

## DRBL/Clonezilla進階使用

DRBL：有關DRBL的進階使用說明

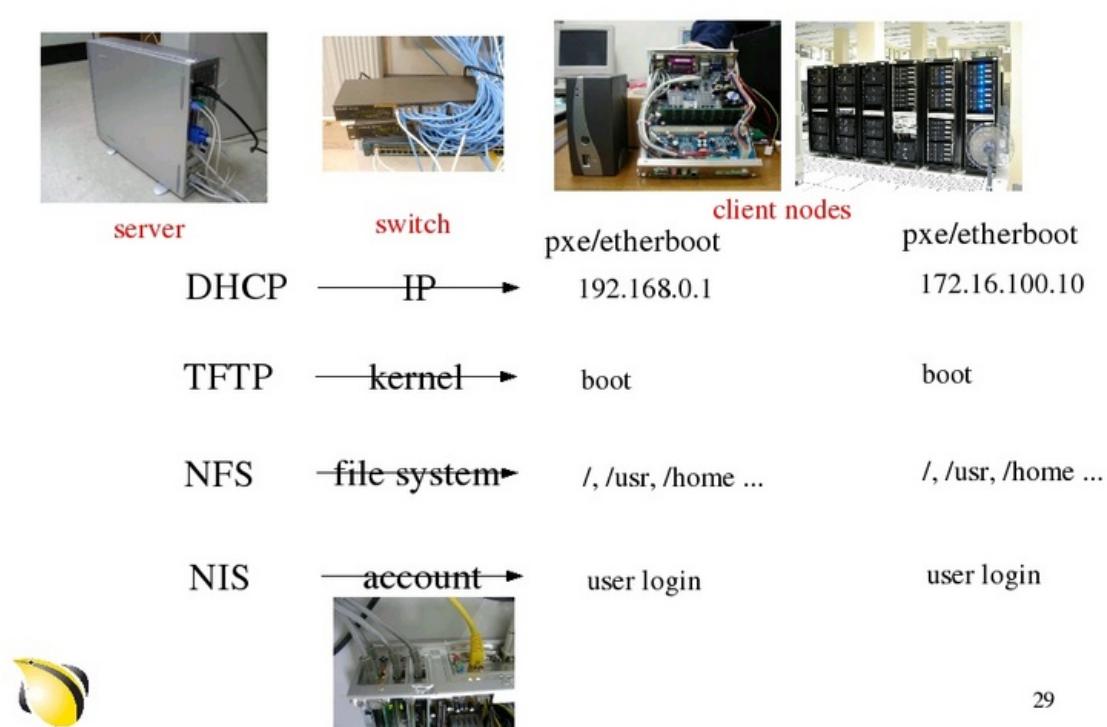
1. [DRBL原理與架構介紹](#)
2. [PXELINUX的設定，包含指定IP address或MAC address作法，修改開機選單，抽換背景圖片](#)
3. [DRBL SSI mode的使用與修改](#)
4. [DRBL client群組管理](#)
5. [將BIOS更新程式放進FreeDOS以便透過PXE開機更新BIOS的作法](#)
6. [在DRBL server上製作Linux套件網路安裝隨身碟](#)
7. [利用virtualbox來做drbl開發與測試環境](#)
8. [如何在DRBL上建置SAN \(AoE, iSCSI\)開機環境？](#)
9. [從無到有製作DRBL live](#)
10. [我的網路環境中已經有一個DHCP伺服器了，所以我不能再使用DRBL伺服器中的DHCP服務，請問有沒有解？](#)

### DRBL原理與架構介紹

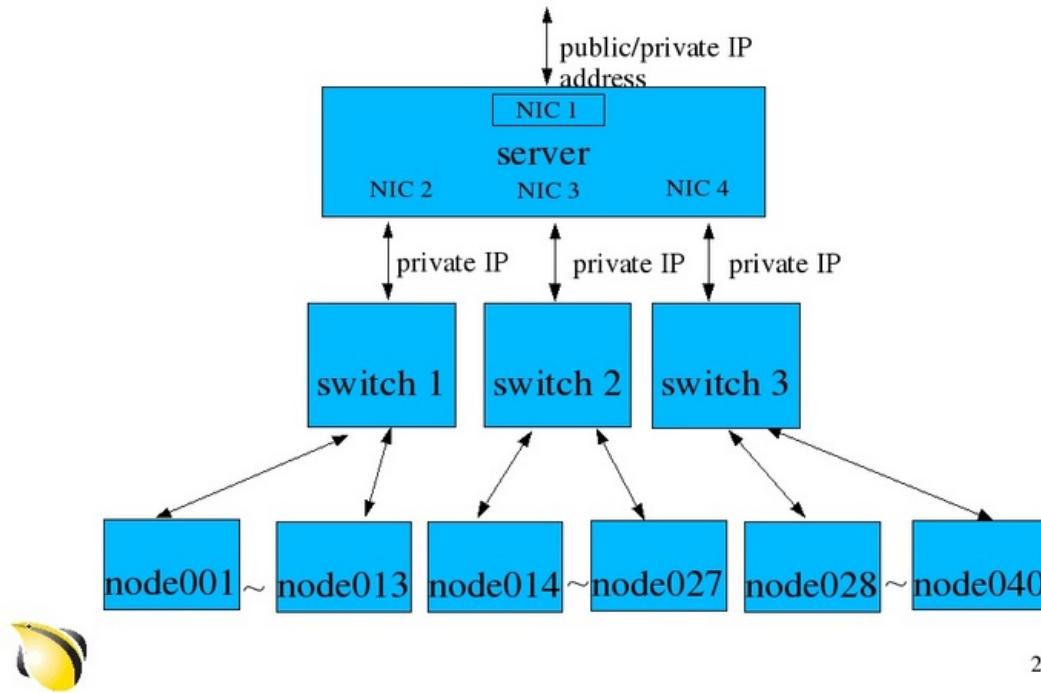


DRBL伺服器提供這些服務: DHCP, TFTP, NFS, NIS, YP/NIS。而用戶端透過PXE網路開機之後，先把網路環境設定好之後，透過NFS掛載根目錄，有了根目錄之後，後續的開機過程就跟一般用硬碟開機過程相同了。DRBL的用戶端基本上都是透過NFS來取得檔案，然後用自己的CPU與記憶體來執行程式。這個架構也就是一般叢集計算環境集中管理所使用的方法。

## DRBL



# DRBL Environment



28

DRBL伺服器上提供給用戶端使用的檔案主要放在目錄/tftpboot/中，其中再分三個目錄：

```
/tftpboot/nbi_img/
/tftpboot/node_root/
/tftpboot/nodes/
```

- 第一個目錄/tftpboot/nbi\_img/主要是放置跟PXELINUX開機有關的檔案，如pxelinux.0的開機檔，linux的核心等等。
- 第二個目錄/tftpboot/node\_root/主要是放置用戶端開機所需要的共通檔案，例如核心模組，等等。
- 第三個目錄/tftpboot/nodes/下會再細分每個用戶端自己的目錄，以IP address為目錄名稱，例如/tftpboot/nodes/192.168.100.1/。裏面會有etc, var等目錄，以便每台用戶端自己可以存系統設定檔以及log檔案等等。

至於在DRBL用戶端的目錄，對應到DRBL伺服器的目錄分佈：

下表標示DRBL環境中，client目錄的位置與server上對應的位置，以及client讀寫權限。由此表可以知道，除了/home與/var/spool/mail外，彼此並沒有覆蓋的問題。而/home與/var/spool/mail是根據使用者帳號而定，如果同一個帳號在不同的用戶端登入，會造成部分檔案重疊。因此，使用同一帳號在不同的用戶端登入時請注意。同時，也不建議用同一帳號在不同用戶端登入X-window環境。

client目錄的位置	server上對應的位置	client讀寫權限
/etc	/tftpboot/nodes/\$IP/etc	RW
/root	/tftpboot/nodes/\$IP/root	RW
/var	/tftpboot/nodes/\$IP/var	RW
/dev	/tftpboot/nodes/\$IP/dev	RW
/	/tftpboot/node_root	R0
/usr	/usr	R0
/opt	/opt	R0
/home	/home	RW
/var/spool/mail	/var/spool/mail	RW
/var/lib/rpm	/var/lib/rpm	R0

註:

1. \$IP表示client的IP address
2. RW表示client可以對該目錄讀寫
3. RO表示client只能對該目錄讀

**PXELINUX**的設定，包含指定**IP address**或**MAC address**作法，修改開機選單，抽換背景圖片



DRBL用戶端是使用PXELINUX來控制的，PXE開機的設定檔放在/tftpboot/nbi\_img/pxelinux.cfg/目錄下，用戶端取得的設定檔會有優先順序，以下我們以PXELINUX 3.7x的版本為例。

以用戶端電腦的UUID是b8945908-d6a6-41a9-611d-74a6ab80b83d(註:這個UUID是來自PXE的stack裡面，有些BIOS並沒有提供正確的UUID，你有可能會得到全部是1的數值)，網路卡MAC address是88:99:AA:BB:CC:DD，然後IP address是192.168.100.1。用戶端PXE開機時候會依序這樣的順序試著抓取設定檔：

```
/tftpboot/nbi_img/pxelinux.cfg/b8945908-d6a6-41a9-611d-74a6ab80b83d
/tftpboot/nbi_img/pxelinux.cfg/01-88-99-aa-bb-cc-dd
/tftpboot/nbi_img/pxelinux.cfg/C0A86401
/tftpboot/nbi_img/pxelinux.cfg/C0A8640
/tftpboot/nbi_img/pxelinux.cfg/C0A864
/tftpboot/nbi_img/pxelinux.cfg/C0A86
/tftpboot/nbi_img/pxelinux.cfg/C0A8
/tftpboot/nbi_img/pxelinux.cfg/C0A
/tftpboot/nbi_img/pxelinux.cfg/C0
/tftpboot/nbi_img/pxelinux.cfg/C
/tftpboot/nbi_img/pxelinux.cfg/default
```

也就是找尋的順序是: (1) UUID (2) MAC address (3) IP address (4) default。

其中Ethernet (ARP type 1)的MAC Address對應的檔名是在最前面有多一個"01-",而且原本的":"取代為"-",採用小寫字母。IP address使用的是16進位的表示方式，字母採用大寫。例如192.168.100.1是用C0A86401來表示。DRBL伺服器上有一個perl的檔案drbl-gehostip(取自PXELINUX)，可以讓你做這樣的計算：

```
$/opt/drbl/bin/drbl-gehostip 192.168.100.1
C0A86401
```

因此如果你有需要針對某台用戶端來做特別的控制，例如設定該台電腦開機選單等，你可以在/tftpboot/nbi\_img/pxelinux.cfg/下放置一個設定檔，不論依據UUID, MAC address 者IP address所對應的檔案都可以達到目的。實做我們可以直接複製DRBL產生的default為設定檔給用戶端使用，例如我們控制用戶端192.168.100.1，以IP address方式控制，我們可以：

```
$cd /tftpboot/nbi_img/pxelinux.cfg/
$cp -f default C0A86401
```

然後修改C0A86401，將選單部份予以調整。

要詳細看DRBL伺服器與用戶端之間在PXE開機時候的訊息，可以修改：

- /etc/dhcp3/dhcpd.conf，將這段設定的註解拿掉成：

```
log(info, concat("Client ",
    binary-to-ascii(16, 8, ":"), substring(hardware, 1, 6)),
    " requests ",
    binary-to-ascii(16, 8, ":"), option dhcp-parameter-request-list),
    " - ",
    pick-first-value(option vendor-class-identifier, "no vendor-id"),
    " - ",
    pick-first-value(option dhcp-client-identifier, "no dhcp-client-id"))
);
```

- /etc/default/tftpd-hpa，將這段設定改成：

```
OPTIONS="-l -s /tftpboot/nbi_img"
```

然後將dhcpd, tftpd重新啟動：

```
/etc/init.d/dhcp3-server restart
```

```
/etc/init.d/tftpd-hpa restart
```

然後在/var/log/daemon.log就可以看到詳細訊息。

PXE LINUX設定檔的語法與SYSLINUX語法一樣，細節可以參考[syslinux提供的資料](#)。在DRBL環境裡面，我們使用menu的語法，因此你會看到類似這樣的語法：

```
default vesamenu.c32
timeout 70
prompt 0
noescape 1
MENU MARGIN 5
MENU BACKGROUND drblwp.png
...
label drbl
  MENU DEFAULT
  # MENU HIDE
  MENU LABEL Debian 4.0 Linux (DRBL mode, mostly local resources)
  # MENU PASSWD
  kernel vmlinuz-pxe
  append initrd=initrd-pxe.img devfs=nomount drblthincl=off selinux=0
  TEXT HELP
  * DRBL version: 1.9.1-26. (C) 2003-2008, NCHC, Taiwan
  * Disclaimer: DRBL comes with ABSOLUTE NO WARRANTY
  ENDTEXT
```

而以上這些語法所指定的檔案都是和pxelinux.0 者gpxelinux.0在同樣目錄，也就是都放在/tftpboot/nbi\_img/下。例如上例中vesamenu.c32, drblwp.png, vmlinuz-pxe以及initrd-pxe.img都是在/tftpboot/nbi\_img/下。

至於要修改用戶端的背景的話，只要在DRBL server將圖檔存成640x480大小，格式使用jpg 者png，然後放到/tftpboot/nbi\_img/下，修改上例MENU BACKGROUND對應的檔名即可。

PS. 詳細資料可以參考PXE LINUX網頁:<http://syslinux.zytor.com/wiki/index.php/PXE LINUX>。

### DRBL SSI mode的使用與修改



從DRBL 1.6.3以後的版本，可以有Full DRBL, Full Clonezilla, DRBL SSI (Single System Image)與Clonezilla Box 四種模式，前兩個Full模式需要多一點的網路流量與硬碟空間，但是用戶端所有的檔案都會回存到伺服器。而DRBL SSI模式，乃是使用單一的主程式，用戶端開機後動態調整需要的設定。這個模式的好處是server設定以及所需的空間減少。但是缺點是每台不能回存各自的系統設定檔(使用者的設定檔與資料仍然可以回存)。

這種模式的原理是在產生DRBL用戶端的時候，由於所有用戶端大部分的檔案都差不多，只有網路設定，主機名稱設定，YP/NIS設定，掛載點等差異。因此我們可以在drblpush的時候，在/tftpboot/node\_root/drbl\_ssi目錄下產生這幾個範本檔案：

template\_etc.tgz, template\_var.tgz template\_opt\_drbl.tgz,

然後在client開機的時候，將這些範本檔解開，放在tmpfs為基礎的目錄(/etc, /var, /opt/drbl/)，產生後根據實際的狀況，/opt/drbl/sbin/drbl-ssi-client-prepare會對這幾個目錄下的檔案做必要的修改。

四種模式的比較：

	Full DRBL	Full Clonezilla	DRBL SSI	Clonezilla Box
用戶端/etc與/var目錄	NFS-based	NFS-based	tmpfs-based	tmpfs-based
用戶端/etc與/var下開機後修改過的檔案於重新開機後	保留	保留	消失	消失
每增加一台用戶端時伺服器需要額外的空間	~ 50 MB	~ 50 MB	0	0
伺服器每張網路卡最多能提供的用戶端數目	253	253	253	253

要切換這四種不同的模式，可以使用/opt/drbl/sbin/dcs -> more -> drbl-mode 來切換企鵝龍機制，或者/opt/drbl/sbin/dcs -> more -> clonezilla-mode 來切換再生龍機制。要注意的是，如果在drblpush的時候沒有選擇Full DRBL 或者Full Clonezilla模式，那麼就無法從DRBL Box 者Clonezilla box切換回Full DRBL 或者Full Clonezilla模式。由於這幾個範本檔產生後，並不會隨時根據/tftpboot/nodes/下的目錄來對應更新，因此如果你有手動修改過/tftpboot/nodes/\$IP/etc/下的檔案，用戶端開機後不見得會對應更新。所以如果你要強迫更新的話，可以使用：/opt/drbl/sbin/dcs -> more -> gen-template-files (自動搜尋第一個用戶端檔案為範本)

或者

drbl-gen-ssi-files

來重新產生DRBL SSI/Clonezilla box範本檔案。如果/tftpboot/nodes/下有多個目錄的話，你也可以手動使用這個指令來強迫使用某個client的IP address目錄來當範本，例如：

/opt/drbl/sbin/drbl-gen-ssi-files -t 192.168.1.1

就是採用/tftpboot/nodes/192.168.1.1這個目錄來產生範本。

這個DRBL SSI 或者Clonezilla box模式的另外一個好處是，用戶端電腦數目有所增加，只要改dhcpd.conf之後，執行 "/opt/drbl/sbin/dcs -> others -> gen\_3N\_conf" 是"/opt/drbl/sbin/drbl-3n-conf generate"即可以馬上增加用戶端電腦的數目。採用此模式，理論上server一張網卡可以支援253台client。另外Clonezilla box 模式乃是DRBL SSI的延伸，以用戶端的RAM降低網路流量，理論上server一張網卡可以支援253台client來clone。<注意>DRBL SSI模式，最好是在你用戶端電腦的RAM夠大，例如512MB，以及所有用戶端的硬體(尤其是顯示卡)都一樣才

建議使用。

### DRBL client群組管理



由於DRBL管裡的是一群機器，因此，如果每次啓動dcs的時候，要勾選一堆機器也是很累人的。所以DRBL裡面提供了一個功能，可以自己定義一個群組。

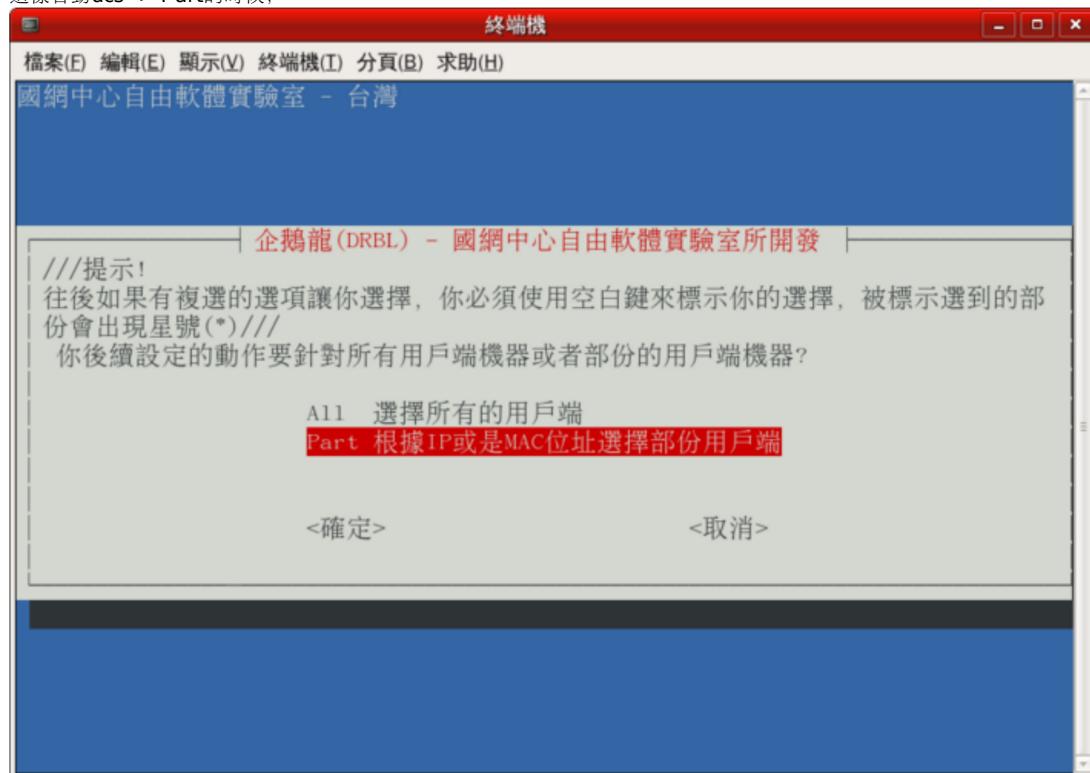
- 如果要自訂IP位址群組，只要把你想要的IP位址，一行一行列好，存成這樣的檔名/etc/drbl/IP-grp-\*, 該檔名\*的部份可以用任何可讀的字元，例如1, DX等。
- 如果要自訂MAC位址組，只要把你想要的MAC位址，一行一行列好，存成這樣的檔名/etc/drbl/MAC-grp-\*, 該檔名\*的部份可以用任何可讀的字元，例如1, DX等。

這樣下次啓動dcs -> Part "根據IP 是MAC位址選擇部份用戶端"的時候，就會列出你定義的群組了。

舉例而言，在DRBL server上，編寫一個檔案"/etc/drbl/IP-grp-classroom1"，內容是：

```
192.168.105.1
192.168.105.2
192.168.105.3
192.168.105.4
192.168.105.5
```

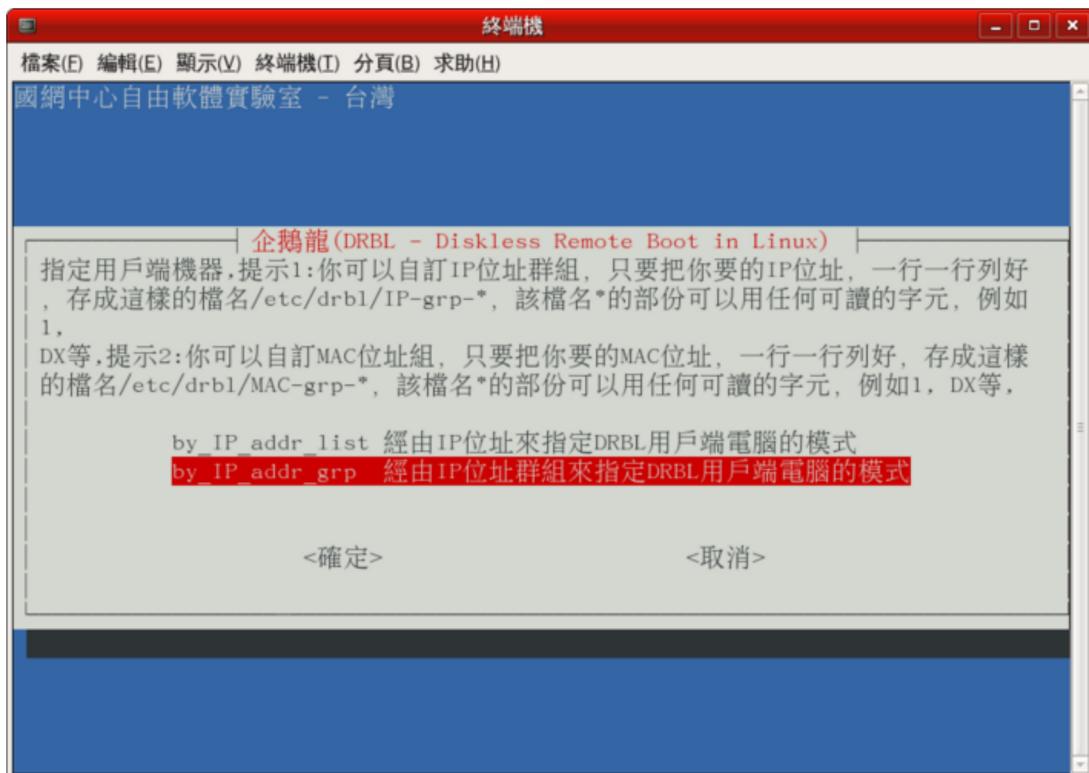
這樣啓動dcs -> Part的時候，



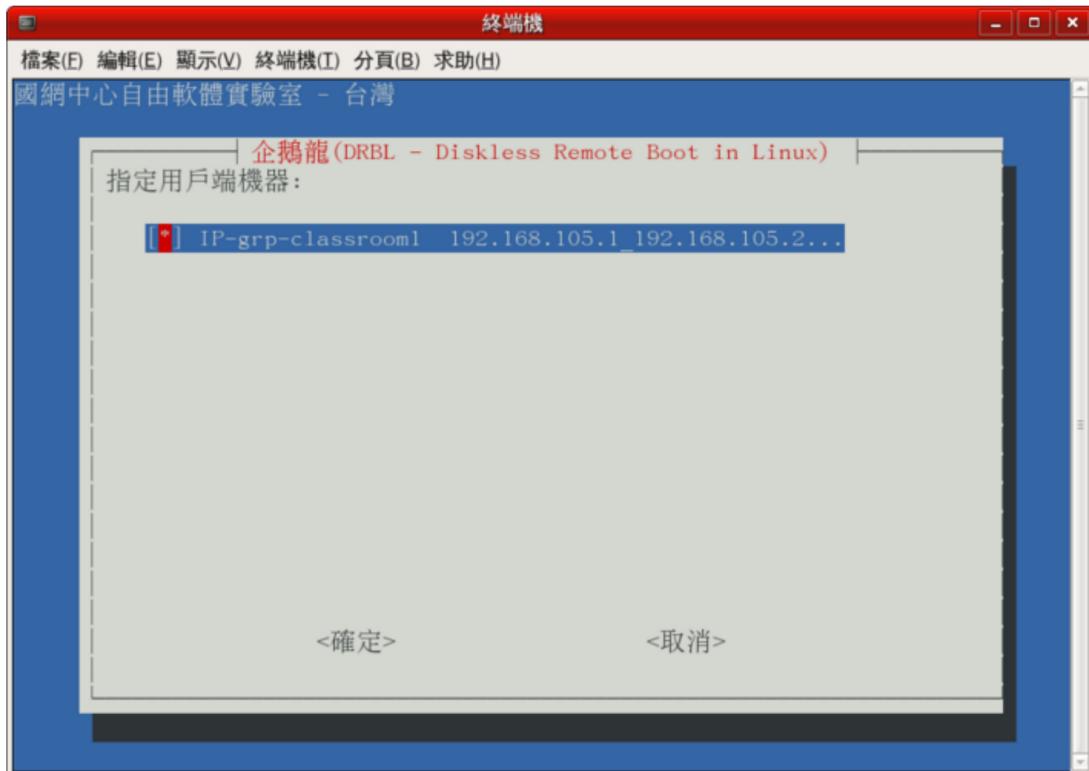
我們就會看到有兩個選項：

- by\_IP\_addr\_list 經由IP位址來指定DRBL用戶端電腦的模式
- by\_IP\_addr\_grp 經由IP位址群組來指定DRBL用戶端電腦的模式

第二個"by\_IP\_addr\_grp"就是因為我們放上/etc/drbl/IP-grp-classroom1這個檔案而新增的選項。如下圖：



然後我們就可以根據我們定義的群組來選擇要的動作了。如下圖：



將 **BIOS** 更新程式放進 **FreeDOS** 以便透過 **PXE** 開機更新 **BIOS** 的作法



由於筆記型電腦越來越便宜，也越來越走輕薄短小風，因此沒有隨機內建光碟機與軟碟機的情形越來越普遍。這時候如果你要更新 **BIOS** 的話，尤其在 **Linux** 環境中，是非常麻煩的一件事情。主機板廠商可能會出 **MS windows** 的 **BIOS** 更新程式，但是卻很少有 **Linux** 上的 **BIOS** 更新程式。幸好現在電腦網路卡變成一個必備的配備，而且都內建 **PXE** 開機功能，因此搭配 **DRBL** 提供的 **FreeDOS** 來做 **BIOS** 更新就非常適當了。

我們可以利用DRBL提供的這個指令:

/opt/drbl/sbin/insert-file-fdos.sh

把主機板更新程式以及firmware塞到/tftpboot/nbi\_img/fdos1440\_drbl.img中，這樣讓要被更新的電腦透過PXE開機後，進入FreeDOS，裡面就內建有主機板更新程式以及firmware了。

以下我們以ASUS Eee PC 900為例，我們可以下載:

1. DOS版的更新程式: wget [http://dlsrv04.asus.com/pub/ASUS/EeePC/EeePC900/AFUDOS\\_XP\\_080514.zip](http://dlsrv04.asus.com/pub/ASUS/EeePC/EeePC900/AFUDOS_XP_080514.zip)
2. BIOS firmware: wget [http://dlsrv04.asus.com/pub/ASUS/EeePC/EeePC900/900\\_ASUS\\_0802.zip](http://dlsrv04.asus.com/pub/ASUS/EeePC/EeePC900/900_ASUS_0802.zip)

下載後，執行:

1. mkdir bios
2. unzip AFUDOS\_XP\_080514.zip -d bios
3. unzip 900\_ASUS\_0802.zip -d bios

由於firmware的檔名"900-ASUS-0802.ROM"太長，最好符合DOS的8.3規則的命名方式，我們把它改名以便後續使用方便:

mv bios/900-ASUS-0802.ROM bios/9000802.rom

然後執行:

cd bios; sudo /opt/drbl/sbin/insert-file-fdos.sh \*

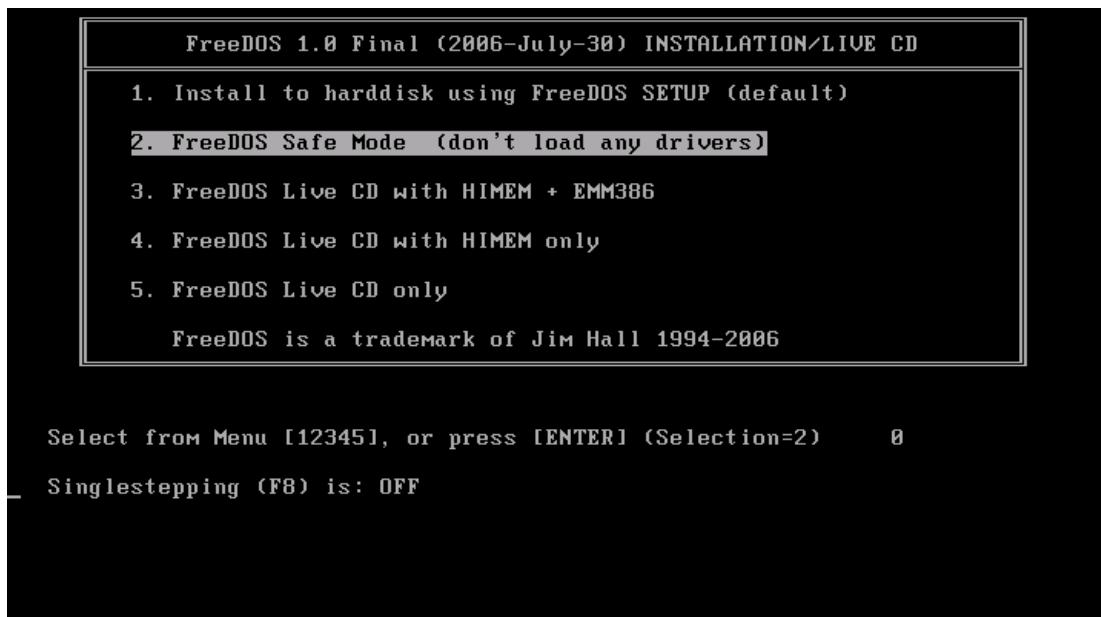
這樣就會把bios這個目錄下的檔案塞到/tftpboot/nbi\_img/fdos1440\_drbl.img中。然後你可以執行:

/opt/drbl/sbin/dcs -> All -> remote-fdos

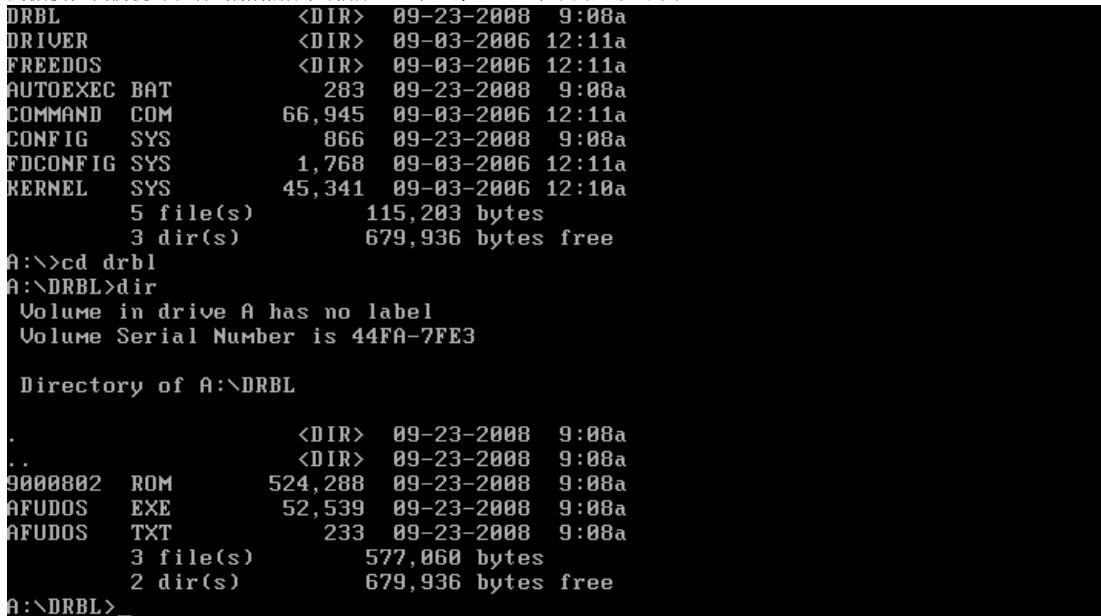
這樣當用戶端透過PXE開機，就會進入FreeDOS，如下圖:



然後我們可以選擇"FreeDOS Safe Mode"，如下圖:



開機後，我們就可以看到剛剛解開的檔案已經在A:\DRBL>裡面了。如下圖：



再來就可以依照ASUS提供的BIOS更新方法來執行BIOS的更新了。

當然這個方法產生的/tftpboot/nbi\_img/fdos1440\_drbl.img你也可以放到Clonezilla live的USB隨身碟中，取代原來的/live/freedos.img，這樣，你也可以用USB隨身碟開機進入FreeDOS來更新Eee PC 900的BIOS。

當你面對整個電腦教室40台電腦要更新BIOS的話，如果你要的話，甚至可以依照FreeDOS的語法，寫一個開機的批次檔，直接執行BIOS更新。這樣整間電腦教室的電腦就可以透過DRBL伺服器直接由PXE開機，更新BIOS一氣呵成。免去準備光碟片軟碟片，一台一台開機，一個指令一個指令執行的痛苦。

若需要 2.8 MB 的 Freedos 做法：

- 參考此篇：[PXE booting a new copy of a slave, nfs, or nfsdata bios image](#)
  - 下載 Freedos 2.8MB [映像檔](#)
- 以上資料感謝學員 Eric Wu 提供。

在**DRBL server**上製作**Linux**套件網路安裝隨身碟



DRBL伺服器提供了用戶端透過PXE開機後，就可以透過網路安裝GNU/Linux套件，這是很方便的作法，只要網路速度不慢，透過這個方式，我們可以完全不需要燒光碟片就可以開始安裝。不過，缺點是，你一定要有一台DRBL伺服器。為了克服這個缺點，從DRBL 1.9.1-25之後，我們提供一個程式：

/opt/drbl/sbin/drbl-syslinux-netinstall

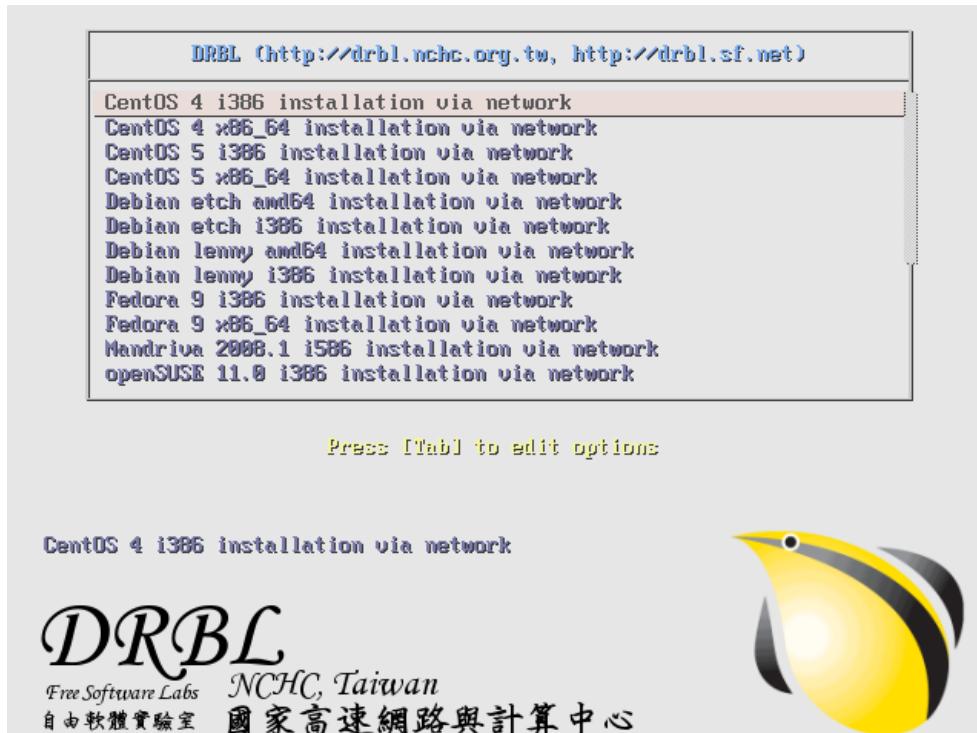
透過這個程式，你可以直接打包一個包含32位元與64位元的CentOS, Debian, Fedora, Mandriva, openSUSE, Scientific Linux, Ubuntu的安裝程式。

只要執行：

sudo /opt/drbl/sbin/drbl-syslinux-netinstall

DRBL就會自動下載上述Linux套件的安裝程式，然後打包成一個類似這樣的檔案："linux-netinstall-20080923.zip"的檔案。再來你就可以把這個檔案解開，放到FAT的隨身碟上，然後依照類似[Clonezilla live隨身碟版的作法](#)把這個隨身碟作成可開機，再來用這個隨身碟在你要安裝Linux的機器上，透過USB開機，就可以由網路來安裝Linux了。完全不需光碟，非常環保。

用這個USB隨身碟來開機的畫面如下圖：



PS.如果你要自己修改某個套件的Linux安裝版本，可以修改/opt/drbl/conf/drbl.conf裡面的某段設定，例如這個安裝程式中你只需要etch的32位元與64位元的安裝程式，那你在這一段設定中：

```
# Debian netinstall
debian_netinstall_ver="etch lenny"
debian_netinstall_arch="i386 amd64"
debian_url_site="http://free.nchc.org.tw"
# Path to Debian where directory $ver (Ex. etch) exists
path_to_debian="debian/dists"
debian_netinstall_kernel="linux"
debian_netinstall_initrd="initrd.gz"
```

將

debian\_netinstall\_ver="etch lenny"

改成

debian\_netinstall\_ver="etch"

即可。

### 利用virtualbox來做drbl開發與測試環境



如果是Ubuntu 8.04的話，其中的scim與virtualbox有相衝的問題，所以我們要把scim輸入法換成gcin。

#### 1. 安裝 gcin virtualbox:

```
sudo apt-get update; sudo apt-get install gcin virtualbox-ose virtualbox-ose-modules-`uname -r`
```

2. 切換預設輸入法為gcin:  
im-switch -s gcin
3. 重新啓動vboxdrv服務:  
sudo /etc/init.d/vboxdrv restart
4. 將自己加入vboxusers的群組, 才能使用virtualbox:  
sudo adduser `whoami` vboxusers
5. 重新登入讓 group 屬性生效

如果是Ubuntu 9.04 者更新的版本的話, 由於上述的問題已經解決了, 所以直接執行這樣來安裝: "apt-get install virtualbox-ose" 即可。這時你在"應用程式" -> "系統工具"裏面就會找到"VirtualBox OSE", 也可以直接在命令列上執行"virtualbox"就可以把virtualbox開啟了。啓動後的控制面板如下圖:



如果要把virtualbox裡的一台虛擬機器當作DRBL client, 直接和實機上安裝DRBL的伺服器溝通, 必須要配置 tun/tap 虛擬網卡裝置, 這樣那台虛擬的client才可以直接跟主機透過網路溝通:

- [下載這個程式](#)
- 然後執行: sudo apt-get install uml-utilities; sudo chmod +x config-tun.sh; sudo ./config-tun.sh start

執行完config-tun.sh之後, 你可以執行ifconfig, 正常就會看到類似這樣的tap0設定:

```
tap0      Link encap:Ethernet HWaddr 00:ff:3e:1e:bd:9e
          inet addr:192.168.125.254 Bcast:192.168.125.255 Mask:255.255.255.0
          inet6 addr: fe80::2ff:3eff:fe1e:bd9e/64 Scope:Link
          UP BROADCAST RUNNING PROMISC MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:6 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
```

此時表示我們的主機和virtualbox中的溝通就是透過tap0這張網卡, 然後以網段192.168.125來連接。不過由於這張虛擬的網卡tap0是剛產生的, 在設定drbl的時候, 執行"/opt/drbl/sbin/drblpush -i"時並沒有tap0這張卡的存在, 因此我們必要要重新執行"/opt/drbl/sbin/drblpush -i"來讓tap0的用戶端可以和DRBL伺服器連接, 使用DRBL伺服器提供的服務。

而在virtualbox新增的虛擬機器中，我們也要在網路卡設定中，在"附掛到"這個選項裡，選擇"主端介面"，然後在"介面名稱"填入"tap0"(設定如下圖)，這樣就完成虛擬機器和我們DRBL伺服器主機的連接了。



有了virtualbox的話，就可以很方便練習架設DRBL與Clonezilla了。

PS.

1. Virtualbox 3.0以上可以在"檔案"->"喜好設定"->"網路"中編輯主機網路(vboxnet0, vboxnet1...)，然後設定一個固定IP address給該網路卡(例如vboxnet0)，並且關閉其中的DHCP伺服器器。不過，這個IP address雖然設定了，但是下次開機後，會自動跳成另外的IP address，這樣DRBL相關的服務就可能失效，因此不建議使用此法。
2. 如果出現執行virtualbox啟動時，出現""Qt FATAL: Failed to create XIM input context! "，那你的環境變數中要設定"QT\_IM\_MODULE=gbm"，也就是你可以在你的家目錄中的.bashrc中加上:"export QT\_IM\_MODULE=gbm"。然後登出再登入就可以生效。
3. [有網友提到說](#) VirtualBox 與scim共存的解法，您可以參考。
4. 如果你要讓你的主機開機都自動啓動tap0網卡的話，以Debian/Ubuntu為例，可以這樣做：  
cp config-tun.sh /et/rc2.d/S99config-tun.sh

### 如何在DRBL上建置SAN (AoE, iSCSI)開機環境？



有關SAN (Storage area network)可以參考[Wikipedia上有關SAN \(Storage area network\)的說明](#)。

至於如何在DRBL上建置SAN (AoE, iSCSI)開機環境？這裡我們以MS Windows XP為例：

在DRBL伺服器上建置MS Windows SAN開機環境: **//注意//** 這個方式還是實驗性質，有些作法未來可能會改變!!!

1. 參考[這裡](#)來準備你的MS Windows XP機器(從現在開始我們稱這台機器為T)。記得將MS Windows XP裝在一個小的分割區 (~ 5 or 10 GB)以便待會把象 檔複製到伺服器可以不用花太多時間。如果你已經裝好的MS Windows XP分割區太大，你可以用[gparted live](#)來縮小分割區。
2. 參考[這裡](#)來把AoE開機的機制裝在T機器的MS Windows XP中，並且把AoE開機的功能在Registry中開啓。
3. 如果你沒有DRBL伺服器，你可以參考[這裡](#)來準備DRBL伺服器。你需要使用drbl 1.9.1-28 之後的版本。如果你已經有一台DRBL伺服

器,但是版本比1.9.1-28舊,在升級drbl之後,記得重新執行"/opt/drbl/sbin/drblsrv -i"與"/opt/drbl/sbin/drblpush -i"。註: **如果你的DRBL伺服器是跑CentOS者RHEL之類套件,你要自己手動安裝vblade這個程式**。你可以在<http://rpm.pbone.net>找到vblade。

4. DRBL伺服器就緒後,在DRBL伺服器上執行"/opt/drbl/sbin/dcs",選擇"remote-linux-txt"。然後將T機器當作DRBL的用戶端,把它經由PXE來開機進入DRBL的用戶端命令列。
5. 在T機器登入成root(如果你的root沒有設定密碼的話,你無法登入該主機,可以先在DRBL伺服器上執行"/opt/drbl/sbin/drbl-client-root-passwd"來設定用戶端root的密碼後再來在T機器登入。)
6. 執行"/etc/init.d/mkswapfile stop"來停止使用swap以免有任何分割區被掛載住。
7. 在T機器上執行"/opt/drbl/sbin/drbl-aoe-img-dump -x",然後選擇要複製到伺服器的硬碟,再來就是輸入象檔名稱。這個命令會將T機器裏面的硬碟的第一個分割區以及MBR,分割區表等資訊在伺服器的/home/sanboot-img裡存成一個類似這樣的檔名"winaoe.aoe-0-0-for-00-50-56-02-01-09"(如果你輸入的象檔名稱是winaoe)。此時會等待一段時間。完成後,建議把這個象檔備份,這樣萬一不小心攪亂這個象檔的時候,不用重新再安裝MS Windows以及再dump一次,可以直接把備份的檔案複製回來即可。註:這裡我們檔名的格式:[name].aoe-[shelf no]-[slot no]-for-[client MAC address]
8. 然後在DRBL伺服器上,登入成root,然後執行"/opt/drbl/sbin/drbl-aoe-serv start"。這樣所有的設定應該已經完成。**現在先移除T機器裏面的硬碟**(一定要移除,要不然SAN硬碟和本機硬碟這兩個開機裝置會相衝突),然後把T機器用PXE開機吧,正常的話,你就可以看到MS Windows XP開機了。
9. 如果不要使用這個AoE服務的話,可以執行"/opt/drbl/sbin/drbl-aoe-serv stop"來停止。

PS. 如果你要監控網路流量的話,可以安裝iptraf(監看TCP/UDP)與wireshark(更多網路底層的資訊,如AoE的封包)來看。

### 從無到有製作DRBL live

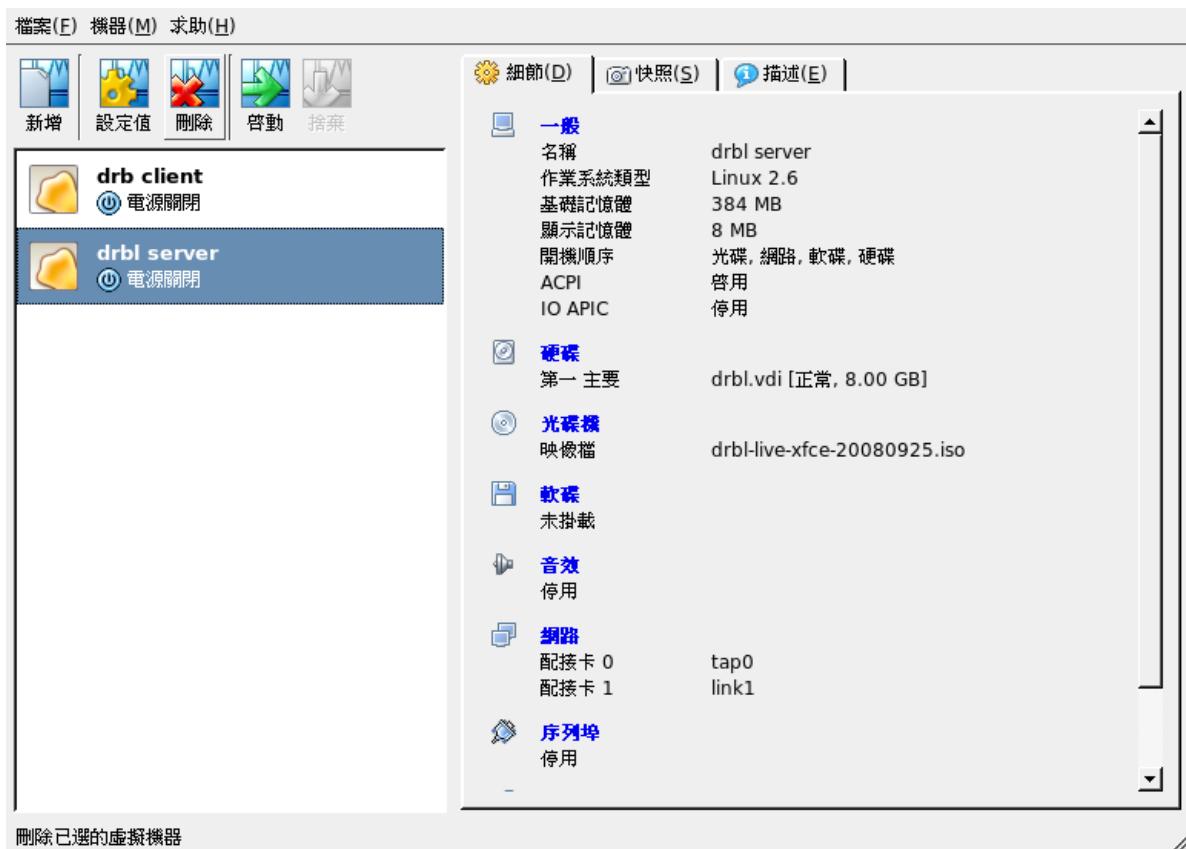


DRBL live是一個尚在測試中的開發版本,主要是可以透過一個光碟者隨身碟把一台機器直接開機,就可以當一台DRBL伺服器,換句話說就是所有的程式不需要裝在那台電腦的硬碟中。從DRBL 1.9.1-25(含)之後的版本提供了一個程式:

/opt/drbl/sbin/create-drbl-live-by-pkg

可以讓你一個指令就可以產生一個DRBL live。不過,這個指令要在Debian者Ubuntu的環境中,才能執行。而且要先裝live helper與cdebootstrap(這兩個都要使用DRBL patch過的版本)才能使用。簡單一點的話,我們可以直接利用Clonezilla live開機後來製作,因為所需要的環境Clonezilla live中都已經有了。做法如下:

1. [下載clonezilla live 1.2.2-31](#)(1.2.3-x的版本還不能適用)
2. 用上述版本來開機之後選擇好語言與鍵盤後,選擇"Start\_Clonezilla" -> "device-image",然後掛載一個空間夠的分割區(**必須是Linux的檔案系統,例如ext3, reiserfs等,不可用FAT是NTFS的分割區**)成為/home/partimag目錄來當工作目錄,建議選用local\_dev,因為NFS之類的網路磁碟目錄和aufs/squashfs等相容性有些問題,製作的時候可能會失敗。
3. 在選擇要備份硬碟/分割區者還原的時候,選擇"exit",進入命令列
4. 執行"sudo su -"
5. 執行ocs-live-netcfg來設定網路,以便待會製作時下載程式需要。
6. cd /home/partimag/
7. apt-get update; apt-get -y install live-helper cdebootstrap
8. 執行:"create-drbl-live-by-pkg" 是"create-drbl-live-by-pkg -m http://free.nchc.org.tw/debian -s http://free.nchc.org.tw/debian-security -e unstable" (預設選項是使用Debian Lenny,使用xfce桌面環境,這些都可以透過適當的參數來指定)等待數十分鐘後(視網路與主機速度而定),就可以產生一個iso檔,以及一個zip檔。前者是給光碟使用,後者是給USB隨身碟使用。更多的create-drbl-live-by-pkg參數可以由執行"create-drbl-live-by-pkg --help"來獲得。
9. 產生好的DRBL live我們就可以讓一台機器開機後,當作是DRBL伺服器。測試時我們可以用Virtualbox,裏面新增兩個虛擬機器,一個DRBL server,另外一個是DRBL client,開機當作DRBL伺服器測試。這兩台機器的設定如下:
  - DRBL server,網路卡我們設定兩張,一個tap0和實際主機溝通,另外一個link1的內部網路卡,光碟機掛載所產生的drbl-live-xfce-20080925.iso檔:



- DRBL client, 設定網路卡1張，並且把"網路"設為開機順序裏面的第一個：



我的網路環境中已經有一個**DHCP**伺服器了，所以我不能再使用**DRBL**伺服器中的**DHCP**服務，請問有沒有  解？

基本上有4個解法：

1. 在DRBL伺服器上使用不同埠的DHCP服務，這樣就不會和現有的服務相衝突。這個作法可以在你架設好DRBL伺服器再來調整。這裡我們以Debian為例，先記得確認你的DRBL版本是**1.9.0-35** 者更新的版本：
  1. 修改DRBL伺服器上的/etc/dhcp3/dhcpd.conf，寫上這兩行：  
`local-port 1067;  
remote-port 1068;`  
 然後重新啟動DHCP服務：  
`/etc/init.d/dhcp3-server restart`
  2. 在DRBL伺服器上執行這個指令：  
`/opt/drbl/sbin/mknic-nbi --udhcpc-port 1068`
  3. 產生特別的開機磁片 者光碟片:到這個網頁<http://www.rom-o-matic.net>,選擇適合你的網卡，然後最重要的是在記得選"3. (optional) To customize ROM configuration press:", 選"ALTERNATE\_DHCP\_PORTS\_1067\_1068". 再來將產生的dsk 者iso檔放到軟碟 著光碟片上。  
 更詳細的細節可以參考這裡的[討論](#)。
2. 另外的方法是透過鎖住用戶端MAC address的方式來提供IP address給用戶端使用。這樣即使是兩個DHCP伺服器也不會互相衝突。你可以在執行"drblpush -i"的時候，選擇要這樣做。
3. 如果你還是喜歡提供浮動IP位址給用戶端，不想鎖定MAC address，從DRBL 1.8.0-15 者更新的版本，你可以在/etc/dhcp3 /dhcpd.conf 者/etc/dhcpd.conf裡，把'allow members of "DRBL-Client";'前面的'#'註解拿掉，然後重新啟動DHCP服務。這個作法是DRBL伺服器只會提供IP位址給PXE, Etherboot 者DRBL的Remote Linux用戶端，其他非這三類的機器 者作業系統，DRBL主機不會發IP位址給它(即使某台機器雖然是DRBL的用戶端，但是是使用本機硬碟裡面的MS Windows 者Linux開機，也會無法從DRBL伺服器要到IP位址)。所以，你要確認所有的PXE 者etherboot開機的機器都是你想要當作DRBL的用戶端使用的機器。
4. 如果你熟悉DHCP語法的話，你可以合併DRBL伺服器上的dhcpd.conf以及你現有的DHCP服務，然後將DRBL伺服器上的DHCP服務關掉 者移除。

者你需要類似例子的話，可以參考這裡：

[https://sourceforge.net/forum/message.php?msg\\_id=5341768](https://sourceforge.net/forum/message.php?msg_id=5341768)

感謝Nikolay Kasatkin提供資訊.



[>>](#)

---

Last modified: February 02 2010 18:21:30.

---

本文件版權屬於[國家高速網路與計算中心自由軟體實驗室](#)所有，且以[GNU FDL](#)授權合約發佈。在遵守 **GNU FDL** 授權合約的情況下，您可以複製、散佈，以及修改本文件。唯請保持作者列表的完整，若您修改過此份文件，也請在作者列表中加入您的大名。  
同時請注意本文件沒有任何擔保，若因本文件而產生任何直接間接 的損失，本文作者群均不負任何法律及道義上的責任。