名稱和密碼不被接受

這顯然是使用者最常遇到的問題，其發生的原因有很多。根據您使用本地使用者管理和驗證，或使用網路驗證，會有不同原因造成登入失敗。

本地使用者管理可能因為下列原因而失敗：

- 使用者輸入的密碼有誤。
- 使用者包含桌面組態檔的主目錄損毀或有防寫保護。
- X Window System 可能無法驗證此特定使用者，尤其是在安裝目前版本的 Linux 之前，此使用者的主目錄已用於其他 Linux 版本的情況下。

若要找出本地登入失敗的原因，請執行下列步驟：

1. 開始進行整個驗證機制的偵錯之前，先確認使用者可以正確記住密碼。若使用者無法正確記住密碼，請使用 YaST 使用者管理模組變更使用者的密碼。請注意 Caps Lock 鍵的使用，並根據需要進行切換。

2. 以 `root` 身分登入，並使用 `journalctl -e` 檢查系統日誌，找出登入程序和 PAM 的錯誤訊息。

3. 嘗試從主控台登入（使用 Ctrl—Alt—F1）。如果成功，表示問題不在 PAM，因為它能夠在此機器上驗證此使用者。嘗試找出 X Window System 或 GNOME 桌面的任何問題。若需更多資訊，請參閱第 40.4.4 節 「登入成功但 GNOME 桌面失敗」。

4. 若使用者的主目錄已由其他 Linux 版本使用，請移除使用者主目錄中的 `Xauthority` 檔案。使用主控台透過 Ctrl—Alt—F1 登入，並以此使用者身份執行 `rm .Xauthority`。這樣應可排除此使用者的 X 驗證問題。重新嘗試圖形登入。

5. 若由於組態檔案毀損導致桌面無法啓動，請繼續執行第 40.4.4 節 「登入成功但 GNOME 桌面失敗」。

下面列出了特定使用者在特定機器上網路驗證失敗的一些常見原因：

- 使用者輸入的密碼有誤。
- 機器的本地驗證檔案中已存在使用者名稱，但網路驗證系統也提供了，兩者產生了衝突。
- 主目錄是存在的，但損毀或無法使用。或許此目錄設為防止寫入，或位於此時無法存取的伺服器上。
- 使用者沒有登入驗證系統特定主機的許可。
- 機器的主機名稱已因某種原因而變更，而使用者沒有登入該主機的許可。
- 機器無法聯繫驗證伺服器，或是含有使用者資訊的目錄伺服器。
- X Window System 可能無法驗證此特定使用者，尤其是在安裝目前版本的 Linux 之前，此使用者的主目錄已用於其他 Linux 版本的情況下。

若要找出登入發生網路驗證失敗的原因，請執行下列步驟：

1. 進行整個驗證機制的除錯之前，請先確認使用者所記的密碼正確無誤。
2. 確定機器賴以進行驗證的目錄伺服器，並確定該伺服器已啓動且正在執行，而且能夠與其他機器正常進行通訊。
3. 確定使用者的使用者名稱和密碼可以在其他機器上使用，以確定其驗證資料存在，而且已正確配送。
4. 再看看在運作不正常的機器上，可否讓其他使用者登入。若其他使用者可以正常登入，或 `root` 可以登入的話，請登入並使用 `journalctl -e>` 檔案檢驗系統日誌。找出嘗試登入所對應的時間戳記，並判斷 PAM 是否已產生任何錯誤訊息。

5. 嘗試從主控台登入（使用 Ctrl—Alt—F1）。若是成功，說明問題不是出在 PAM 或使用者主目錄所在的目錄伺服器，因為能夠在此機器上驗證此使用者。嘗試找出 X Window System 或 GNOME 桌面的任何問題。若需更多資訊，請參閱第 40.4.4 節 「登入成功但 GNOME 桌面失敗」。

6. 若使用者的主目錄已由其他 Linux 版本使用，請移除使用者主目錄中的 `Xauthority` 檔案。使用主控台透過 Ctrl—Alt—F1 登入，並以此使用者身份執行 `rm .Xauthority`。這樣應可排除此使用者的 X 驗證問題。重新嘗試圖形登入。

7. 若由於組態檔案毀損導致桌面無法啓動，請繼續執行第 40.4.4 節 「登入成功但 GNOME 桌面失敗」。

40.4.3 無法登入加密的主分割區

建議對筆記型電腦使用加密的主分割區。如果無法登入您的筆記型電腦，通常只是因為無法解除鎖定您的分割區。

開機期間，您需要輸入密碼片語以解除鎖定加密的分割區。如果不輸入密碼片語，則開機程序會繼續，但分割區將處於鎖定狀態。

若要解除鎖定加密的分割區，請執行以下步驟：

1. 按 Ctrl—Alt—F1 切換到文字主控台。

2. 以 `root` 使用者身分登入。

3. 使用以下指令重新啓動解除鎖定程序：

```
systemctl restart home.mount
```

4. 輸入可解除鎖定加密分割區的密碼片語。

5. 按 Alt—F7 離開文字主控台並切換回登入畫面。

6. 像往常一樣登入。

40.4.4 登入成功但 GNOME 桌面失敗

若出現此狀況，可能是 GNOME 組態檔案已損毀。其他症狀還包括鍵盤無法工作、畫面幾何錯亂，甚至畫面呈現空白的灰色區塊。此問題最重要的區隔是，若其他使用者可以登入，則此機器是正常運作的。此類問題或許可以較快解決，只要將使用者的 GNOME 組態目錄移到新的位置，讓 GNOME 啓始化新的組態目錄即可。雖然這樣算是强制使用者重新設定 GNOME，但並沒有資料因此遺失。

1. 按 Ctrl—Alt—F1 切換到文字主控台。
2. 使用您的使用者名稱登入。
3. 將使用者的 GNOME 組態目錄移到一個暫時位置：

```
mv .gconf  .gconf-ORIG-RECOVER
mv .gnome2 .gnome2-ORIG-RECOVER
```

4. 登出。
5. 再次登入，勿執行任何應用程式。
6. 依照下列方式將 `~/.gconf-ORIG-RECOVER/apps/` 目錄複製回新的 `~/.gconf` 目錄，修復您的個別應用程式組態資料（包括 Evolution 電子郵件用戶端資料）：

```
cp -a .gconf-ORIG-RECOVER/apps .gconf/
```

若這樣導致登入問題，請嘗試僅修復重要的應用程式資料，並重新設定其他的應用程式。

40.5 網路問題

您系統的許多問題可能都與網路有關，但可能一開始看不出來。例如，系統不允許使用者登入，可能就是某種網路問題所致。本節會介紹簡單的檢查清單，可讓您用來辨識所遇到網路問題的原因。

程序 40.6 如何識別網路問題

檢查機器網路連接時，請如下執行：

1. 若使用乙太網路連線，請先檢查硬體。確認您的網路線正確插在電腦和路由器（或集線器等）上。正常情況下，乙太網路連接器旁的兩個控制燈都會亮起。

若連接失敗，請檢查網路線在其他機器上可否使用。若可以的話，就是您的網路卡造成的問題。若您的網路設定中包含集線器或交換器，也可能是這些裝置的問題。

2. 若使用無線連接的話，請檢查可否由其他機器建立無線連結。若無法建立，請聯絡無線網路管理員。

3. 檢查完基本網路連接之後，請嘗試找出未回應的服務為何。收集您設定中所需所有網路伺服器的位址資訊。您可在適當的 YaST 模組中查詢，或詢問您的系統管理員。下列清單提供了設定中所包含的一些基本的網路伺服器，以及其故障的症狀。

   DNS（名稱服務）
   : 名稱服務損壞或故障會從許多方面影響網路的運作。如果本地機器依賴任意網路伺服器進行驗證，而這些伺服器由於名稱解析問題而無法找到，則使用者甚至還不能登入。網路中由故障名稱伺服管理的機器將無法「看到」彼此，也不能相互通訊。

   NTP（時間服務）
   : NTP 服務的損壞或完全故障會影響 Kerberos 驗證以及 X 伺服器的功能。

   NFS（檔案服務）
   : 若應用程式所需的資料儲存於掛接 NFS 的目錄中，萬一此服務關閉或設定錯誤，該應用程式將無法啓動或無法正常運作。最糟糕的情況是，如果因 NFS 伺服器出現故障，而找不到包含 `.gconf` 子目錄的使用者主目錄，則使用者的個人桌面組態將無法起作用。

   Samba（檔案服務）
   : 若應用程式所需的資料儲存於出現故障之 Samba 伺服器的目錄中，該應用程式將無法啓動或無法正常運作。

   NIS（使用者管理）
   : 若您的 SUSE Linux Enterprise Server 系統依賴出現故障之 NIS 伺服器提供使用者資料，使用者將無法登入這部機器。

   LDAP（使用者管理）
   : 若您的 SUSE Linux Enterprise Server 系統依賴出現故障之 LDAP 伺服器提供使用者資料，使用者將無法登入這部機器。

Kerberos（驗證）
: 不進行驗證，也無法登入任何機器。

CUPS（網路列印）
: 使用者無法列印。

4. 請檢查網路伺服器是否運作，且您的網路設定可否讓您建立連接：

重要：限制

下述偵錯程序只適用於不涉及任何內部路由的簡易網路伺服器/用戶端設定。假設伺服器和用戶端都是相同子網路的成員，不需要其他路由。

a. 使用 `ping IP_ADDRESS/HOSTNAME`（以伺服器的主機名稱或 IP 位址取代該項）來檢查各伺服器是否在正常運作，且能夠回應網路。若此指令成功的話，就會告知您的主機您正在尋找並執行它，且您網路的名稱服務設定是正確的。

   若 ping 的結果失敗且傳回 `destination host unreachable`（無法聯繫目的地主機），則您的系統或想找的伺服器可能設定錯誤或故障。從另一部機器執行 `ping IP address` 或 `YOUR_HOSTNAME` 指令，以檢查是否可連接您的系統。如果您可以從另一台機器存取您的機器，可能是伺服器未執行或設定錯誤。

   若 ping 失敗且傳回 `unknown host`，則是名稱服務設定錯誤，或使用的主機名稱不正確。如須對此問題做進一步檢查，請參閱步驟 4.b。若 ping 仍然失敗，則是您的網路卡未設定正確，或網路硬體故障。

b. 請使用 `host HOSTNAME` 來檢查您嘗試連接的伺服器的主機名稱是否正確地轉譯為 IP 位址，反之亦然。若此指令傳回主機的 IP 位址，則名稱服務是啓動且執行中的。如果 `host` 指令失敗，請在您的主機上檢查所有與名稱及位址解析有關的網路組態檔案：

   `/etc/resolv.conf`
   : 此檔案用於追蹤您目前使用的名稱伺服器與領域。您可以手動修改此檔案，或以 YaST 或 DHCP 自動調整。建議您採用自動調整。然而，請確定此檔案的結構如下，且所有的網路位址與領域名稱均正確：

```
search FULLY_QUALIFIED_DOMAIN_NAME
nameserver IPADDRESS_OF_NAMESERVER
```

此檔案會包含多個名稱伺服器位址，其中至少有一個必須是正確的，才能為您的主機提供名稱解析。需要時，請使用 YaST 的「網路設定」模組（「主機名稱/DNS」索引標籤）調整此檔案。

若您是透過 DHCP 處理網路連接的，請在 YaST 的「網路設定」模組（「主機名稱/DNS」索引標籤）中選取透過 DHCP 設定主機名稱（可針對所有介面全域設定，也可以逐個介面設定）和透過 DHCP 更新名稱伺服器和搜尋清單，以允許 DHCP 變更主機名稱和名稱服務資訊。

`/etc/nsswitch.conf`

此檔案會告知 Linux 何處可找到名稱服務資訊。其內容如下：

```
 ...
hosts: files dns
networks: files dns
...
```

`dns` 項目是必備的。這會告訴 Linux 使用外部名稱伺服器。正常情況下，YaST 會自動管理這些項目，但檢查很謹慎。

若主機上所有相關的項目都正確的話，請要求您的系統管理員檢查 DNS 伺服器組態是否具備正確的時區資訊。如需關於 DNS 的詳細資訊，請參閱第 25 章 「網域名稱系統」。若您確定主機和 DNS 伺服器的 DNS 組態正確物誤，請繼續檢查網路組態和網路裝置。

c. 若您的系統無法建立與網路伺服器的連接，且您已經從問題可能原因清單中排除名稱服務的問題，則請檢查網路卡的組態。

使用 `ip addr show NETWORK_DEVICE` 指令來檢查是否已正確設定此裝置。確定已正確設定帶有網路遮罩（`/MASK`）的 `inet address`。IP 位址有錯誤或網路遮罩有位元遺失的話，都可能造成網路組態無法使用。必要的話，請一併於伺服器上執行此檢查。

d. 如果已正確設定且正在執行名稱服務和網路硬體，但有些外部網路連接仍然長時間逾時或完全失敗，請使用 `traceroute FULLY_QUALIFIED_DOMAIN_NAME` 指令（以 `root` 使用者的身分執行）來追蹤這些要求所採用的網路路由。此

指令會列出請求從您機器傳送到其目的地所經的所有閘道（躍程）。其會列出各躍程的回應時間，以及是否可連接此躍程。請使用 traceroute 加上 ping 找出問題的原因，並告知管理員。

一旦您辨識出網路問題的原因，就可以自己解決（若問題發生在您機器上的話），或將您的發現告訴網路系統管理員，讓他們可以重新設定服務，或修復必要的系統。

40.5.1 NetworkManager 問題

若您有網路連接的問題，請依程序 40.6 「如何識別網路問題」所述將範圍調窄。若 NetworkManager 似乎有問題，請執行下列步驟，取得 NetworkManager 故障原因的提示記錄：

1. 開啓外圍程序並以 `root` 身份登入。
2. 重新啓動 NetworkManager：

   ```
   systemctl restart Network.Manager
   ```

3. 以一般使用者身分開啓網頁，例如 http://www.opensuse.org，看看是否可以連接。
4. 在 `/var/log/NetworkManager` 中收集 NetworkManager 狀態的所有相關資訊。

如需關於 NetworkManager 的詳細資訊，請參閱第 36 章 「使用 NetworkManager」。

40.6 資料問題

資料問題是指，機器或許可以（或無法）正確開機，但系統上有著明顯的資料損毀，且需要修復。遇到這些情況時，需要用到您重要資料的備份檔案，以讓您的系統回復到故障前的狀態。

40.6.1 管理分割區影像

有時，您需要對整個分割區甚至是硬碟執行備份。Linux 附帶 `dd` 工具，可為磁碟建立完全一致的副本。結合 `gzip`，還能為您節省一些空間。

程序 40.7 備份和還原硬碟

1. 以 `root` 使用者身分啓動外圍程序。
2. 選取來源裝置。一般類似 `/dev/sda`（標示為 `SOURCE`）。
3. 決定要儲存影像（標示為 `BACKUP_PATH`）的位置。此位置必須不同於來源裝置的位置。換言之，如果您對 `/dev/sda` 進行備份，就不能將影像檔案儲存在 `/dev/sda` 下。
4. 執行以下指令建立壓縮影像檔：

   ```
   dd if=/dev/SOURCE | gzip > /BACKUP_PATH/image.gz
   ```

5. 使用以下指令還原硬碟：

   ```
   gzip -dc /BACKUP_PATH/image.gz | dd of=/dev/SOURCE
   ```

如果您只需備份分割區，請使用各個分割區取代指令中預留的 `SOURCE`。如此，您的影像檔便會存放於同一個硬碟的不同分割區中。

40.6.2 使用救援系統

系統無法啓動並正常運作的原因可能有幾種。最常見的原因是系統當機後檔案系統損毀、組態檔案損毀，或開機載入程式組態損毀。

為了幫助您解決這些狀況，SUSE Linux Enterprise Server 包含可以開機的救援系統。救援系統是一個小型的 Linux 系統，可以載入到 RAM 磁碟上並裝載為根目錄檔案系統，好讓您從外部存取 Linux 分割區。藉由此救援系統，您可以復原或修改任何重要的系統項目。

- 操作任何類型的組態檔案。
- 檢查檔案系統有無缺失並啓動自動修復程序。
- 在「變更根目錄」環境中存取已安裝的系統.
- 檢查、修改和重新安裝開機載入程式組態。

- 從錯誤安裝的裝置驅動程式或無法使用的核心復原。
- 使用 Parted 指令來調整分割區大小。如需此工具的詳細資訊，請造訪 GNU Parted 網站 http://www.gnu.org/software/parted/parted.html 。

救援系統可以從各種來源與位置載入。最簡單的方法就是從原始安裝媒體將救援系統開機。

注意：在 IBM z Systems 上啓動救援系統

在 IBM z Systems 上，安裝系統可用於救援目的。若要啓動救援系統，請遵照第 40.7 節 「IBM z Systems：將 initrd 當成救援系統」中的指示。

1. 將安裝媒體插入 DVD 光碟機。
2. 重新啓動系統。
3. 在開機畫面中按 F4，然後選擇DVD-ROM。之後從主功能表中選擇救援系統。
4. 在 `Rescue:` 提示輸入 `root`。無須輸入密碼。

如果硬體設定沒有包含 DVD 光碟機，您可以從網路來源將救援系統開機。以下範例適用於遠端開機案例 — 如果使用其他開機媒體（例如 DVD），請相應修改 `info` 檔案，並按照一般安裝方式開機。

1. 進入 PXE 開機設定的組態中，新增下面的行：`install=PROTOCOL://INSTSOURCE` 和 `rescue=1`。如果需要啓動系統修復，請使用 `repair=1`。如同正常的安裝一樣，`PROTOCOL` 代表任何受支援的網路通訊協定（NFS、HTTP、FTP 等），`INSTSOURCE` 代表網路安裝來源的路徑。
2. 依照《部署指南》，第 9 章「準備目標系統的啓動」，第 9.7 節「區域網路喚醒」 中的說明，使用「網路喚醒」啓動系統。
3. 在 `Rescue:` 提示輸入 `root`。無須輸入密碼。

一旦進入救援系統後，您可以利用 Alt—F1 至 Alt—F6 來使用虛擬主控台。

`/bin` 目錄中提供了一個外圍程序和其他有用的公用程式，如 mount 程式。`/sbin` 目錄中包含重要的檔案與網路公用程式，以便檢視及修復檔案系統。此目錄中也有最重要的二進位系統維護程式，例如 `fdisk`、`mkfs`、`mkswap`、`mount` 和 `shutdown`，以及維護網路的 `ip` 和 `ss`。目錄 `/usr/bin` 包含 vi 編輯器、find、less 和 SSH。

若要檢視系統訊息，請使用指令 `dmesg`，或者使用 `journalctl` 來檢視系統記錄。

40.6.2.1 檢查和操作組態檔案

為了舉例說明使用救援系統如何修正組態檔案，請想像系統由於組態檔案損毀而無法正常開機。您可以使用救援系統來解決這個問題。

若要操作組態檔案，請執行下列步驟：

1. 使用上述的其中一個方法啓動救援系統。
2. 若要將 `/dev/sda6` 下的開機檔案系統裝載到救援系統，請使用下列指令：

   ```
   mount /dev/sda6 /mnt
   ```

   系統的所有目錄現在都存放在 `/mnt` 中
3. 將此目錄變更到裝載的開機檔案系統中：

   ```
   cd /mnt
   ```
4. 在 vi 編輯器中開啓有問題的組態檔案。調整並儲存設定。
5. 從救援系統解除裝載開機檔案系統：

   ```
   umount /mnt
   ```
6. 重新開機。

40.6.2.2 修復和檢查檔案系統

一般而言，檔案系統無法在執行中的系統上修復。如果發生了嚴重的問題，您可能甚至無法裝載開機檔案，而且系統可能會因為「核心異常」而無法開機。在此情況下，唯一的方法就是從外部修復系統。系統包含的公用程式可檢查和修復

`btrfs`、`ext2`、`ext3`、`ext4`、`reiserfs`、`xfs`、`dosfs` 以及 `vfat` 檔案系統。尋找指令 `fsck.` 檔案系統，例如，如果您需要某個檔案系統，請檢查 `btrfs`，使用 `fsck.btrfs`。

40.6.2.3 存取已安裝的系統

如果需要從救援系統存取已安裝的系統，您需要在變更根目錄環境中執行此操作。例如，若要修改開機載入程式組態或執行硬體組態公用程式。

若要根據已安裝的系統設定變更根目錄環境，請執行下列步驟：

1. 

   提示：輸入 LVM 磁碟區群組

   如果您使用的是 LVM 設定（如需更多一般性詳細資料，請參閱《儲存管理指南》），請輸入所有現有的磁碟區群組，以便能夠尋找和掛接裝置：

   ```
   rootvgimport -a
   ```

   執行 `lsblk` 以檢查哪個節點對應於根分割區。在本例中，該節點為 `/dev/sda2`：

   ```
   lsblk
   NAME        MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINT
   sda           8:0    0 149,1G  0 disk
   ├─sda1        8:1    0     2G  0 part  [SWAP]
   ├─sda2        8:2    0    20G  0 part  /
   └─sda3        8:3    0   127G  0 part
     └─cr_home 254:0    0   127G  0 crypt /home
   ```

2. 從安裝的系統掛接根分割區：

   ```
   mount /dev/sda2 /mnt
   ```

3. 掛接 `/proc`、`/dev` 和 `/sys` 分割區：

   ```
   mount -t proc none /mnt/proc
   mount --rbind /dev /mnt/dev
   mount --rbind /sys /mnt/sys
   ```

4. 現在可以「變更根分割區」為新的環境，並保留 `bash` 外圍程序：

   ```
   chroot /mnt /bin/bash
   ```

5. 最後，從已安裝的系統裝載其餘分割區：

```
mount -a
```

6. 現在您可以存取已安裝的系統。重新啓動系統之前，請先使用 `umount -a` 來解除裝載分割區，並以 「exit」 離開變更根目錄環境。

警告：限制

雖然您可以完全存取已安裝系統的檔案和應用程式，但必須遵守某些限制。執行中的核心是使用救援系統啓動的核心，而不是使用變更根目錄環境啓動的核心。它只支援基本硬體，而且無法從已安裝系統新增核心模組，除非核心版本完全一致。一律使用 `uname -r` 檢查目前執行的（救援）核心，然後確定變更根目錄環境的 `/lib/modules` 目錄中是否有相符的子目錄。如果有，您便可以使用已安裝的模組，否則，需要在其他媒體（例如快閃磁碟機）上提供模組的正確版本。多數情況下，救援核心版本與已安裝的版本並不相同，因此，舉例來說，您便不能像平常一樣存取聲卡。您也無法啓動圖形使用者介面。

另外請注意，當您使用 「F1」 至 Alt—F6 來切換主控台時，將會離開 Alt—變更根 目錄環境。

40.6.2.4 修改和重新安裝開機載入程式

有時候系統無法開機是因為開機載入程式組態已損毀。譬如說，開機載入程式若未執行，啓動常式便無法將實體磁碟機轉譯為 Linux 檔案系統中的實際位置。

若要檢查開機載入程式組態和重新安裝開機載入程式，請執行下列步驟：

1. 執行存取已安裝系統所需的必要步驟，如 第 40.6.2.3 節 「存取已安裝的系統」 所述。

2. 檢查系統上是否已安裝 GRUB 2 開機載入程式。如果未安裝，請安裝 `grub2` 套件並執行

```
grub2-install /dev/sda
```

3. 根據第 12 章 「開機載入程式 GRUB 2」中概述的 GRUB 2 組態原則，檢查下列檔案是否正確設定，並在必要時加以修正。

- `/etc/default/grub`
- `/boot/grub2/device.map`（選用檔案，手動建立後才存在）
- `/boot/grub2/grub.cfg`（此檔案是產生的，不要編輯）
- `/etc/sysconfig/bootloader`

4. 依序使用下列指令來重新安裝開機載入程式：

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

5. 解除裝載分割區，從「變更根目錄」環境登出，並重新啓動系統：

```
umount -a
exit
reboot
```

40.6.2.5 修正核心安裝

核心更新可能會帶來新的錯誤，進而會影響系統作業。例如，系統中某個硬體的驅動程式可能有錯誤，致使您無法存取和使用系統。在這種情況下，請復原至上回正常運作的核心（如果系統中提供的話），或者從安裝媒體安裝原始核心。

提示：如何在更新後保留之前的核心

為了防止在錯誤的核心更新後無法開機，請使用核心多版本功能，並告知 `libzypp` 您要在更新後保留哪些核心。

例如，若要永遠保留最後兩個核心和目前執行中的核心，請新增

```
multiversion.kernels = latest,latest-1,running
```

至 `/etc/zypp/zypp.conf` 檔案。如需相關資訊，請參閱《部署指南》，第 15 章「安裝多個核心版本」。

另一個類似的情況是，當您需要重新安裝或更新 SUSE Linux Enterprise Server 不支援之裝置的已損毀驅動程式時。例如，當硬體廠商使用特定裝置時，比如使用硬體 RAID 控制器，這就要求作業系統能夠識別二進位驅動程式。廠商一般會發行「驅動程式更新磁碟」(DUD)，內含所需驅動程式的修復或更新版本。

在這兩種情況下，您都需要以救援模式存取安裝的系統，並修正與核心相關的問題，否則系統可能無法正常開機：

1. 從 SUSE Linux Enterprise Server 安裝媒體開機。
2. 如果您要在錯誤的核心更新之後復原，請跳過此步驟。如果需要使用驅動程式更新磁碟 (DUD)，請在開機功能表出現之後按 F6 以載入驅動程式更新，並選擇驅動程式更新的路徑或 URL，然後按一下是確認。
3. 從開機功能表中選擇救援系統，然後按 Enter。如果您之前選擇使用 DUD，系統將會要求您指定儲存驅動程式更新的位置。
4. 在 `Rescue:` 提示輸入 `root`。無須輸入密碼。
5. 手動將目標系統和「變更根目錄」掛接至新環境：如需詳細資訊，請參閱第 40.6.2.3 節 「存取已安裝的系統」。
6. 如果使用 DUD，請安裝/重新安裝/更新錯誤的裝置驅動程式套件。請務必確定已安裝的核心版本與要安裝的驅動程式版本完全相符。
   如果要修正錯誤的核心更新安裝，可以按照以下程序從安裝媒體安裝原始核心。
   a. 使用 `hwinfo --cdrom` 識別 DVD 裝置，並使用 `mount /dev/sr0 /mnt` 掛接裝置。
   b. 導覽到 DVD 上儲存核心檔案的目錄，例如 `cd /mnt/suse/x86_64/`。
   c. 使用 `rpm -i` 指令安裝您的產品類別所需的 `kernel-*`、`kernel-*-base` 以及 `kernel-*-extra` 套件。
7. 根據需要更新組態檔，然後重新啓動開機載入程式。如需詳細資訊，請參閱第 40.6.2.4 節 「修改和重新安裝開機載入程式」。
8. 從系統磁碟機中取出任何可開機的媒體，然後重新開機。

40.7 IBM z Systems：將 initrd 當成救援系統

如果升級或修改 SUSE® Linux Enterprise Server for IBM z Systems 的核心，可能會意外以不一致的狀態將系統重新開機，使得對所安裝的系統執行 IPL 的標準程序失敗。在這種情況下，您可以使用安裝系統來提供救援。

依《部署指南》，第 4 章「在 IBM z Systems 上安裝」，第 4.2 節「準備安裝」所述，對 SUSE Linux Enterprise Server for IBM z Systems 安裝系統執行 IPL。選擇開始安裝，然後輸入所有需要的參數。載入安裝系統後，系統會詢問您要使用哪個顯示類型來控制安裝，此時請選取 `SSH`。現在，您可以不輸入密碼直接以 `root` 身分透過 SSH 登入系統。

在此狀態下，尚未設定任何磁碟。您必須先設定磁碟，才能繼續。

程序 40.8 設定 DASD

1. 使用下列指令來設定 DASD：

   ```
   dasd_configure 0.0.0150 1 0
   ```

   0.0.0150 是連接 DASD 的通道。`1` 表示啓用磁碟（此處的 `0` 會停用磁碟）。`0` 表示磁碟的「無 DIAG 模式」（此處的 `1` 會啓用磁碟的 DAIG 存取）。

2. 現在 DASD 已經上線（請使用 `cat /proc/partitions` 來檢查），而且可以用於後續指令。

程序 40.9 設定 ZFCP 磁碟

1. 若要設定 zFCP 磁碟，您必須先設定 zFCP 介面卡。使用下列指令來執行此動作：

   ```
   zfcp_host_configure 0.0.4000 1
   ```

   `0.0.4000` 是連接介面卡的通道，而 `1` 表示啓動（此處的 `0` 會停用介面卡）。

2. 啓用介面卡之後，便可以設定磁碟。使用下列指令來執行此動作：

   ```
   zfcp_disk_configure 0.0.4000 1234567887654321 8765432100000000  1
   ```

`0.0.4000` 是之前使用的通道 ID，`1234567887654321` 是 WWPN（全球連接埠號碼），而 `8765432100000000` 則是 LUN（邏輯單元編號）。`1` 表示啓用磁碟（此處的 `0` 會停用磁碟）。

3. 現在 zFCP 磁碟已經上線（請使用 `cat /proc/partitions` 來檢查），而且可以用於後續指令。

現在，救援系統已完全設定好，您可以開始修復安裝的系統。如需關於如何解決最常見問題的指示，請參閱第 40.6.2 節 「使用救援系統」。

A 文件更新

本章列出了本文件的內容變更。

本手冊在以下日期進行了更新：

- 第 A.1 節 「2018 年 9 月 (SUSE Linux Enterprise Server 12 SP3 的文件維護版本)」
- 第 A.2 節 「2018 年 6 月 (SUSE Linux Enterprise Server 12 SP3 的文件維護版本)」
- 第 A.3 節 「2017 年 12 月 (SUSE Linux Enterprise Server 12 SP3 的維護版本)」
- 第 A.4 節 「2017 年 9 月 (SUSE Linux Enterprise Server 12 SP3 的初始版本)」
- 第 A.5 節 「2016 年 11 月 (SUSE Linux Enterprise Server 12 SP2 的初始版本)」
- 第 A.6 節 「2016 年 3 月 (SUSE Linux Enterprise Server 12 SP1 的維護版本)」
- 第 A.7 節 「2015 年 12 月 (SUSE Linux Enterprise Server 12 SP1 的初始版本)」
- 第 A.8 節 「2015 年 2 月 (文件維護更新)」
- 第 A.9 節 「2014 年 10 月 (SUSE Linux Enterprise Server 12 的初始版本)」

A.1 2018 年 9 月 (SUSE Linux Enterprise Server 12 SP3 的文件維護版本)

錯誤修復

- 在第 32 章 「使用 YaST 設定 FTP 伺服器」中移除了 `pure-ftpd` 的參考內容 (https://bugzilla.opensuse.com/show_bug.cgi?id=1101631)。
- 在第 21 章 「使用 udev 進行動態核心裝置管理」中說明了 `udev` 規則檔案的不同位置，並更新了預設規則檔案的名稱 (https://bugzilla.suse.com/show_bug.cgi?id=1103082)。
- 在第 13.6.2 節 「系統記錄」中更正了指令及其 man 頁面參考 (https://bugzilla.suse.com/show_bug.cgi?id=1104266)。

A.2 2018 年 6 月 (SUSE Linux Enterprise Server 12 SP3 的文件維護版本)

第 10 章 「開機程序簡介」

全面重寫了該章，並新增了 IBM z Systems 專屬的資訊 (https://bugzilla.opensuse.com/show_bug.cgi?id=1046514)。

第 14 章 「64 位元系統環境的 32 位元和 64 位元應用程式」

移除了關於在 64 位元系統上編譯 32 位元應用程式的文件。SUSE Linux Enterprise Server 不支援此操作 (https://bugzilla.suse.com/show_bug.cgi?id=1092434)。

第 16.7 節 「設定 Bonding 裝置」

新增了所有結合模式的描述 (文件備註 #35319)

第 12 章 「開機載入程式 GRUB 2」

- 新增了第 8.4.2 節 「使用 vncmanager 啓動的 VNC 工作階段」(Fate#319319)。
- 使用 `videoinfo`、`vbeinfo` 列出 grub 中視訊模式的方法已過時。請參閱 第 12.2.2 節 「檔案 /etc/default/grub」(https://bugzilla.opensuse.com/show_bug.cgi?id=1074026)。

第 22 章 「使用 kGraft 即時修補 Linux 核心」

- kGraft 即時修補支援 POWER，具體請參閱第 22 章 「使用 kGraft 即時修補 Linux 核心」。(https://bugzilla.opensuse.com/show_bug.cgi?id=1074844)

第 12 章 「開機載入程式 GRUB 2」

- 以 `grub2-once` 取代 `grub2-reboot`，以便僅設定下次開機的預設開機項目，具體請參閱第 12.2.2 節 「檔案 /etc/default/grub」。(https://bugzilla.opensuse.com/show_bug.cgi?id=1071210)

A.3 2017 年 12 月 (SUSE Linux Enterprise Server 12 SP3 的維護版本)

一般

- 依據技術回饋，修復了文件中的大量小問題並新增了大量內容。
- 移除了 `faillog` 套件的所有參考內容，因為不再隨附該套件 (https://bugzilla.suse.com/show_bug.cgi?id=710788)。

第 22 章 「使用 kGraft 即時修補 Linux 核心」

在第 22.9 節 「SLE Live Patching 的應用範圍」中提到，現已推出針對 CVSS 7+ 弱點的 kGraft 修補程式。(https://bugzilla.suse.com/show_bug.cgi?id=1068181)

錯誤修復

- 在程序 8.2 「使用 `vncserver` 啓動持續 VNC 工作階段」中新增了用於建立永久 VNC 連接的 `-alwaysshared` 選項（https://bugzilla.suse.com/show_bug.cgi?id=1081409）。
- 在第 12.2.2 節 「檔案 `/etc/default/grub`」中移除了參考 YaST *DEFAULT 選項的備註（https://bugzilla.suse.com/show_bug.cgi?id=1017728）。
- 在第 12.3.1 節 「開機載入程式位置和開機碼選項」中調整了章節並提供了更多詳細資料（https://bugzilla.suse.com/show_bug.cgi?id=1017737）。
- 在第 1.1.1 節 「瞭解 Bash 組態檔案」中新增了 `~/.alias`。(https://bugzilla.suse.com/show_bug.cgi?id=1062209)。
- 在第 40.6.2.3 節 「存取已安裝的系統」中新增了關於輸入磁碟區群組的提示（https://bugzilla.suse.com/show_bug.cgi?id=1051369）。
- 在第 12.3.2 節 「調整磁碟順序」中描述了開機磁碟順序（https://bugzilla.suse.com/show_bug.cgi?id=1017731）。
- 在圖形 12.5 「核心參數」中取代了 VGA 模式（https://bugzilla.suse.com/show_bug.cgi?id=1017753）。
- 更新並簡化了第 12.2.3 節 「`/etc/grub.d` 中的程序檔」(https://bugzilla.suse.com/show_bug.cgi?id=1017726)。

A.4 2017 年 9 月 (SUSE Linux Enterprise Server 12 SP3 的初始版本)

一般

- 依據技術回饋，修復了文件中的大量小問題並新增了大量內容。
- 移除了 `faillog` 套件的所有參考內容，因為不再隨附該套件（https://bugzilla.suse.com/show_bug.cgi?id=710788）。

第 4 章 「YaST」

- 新增了關於 YaST GUI 的新章，並提到了進階按鍵組合（https://bugzilla.suse.com/show_bug.cgi?id=1010039）。

第 5 章 「文字模式的 YaST」

- 新增了第 5.2 節 「進階按鍵組合」(https://bugzilla.suse.com/show_bug.cgi?id=1010039)。

第 6 章 「使用指令行工具管理軟體」

- 新增了第 6.1.5.2 節 「重新整理儲存庫」(Fate#319486)。
- 更新了第 6.1.3.1 節 「安裝全部所需的修補程式」(Fate #320653)。
- 新增了第 6.1.6.3 節 「`zypper info` 用法」(Fate#321104)。

第 7 章 「使用 Snapper 進行系統復原和快照管理」

- 提到了系統會自動刪除 snapper 復原快照。請參閱第 7.3 節 「透過從快照開機來執行系統復原」和第 7.3.1 節 「復原後的快照」(Fate #321773)。
- 在第 7.4 節 「建立和修改 Snapper 組態」中新增了關於如何計算啓用快照所需的最小根檔案系統大小的詳細資訊（https://bugzilla.suse.com/show_bug.cgi?id=1036175）。
- 提到了 Btrfs 預設子磁碟區及其限制（https://bugzilla.suse.com/show_bug.cgi?id=1045884）。

第 8 章 「透過 VNC 進行遠端存取」

- 修正了關於加密通訊的資訊，並新增了第 8.5 節 「加密 VNC 通訊」來介紹如何設定加密（https://bugzilla.suse.com/show_bug.cgi?id=1029117）。

第 9 章 「使用 RSync 複製檔案」

- 全面修訂了以前的「檔案同步」一章，並著重介紹 Rsync。

第 13 章 「systemd 精靈」

- 在第 13.2.2.1 節 「在指令行上啓用/停用服務」的比較表中新增了 System V init 指令 `chkconfig`（文件備註 #30251）。

第 16 章 「基本網路功能」

- 修正了第 16.8 節 「設定用於網路組合的組合裝置」中的多個文件備註。

第 21 章 「使用 udev 進行動態核心裝置管理」

- 修正了 `udevadm` 指令。

第 22 章 「使用 kGraft 即時修補 Linux 核心」

更新了第 22.4 節 「修補程式生命週期」(Fate #322212)。

第 23 章 「特殊系統功能」

- 從第 23.1.4 節 「記錄檔：套件 logrotate」中移除了重複的內容。

第 II 部分 「Linux 系統開機」

- 重新編排了所含的章節，以便它們遵循開機程序的順序。

第 33 章 「代理伺服器 Squid」

- 介紹了 YaST Squid 模組。

錯誤修復

- 以 `apache2ctl` 取代了用於啓動 Apache2 的指令 `httpd2`（https://bugzilla.suse.com/show_bug.cgi?id=1042437）。
- 在第 16.2.5 節 「更多資訊」中，更正了所參考 RFC 文件中的一處拼字錯誤（https://bugzilla.suse.com/show_bug.cgi?id=1045881）。
- 在第 40.6 節 「資料問題」中，移除了某個不再隨附的 YaST 模組的參考內容（https://bugzilla.suse.com/show_bug.cgi?id=1052675）。

A.5 2016 年 11 月（SUSE Linux Enterprise Server 12 SP2 的初始版本）

一般

- 文件回饋電子郵件地址已變更為 `doc-team@suse.com`。
- 增强了 Docker Open Source Engine 的文件並將其重新命名為《Docker Guide》（Docker 指南）。

第 3 章 「YaST 線上更新」

- 如第 3.3 節 「自動線上更新」中所述，自動線上更新之後不會自動重新啓動系統（文件備註 #30116）。

第 6 章 「使用指令行工具管理軟體」

- `zypper patch` 預設不再安裝可選的修補程式。若要安裝可選的修補程式，可使用 `--with-optional` 參數（FATE#320447）。

第 7 章 「使用 Snapper 進行系統復原和快照管理」

- 在第 7.1.2 節 「從快照中排除的目錄」中新增了 `/var/cache` 和 `/var/lib/libvirt/images`（Fate #320834）。
- 新增了第 7.6 節 「自動快照清理」，其中還包括關於 Snapper 新定額支援的文件（Fate #312751）。
- 新增了問：（Fate#318799）。

第 10 章 「開機程序簡介」

- 建議使用者修復檔案系統，以避免根檔案系統於開機期間發生錯誤（FATE#320443）。

第 12 章 「開機載入程式 GRUB 2」

- 在第 12.2 節 「組態檔案結構」中新增了關於 `/boot/grub2/custom.cfg` 對 `grub-once` 的支援的提示（Fate #319632）。
- 新增了第 7.3.2 節 「存取和識別快照開機項目」（Fate #317972 和 #318101）。
- 在第 12.3.1 節 「開機載入程式位置和開機碼選項」中新增了受信任開機支援的相關資訊（Fate #316553）。

第 16 章 「基本網路功能」

- 新增了關於網路組合的一節（FATE#320468），請參閱第 16.8 節 「設定用於網路組合的組合裝置」。
- 提到了用於 Wicked 的 `TUNNEL_DEVICE`（FATE#317977，第 16.5.1.5 節 「通道與 Wicked 配合使用」）。

第 24 章 「使用 NTP 進行時間同步化」

- 新增了同步但不啓動精靈啓動選項的相關資訊。Chroot jail 不再是預設設定 (FATE #320392)。

注意：NFSv2

- 新增了關於啓用 NFSv2 的備註 (https://bugzilla.suse.com/show_bug.cgi?id=919708)。

第 33 章 「代理伺服器 Squid」

- 更新了針對 Squid 3.5 的一章 (FATE#319674)。

第 39.5 節 「在安裝期間收集資訊」

- 新增了關於安裝期間建立的記錄檔案的一節 (FATE#320015)。

錯誤修復

- 使用 Kerberos 之 NFS 的錯誤服務名稱 (https://bugzilla.suse.com/show_bug.cgi?id=983230)。
- 線上修補程式是依據 SUSE CVSS 分數發行的 (https://bugzilla.suse.com/show_bug.cgi?id=992101)。

A.6 2016 年 3 月 (SUSE Linux Enterprise Server 12 SP1 的維護版本)

第 10 章 「開機程序簡介」

新增了關於從交換到 LVM 的 initramfs 移轉的備註 (https://bugzilla.suse.com/show_bug.cgi?id=)。

A.7 2015 年 12 月 (SUSE Linux Enterprise Server 12 SP1 的初始版本)

一般

- 《Subscription Management Tool for SLES 12 SP5》SUSE Linux Enterprise Server 的文件中現已包含。
- SUSE 提供的附加產品已重新命名為模組與延伸。手冊已更新，以反映這項變更。
- 依據技術回饋，修復了文件中的大量小問題並新增了大量內容。
- 註冊服務已從 Novell Customer Center 變更為 SUSE Customer Center。
- 在 YaST 中，現在您可以透過系統群組到達網路設定。網路裝置為 gone (https://bugzilla.suse.com/show_bug.cgi?id=867809)。

第 7 章 「使用 Snapper 進行系統復原和快照管理」

- 在第 7.5.4 節 「刪除快照」中新增了關於 `snapper delete` 的新 `--sync` 參數的資訊 (Fate#317066)。
- 新增了第 7.3.2 節 「存取和識別快照開機項目」 (Fate#317972 和 Fate#318101)。
- 在第 7.3 節 「透過從快照開機來執行系統復原」中新增了關於如何復原到初始安裝狀態或復原到系統更新之前狀態的提示 (Fate#317973 和 Fate#317900)。
- 新增了第 7.1.3.3 節 「建立和掛接新子磁碟區」 (Fate#318805, https://bugzilla.suse.com/show_bug.cgi?id=910602)。

第 8 章 「透過 VNC 進行遠端存取」

- 將一則說明延伸為一節內容，新增了關於預設使用安全通訊協定之 VNC 的資訊（Fate#318936），並移除了 `tightvnc`，因為它已完全由 `tigervnc` 取代。所有這些資訊都在第 8.3.1 節 「可用的組態」中介紹。

第 6 章 「使用指令行工具管理軟體」

- 新增了第 6.1.4 節 「識別使用已刪除檔案的程序和服務」（Fate#318827）。
- 在第 6.1.3.1 節 「安裝全部所需的修補程式」中新增了 `zypper list-patches --cve` 的更多範例（Fate#319053）。
- 在第 6.1.2 節 「使用 Zypper 安裝和移除軟體」中新增了第 6.1.2.6 節 「從已停用的儲存庫安裝套件」，以及關於移除所有 `debuginfo` 套件的提示（Fate#316287）。
- 新增了一句說明，告知在套用特定修補程式後需要將系統重新開機。（Fate#317872）。

第 15 章 「`journalctl`：查詢 `systemd` 日誌」

- 新增了第 15.6 節 「使用 YaST 過濾 `systemd` 記錄」一節（Fate#318486）。

第 12 章 「開機載入程式 GRUB 2」

- 更新/簡化了整章，以便與最新的 GRUB 版本（指令行與 YaST 版本）相符。

第 11 章 「UEFI（整合可延伸韌體介面）」

- 新增了第 11.1.4 節 「使用非內建的驅動程式」（Fate#317593）。

第 16 章 「基本網路功能」

- 第 16.5.1.3 節 「Nanny」中指出，現在預設會啓用 Nanny（Fate#318977）。
- 新增了第 16.6 節 「基本路由器設定」（Fate#317121，https://bugzilla.suse.com/show_bug.cgi?id=870132↗）。
- 新增了第 16.9 節 「使用 Open vSwitch 的軟體定義網路」（Fate#318497）。

第 25 章 「網域名稱系統」

- 新增了第 25.3.2.9.1 節 「新增反向區域」(文件備註 #1356)。

第 27 章 「使用 NFS 共享檔案系統」

- 第 27.3.2 節 「手動輸出檔案系統」中新增了提示，指出 NFSv4 掛接不再需要 `--bind` 掛接 (Fate#316311)。

可用的資料同步化軟體

- 提到了使用雲端運算進行檔案同步。

第 31 章 「Apache HTTP 伺服器」

- 第 31.6.1.3 節 「取得官方簽發證書」中已將 `CA.sh` 取代為明確的 `openssl` 指令 (文件備註 #28367)。
- 新增了第 31.7 節 「在同一部伺服器上執行多個 Apache 例項」(Fate#317786)。
- 已更新章節內容，以便與最新的 Apache 版本 2.4 相符 (Fate#319012)。

第 40 章 「一般問題和解決方案」

- 在第 40.6.2.4 節 「修改和重新安裝開機載入程式」中改善了 GRUB 2 的重新安裝程序。

第 III 部分 「系統」

- 新增了第 22 章 「使用 kGraft 即時修補 Linux 核心」 (Fate#313296 和 Fate#313438)。

錯誤修復

- 移除了過時的 `acpid.service` (https://bugzilla.suse.com/show_bug.cgi?id=918655)。
- 修正了第 22 章 「使用 kGraft 即時修補 Linux 核心」中錯誤的標題 (https://bugzilla.suse.com/show_bug.cgi?id=954250)。

- 修正了第 31 章 「Apache HTTP 伺服器」中錯誤的路徑名稱（https://bugzilla.suse.com/show_bug.cgi?id=949395）。
- 在第 11.1.1 節 「在 SUSE Linux Enterprise Server 上實作」中新增了關於預設啓用安全開機的段落（https://bugzilla.suse.com/show_bug.cgi?id=879486）。
- 在第 8.4 節 「永久 VNC 工作階段」中移除了有關 VNC 僅可檢視之密碼的文件，因為這種密碼在 SUSE Linux Enterprise Server 中不可用（https://bugzilla.suse.com/show_bug.cgi?id=941307）。
- 在第 40.6.2.3 節 「存取已安裝的系統」中修正了關於在救援模式下存取已安裝系統的程序（https://bugzilla.suse.com/show_bug.cgi?id=918217）。
- 在第 10.2.2.1 節 「initramfs 檔案」中新增了關於變更預設 `sysctl` 組態後更新 initramfs 檔案的提示（https://bugzilla.suse.com/show_bug.cgi?id=927506）。
- 在第 27.4.1 節 「以 YaST 輸入檔案系統」和第 16.4.1.2.5 節 「啓動網路裝置」中新增了關於防止 Wicked 停用 NFS 根目錄中網路裝置的提示（https://bugzilla.suse.com/show_bug.cgi?id=938152）。
- 在第 39.6 節 「核心模組支援」中修正了關於 `kernel-FLAVOR-extra` 的誤導性陳述（http://bugzilla.suse.com/show_bug.cgi?id=922976）。
- Btrfs/Snapper：將不會刪除包含新子磁碟區的快照（https://bugzilla.suse.com/show_bug.cgi?id=910602）。
- 關於 /var/lib 上的個別子磁碟區和可支援性的 Btrfs 文件（https://bugzilla.suse.com/show_bug.cgi?id=930424）。

A.8 2015 年 2 月（文件維護更新）

第 13 章 「systemd 精靈」

修正了某個指令的拼字錯誤（https://bugzilla.suse.com/show_bug.cgi?id=900219）。

A.9 2014 年 10 月 (SUSE Linux Enterprise Server 12 的初始版本)

一般

- 由於不再提供 KDE，移除了所有 KDE 文件和參考內容。
- 由於不再支援 SuSEconfig，移除了所有相關參考內容 (Fate#100011)。
- 從 System V init 移至 systemd (Fate#310421)。更新了文件受影響的部分。
- YaST 執行層級編輯器已變更為服務管理員 (Fate#312568)。更新了文件受影響的部分。
- 由於 ISDN 支援已被移除，移除了 ISDN 的所有參考內容 (Fate#314594)。
- 由於不再提供 YaST DSL 模組，移除了所有相關參考內容 (Fate#316264)。
- 由於不再提供 YaST 數據機模組，移除了所有相關參考內容 (Fate#316264)。
- Btrfs 已變為根分割區的預設檔案系統 (Fate#315901)。更新了文件受影響的部分。
- `dmesg` 現在提供與 `ctime()` 格式類似的可讀時間戳記 (Fate#316056)。更新了文件受影響的部分。
- syslog 和 syslog-ng 已被 rsyslog 取代 (Fate#316175)。更新了文件受影響的部分。
- MariaDB 現在做為關聯式資料庫而非 MySQL 提供 (Fate#313595)。更新了文件受影響的部分。
- SUSE 相關產品不再在 http://download.novell.com 上提供，而是在 http://download.suse.com 上提供。連結已相應地調整。
- Novell Customer Center 已取代為 SUSE Customer Center。更新了文件受影響的部分。

- `/var/run` 掛接為 tmpfs（Fate#303793）。更新了文件受影響的部分。
- 不再支援以下架構：IA64 與 x86。更新了文件受影響的部分。
- 使用 `ifconfig` 設定網路的傳統方法已由 `wicked` 取代。更新了文件受影響的部分。
- 許多網路指令已過時，現已由更新的指令取代（通常使用 `ip`）。更新了文件受影響的部分。

  `arp`：`ip neighbor`
  `ifconfig`：`ip addr`、`ip link`
  `iptunnel`：`ip tunnel`
  `iwconfig`：`iw`
  `nameif`：`ip link`、`ifrename`
  `netstat`：`ss`、`ip route`、`ip -s link`、`ip maddr`
  `route`：`ip route`
- 依據技術回饋，修復了文件中的大量小問題並新增了大量內容。

第 3 章 「YaST 線上更新」

- YaST 提供了一個選項用於啓用或停用 delta RPM（Fate#314867）。
- 在安裝需要重新開機的修補程式之前，YaST 將會通知您，並且您可以選擇如何繼續操作。

第 39 章 「收集系統資訊以供支援所用」

- 新增了一節：第 39.1 節 「顯示目前系統資訊」（Fate#315869）。
- 新增了關於 Supportconfig 分析（SCA）工具和裝置的一節：第 39.4 節 「分析系統資訊」（Fate#315699）。
- 新增了一節：第 39.6 節 「核心模組支援」（http://bugzilla.suse.com/show_bug.cgi?id=869159）。
- 更新並重新組織了該章。

第 5 章 「文字模式的 YaST」

- 新增了關於如何在軟體安裝模組中過濾和選取套件的資訊。

第 6 章 「使用指令行工具管理軟體」

- 移除了關於 Zypper rug 相容模式的文件 (Fate#317708).
- 重新編寫了第 6.1.6 節 「使用 Zypper 查詢儲存庫和套件」。

第 7 章 「使用 Snapper 進行系統復原和快照管理」

- 更新了該章，並新增了功能 (Fate#312751、Fate#316238、Fate#316233、Fate#316232、Fate#316222、Fate#316203、Fate#316222)。
- 新增了一節：第 7.3 節 「透過從快照開機來執行系統復原」(Fate#316231、Fate#316221、Fate#316541、Fate#316522)。

第 8 章 「透過 VNC 進行遠端存取」

- 預設 VNC 檢視器現為 `tigervnc`。
- 新增了關於在永久 VNC 工作階段中啓動視窗管理員的修正內容。

第 10 章 「開機程序簡介」

- 由於 System V init 已被 systemd 取代，該章內容進行了大幅縮減。systemd 現在放在一個單獨的章中介紹：第 13 章 「systemd 精靈」。

第 13 章 「systemd 精靈」

- 新增了一章關於 systemd 和 YaST 服務管理員的內容 (Fate#316631、Fate#312568)。
- 載入核心模組上的新區段 (http://bugzilla.suse.com/show_bug.cgi?id=892349)。

第 15 章 「journalctl：查詢 systemd 日誌」

新增了一章 (http://bugzilla.suse.com/show_bug.cgi?id=878352)。

第 12 章 「開機載入程式 GRUB 2」

- GRUB Legacy 文件已被新的一章關於 GRUB 2 的內容取代。
- 捨棄了對 LILO 的支援。
- 新增了一節：第 12.4 節 「z Systems 上終端機使用方式的差異」。

第 11 章 「UEFI (整合可延伸韌體介面)」

- 更新了該章，並新增了功能 (Fate#314510、Fate#316365)。
- 新增了關於 SUSE 金鑰證書所在位置的指示 (文件備註 #25080)。

第 17 章 「印表機操作」

已根據新的 CUPS 版本更新了章節，現在提供通用列印資料格式的 PDF (Fate#314630)。

第 18 章 「X Window System」

- 更新了該章，以反映每次啓動期間的動態組態。
- 更新了第 18.1 節 「安裝與設定字型」。

第 16 章 「基本網路功能」

- 現在，NetworkManager 是工作站延伸的組成部份：第 16.4.1.1 節 「設定全域網路選項的組態」 (Fate#316888)。
- 新增了關於網路組態的新 `wicked` 架構的小節：第 16.5 節 「手動設定網路連接」 (Fate#316649)。
- 提到了可新增至 `/etc/resolv.conf` 的其他選項：第 16.5.2 節 「組態檔案」(Fate#316048)。

第 30 章 「SLP」

- 重新編寫了該章，大幅增加了關於 `slptool` 指令的資訊。

第 25 章 「網域名稱系統」

- YaST DNS 模組現在支援設定轉遞者（Fate#309036）。

第 26 章 「DHCP」

- 由於不再提供 dhcpcd，現已將其移除（Fate#316111）。

第 28 章 「Samba」

- 新增了一節：第 28.8 節 「進階主題」。
- 新增了一節：第 28.8.1 節 「Btrfs 上的透明檔案壓縮」。
- 新增了一節：第 28.8.2 節 「快照」。

第 27 章 「使用 NFS 共享檔案系統」

- NFSv4 共用的設定現在基本上與 NFSv3 類似，特別是以前必需的結合掛接設定現已過時（Fate#315589）。
- 不支援在輸出伺服器本地掛接 NFS 磁碟區。

第 29 章 「使用 Autofs 按需掛接」

- 新增了關於 `autofs` 的一章（Fate#316185）。

第 31 章 「Apache HTTP 伺服器」

- 由於套裝作業系統中已移除 mono 和 mod_mono，因此已移除其參考內容。
- 本章內容已更新為與 Apache 2.4 版相關（Fate#316067）。
- 移除了過時的指令 `NameVirtualHost`，並相應地更新了第 31.2.2.1 節 「虛擬主機組態」。
- 使用標準的 `Require` 更新了 `Order`、`Allow` 和 `Deny` 指令。
- 從第 31.6 節 「設定提供 SSL 的安全網頁伺服器」中移除了虛構的「Snake Oil」公司。

第 32 章 「使用 YaST 設定 FTP 伺服器」

- 捨棄了 `pure-ftpd` (Fate#315176、Fate#316308)。

第 37 章 「電源管理」

- 移除了對 `pm-utils` 套件的過時參考。

第 40 章 「一般問題和解決方案」

- 新增了一節：第 40.3.3 節 「無法掛接 Btrfs 根分割區」(Fate#308679、Fate#315126)。
- 移除了關於已過時 YaST 修復模組的小節 (Fate#308679)。

Wi-Fi 組態

- 移除了關於使用 YaST 進行 Wi-Fi 組態的章節，因為可以使用 NetworkManager 完成 Wi-Fi 組態：第 36 章 「使用 NetworkManager」。

平板電腦

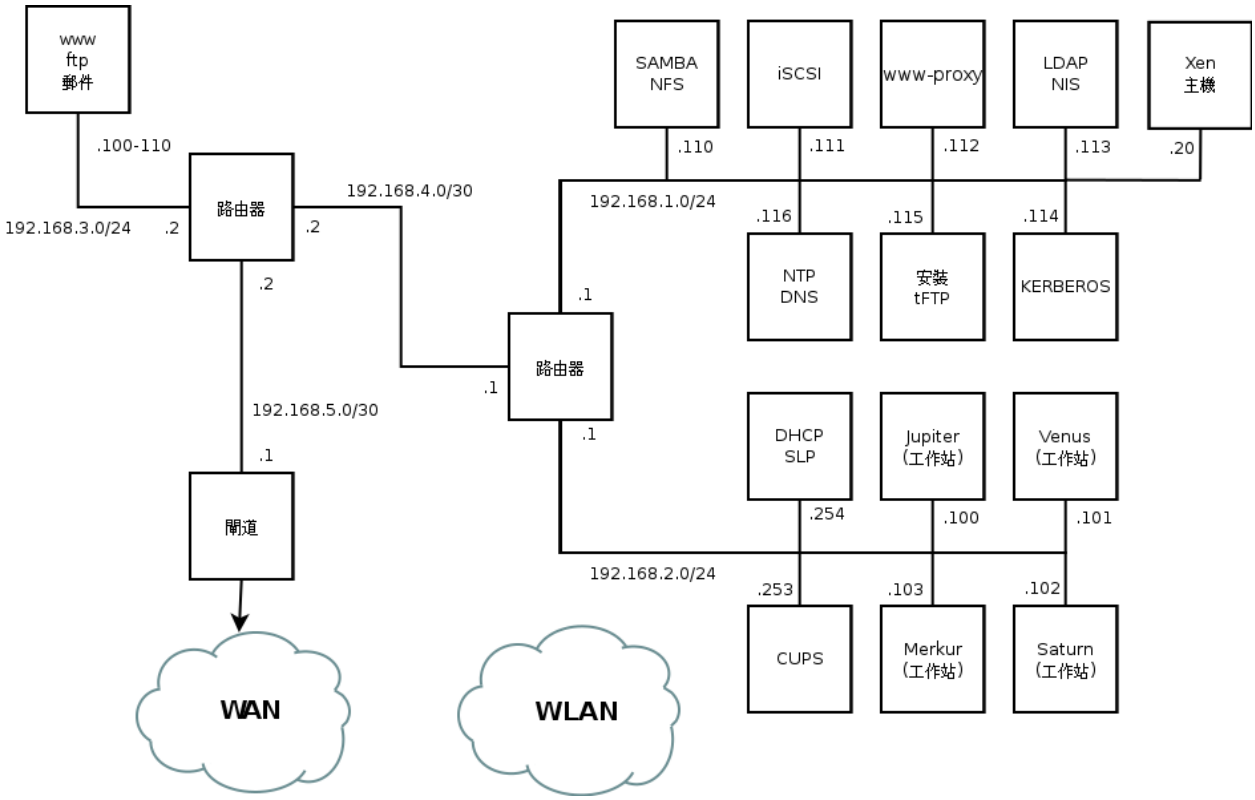
- 移除了關於平板電腦的已過時章節。

錯誤修復

- 新增了一節：第 39.6 節 「核心模組支援」(http://bugzilla.suse.com/show_bug.cgi?id=869159)。
- 新增了一章：第 15 章 「`journalctl`：查詢 `systemd` 日誌」(http://bugzilla.suse.com/show_bug.cgi?id=878352)。

B 網路範例

此網路範例會貫穿 SUSE® Linux Enterprise Server 文件中所有與網路相關的章節。



C GNU 授權

本附錄包含 GNU Free Documentation License（GNU 自由文件授權）1.2 版。

GNU Free Documentation License

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their

titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document

does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this
 document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-
Cover Texts.
A copy of the license is included in the section entitled “GNU
Free Documentation License”.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being
 LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.