

PulseAudio專門負責收集所有來自上層媒體播放程式所送出的聲音，以及接收來自下層經過硬體和驅動程式接收而來的聲音，不僅僅是本機的聲音，也可以一併處理網路所傳送的聲音。但其中最重要的一點，就是所有PulseAudio可以處理的聲音，都是PCM的格式，也就是純數位聲音，所以不管是要進到PulseAudio或是要從PulseAudio出去的聲音，都必須是PCM的聲音。

PulseAudio是一個總管的概念，主宰著整個聲音的流程，用來簡化聲音系統架構的複雜性與不便利性，因此對使用者來說，PulseAudio給了兩個介面，讓使用者可以決定聲音是進去還是出來：Sink和Source（如圖3-15）。

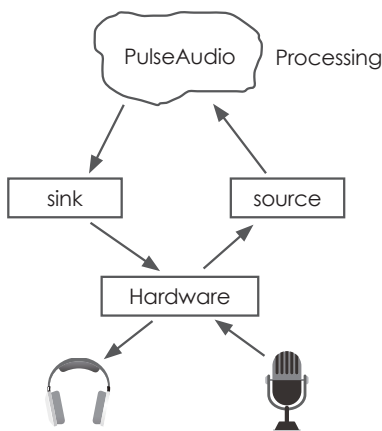


圖3-15：Sink和Source的示意圖

基本上來說，sink和source的意思如下：

- Sink：是輸出的介面，對底層（像是ALSA）不論是傳統介面或是網路上的傳輸，sink代表的就是輸出的介面，也就是從PulseAudio要轉到聲音設備的介面。

- **Source**：相反地，**source**所代表的則是輸入的介面，任何需要進入到PulseAudio的聲音，都會經由**source**介面而導入。

PulseAudio這個服務軟體，主要就是將來自各個不同的音源轉到正確的窗口，除了這個最主要的任務外，PulseAudio還有很多些重要的特點，以下是筆者認為大家比較需要知道的幾個重點：

- 可以針對不同的應用程式做音量的調整。
- 可同時接受多個音源的輸入（**source**）或是輸出（**sink**）。
- 大部分聲音相關的應用軟體都已經支援PulseAudio。
- 可整合多張音效卡。
- 可動態偵測藍牙的聲音介面。
- 有能力自動偵測網路上正在使用PulseAudio的電腦，並透過該電腦的喇叭播放。
- 若應用程式在播放音樂中，PulseAudio可以即時的改變其原本的輸出介面。
- 模組的自動載入。

在比較新的作業系統版本，應該都已經預設以PulseAudio為主要的聲音控制軟體，所以在圖形介面下所看到的音量控制，應該都是透過PulseAudio，圖3-16是Ubuntu 10.10作業系統中預設的聲音管理畫面，就是以PulseAudio為主要的管道，所以畫面中可以同時看到兩塊不同的管理區塊。

上面紅色圈選的部分，就是整個系統的總音量控制，基本上所有系統相關的音量都會以該音量設定為主。

下面所圈選的區塊，則是顯示個別正在執行中軟體的音量控制，使用者可以透過這個畫面直接調整個別音量，而不會影響到系統的預設音量，非常方便，對筆者來說已經是非常人性化的設計。

## 03-04

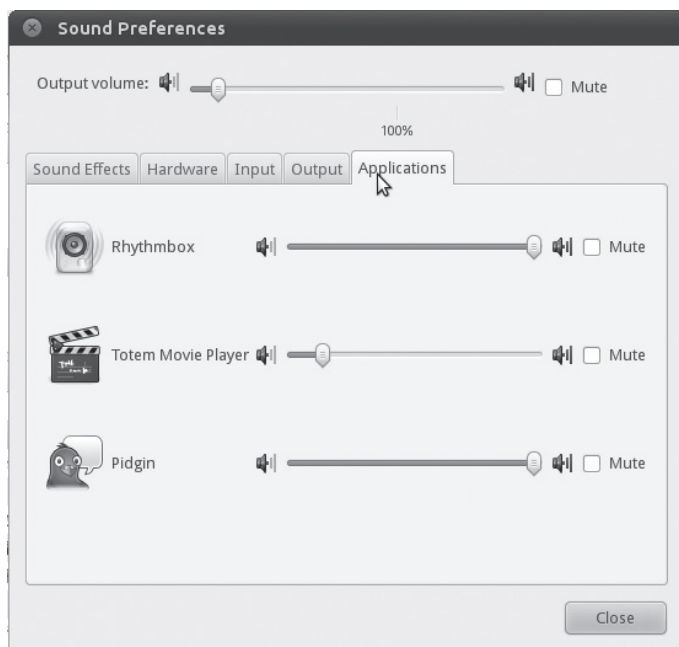


圖3-16：Ubuntu下預設的聲音管理程式

從圖3-16可以很明顯的知道，PulseAudio可以對總聲音的音量做管理，也可以針對不同的應用程式做聲音的控制，這種多角化的管理，也是後來的PulseAudio的功能。此外，無論來自聲音相關的應用程式的音源，或是底層聲音處理系統的音源，只要是聲音相關的轉換，都可以丟給PulseAudio，而PulseAudio在接收到之後，就可以幫忙分配工作，看由哪一個應用程式所接收，或是哪一種聲音介面來處理。

所以，其實PulseAudio就是一個聲音的「伺服器」（如圖3-17所示）。

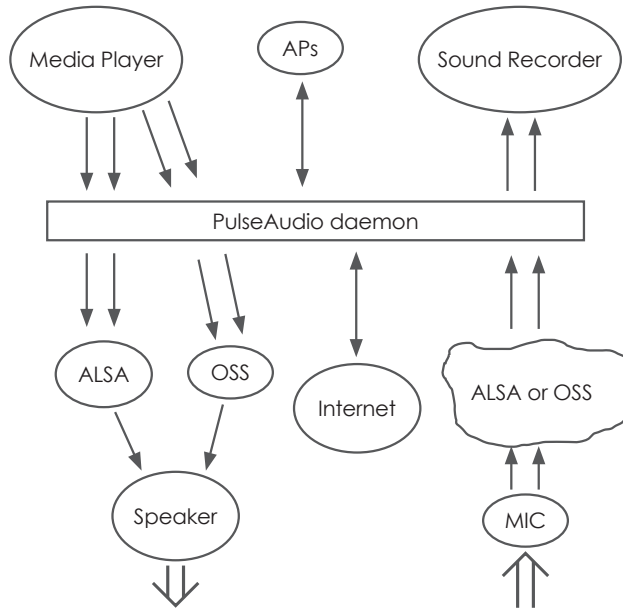


圖3-17：PulseAudio daemon的角色

對上而言，所有聲音相關的應用程式，PulseAudio都提供了單一窗口，讓應用程式的開發者不需要考慮底層的問題，可以專心於應用程式功能上的開發，而不需要針對底層協定或是驅動程式的支援程度，因為PulseAudio就可以幫忙做到。

對下而言，PulseAudio支援許多音源處理方式，像是ALSA比較底層的聲音系統，以及透過網路所傳輸的訊號。

對底層支援的程度上，近幾年的Linux改變蠻多的，因為早在幾年前，Linux的使用者應該很容易感受到，Linux在聲音處理上的效果不是很好（像是用媒體播放程式或是音樂軟體，會嚴重影響CPU的使用率，或是聲

音大小差很多），往往在耗費許多精、力、神之後，就會發現常用的解決方案不外乎幾種：

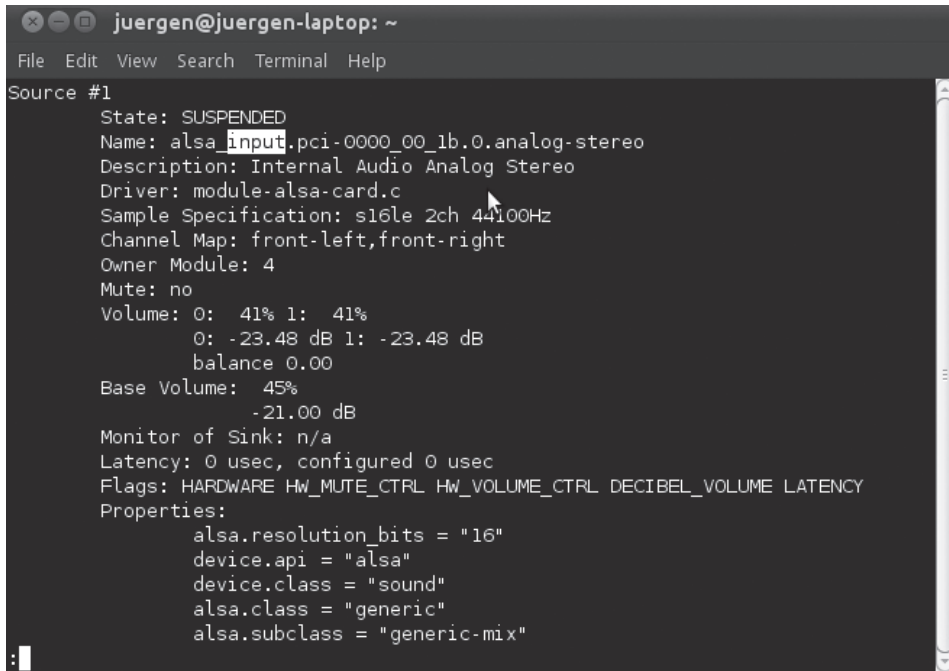
- 將原本的ALSA換成OSS，或是其他支援的底層聲音架構。
- 更新PulseAudio，比較底層的一些模組或是函式庫更新。
- 從聲音管理介面中，將PulseAudio換掉，換回直接指定到ALSA或是不經由PulseAudio管理。

這些方式都是一些常見的處理程序，大部分的小問題都可以解決。在這個階段應該是因為PulseAudio的成熟度不夠，又需要相容這些直接和硬體相關的軟體，所以才會造成在一開始的效果不彰。不過如果是現在各位所使用的Linux，應該不太會有這一類的問題了，因為PulseAudio的支援度已經成熟許多。

目前因為Linux作業系統已經將功能以及圖形化結合的很好，所以反而開放給使用者使用的設定介面愈來愈少（感覺好像慢慢變Windows...），不過PulseAudio還是有提供一些主要的指令供使用者使用，如果讀者想知道目前PulseAudio預設的聲音處理系統是哪一套，可以透過PulseAudio的指令查詢（甚至修改）的。

下面是其中一個「`pactl`」指令的結果，將該系統input的介面列印出來（若讀者想要看所有介面的詳細資料，可以使用「`pactl list`」，`list`參數會將所有PulseAudio所使用的模組、輸出介面、輸入介面等等相關的一併列印出來），在顯示這些資訊時是在沒有任何應用程式被開啟的狀態，所以從圖3-18中可以看到該輸入介面（也就是Source #1，也就是麥克風）的兩個基本資訊：

1. 狀態 (State) : 暫停中 (SUSPENDED) 。
2. 靜音 (Mute) : 沒有被打開 (NO) 。



```
juergen@juergen-laptop: ~  
File Edit View Search Terminal Help  
Source #1  
State: SUSPENDED  
Name: alsa_input.pci-0000_00_1b.0.analog-stereo  
Description: Internal Audio Analog Stereo  
Driver: module-alsa-card.c  
Sample Specification: s16le 2ch 44100Hz  
Channel Map: front-left,front-right  
Owner Module: 4  
Mute: no  
Volume: 0: 41% 1: 41%  
         0: -23.48 dB 1: -23.48 dB  
         balance 0.00  
Base Volume: 45%  
            -21.00 dB  
Monitor of Sink: n/a  
Latency: 0 usec, configured 0 usec  
Flags: HARDWARE HW_MUTE_CTRL HW_VOLUME_CTRL DECIBEL_VOLUME LATENCY  
Properties:  
    alsa.resolution_bits = "16"  
    device.api = "alsa"  
    device.class = "sound"  
    alsa.class = "generic"  
    alsa.subclass = "generic-mix"
```

圖3-18 : pactl所顯示的PulseAudio的input介面資訊

接著筆者將錄音程式打開 (Sound Recorder) 進行錄音，但在同時筆者利用聲音設定程式 (Sound Preferences，也就是一般Linux系統右上角的喇叭按鈕)，將輸入部分的選項設定為「Mute」，接著再看一次「pactl list」的結果，很清楚地看到結果已經改變，呈現以下的內容：

- 狀態 (State) : 執行中 (RUNNING) 。
- 靜音 (Mute) : 打開 (yes) 。

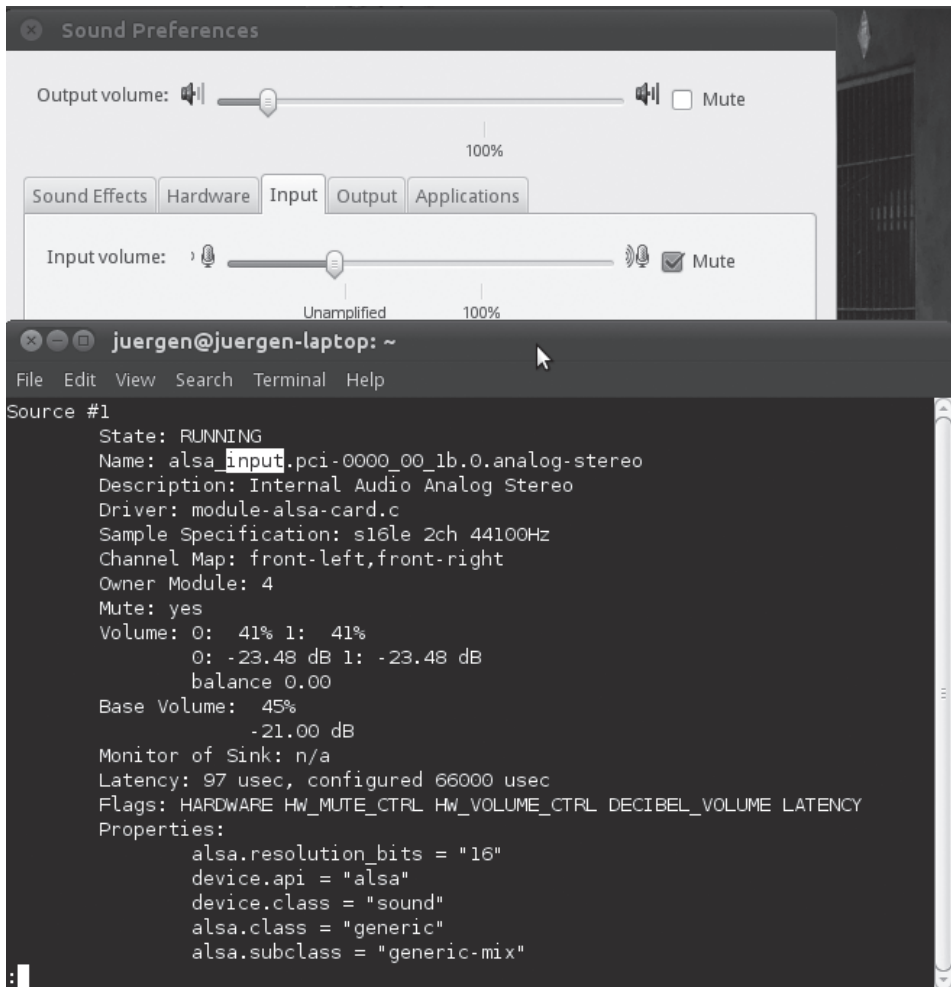


圖3-19：輸入介面被靜音時PulseAudio的改變

因此，既然PulseAudio是一個聲音的伺服器，該有的功能還是不能少，當然需要有一套管理伺服器的軟體或指令，剛剛看到的「pactl」指令其實就是這個指令，透過pactl可以修改伺服器的設定以及配置，但pactl僅限於侷限的範圍，若要完整的功能，就必須要回歸到最基本的指令「pacmd」。透過pacmd指令，才可以完全的控制PulseAudio的服務核心。

但是這一切都僅供參考，畢竟現在的PulseAudio已經蠻成熟的，在大部分的聲音處理上都沒有太大問題，除非是底層的開發人員，不然一般使用者應該是直接套用基本的設定即可，也就是不做任何變動的情況下，應該可以正常使用錄音以及放音的功能。

## 03-05 ALSA

ALSA的全名為「Advanced Linux Sound Architecture」，是一個聲音的組織，主要是希望所有聲音相關的設備廠商可以遵循一樣的標準，讓開發者可以簡化，之前比較有名的是OSS，現在大部分都被ALSA取代了。

ALSA這個組織中有非常多的成員，也都是聲音相關的廠商，如圖3-20所示，這些全部都是從ALSA官方網站上看到的，都是ALSA的供應商，也就是配合ALSA架構而持續在提供支援ALSA聲音架構的軟體，給使用者使用的廠商，所以在這邊看到的名單，應該都可以對Linux有非常好的支援。

在Linux的世界中，要聲音的硬體可以正常播放，最重要的元件大概就是ALSA，因為一般硬體的驅動程式或是一些聲音相關的函式庫，都是由ALSA所提供的，早期甚至連播放聲音都需要透過ASLA才可以做到，對Linux而言，ALSA可以說是舉足輕重的一個聲音架構。



A-trend	AKAI	ALSA
ALI	AMD	ASOUND
ATI	Abit	AdLib
Airis	Alesis	Analog_Devices
Aopen	Apple	Apple_Computer_Inc
Asound	Asus	AudioExcel
AudioScience	AudioTrak	Aureal
AuzenTech	Avance_Logix	AzTech_System_Ltd
BOS	Best_Union	Byears
Broadtree	C.Media	CME
Chic_Technology	Cirrus_Logix	Club3D
Compustar	Core_Sound	Creative_Labs
Cyrix	Diamond_Multimedia	Digigram
Digital_Audio_Labs	Dream	Dynasonic
E-mu	ESI	ESS_Technology
Echo_Corporation	Edrol	Ego_Sys
Emagic	Ensoniq	Evolution
Fortemedia	Freemate	Frontier_Design
Gadget_Labs	Genius	Gibson
Gravis	Griffin	Guillemot
HIS	HT_Omega	Hercules
Hoontech	IC_Ensemble	IDT
Intel	Jaton	Korg
LaCie	Labway	Lexicon
Lyrix_Studio_Technology	MAudio	Mackie
Mad-Dog-Multimedia	Marian	Mark_Of_The_Unicorn
MediaTek	Midtech	MixVibes
NEC	Native_Instruments	Neomagic
NuForce	Nvidia	Oak_Technology
Onkyo	Opcode	Opti
PME	Philips	RME
Razer	Roland_Edrol	S3
SEKD	STB	Seasound
Sega	Sennheiser	Shark_Multimedia
Shuttle	SIS	Sims
Sonidgo	Sonorus	Sony
Steinberg	Stereo-Link	Tascam
TempoTec	Terratec	Tashba
Trident	Trust	Turtle_Beach
VIA	VLSI	VideoLogic
Xitel	Yamaha	Zelfro
Zoltrix	雅王	真人志响

圖3-20：ALSA Vendors

來源：ALSA官方網站（<http://www.alsa-project.org/main/index.php/Matrix:Main>）

從圖3-21的聲音架構圖來看，聲音從硬體到PulseAudio，再到應用程式，這中間不斷地經過ALSA相關的軟體，所以ALSA所涵蓋的範圍是很廣的，尤其值得注意的是，整個底層的支援都是由ALSA所提供，所以如果ALSA對某一張音效卡的支援有問題，除非原廠可以提供正確的驅動程式，否則使用者沒有辦法透過該音效卡播放或錄製聲音，對桌上型電腦的嚴重程度可想而知。

雖然之前提過很多像GStreamer或是PulseAudio帶給使用者的方便，但回歸到最基本面，畢竟硬體要被支援，聲音要可以被正確的處理，這一塊始終還是必需要仰賴ALSA的支援程度，所以其實ALSA也不是孤軍奮戰，很多的音效卡廠商也必須有人直接可用的驅動程式，放在ALSA的套件中。

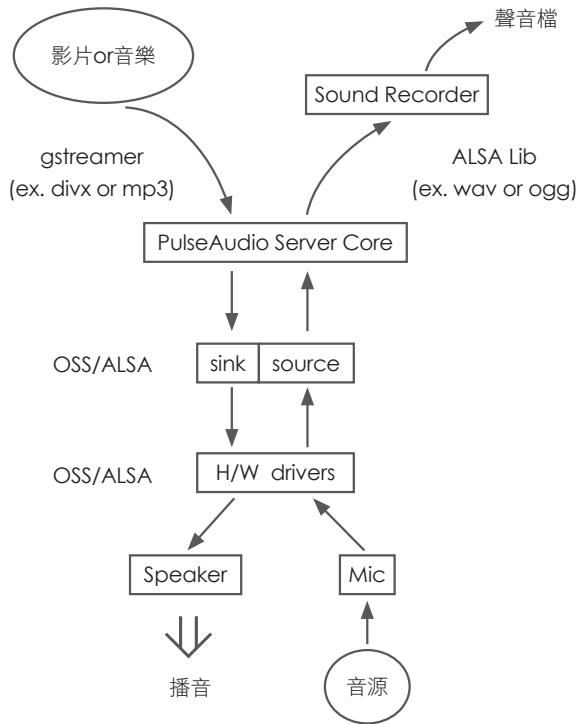


圖3-21：ALSA在系統中所涵蓋的範圍

另外，最容易需要更新的套件就是ALSA，因為當使用者使用比較新的音效設備時，基本上有可能會遇到不支援的問題，這個時候就需要更新驅動的版本，而這部分是由ALSA負責，所以在ALSA的官方網站上也一直持續更新讓使用者下載最新的ALSA套件版本。

ALSA的架構（如圖3-22所示）分得很清楚，因此在官方網站的下載區塊中（如下圖3-23所示），其中所列出的檔案，其實某方面也代表其所支援的層面，因此，若是要更新ALSA也必需要知道哪一個檔案才是符合需要更新的地方。