# SMB HOWTO

David Wood, *dwood@plugged.net.au*
Adaptation française par Mathieu Arnold *arn_ mat@club-internet.fr* v1.1.1 25 Juin 1999

Voici le SMB-HOWTO, décrivant comment utiliser sous Linux le protocole SMB (« Server Message Block ou Session Message Block ») également appelé le protocole NetBIOS ou LanManager.

## Table des matières

**1 Introduction** **1**

**2 Pour plus d'informations** **2**

**3 Installation** **2**

**4 Lancer les daemons** **4**

**5 Configuration générale (/etc/smb.conf)** **5**

**6 Partager un lecteur Linux avec des machines Windows** **6**

**7 Partager un répertoire Windows pour des machines Linux** **8**

**8 Partager une imprimante Linux pour des stations Windows** **10**

**9 Partager une imprimante Windows pour des clients Linux** **13**

**10 Sauvegarder un poste Windows depuis une machine Linux** **21**

**11 Copyright** **24**

**12 Remerciements** **24**

## 1 Introduction

Voici le SMB-HOWTO, décrivant comment utiliser sous Linux le protocole SMB (« Server Message Block ou Session Message Block ») également appelé le protocole NetBIOS ou LanManager. Bien que ce document ne porte que sur Linux, Samba fonctionne aussi sur bien d'autres Unix.

Ce document est maintenu par David Wood (*dwood@plugged.net.au*). Veuillez envoyer vos ajouts, modifications ou corrections à cette adresse, pour qu'ils soient inclus dans la version suivante.

Il existe un nombre impressionnant de documentations sur le site web de samba, situé à : *http ://www.samba.org/* `<http://www.samba.org/>`. Vous pouvez aussi essayer le groupe de discussions `comp.protocols.smb`.

Le protocole SMB est utilisé par Microsoft Windows 3.11, NT et 95/98 pour partager des disques et des imprimantes. En utilisant les outils Samba d'Andrew Tridgell(*Andrew.Tridgell@anu.edu.au*), les systèmes Unix (Linux inclus) peuvent également partager des disques et des imprimantes avec des hôtes Windows. Les smbfs tools de Paal-Kr. Engstad (*engstad@intermetrics.com*) et Volker Lendecke (*lendecke@namu01.gwdg.de*) permettent aux machines Unix de monter des partages venant d'hôtes Windows ou Samba.

Vous pouvez faire quatre choses avec Samba :

1. Partager un disque Linux pour des machines Windows ;
2. Accéder à disque Windows depuis une machine Linux ;
3. Partager une imprimante Linux pour des machines Windows ;
4. Utiliser une imprimante Windows à partir d'un hôte Linux.

Tous ces points sont abordés dans ce document.

Mise en garde : Les procédures décrites et les scripts présentés fonctionnent pour l'auteur ou les personnes qui les ont écrits. Les informations données dans ce document peuvent ne pas fonctionner sur une configuration différente. Si vous rencontrez un problème, écrivez un mail à l'auteur avec vos suggestions pour l'amélioration de ce document, mais l'auteur ne garantit rien. Qu'attendiez-vous ? Après tout, l'auteur n'est qu'un consultant...

Notez que pour que les machines Windows 3.x puissent accéder aux partages SMB, elles doivent avoir une couche TCP/IP d'installée, et les DLL Win32s. Ces deux choses sont disponibles sur le site de Microsoft (*http ://www.microsoft.com* <http://www.microsoft.com>).

# 2 Pour plus d'informations

Cet HOWTO tente d'expliquer comment configurer de manière basique les services SMB de partage de fichier et d'imprimantes sous Linux. Samba est un ensemble de programme très complet, et très complexe également. Cela ne servirait à rien de recopier la documentation de Samba dans ce document.

Pour de plus amples informations, veuillez vous référer aux documents suivants :

- La documentation de Samba, disponible au sein de la distribution de Samba. Cette distribution est disponible à : *ftp ://www.samba.org/* <ftp://www.samba.org/>
- Le Printing-HOWTO.
- Le Print2Win Mini-HOWTO.
- Protocoles standard pour un service NetBios via une couche TCP/UDP.

**RFC 1001**

Concepts et méthode.

**RFC 1002**

Détails sur les spécifications.

# 3 Installation

Tout d'abord, pour pouvoir utiliser Samba sur vos machines, celles ci doivent être sur un même segment ethernet utilisant le protocole TCP/IP. Samba ne marchera pas si vous utilisez un autre protocole. Cela est généralement facile puisque Linux et Windows 95/98/NT sont livrés avec une couche TCP/IP. Toutefois, si vous utilisez une machine Windows 3.x, la couche TCP/IP devra être ajoutée.

Les services SMB ne peuvent pas être utilisés a travers des routeurs. Si vous voulez faire une telle chose, Vous aurez a mettre en place un tunnel IP, ce qui hors de propos dans ce document..

Pour récupérer la dernière version des sources de Samba, allez a cette adresse, et choisissez le site le plus proche de vous : *ftp ://www.samba.org/* `<ftp://www.samba.org/>`

Cependant, si vous avez installé la distribution RedHat de Linux, vous pour pouvez l'installer comme paquetage. Certaines autres distributions proposent également un paquetage Samba (NdT : dont Slackware et Debian).

Les deux « daemons » suivant sont nécessaires pour Samba. Ils sont généralement installé dans le répertoire `/usr/sbin` et sont lancés soit au démarrage soit par inetd. Des scripts d'exemple sont présentés dans la section 4 (Lancer les daemons).

**smbd**
: Le daemon SMB ;

**nmbd**
: Propose le support du serveur de nom NetBIOS aux clients.

Notez que le serveur de noms fournis par le démon nmbd est différent de celui fourni par le DNS. Le serveur de noms NetBIOS est un serveur de nom a la Windows utilisé pour le SMB. En d'autres termes, le fait d'avoir un serveur DNS ne vous garantis pas que Samba réussira a résoudre les noms.

En général, les binaires Samba suivant sont installés dans `/usr/bin` ou `/usr/local/samba/bin`, bien que leur emplacement soit variable.

**smbclient**
: Un client SMB pour machines Unix ;

**smbprint**
: Un script pour imprimer sur l'imprimante d'un hôte SMB ;

**smbprint.sysv**
: Comme ci-dessus, mais pour des Unix SVR4 ;

**smbstatus**
: Liste les connexions SMB présentes sur localhost ;

**smbrun**
: Un script pour faciliter le lancement d'applications sur des hôtes SMB.

Les exécutables pour le smbfs sont décris plus loin dans le document.

De plus, un script nommé « print » est inclus dans cet HOWTO. Il sert d'interface au script smbprint.

Le paquetage Samba est simple à installer. Il suffit de récupérer les sources depuis les sites mentionnés plus haut, et de lire les fichiers `README` fournis dans la distribution. Le fichier `docs/INSTALL.txt` fourni un ensemble d'instruction simple pour une configuration pas à pas.

Installez les daemons dans `/usr/sbin` et les autres programmes dans `/usr/bin`. Installez les pages *man* dans `/usr/local/man`.

Lors de la compilation du paquetage Samba, vous avez dû spécifier dans le Makefile l'emplacement du fichier de configuration, `smb.conf`. Il se trouve généralement dans `/etc`, mais vous pouvez le placer où bon vous semble. Dans la suite du document, nous considérerons que ce fichier de configuration est `/etc/smb.conf`, que le fichier de log est `/var/log/samba.log.%m` (valeur de « logfile » dans le Makefile) et que le répertoire des *locks* (verrous) est `/var/lock/samba` (« lock directory » dans le Makefile).

Installez le fichier de configuration, `smb.conf`. Allez dans le répertoire dans lequel vous avez compilé Samba. Consultez le fichier `README` dans le répertoire `examples/simple`. Copiez le fichier `smb.conf` de ce répertoire

vers /etc. ATTENTION ! Si vous utilisez une distribution Linux et que vous avez déjà installé Samba, il doit déjà exister un fichier smb.conf dans /etc. Vous devriez sans doute commencer avec celui-là.

Si vous ne désirez pas placer votre fichier de configuration dans /etc, mettez le dans le répertoire que de votre choix, et faites un lien symbolique dans /etc :

```
ln -s /path/vers/smb.conf /etc/smb.conf
```

# 4 Lancer les daemons

Les deux daemons SMB sont /ust/sbin/smbd et /usr/sbin/nmbd.

Vous pouvez lancer les daemons Samba depuis l'inetd ou en tant que processus indépendants. Samba sera légèrement plus rapide a répondre si il est configuré en tant que démon indépendant.

Dans les deux cas, vous devriez vérifier que vous avez bien les lignes suivantes dans votre /etc/services :

```
netbios-ns      137/tcp         nbns
netbios-ns      137/udp         nbns
netbios-dgm     138/tcp         nbdgm
netbios-dgm     138/udp         nbdgm
netbios-ssn     139/tcp         nbssn
```

Assurez vous que ces lignes ne sont pas commentées, et suivant votre distribution, vous aurez peut être même a les ajouter. Samba ne sera pas capable de lier les bons ports si votre /etc/services n'a pas ces entrées.

Pour lancer les daemons depuis l'inetd, mettez les lignes suivantes dans le fichier de configuration d'inetd, /etc/inetd.conf :

```
# SAMBA NetBIOS services (for PC file and print sharing)
netbios-ssn stream tcp nowait root /usr/sbin/smbd smbd
netbios-ns dgram udp wait root /usr/sbin/nmbd nmbd
```

Et relancez le daemon inetd avec la commande :

```
kill -HUP ‘cat /var/run/inetd.pid‘
```

Pour lancer les daemons depuis les scripts de démarrage, copier le script suivant sous le nom /etc/rc.d/init.d/smb (sur une RedHat) et faites des liens depuis les fichiers spécifiés dans les commentaire sur ce script :

```
#!/bin/sh

#
# /etc/rc.d/init.d/smb - démarre et stoppe les services SMB.
#
# Les fichiers suivants doivent être des liens symboliques vers ce fichier~:
#    /etc/rc.d/rc1.d/K35smb  (Kille les services SMB à l'extinction)
#    /etc/rc.d/rc3.d/S91smb  (Démarre les services SMB en mode
#                             multi-utilisateur)
#    /etc/rc.d/rc6.d/K35smb  (Kille les services SMB au redémarrage)

# Charge la librairie de fonctions
. /etc/rc.d/init.d/functions

# Charge la configuration réseau
```

```
. /etc/sysconfig/network

# Vérifie que le réseau fonctionne
[ ${NETWORKING} = "no" ] && exit 0

# Traite les arguments
case "$1" in
   start)
      echo -n "Starting SMB services: "
      daemon smbd -D
      daemon nmbd -D
      echo
      touch /var/lock/subsys/smb
     ~;;
   stop)
      echo -n "Shutting down SMB services: "
      killproc smbd
      killproc nmbd
      rm -f /var/lock/subsys/smb
      echo ""
     ~;;
   *)
      echo "Usage: smb {start|stop}"
      exit 1
esac
```

Si au lancement de ce script, vous avez une erreur qui dit que le démon n'a pas pu lier le port 139, cela signifie que vous avez très certainement un autre samba qui n'a pas été arrêté. Vérifiez la liste des processus (avec 'ps auxww | grep mbd') pour savoir si un autre samba est déjà en route.

# 5 Configuration générale (/etc/smb.conf)

La configuration de Samba sur une machine Linux (ou sur un autre Unix) est contrôlée par un seul fichier, `/etc/smb.conf`. Ce fichier indique quelle ressources système vous désirez partager avec le monde extérieur, et quelle restrictions vous voulez mettre dessus.

Puisque les chapitres suivants vont traiter du partage de fichier et d'imprimantes avec des machines Windows, le fichier `smb.conf` présenté dans ce chapitre est aussi simple que possible, en guise d'introduction.

Ne vous inquiétez pas pour les détails, en tout cas pas encore. Les chapitres suivants vont présenter les concepts fondamentaux.

Chaque partie du fichier commence par une entête, comme par exemple `[global]`, `[homes]`, `[printers]`, etc...

La section `[global]` définit quelques variables communes pour le partage de toutes les ressources.

La section `[homes]` permet à un utilisateur distant d'accéder à son répertoire d'accueil (et uniquement aux sien) sur la machine Linux locale. C'est-à-dire que si un utilisateur essaye de se connecter à ce répertoire partagé depuis sa machine Windows, il sera connecté sur sa répertoire d'accueil personnel. Notez que pour cela, il doit posséder un compte sur l'hôte Linux.

Le fichier `smb.conf` suivant donné en exemple permet à des utilisateurs distants d'accéder à leur répertoire *home* sur la machine local et d'écrire dans un répertoire temporaire. Pour qu'un utilisateur Windows puisse voir ces partages, il faut que l'hôte Linux soit sur le réseau local. Ainsi l'utilisateur connecte un lecteur réseau

depuis le gestionnaire de fichiers Windows ou l'Explorateur.

Veuillez noter que dans les chapitres suivants, des champs seront ajoutés à ce fichier pour permettre à plus de ressources d'être partagées.

```
; /etc/smb.conf
;
; Assurez vous de relancer le serveur après avoir fait des changement , dans
; ce fichier. Par exemple~:
; /etc/rc.d/init.d/smb stop
; /etc/rc.d/init.d/smb start

[global]
; décommentez cette ligne si vous désirez autoriser les invités (comptes
; "guest") à se connecter
; guest account = nobody
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba
   share modes = yes

[homes]
   comment = Répertoire homes
   browseable = no
   read only = no
   create mode = 0750

[tmp]
   comment = Espace disque temporaire
   path = /tmp
   read only = no
   public = yes
```

Apres avoir écrit un nouveau smb.conf, il est utile de le tester pour vérifier qu'il ne contient pas de fautes. Il est possible de vérifier la syntaxe d'un fichier smb.conf avec le programme `testparm` (`man testparm`) ; si `testparm` ne trouve pas de problemes, `smbd` chargera correctement le fichier de configuration.

Voici un petit truc : si votre serveur Samba a plus d'une carte ethernet, `smbd` liera peut être la mauvaise. Si c'est le cas, vous pouvez le forcer a lier la bonne en ajoutant une ligne qui ressemble a cela dans la section [global] du `/etc/smb.conf` :

```
   interfaces = 192.168.1.1/24
```

Ou vous remplacez l'adresse IP ci dessus par celle assignée a l'interface réseau que vous voulez utiliser. Le "24" est correct pour un réseau de classe C, mais devra peut être être recalculé si vous avez découpé votre sous réseau. Ce nombre est relatif au masque de sous réseau. Les nombres a utiliser pour les autres classes de réseaux sont donnés dans l'*IP-Masquerade mini-HOWTO.* <http://www.freenix.org/unix/linux/HOWTO/mini/IP-Masquerade.html>

Il y a maintenant un outil de configuration graphique pour Samba : GtkSamba. Allez faire un tour sur *http ://www.open-systems.com/gtksamba.html* <http://www.open-systems.com/gtksamba.html>.

# 6 Partager un lecteur Linux avec des machines Windows

Comme vous pouvez le constater avec le fichier `smb.conf` ci-dessus, partager des disques Linux avec des utilisateurs est simple. Néanmoins, comme pour tout avec Samba, vous pouvez tout contrôler finement. Voici quelques exemples :

Pour partager un répertoire public, créez un clone de la section [tmp] ci-dessus en ajoutant le code suivant dans le smb.conf :

```
[public]
   comment = Partage public
   path = /home/public
   public = yes
   writable = yes
   printable = no
```

Pour que le répertoire ci-dessus soit en lecture pour tout le monde mais uniquement en écriture pour les personnes du groupe staff, modifiez l'entrée comme ci-dessous :

```
[public]
   comment = Partage public
   path = /home/public
   public = yes
   writable = yes
   printable = no
   write list = @staff
```

Il fut un temps ou c'était aussi facile que cela ; vous auriez tout de suite pu lancer Samba et parcourir vos partages depuis un PC sous Windows. Toutefois, Microsoft a récemment rendu les choses plus difficiles pour ceux qui utilisent Samba. Windows 98, Windows NT (service pack 3 et ultérieurs) et les dernières version de Windows 95 utilisent des mots de passe cryptés par défaut. Samba utilise des mots de passe non cryptés par défaut. Vous ne pouvez donc pas parcourir les serveurs quand soit le client, soit le serveur utilisent des mots de passe cryptés, car les connexions ne peuvent se faire de façon anonymes.

Vous saurez si vous avez un problème de mots de passe entre les clients et les serveurs si quand vous essayez de vous connecter a un partage, vous avez une boite de dialogue qui vous dit quelque chose comme "Vous n'avez pas l'autorisation d'accéder a ce compte depuis cette machine"

Vous avez deux possibilités, soit configurer Samba pour utiliser des mots de passes cryptés, soit configurer les machines Windows pour utilisés des mots de passe en clair.

Pour faire en sorte que les Windows marche avec des mots de passe non cryptés :

```
Windows 95/98
=============

Using the registry editor (regedit), create the registry setting
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP
Add a new DWORD value:
Value Name:  EnablePlainTextPassword
Data:        0x01.

Windows NT
==========

Using the registry editor (regedit), create the registry setting
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Rdr\Parameters
Add a new DWORD value:
Value Name:  EnablePlainTextPassword
Data:        0x01
```

Lorsque ces changements auront été réalisés, redémarrez les machines Windows, et essayez de récupérer un partage Samba. Cela devrais marcher, si le serveur Samba utilise des mots de passe en clair.

Pour que Samba utilise des mots de passe cryptés :

Dans la section [global] de votre `/etc/smb.conf`, ajoutez les lignes suivantes :

```
encrypt passwords = yes
smb passwd file = /etc/smbpasswd
```

Je vous conseille fortement de lire les fichiers ENCRYPTION.txt, Win95.txt et WinNT.txt de la documentation de Samba avant de faire ça !

Si votre client et votre serveur utilisent des mots de passe cryptés, vous ne pourrez pas voir les partage disponibles sur le serveur tant qu'une connexion initiale n'aura pas été établie avec une authentification réussie. Pour faire une connexion initiale, entrez le nom du partage manuellement dans le gestionnaire de fichiers de Windows, ou dans la boite de dialogue de l'Explorateur, sous la forme : '\\<machine$gt;\<partage>'. Logez vous sur le serveur avec un nom d'utilisateur et un mot de passe qui sont valides sur le serveur !

Si vous suspectez que votre service de noms NetBios n'est pas correctement configuré (par exemple si vous avez des 'hôte non trouvé' lorsque vous tentez de vous connecter), essayez d'utiliser juste l'adresse IP du serveur : '\\<adresse ip>\<partage>

Pour que les noms de fichiers apparaissent correctement, vous aurez peut etre aussi besoin de mettre quelques options dans les sections appropriés. Pour des clients sous Windows 95/98/NT, les suivantes sont correctes, mais elles auront a etre modifiées pour si vous avez des clients Windows 3.x :

```
; Mangle case = yes semble donner des noms de fichiers corrects
; sous Win95/98/NT.
mangle case = yes

; Si Samba fera attention a la casse en cherchant les fichiers.
case sensitive = no

; Casse par défaut des fichiers crés.
default case = lower

; Préserver la casse de tous les fichiers
preserve case = yes

; Préserver la casse des fichiers 8.3 (DOS)
short preserve case = no
```

Pour d'autres trucs et astuces concernant les partages de répertoires, conférez vous à la documentation de Samba ou aux pages de manuel.

```
interfaces = 192.168.1.1/24
```

Note : L'octet apres le / est la référence du masque de sous réseau. "24" est la valeur a utiliser pour une classe C complete. Pour plus d'informations sur comment calculer les sous réseaux, référez vous à : *http ://www.zipli\nk.net/˜ralphb/IPSubnet/index.html* <http://www.ziplink.net/~ralphb/IPSubnet/index.html>.

Il y a bien plus d'options de configuration dans Samba, mais cela vous permettra de démarrer. Si vous voulez faire quelque chose de plus avancé, référez vous au site web Samba mentionné plus haut.

# 7 Partager un répertoire Windows pour des machines Linux

Un client SMB pour un hôte Unix est inclus dans la distribution de Samba. Il fournit une interface semblable au ftp, en ligne de commande. Vous pouvez utiliser cette utilitaire pour transférer des fichiers entre un

« serveur » Windows et un client Linux.

Pour voir ce que partage une machine donnée, utilisez :

```
/usr/sbin/smbclient -L hôte
```

où « hôte » est le nom NetBIOS de la machine dont vous voulez voir les partages. Vous obtiendrez une liste des « services » en partage, c'est-à-dire le nom des répertoires ou des machines qu'il partage pour vous. A moins que le serveur SMB n'aient aucune protection, vous allez devoir saisir un mot de passe. Utilisez le mot de passe pour le compte d'invité ou pour votre compte personnel sur cette machine.

Par exemple :

```
smbclient -L zimmerman
```

Ce qui devrait donner quelque chose comme cela :

```
Server time is Sat Aug 10 15:58:27 1996
Timezone is UTC+10.0
Password:
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]

Server=[ZIMMERMAN] User=[] Workgroup=[WORKGROUP] Domain=[]

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        public          Disk      Public
        C$              Disk      Default share
        IPC$            IPC       Remote IPC
        OReilly         Printer   OReilly
        print$          Disk      Printer Drivers


This machine has a browse list:

        Server               Comment
        ---------            -------
        HOPPER               Samba 1.9.15p8
        KERNIGAN             Samba 1.9.15p8
        LOVELACE             Samba 1.9.15p8
        RITCHIE              Samba 1.9.15p8
        ZIMMERMAN
```

La « browse list » (liste des machines) montre quels sont les autres serveurs SMB partageant des ressources sur le réseau.

Pour utiliser le client, lancez :

```
/usr/sbin/smbclient service <mot de passe>
```

ou « service » est une machine et un nom de partage. Par exemple, si vous essayez d'accéder à un répertoire en accès public sur une machine appelée zimmerman, le service sera nommé \\zimmerman\public. Néanmoins, à cause des restrictions de l'interpréteur de commandes (le *shell*), vous allez devoir redoubler les *backslashs*, pour obtenir la ligne suivante :

```
/usr/sbin/smbclient \\\\zimmerman\\public mon_mot_de_passe
```

où « mon_mot_de_passe » est votre mot de passe écrit tel quel.

Vous allez obtenir la ligne de commande suivante :

```
Server time is Sat Aug 10 15:58:44 1996
Timezone is UTC+10.0
Domain=[WORKGROUP] OS=[Windows NT 3.51] Server=[NT LAN Manager 3.51]
smb: \>
```

Tapez « h » pour obtenir de l'aide sur smbclient :

```
smb: \> h
ls             dir            lcd            cd             pwd
get            mget           put            mput           rename
more           mask           del            rm             mkdir
md             rmdir          rd             prompt         recurse
translate      lowercase      print          printmode      queue
cancel         stat           quit           q              exit
newer          archive        tar            blocksize      tarmode
setmode        help          ~?             ~!
smb: \>
```

Si vous savez utiliser ftp, vous ne devriez pas avoir besoin de lire la page de manuel de smbclient.

Bien que vous pouvez utiliser smbclient pour des tests, vous en aurez vite assez si vous voulez travailler sérieusement. Pour cela, vous voudrez très certainement utiliser le paquetage smbfs qui vient avec deux utilitaires très simples, smbmount et smbumount. Ils marchent de façon similaire a mount et umount pour des partages SMB.

Le paquetage smbfs est dorénavent inclu dans la majorité des distributions Linux. Une chose importante a noter : vous devez avoir le support smbfs de compile dans votre noyau (ou en module) pour utiliser ces programmes.

# 8 Partager une imprimante Linux pour des stations Windows

Pour partager une imprimante Linux pour des stations Windows, vous devez être sûr que votre imprimante est configurée pour fonctionner sous Linux. Si vous savez imprimer depuis Linux, la mise en place d'un partage de l'imprimante par SMB est quasi-automatique.

Veuillez vous référer au Printing HOWTO pour la configuration de l'imprimante.

Puisque l'auteur utilise une imprimante connectée à une machine sous Windows NT (Il y a bien longtemps, avant de convertir presque tout notre réseau à Linux), cette section ne doit pas être prise à la lettre, mais simplement comme une suggestion. Quiconque ayant des détails sur ce point est invité à les envoyer à *dwood@plugged.net.au* afin que ce chapitre puisse être complété.

Ajoutez la configuration d'impression à votre `smb.conf` :

```
[global]
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes
   log file = /var/log/samba-log.%m
   lock directory = /var/lock/samba

[printers]
```

```
   comment = Toutes les imprimantes
   security = server
   path = /var/spool/lpd/lp
   browseable = no
   printable = yes
   public = yes
   writable = no
   create mode = 0700

[ljet]
   security = server
   path = /var/spool/lpd/lp
   printer name = lp
   writable = yes
   public = yes
   printable = yes
   print command = lpr -r -h -P %p %s
```

Assurez vous que le « path » (dans cet exemple, à l'intérieur de la section `[ljet]`) corresponde au répertoire de *spool* défini dans `/etc/printcap` !

Les lignes :

```
printcap name = /etc/printcap
load printers = yes
```

Contrôle si toutes les imprimantes du `/etc/printcap` doivent être chargées par défaut. Si vous faites cela, il ne sera pas nécessaire de mettre en place les imprimantes unes a unes. La section [printers] spécifie les options pour les imprimantes que vous voulez définir explicitement. Si le système d'impression que vous utilisez ne marche pas de cette manière (BSD), vous aurez a faire un faux fichier printcap (ou utiliser la technique du 'print command' décrite ci dessous). Pour plus d'informations sur le système printcap, référez vous au Printing HOWTO.

Une technique utile pour tester les connexions réseau est de changer la commande print par :

```
print command = cp %S /tmp/print.%P.%S
```

Le fichier résultant pouvant alors être analysé.

NB : On constate quelques problèmes pour le partage d'imprimante sur des serveurs Unix, utilisant Samba, pour des clients sous Windows NT. Un des problèmes est que NT ne voit pas l'imprimante partagée correctement. Pour résoudre ce problème, lisez les remarques dans le fichier `docs/WinNT.txt` de la distribution de Samba. Un autre problème existe avec les mots de passe. Voyez également dans ce fichier comment résoudre le problème.

Oleg L. Machulskiy (*machulsk@shade.msu.ru*) suggère une meilleur commande print a mettre dans l'exemple précédent :

```
print command = smb2ps %s | lpr -r -h -P %p
```

Ou 'smb2ps' est un script qui transforme le fichier reçu d'un poste Windows en un fichier postscript utilisable. Il doit enlever les 3 premières lignes ainsi que les 2 dernières, car celles ci contienent quelques codes PJL ou PCL.

Cette approche n'est nécessaire que si votre imprimante n'imprime que du PCL et non du vrai postscript. J'ai découvert que Windows 95/98/NT n'ont pas de pilote postscript générique, mais le "Digital turbo

Printserver 20" est un bon pilote générique pour la majorité des installations. J'ai aussi entendu dire que le "Apple LaserWriter II NTX" marchais plutôt bien aussi.

Jeff Stern (*jstern@eclectic.ss.uci.edu*) a rapporté ce qui suit et qui pourrais intéresser certains d'entre vous :

————————%<————————-

Le probleme que j'avais etais que je pouvais imprimer avec lpd/lpr sur mon imprimante sous Linux, depui la machine elle meme. Mais je n'arrivais pas a la faire imprimer depuis une machine Win95, quand j'essayais :

```
smbclient \\eclectic\belen -P
```

Puis, j'ai essayé :

```
print mon_fichier
```

J'ai eu des erreurs a propos d'un accès non autorisé. J'ai donc mis les permissions sur le `/var/spool/lpd/lp1` a 777. Je suis désolé, mais c'est la seule manière que j'ai trouvé. Je suppose que j'aurais pu me débrouiller en mettant tout le monde dans le groupe 'lp', ou en donnant au répertoire le groupe 'users', etc. Mais pour l'instant, cela marche (avec root :lp comme droits).

Enfin, l'autre chose que les administrateurs devraient savoir est que le nom du compte sur la machine windows (qui essaye d'acceder a la machine linux) devrais etre le meme que le compte sur la machine linux. Par conséquent, si il y a un utilisateur 'joe' sur la machine windows, 'mamachinewin' et qu'il essaye d'imprimer sur l'imprimante de la machine linux 'eclectic' (\\eclectic\belen), il doit y avoir un utilisateur 'joe' sur la machine linux. De plus, le mot de passe de joe sera celui pour acceder a l'imprimante belen sur eclectic. Ce mot de passe sera demandé lors de l'ajout de l'imprimante sur la machine windows.

Je suppose que cela n'est pas vraiment la cas car j'ai déclaré mon imprimante en tant que 'public' dans mon `smb.conf`. Mais apparament, il continue a demander le mot de passe. (la boite de dialogue de windows ne vous permet de pas spécifier un autre nom d'utilisateur pour la machine distante. Il vous donne juste le nom de l'utilisateur local que vous avez donné lorsque vous vous etes logé au démarrage de windows.

————————%<————————-

Dr. Michael Langner (*langner@fiz-chemie.de* souligne que les droits d'acces sur `/var/spool/lpd` peuvent etre évités en utilisant un `"path = /tmp"` et un `"print command = lpr -r -P%p %s"` a la place.

Quelquefois, une erreur d'interprétation Postscript arrivera lorsque vous imprimerez du Postscript d'une imprimante Windows, qui fera qu'une page supplémentaire sera imprimée a la fin de chaque impression. La derniere page aura toujours "%%[ Lastpage ]%%" d'inscrit en haut. Cela semble arriver uniquement avec Windows 95 et 98 car le Postscript est mal formé.

L'une des manieres d'empecher cela est d'utiliser un petit script qui supprime le bout de Postscript des travaux d'impression. Une autre maniere est de trouver un pilote Postscript pour windows qui sache faire du Postscript. La meilleur maniere est semble-t-il d'utiliser LPRng a la place de Postscript pour imprimer sur l'imprimante Samba.

Erik Ratcliffe (*erik@caldera.com*) de Caldera me dit que le fait d'utiliser le pilote LPRng impliquera qu'il faudra utiliser le bon pilote sur les machines Windows. Sur le serveur Samba, il y a un entrée dans leur `/etc/printcap` qui ressemble a ça :

```
raw:\
        :rw:sh:
        :lp=/dev/lp1
        :sd=/var/spool/lpd/raw
        :fx=flp
```

LPRng ne necessite pas :\ a la fin de chaque ligne. Une entrée d'imprimante sera toujours necessaire dans le `/etc/smb.conf` pour l'imprimante physique. La commande `print` devra utiliser l'entrée "raw" du fichier `/etc/printcap` et les données devront etre envoyéees a l'imprimant sous forme binaire de la maniere suivante :

```
print command = lpr -b -Praw %s
```

Vous aurez peut etre aussi a mettre la file d'attente sur la machine Windows 95, et a imprimer directement sur l'imprimante plutôt que d'utiliser une file d'attente.

# 9 Partager une imprimante Windows pour des clients Linux

Pour partager une imprimante sur une machine Windows, vous devez suivre les points suivants :

1. Vous devez avoir les entrées correspondantes à l'imprimante dans `/etc/printcap` et elles doivent correspondre à la structure locale des répertoires (pour le répertoire de *spool*, etc...).
2. Vous devez avoir le script `/usr/bin/smbprint`. Il est fournit avec les sources de Samba, mais pas avec toutes les distributions binaires. Une version légèrement modifiée de ce script est présentée plus loin.
3. Si vous voulez convertir des fichiers ASCII en Postscript, vous devez avoir `nenscript`, ou équivalent. `nenscript` est un convertisseur Postscript et se trouve généralement dans `/usr/bin`.
4. Vous voudrez peut-être simplifier l'impression à l'aide de Samba en utilisant une interface simple d'emploi. Un script simple, écrit en perl, pour gérer l'ASCII ou le PostScript est présenté ci-dessous.
5. Vous pouvez aussi utiliser MagicFilter pour faire la chose précédente. Les instruction sur comment mettre en place MagicFilter sont données après le script perl. MagicFilter a comme avantage de connaître maints formats de fichiers.

L'entrée dans `/etc/printcap` est pour une imprimante HP 5 MP sur une machine utilisant Windows NT. Les entrées sont comme suit :

```
cm - commentaire
lp - nom du périphérique à ouvrir en écriture
sd - le répertoire de spool de l'imprimante (sur la machine locale)
af - le fichier d'accounting
mx - la taille maximum de fichier (zéro pour aucune limite)
if - le nom du filtre en entrée (un script)
```

Pour plus ample information, lisez le Printing HOWTO ou la page de manuel de printcap.

```
# /etc/printcap
#
# //zimmerman/oreilly avec smbprint
#
lp:\
       ~:cm=HP 5MP Postscript OReilly sur zimmerman:\
       ~:lp=/dev/lp1:\
       ~:sd=/var/spool/lpd/lp:\
       ~:af=/var/spool/lpd/lp/acct:\
       ~:mx#0:\
       ~:if=/usr/bin/smbprint:
```

Assurez que le répertoire de spool et celui d'accounting existent et son accessibles en écriture, que le chemin correct vers le script smbprint (donné ci-dessous) est indiqué par la ligne « if » et que vous avez sélectionné le bon fichier de périphérique (le fichier spécial dans `/dev`).

Vient ensuite le script smbprint. Il est généralement mis dans le répertoire `/usr/bin` et a été créé par Andre Tridgell, le créateur de Samba pour autant que je sache. Il est fourni avec la distribution sous forme de code source de Samba, mais est absent de certaines distribution binaires. Je l'ai donc recopié ici.

Examinez le avec attention. Certains changement fait à ce script se sont avérées être utiles.

```
#!/bin/sh -x

# Ce script est un filtre d'entrée sur l'impression avec printcap sur une
# machine Linux. Il utilise le programme smbclient pour imprimer le fichier au
# serveur et service spécifié.
# Par exemple, vous pouvez avoir une entrée printcap comme celle-ci~:
#
# smb:lp=/dev/null:sd=/usr/spool/smb:sh:if=/usr/local/samba/smbprint
#
# qui créerai une imprimante Unix appelée "smb" qui imprimerait par
# l'intermédiaire de ce script. Vous devrez créer le répertoire de spool,
# /usr/spool/smb avec les permissions qui conviennent et le bon propriétaire,
# pour votre système.

# Mettez les valeurs pour le serveur et le service sur lequel vous voulez
# imprimer.  Dans cet exemple, j'utilise un PC sous Windows pour Workgroups
# nommé "laplan" ayant une imprimante appelée "printer" sans mot de passe.

#
# Script modifié par hamiltom@ecnz.co.nz (Michael Hamilton) afin que le
# serveur, le service et le mot de passe puissent être lus depuis un fichier
# /usr/var/spool/lpd/PRINTNAME/.config
#
# Pour que ceci puisse fonctionner, l'entrée du /etc/printcap doit inclure un
# fichier d'accounting (af=...)~:
#
#   cdcolour:\
#       :cm=CD IBM Colorjet au 6eme etage:\
#       :sd=/var/spool/lpd/cdcolour:\
#       :af=/var/spool/lpd/cdcolour/acct:\
#       :if=/usr/local/etc/smbprint:\
#       :mx=0:\
#       :lp=/dev/null:
#
# Le fichier /usr/var/spool/lpd/PRINTNAME/.config devrait contenir~:
#   server=SERVEUR_PC
#   service=NOM_IMPRIMANTE
#   password="mot_de_passe"
#
# Pas exemple~:
#   server=MON_BO_PC
#   service=CJET_371
#   password=""

#
# Fichier de log pour debuggage, changez le à /dev/null si vous le voulez
#
logfile=/tmp/smb-print.log
# logfile=/dev/null
```

```
#
# Le dernier paramètre du filtre est le nom du fichier d'accounting
#
spool_dir=/var/spool/lpd/lp
config_file=$spool_dir/.config

# Les variables suivantes devraient être lues depuis le fichier de
# configuration~:
#   server
#   service
#   password
#   user
eval `cat $config_file`

#
# Des informations de débogage, changez le >> en > si vous voulez économiser
# de la place.
#
echo "server $server, service $service" >> $logfile

(
# NOTE Vous voudrez peut être ajouter la ligne "echo translate" si vous voulez
# une conversion automatiques des CR/LF lors de l'impression
        echo translate
        echo "print -"
        cat
) | /usr/bin/smbclient "\\\\$server\\$service" $password -U $user -N -P >> $logfile
```

La plupart des distributions Linux sont fournies avec nenscript pour convertir des documents ASCII en Postscript. Le script perl qui suit simplifie la vie en fournissant une interface simple à smbprint pour l'impression sous Linux.

```
Usage: print [-a|c|p] <fichier>
       -a imprime <fichier> comme un fichier ASCII
       -c imprime <fichier> formatté en code source
       -p imprime <fichier> en tant que fichier Postscript
       Si aucun paramètre n'est donné, print tente de
       deviner le type de fichier et imprime en conséquence.
```

smbprint a tendance à tronquer les longues lignes lors de l'impression de fichiers ASCII. Ce script coupe les longues lignes sur les espaces (plutôt qu'au milieu d'un mot), si possible.

Le formatage en code source est réalisé par nenscript.Il prend en entrée un fichier ASCII et le formatte sur deux colonnes avec une entête (date, nom du fichier, etc...). Il numérote également les lignes. En prenant ce script comme exemple, on peut faire d'autres types de formatage.

Les documents sont déjà correctement formatés, donc ils passent directement à travers le filtre.

```
#!/usr/bin/perl

# Script:   print
# Auteurs:  Brad Marshall, David Wood
#           Plugged In Communications
# Date:     960808
#
```

```
# Script pour imprimer sur une imprimante postscript via samba.
# But:      Prendre différentes sortes de fichier en argument et les
#  traiter pour les injecter dans le script d'impression de Samba.
#
# Types de fichier supportés pour l'instant~:
#
# ASCII      - vérifie que les lignes plus longues que $line_length
#              caractères sont coupés sur un espace.
# Postscript - Aucune action.
# Code       - Formatte en Postscript (à l'aide de nenscript) pour un
#              affichage correct (orientation, fonte, etc...).
#

# Fixe la longueur maximale d'une ligne de texte ASCII
$line_length = 76;

# Le chemin d'accès vers le script d'impression de Samba
$print_prog = "/usr/bin/smbprint";

# Le chemin vers le programme nenscript (le convertisseur
# ASCII->Postscript)
$nenscript = "/usr/bin/nenscript";

unless ( -f $print_prog ) {
        die "Je ne peux pas trouver $print_prog!";
}
unless ( -f $nenscript ) {
        die "Je ne peux pas trouver $nenscript!";
}

&ParseCmdLine(@ARGV);

# Débug
print "filetype is $filetype\n";

if ($filetype eq "ASCII") {
        &wrap($line_length);
} elsif ($filetype eq "code") {
        &codeformat;
} elsif ($filetype eq "ps") {
        &createarray;
} else {
        print "Désolé, ce n'est pas un type de fichier que je connais";
        exit 0;
}
# Envoie le tableau à smbprint
open(PRINTER, "|$print_prog")
         || die "Je ne peux pas ouvrir $print_prog: $!\n";

foreach $line (@newlines) {
        print PRINTER $line;
}
# Envoie un retour à la ligne supplémentaire si jamais le fichier a sa
# dernière ligne incomplète
print PRINTER "\n";
```

```
close(PRINTER);
print "Achevé\n";
exit 0;

# ------------------------------------------------- #
#       Tout ce qui suit est un sous programme      #
# ------------------------------------------------- #

sub ParseCmdLine {
        # Traite la ligne de commande, détermine le type de fichier

        # $arg et $file sont respectivement les arguments (s'ils
        # existent) et le nom de fichier
        if ($#_ < 0) {
                &usage;
        }
        # Débug
#       foreach $element (@_) {
#               print "*$element* \n";
#       }

        $arg = shift(@_);
        if ($arg =~ /\-./) {
                $cmd = $arg;
        # Débug
#       print "\$cmd trouvé.\n";

                $file = shift(@_);
        } else {
                $file = $arg;
        }

        # Définition du type de fichier
        unless ($cmd) {
                # Aucun argument

                if ($file =~ /\.ps$/) {
                        $filetype = "ps";
                } elsif ($file =~ /\.java$|\.c$|\.h$|\.pl$|\.sh$|\.csh$|\.m4$|\.inc$|\.html$|\.htm$/) {
                        $filetype = "code";
                } else {
                        $filetype = "ASCII";
                }

                # Traite $file selon le type de fichier et retourne
                # le type de fichier ($filetype)
        } else {
                # Nous utilisons ltype de fichier décrit dans $arg
                if ($cmd =~ /^-p$/) {
                        $filetype = "ps";
                } elsif ($cmd =~ /^-c$/) {
                        $filetype = "code";
                } elsif ($cmd =~ /^-a$/) {
                        $filetype = "ASCII"
                }
```

```
        }
}

sub usage {
        print "
Usage: print [-a|c|p] <fichier>
       -a imprime <fichier> comme un fichier ASCII
       -c imprime <fichier> formaté en code source
       -p imprime <fichier> en tant que fichier Postscript
       Si aucun paramètre n'est donné, print tente de
       deviner le type de fichier et imprime en conséquence.\n
";
        exit(0);
}

sub wrap {
        # Crée un table contenant les lignes du fichier, avec chaque
        # ligne ayant une longueur < au nombre de caractères
        # spécifiés, et coupée uniquement sur un espace.

        # Récupère la longueur maximum d'une ligne
        $limit = pop(@_);

        # Débug
        #print "Entrée dans la procédure wrap\n";
        #print "La longueur maximum d'une ligne est $limit\n";

        # Lit le fichier, le traite et le stocke dans le tableau
        open(FILE, "<$file") || die "Impossible d'ouvrir $file: $!\n";
        while(<FILE>) {
                $line = $_;

                # Débug
                #print "La ligne est~:\n$line\n";

                # Coupe la ligne si celle-ci dépasse la limite
                while ( length($line) > $limit ) {

                        # Débug
                        #print "Je coupe...";

                        # Prend les premiers $limit +1 caractères.
                        $part = substr($line,0,$limit +1);

                        # Débug
                        #print "La ligne partielle est~:\n$part\n";

                        # Vérifie si le dernier caractère est un
                        # espace
                        $last_char = substr($part,-1, 1);
                        if ( " " eq $last_char ) {
                            # Oui, on imprime le reste

                            # Débug
                            #print "Le dernier caractère était un espace\n";
```

```
                substr($line,0,$limit + 1) = "";
                substr($part,-1,1) = "";
                push(@newlines,"$part\n");
            } else {
                # Non, on cherche le dernier espace de la
                # ligne et on imprime jusqu'à lui

                # Débug
                #print "Le dernier caractère n'était pas un espace\n";

                # Supprime le caractère après $limit
                substr($part,-1,1) = "";
                # Inverse la ligne pour trouver plus
                # facilement l'espace
                $revpart = reverse($part);
                $index = index($revpart," ");
                if ( $index > 0 ) {
                  substr($line,0,$limit-$index) = "";
                  push(@newlines,substr($part,0,$limit-$index)
                      . "\n");
                } else {
                   # Aucun espace dans la ligne
                   # Imprime jusqu'à $limit
                   substr($line,0,$limit) = "";
                   push(@newlines,substr($part,0,$limit)
                       . "\n");
                 }
            }
        }
        push(@newlines,$line);
    }
    close(FILE);
}

sub codeformat {
    # Appelle la procédure wrap et filtre par nenscript
    &wrap($line_length);

    # Envoie le résultat à nenscript pour créer un fichier
    # Postscript qui respecte un format décent d'impression pour
    # du code source (orientation paysage, font Courier,
    # numérotation des lignes).
    # Imprime d'abord dans un fichier temporaire.
    $tmpfile = "/tmp/nenscript$$";
    open(FILE, "|$nenscript -2G -i$file -N -p$tmpfile -r") ||
        die "Je ne peux pas ouvrir nenscript~: $!\n";
    foreach $line (@newlines) {
        print FILE $line;
    }
    close(FILE);

    # Relis le fichier temporaire dans un tableau pour pouvoir
    # être passé au script smbprint de Samba.
    @newlines = ("");
```

```
        open(FILE, "<$tmpfile") || die "Je ne peux pas ouvrir $file~: $!\n";
        while(<FILE>) {
                push(@newlines,$_);
        }
        close(FILE);
        system("rm $tmpfile");
}

sub createarray {
        # Crée le tableau pour un fichier postscript
        open(FILE, "<$file") || die "Can't open $file: $!\n";
        while(<FILE>) {
                push(@newlines,$_);
        }
        close(FILE);
}
```

Maintenant, avec MagicFilter. Merci a Alberto Menegazzi (*flash.egon@iol.it*) pour ces informations.

Alberto nous dit :

————————%<————————-

1) Installez MagicFilter avec le filtre pour l'imprimante que vous comptez utiliser dans `/usr/local/bin` mais **ne mettez pas** dans votre `/etc/printcap` ce qui est conseillé par la documentation de MagicFiler.

2) Ecrivez un `/etc/printcap` comme celui la (c'est pour ma LaserJet 4L) :

```
lp|ljet4l:\
        :cm=HP LaserJet 4L:\
        :lp=/dev/null:\                         # ou /dev/lp1
        :sd=/var/spool/lpd/ljet4l:\
        :af=/var/spool/lpd/ljet4l/acct:\
        :sh:mx#0:\
        :if=/usr/local/bin/main-filter:
```

Le `lp=/dev/..` n'est la que pour servir de vérrou, par conséquent, un périphérique "virtuel" est necessaire pour chaque imprimante distante.

Par exemple, créez le avec un `touch /dev/ljet4l`.

3) Ecrivez un filtre `/usr/local/bin/main-filter` de cette maniere la, en utilisant `ljet4l-filter` a la place de `cat` :

Voici le mien :

```
#! /bin/sh
logfile=/var/log/smb-print.log
spool_dir=/var/spool/lpd/ljet4l
(
  echo "print -"
  /usr/local/bin/ljet4l-filter
) | /usr/bin/smbclient "\\\\SHIR\\HPLJ4" -N -P >> $logfile
```

P.S. : Voici l'extrait du mini howto Print2Win a propos des vérous et des imprimantes virtuelles.

—Ca commence ici

deux petits trucs de Rick Bressler :

Voici un bon truc. J'utilise quelque chose d'assez similaire. Un truc qui aide pas mal, voici la chose a ne pas faire :

```
:lp=/dev/null:\
```

lpr ouvre de maniere exclusive le fichier que vous spécifiez comme `lp=`. Il fait cela de maniere a empecher plusieurs process d'imprimer sur la même imprimante en même temps.

L'effet de bord dans ce cas est que deux imprimantes ayant le meme `lp=` ne pourront pas imprimer en même temps, (c'est habituellement assez transparent car elles sont peut etre rapides et comme elles utilisent une file d'attente, vous ne vous en rendez peut etre pas compte) mais tout autre processus qui tentera d'écrire sur /dev/null ne marchera plus !

Sur un systeme monoutilisateur, ce n'est probablement pas un probleme. J'ai un systeme avec plus de 50 imprimantes, et la, c'est un probleme

La solution est de créer une imprimante virtuelle pour chacun. Ex : touch /dev/couleur

J'ai modifié les entrées lp dans le fichier `printcap` pour prendre en compte les suggestions de Rick. J'ai fait les choses suivantes :

```
#touch /dev/laser
#touch /dev/couleur
```

—Fin

————————%<————————-

# 10 Sauvegarder un poste Windows depuis une machine Linux

Adam Neat (*adamneat@ipax.com.au*) a gracieusement fourni le script suivant pour sauvegarder une machine Windows depuis une machine Linux, en utilisant l'utilitaire smbclient. Adam dit qu'il l'utilise pour sauvegarder des Windows 3.x et NT sur un linux avec un DAT SCSI.

Adam n'est pas fier du code qu'il a pondu, mais il marche. Comme j'ai l'habitude de le dire, "Si ca marche et que c'est stupide, alors, ce n'est pas stupide".

Dans ce script, la chaîne 'agnea1' est l'utilisateur de la machine Linux qui fait les sauvegardes.

```
#!/bin/bash

clear
echo Initialising ...
checkdate=‘date | awk ’{print $1}’‘

if [ -f "~agnea1/backup-dir/backup-data" ]; then

        echo "ERROR: No config file for today!"
        echo "FATAL!"
        exit 1
fi

if [ -d "~agnea1/backup-dir/temp" ]; then

        echo "ERROR: No tempoary directory found!"
        echo
        echo "Attempting to create"
```

```
        cd ~agnea1
        cd backup-dir
        mkdir temp
        echo "Directory Made - temp"
fi

if [ "$1" = "" ]; then

        echo "ERROR: enter in a machine name (ie: cdwriter)"
        exit 1
fi

if [ "$2" = "" ]; then

        echo "ERROR: enter in a SMB (Lan Manager) Resource (ie: work)"
        exit 1
fi

if [ "$3" = "" ]; then

        echo "ERROR: enter in an IP address for $1 (ie:
        130.xxx.xxx.52)" exit 1
fi


#############################################################################
# Main Section
#
#############################################################################

cd ~agnea1/backup-dir/temp
rm -r ~agnea1/backup-dir/temp/*
cd ~agnea1/backup-dir/

case "$checkdate"
in
        Mon)
                echo "Backuping for Monday"
                cat backup-data | /usr/local/samba/bin/smbclient
                \\\\$1\\$2 -I$3 -N echo "Complete"

                        if [ -d "~agnea1/backup-dir/Monday" ]; then
                                echo "Directory Monday Not found ...
                                making" mkdir
                                ~agnea1/backup-dir/Monday
                        fi

                echo "Archiving ..."
                cd ~agnea1/backup-dir/temp
                tar -cf monday.tar *                echo "done ..."
                rm ~agnea1/backup-dir/Monday/monday.tar
                mv monday.tar ~agnea1/backup-dir/Monday
                ;;
```

```
Tue)
        echo "Backuping for Tuesday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\\\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agnea1/backup-dir/Tuesday" ]; then
                        echo "Directory Tuesday Not found ...
                        making" mkdir
                        ~agnea1/backup-dir/Tuesday
                fi
        echo "Archiving ..."
        cd ~agnea1/backup-dir/temp
        tar -cf tuesday.tar *
        echo "done ..."
        rm ~agnea1/backup-dir/Tuesday/tuesday.tar
        mv tuesday.tar ~agnea1/backup-dir/Tuesday
        ;;

Wed)
        echo "Backuping for Wednesday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\\\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agnea1/backup-dir/Wednesday" ]; then
                        echo "Directory Wednesday Not found
                        ... making" mkdir
                        ~agnea1/backup-dir/Wednesday
                fi
        echo "Archiving ..."
        cd ~agnea1/backup-dir/temp
        tar -cf wednesday.tar *
        echo "done ..."
     rm ~agnea1/backup-dir/Wednesday/wednesday.tar
        mv wednesday.tar ~agnea1/backup-dir/Wednesday
        ;;

Thu)
        echo "Backuping for Thrusday"
        cat backup-data | /usr/local/samba/bin/smbclient
        \\\\$1\\$2 -I$3 -N echo "Complete"

                if [ -d "~agnea1/backup-dir/Thursday" ]; then
                        echo "Directory Thrusday Not found ...
                        making" mkdir
                        ~agnea1/backup-dir/Thursday
                fi
        echo "Archiving ..."
        cd ~agnea1/backup-dir/temp
        tar -cf thursday.tar *
        echo "done ..."
        rm ~agnea1/backup-dir/Thursday/thursday.tar
        mv thursday.tar ~agnea1/backup-dir/Thursday
        ;;
```

```
        Fri)
                echo "Backuping for Friday"
                cat backup-data | /usr/local/samba/bin/smbclient
                \\\\$1\\$2 -I$3 -N echo "Complete"

                        if [ -d "~agnea1/backup-dir/Friday" ]; then
                                echo "Directory Friday Not found ...
                                making" mkdir
                                ~agnea1/backup-dir/Friday
                        fi
                echo "Archiving ..."
                cd ~agnea1/backup-dir/temp
                tar -cf friday.tar *
                echo "done ..."
                rm ~agnea1/backup-dir/Friday/friday.tar
                mv friday.tar ~agnea1/backup-dir/Friday
                ;;

        *)
                echo "FATAL ERROR: Unknown variable passed for day"
                exit 1;;

esac
###########
```

# 11 Copyright

Cet HOWTO est copyright © 1996-1999 par David Wood. Il peut être reproduit sous quelque forme que ce soit et être distribué gratuitement aussi longtemps que le fichier reste intact, cette notice y compris.

# 12 Remerciements

Brad Marshall (*bmarshall@plugged.net.au*) et Jason Parker (*jparker@plugged.net.au*) ont fourni du temps, de la patience et des scripts.

Adam Neat (*adamneat@ipax.com.au*) à donné le script bash pour sauvegarder une machine Windows sur une machine Linux.

Matthew Flint (*matthew@philtrum.demon.co.uk*) m'explica l'interet de l'option "`interfaces`" dans le `smb.conf`.

Oleg L. Machulskiy (*machulsk@shade.msu.ru*), Jeff Stern (*jstern@eclectic.ss.uci.edu*), Dr. Michael Langner (*langner@fiz-chemie.de*) et Erik Ratcliffe (*erik@caldera.com*) suggererent des modification de la section Partager une imprimant Linux sur une machine Windows.

Alberto Menegazzi (*flash.egon@iol.it*) a fourni la parti sur MagicFilter pour permettre a une machin linux d'acceder a une machine Windows.

Andrea Girotto (*icarus@inca.dei.unipd.it*) a fait beaucoup de suggestions tout au long du document.

Merci aussi au tous les traducteurs qui ont apporté ce HOWTO aux personnes ne parlant pas l'anglais : Takeo Nakano (*nakano@apm.seikei.ac.jp*), Klaus-Dieter Schumacher (*Klaus-Dieter.Schumacher@fernuni-hagen.de*), Andrea Girotto (*icarus@inca.dei.unipd.it*), Mathieu Arnold (*arn_mat@club-internet.fr*) et tous les autres dont je n'ai pas les coordonnées.