AFPS to Samba Migration Guide

Release 1.0

December 15, 2005

1 GETTING STARTED ... 3

1.1 HOW TO USE THIS GUIDE ... 3
1.2 OVERVIEW OF THE MIGRATION PROCESS ... 3
1.2.1 AFPS and Samba cannot run together on same machine ... 4
1.2.2 Preserving user accounts and groups ... 4
1.2.3 Preserving file and printer shares ... 4
1.2.4 Preserving access control information ... 4
1.2.5 Reassigning DOS---- group file and directory ownership ... 4
1.3 SAMBA INSTALLATION ... 5
1.4 SAMBA ADMINISTRATION ... 5

2 AFPS DATA GATHERING ... 7

3 MIGRATING USER ACCOUNTS AND GROUPS ... 8

3.1 IMPLEMENTATION CHOICES ... *8*
3.1.1 Authentication ... 8
3.1.2 Security modes ... 8
3.1.3 Server Configuration ... 9
3.2 REPLACING A STAND-ALONE AFPS SERVER ... *10*
3.3 REPLACING AN AFPS DOMAIN CONTROLLER IN AN NT DOMAIN ... *11*

4 MIGRATING DIRECTORY SHARES ... 17

4.1 ACCESS CONTROL INFORMATION (PERMISSIONS) ... *17*
4.2 REPLACING DOS---- GROUP OWNERSHIP ... *17*
4.3 USER HOME DIRECTORY SHARES ... *18*
4.4 OTHER SHARED DIRECTORIES ... *19*

5 MIGRATING PRINTER SHARES ... 20

APPENDIX A. INSTALLING SAMBA 3 ON SCO OPENSERVER RELEASE 6 ... 22

APPENDIX B. INSTALLING SAMBA 3 ON UNIXWARE 7 ... 24

APPENDIX C. COMPARING AFPS 4.0 AND SAMBA 3.0 ... 26

1 Getting Started

1.1 How to use this guide

The objective of this guide is to assist in the process of migrating from AFPS to Samba. While there are many issues to consider, most of these issues are similar to those encountered by administrators migrating from Windows NT4 to Samba. These common issues are covered in detail in the following Samba documentation and therefore not covered in this guide.

- **The Official Samba-3 HOWTO and Reference Guide**, edited by Jelmer R. Vernooij, John H. Terpstra, Gerald (Jerry) Carter; available at www.samba.org

- **Samba-3 by Example - Practical Exercises in Successful Samba Deployment**, John H. Terpstra; available at www.samba.org

Also, before proceeding with the process of migrating to Samba, you can acquire a good basic understanding of the capabilities of Samba from the following book:

- **Using Samba, 2nd Edition**, by Jay Ts, Robert Eckstein, and David Collier-Brown; available at www.samba.org, and available in print from O'Reilly & Associates

If you are migrating from AFPS to Samba as part of an operating system upgrade, you should reference the appropriate operating system migration guide for details on migrating hardware and software configurations, user data, and UNIX user accounts and groups.

1.2 Overview of the migration process

AFPS and Samba are very similar in that they both provide UNIX file and printer sharing capabilities, using native Microsoft SMB and CIFS protocols, for interoperability with Microsoft operating systems. There are, however, significant differences in their implementations. The purpose of this guide is to help you address those differences as you migrate from an AFPS server to a Samba server.

The following areas need to be considered when migrating from AFPS to Samba:

- preserving user accounts and groups
- preserving file and printer shares
- preserving access control information
- authentication mechanisms

The following sections provide an overview of issues essential to the migration process.

1.2.1 AFPS and Samba cannot run together on same machine

AFPS and Samba both depend on an implementation of the TCP/IP NetBIOS. AFPS uses the NetBIOS provided with the SCO platform, whereas Samba provides its own NetBIOS implementation. Unfortunately, it is not possible to have two NetBIOS implementations running concurrently on the same system. This means that AFPS and Samba cannot run on the same system at the same time.

1.2.2 Preserving user accounts and groups

User account and group information stored in the AFPS SAM database cannot be copied directly from an AFPS to a Samba server. However, Samba is able to retrieve user account and group information from a Windows NT4 Primary Domain Controller (PDC). So, if AFPS is part of a domain that includes a Windows NT4 PDC, or a Backup Domain Controller (BDC) that can be promoted to a PDC, that server can provide an indirect way for Samba to obtain the AFPS user account and group information, including each user's password.

1.2.3 Preserving file and printer shares

There is no process for automatically transferring the file and printer shares. Shares need to be recreated in Samba using the SWAT administration tool, or by editing the *smb.conf* file. However, the task of recreating shares in Samba is minimized by the fact that the default behavior of Samba is to automatically create shares for the user home directories (as defined in */etc/passwd*) and UNIX printers (those that exist in the System V spooler sub-system). Also, a simple UNIX shell script can capture the AFPS share information and provide a quick way for recreating the shares on the Samba server. (See section 2 below.)

1.2.4 Preserving access control information

There is no automated process available for preserving the access control information that limits user access to files, directories, and printer resources. Also, unless the UNIX platform supports POSIX ACLs, Samba itself does not provide ACL support, relying instead on the UNIX file and directory ownership and permissions to control access. Therefore, during the migration process, it is important to pay close attention to the UNIX file and directory permissions.

1.2.5 Reassigning DOS---- group file and directory ownership

AFPS uses the group ownership of files and directories as a way of storing the Windows archive, hidden, and system file attributes. The group ownership is set to one of the *DOS*---- group names, depending on the attributes that have been set. The default behavior of AFPS is to otherwise ignore group ownership, preferring instead to maintain a database of Windows-style ACLs to control group access. In contrast, Samba relies on the group ownership to determine whether a user has access to files and directories that they don't otherwise own. This means that part of the migration process must be to reassign the group ownership for all files and directories that have one of the AFPS *DOS*---- group assignments.

1.3 Samba installation

If you are installing SCO OpenServer Release 6, Samba is automatically installed. However, Samba is not activated until you run **mkdev samba**. It is strongly recommended that you do not attempt to make any changes to the Samba configuration file, *smb.conf*, until after you have activated Samba. Refer to Appendix A, ``Installing Samba 3 on SCO OpenServer Release 6'' for further details. Check the SCO Support Download web site for the latest available version of Samba for OpenServer 6.

For UnixWare 7.1.4, Samba is available as part of the base operating system installation. Updated versions are available on the UnixWare 7.1.4 Maintenance Pack CD's. Installation instructions are provided in the *Release and Installation Notes* that accompany the Maintenance Pack. Refer to Appendix B, ``Installing Samba 3 on UnixWare 7'' for further details.

NOTE: On UnixWare 7, the Samba components are installed in */usr/lib/samba*. Samba utilities such as **smbpasswd** are found in */usr/lib/samba/bin*.

1.4 Samba administration

AFPS server administrators are accustomed to using the Windows NT 4.0 Server Manager or the Windows 2000 Microsoft Management Console to manage shares, view server status (sessions and open files), stop and start services, and view the event logs. This gives a Windows administrator a consistent view of all servers in a domain that includes a mix of AFPS and NT4 servers. For Samba servers, however, use of these tools are limited to creating and managing shares, and viewing lists of sessions and open files. In place of the traditional tools, Samba provides a sophisticated web-based server administration tool called ``SWAT''.

The SWAT tool can be used to administer most aspects of the Samba server, including configuring servers, managing filesystem and printer shares, managing users, starting and stopping the server, and monitoring the status of sessions and open files. It includes a wizard that assists in selecting the required Samba configuration, provides context-sensitive help on each configuration parameter, monitors the current state of connection information, and provides user account password management.

To connect to SWAT, launch a web browser and enter the URL: **http://*hostname*:901**, where ***hostname*** is either the host name of the UNIX system, or ``localhost'' if the browser is running on the same system. When prompted to log in, enter the *root* user name and password.

Configuration changes made using SWAT are stored as updates to the *smb.conf* file. The *smb.conf* file is a text file that can be modified using a text editor. In some situations, it may be more convenient to update *smb.conf* directly, but it is recommended that you use SWAT wherever possible to avoid introducing errors in the file.

AFPS server administrators may also be familiar with the **net** utility, which provides a rich command set similar to that provided on a Microsoft Windows NT server. All server administration functions can be performed using this utility. Samba also provides a **net** utility similar to that provided with AFPS. The Samba **net** utility allows you to perform a number of server administration tasks directly from the UNIX command line.

NOTE: For Samba on SCO OpenServer, the **net** command has been renamed to **smbnet**, to avoid conflict with an existing SCO OpenServer administrative tool.

Another useful Samba administration tool is **pdbedit.** This tool is used to manage the user accounts stored in the SAM-like database.

Details on all of the Samba administration tools can be viewed from the SWAT home page.

2 AFPS data gathering

Because you cannot run AFPS and Samba on the same system, it is necessary to gather some data about AFPS before you replace AFPS with Samba, unless you are installing Samba on a different system than the one currently running AFPS. The following commands enable you to capture information about the AFPS server configuration, to help you set up the new Samba server.

- To capture the AFPS server name and domain name and a list of AFPS shares, run:

```
# mkdir /tmp/afps
# cd /var/opt/lanman/bin
# ./srvconfig -g server,listenname > /tmp/afps/servername
# ./srvconfig -g workstation,domain > /tmp/afps/domainname
# ./lmshare -l | cut -f2- -d' '> /tmp/afps/shares
```

- To capture data about AFPS users and groups, run:

```
# cd /var/opt/lanman/bin
# ./mapuname > /tmp/afps/mapuname
# net users > /tmp/afps/users
# net groups > /tmp/afps/groups
# net localgroups > /tmp/afps/localgroups
```

NOTE: The AFPS server must be running at the time the above **net** commands are issued.

Copy the contents of the */tmp/afps* directory to removable media (e.g. a floppy disk or CD), then restore these files to the */tmp/afps* directory once the UNIX system upgrade is complete so that they can be referenced during the AFPS to Samba migration process.

3 Migrating user accounts and groups

3.1 Implementation choices

3.1.1 Authentication

Samba offers a variety of authentication mechanisms:

- Local authentication against a backend database
  - **smbpasswd** (does not store SAM information)
  - **tdbsam**
  - **ldapsam**
- Remote authentication against an external Samba, Windows NT4, or Windows 200x server

The **tdbsam** option is the closest to the mechanism used by AFPS and is the recommended choice if you wish to continue managing users on the local server. This guide assumes you are adopting the default **tdbsam** implementation.

NOTE: When you run **mkdev samba** to activate Samba on SCO OpenServer Release 6, the default backend is set to **smbpasswd**. You need to modify the *smb.conf* file, or use SWAT, to explicitly set the value to **tdbsam**.

If your organization has adopted LDAP (or is considering a move to LDAP) for user account management, you should consider the **ldapsam** backend option. *The Official Samba-3 HOWTO and Reference Guide* includes instructions for configuring Samba servers to make use of an LDAP server for storing user account information.

If your organization has adopted Microsoft Active Directory for user account management, you may prefer to use the remote authentication mechanism to authenticate users against a Windows 200x server. *The Official Samba-3 HOWTO and Reference Guide* includes instructions for configuring Samba servers to make use of remote authentication servers.

3.1.2 Security modes

Samba implements several “Security Modes”. They are known as: SHARE, USER, DOMAIN, ADS, and SERVER modes. Each has its merits and it is worth reviewing the Samba documentation to determine the best fit for your organization. For the purpose of this document, the USER security mode is the preferred mode since it provides the facilities that most closely emulate the capabilities of AFPS operating as a Primary or Backup Domain Controller.

NOTE: The DOMAIN security mode does not set up Samba to operate as a Domain Controller. The DOMAIN security mode actually means that the Samba server passes all authentication requests through to a separate Domain Controller. In other words, this configuration makes the Samba server a Domain Member server.

The USER security mode alone does not define the role of the server as a Primary or Backup Domain Controller. Additional parameters must be set to configure Samba for the role of the Domain Controller.

3.1.3 Server Configuration

The platform-specific instructions for installing and configuring your Samba server are provided in the appendices of this document. Following these instructions results in the creation of an *smb.conf* file which may require further modification to support the migration of users from a Windows server.

To support user account migration, you need to ensure that the following *smb.conf* parameters have been correctly defined:

```
[global]
      netbios name = SERVER_NAME
      workgroup = DOMAIN_NAME
      domain master = {yes for PDC, no for BDC}
      security = user
      domain logons = yes
      passdb backend = tdbsam
```

NOTE: When you run **mkdev samba** to activate Samba on SCO OpenServer Release 6, you are asked to enter the *SERVER_NAME* and *DOMAIN_NAME*, and whether the server is to participate in an existing domain or is to be a server in a new domain (select this option when replacing a standalone AFPS server). Providing the appropriate answers to these questions may not result in the correct settings, so you should check the contents of *smb.conf* to ensure that the correct settings have been applied.

You can easily configure these parameters using the SWAT configuration Wizard:

1. Click on the **Wizard** button on the main button bar, then click on the **Domain Controller** server type radio button and press the **Commit** button.
2. Press the **Edit Parameter Values** button to display a form for entering the **netbios name** (*SERVER_NAME)* and **workgroup** (*DOMAIN_NAME)* information. Also, set the **passwd backend** parameter for the required authentication mechanism (see Section 3.1.1, Authentication).
3. Once you have entered the required information, press the **Commit Changes** button.

4. Next, click on the **Globals** button and verify that all the parameters listed above have been set correctly. Make any necessary adjustments and then press the **Commit Changes** button to save your changes.
5. Click on **Status** on the main button bar and restart the Samba server using the **Restart All** button.

3.2 Replacing a stand-alone AFPS server

Migration from AFPS to Samba cannot leverage the Samba migration tools unless a Windows NT4 or 200x system can be temporarily added to the network to act as an intermediary during the migration process.

If a Windows NT4 or 200x system is available, it must be added to the AFPS server domain as a Backup Domain Controller. It can then be promoted to the role of Primary Domain Controller using the NT Server Tools. Refer to the AFPS documentation for adding a BDC to the AFPS server domain. To complete the migration to Samba, refer to the ``Replacing an AFPS domain controller in an NT domain'' section, later in this document.

If a Windows NT4 or 200x system is not available, you will need to install Samba as a Primary Domain Controller for a new domain. You then need to set up each UNIX user as a Samba user by running the following command:

```
# smbpasswd –a username
```

where ***username*** is an existing UNIX user account name. You are prompted to specify a new password for each user.

The first user account to add should be the *root* user, or another UNIX user with *root* privileges. You may later be asked to specify this user account when adding a workstation to the domain.

If you have users connecting from Windows NT4, 200x, or XP Professional workstations, you need to create machine accounts for each system. Each machine account requires a UNIX user account with the name of the machine followed by ‘$’. For example, if the workstation is named *wkstat*, you need to create a UNIX user account with the name *wkstat$* using the command:

```
# useradd -g machines -d /var/nobody -c "comment" -s /bin/false wkstat$
```

where ***machines*** is a group created previously using the command:

```
# groupadd machines
```

Add the machine account to the Samba user account database using the command:

# pdbedit -a -m –u ***wkstat$***

Once the user accounts and machine accounts have been created, the users should be able to log in to the Samba server from their workstations. They must specify the new password assigned when their Samba account was created by the **smbpasswd** command. However, if they encounter a message similar to the following:

```
Windows cannot connect to the domain, either because the domain
controller is down or otherwise unavailable, or because your
computer account was not found
```

then it will be necessary for someone to log in to the workstation using the local administrator account and repeat the process of adding the workstation to the domain. This is necessary to establish the machine trust relationship with the Samba server.

Once the user has successfully logged in to their workstation, they can change their password using the normal Windows password management tools. (In the case of Windows XP, the **Change Password** option is provided in the popup window that appears when the user presses <Ctrl><Alt><Delete>.)

Chapter 6, ``Domain Membership'' in *The Official Samba-3 HOWTO and Reference Guide* includes more detailed instructions for adding workstations to domains.

3.3 Replacing an AFPS domain controller in an NT domain

The Official Samba-3 HOWTO and Reference Guide includes a section on migrating to Samba from an NT4 server-based domain. Unfortunately, incompatibilities between AFPS and Samba prevent the direct migration of user accounts from an AFPS Primary Domain Controller to a Samba Domain Server. However, if AFPS is the PDC in a domain with NT4 Domain Controllers, one of the NT4 servers can be promoted to the role of PDC, enabling users and groups to be migrated using the steps provided in the *Samba HOWTO.* You can change the role of the domain controller using the Server Manager For Domains, supplied in the NT Server Tools package provided with Windows NT4, Windows 200x, and AFPS.

Chapter 33, "Migration from NT4 PDC to Samba-3 PDC", in *The Official Samba-3 HOWTO and Reference Guide* provides instructions for migrating to Samba from an NT4 server-based domain. To prepare for this migration process, you need to do the following:

- Create group mappings for each of the built-in and well-known domain groups, as per the instructions in *The Official Samba-3 HOWTO and Reference Guide.* For your convenience, the following script, **initGroups.sh**, can be used to create the required UNIX groups and group mappings:

initGroups.sh:

```
#!/bin/sh

if [ -f /bin/smbnet ]
then
        NET=/bin/smbnet
        NOGROUP=nogroup
else
        NET=/usr/lib/samba/bin/net
        NOGROUP=nobody
fi

$NET groupmap cleanup

# First assign well known domain global groups
groupadd users
$NET groupmap modify ntgroup="Domain Admins" unixgroup=root
$NET groupmap modify ntgroup="Domain Users"  unixgroup=users
$NET groupmap modify ntgroup="Domain Guests" unixgroup=$NOGROUP

# Add the group for the machine accounts
groupadd machines
$NET groupmap add ntgroup="Domain Controllers" unixgroup=machines type=d rid=515
$NET groupmap add ntgroup="Domain Computers" unixgroup=machines type=d rid=516

# Add other Windows 2000 specific groups
$NET groupmap add ntgroup="Cert Publishers" unixgroup=$NOGROUP type=d rid=518
$NET groupmap add ntgroup="Schema Admins" unixgroup=$NOGROUP type=d rid=519
$NET groupmap add ntgroup="Enterprise Admins" unixgroup=$NOGROUP type=d rid=520

# Assign builtin local groups
$NET groupmap modify ntgroup="Administrators" unixgroup=root
$NET groupmap modify ntgroup="Users" unixgroup=users
$NET groupmap modify ntgroup="Guests" unixgroup=$NOGROUP

# Now create the operator groups, including the associated UNIX group
groupadd operator
$NET groupmap modify ntgroup="Account Operators" unixgroup=operator
$NET groupmap modify ntgroup="System Operators" unixgroup=operator
$NET groupmap modify ntgroup="Print Operators" unixgroup=operator
$NET groupmap modify ntgroup="Backup Operators" unixgroup=operator
$NET groupmap modify ntgroup="Replicators" unixgroup=operator

# Finally, add any of your own domain global and local groups
# groupadd engineer
# $NET groupmap add ntgroup="Engineering Team" unixgroup=engineer type=d rid=1114

$NET groupmap list
```

As described in the Samba documentation, execute the **initGroups.sh** script at the appropriate step in the NT4 to Samba migration process.

- Create a directory named */etc/samba/scripts* and populate it with the following scripts:

  adduser.sh:

```
#!/bin/sh

# This script is called by Samba to create UNIX user accounts
# when importing Windows user accounts and machine accounts from
# a Primary Domain Controller

# Modify following variables as needed
USER_GROUP=users

# Locate the Samba configuration files
if [ -f /etc/samba/smb.conf ]
then
        # OpenServer
        ETCSAMBA=/etc/samba
else
        # UnixWare
        ETCSAMBA=/usr/lib/samba/private
fi
SMBUSERS_FILE=$ETCSAMBA/smbusers

UNIXUSER=`echo $1 | tr [:upper:] [:lower:] | tr -d '_ '`
NUMCHARS=`expr "$UNIXUSER" : '.*'`
if [ "$UNIXUSER" != "$1" -o $NUMCHARS -gt 8 ]
then
        echo Windows user name too long or includes invalid characters
        exit 99
fi

# If the UNIX user does not exists, add the user
userls -l $UNIXUSER > /dev/null 2>&1
if [ $? -ne 0 ]
then
        useradd -d /tmp -g $USER_GROUP -G $USER_GROUP -s /bin/false $UNIXUSER
fi

exit 0
```

  NOTE: Former AFPS user accounts, created using UNIX administration tools such as the SCOAdmin Account Manager, should migrate to Samba without any problems. However, if the user accounts were created using Windows NT administration tools, such as the NT Server Tools User Manager, those names that do not conform to the restrictions on UNIX account names will not migrate correctly. UNIX accounts are limited to 8 lowercase alpha or numeric characters; other characters allowed in Windows user names, such as space and underscore, are not permitted in UNIX user account names.

addgroup.sh:

```
#!/bin/sh
# Convert group name to lowercase and remove spaces and underscores
UNIXGROUP=`echo $1 | tr [:upper:] [:lower:] | tr -d '_ '`
NUMCHARS=`expr "$UNIXGROUP" : '.*'`
if [ "$UNIXGROUP" != "$1" -o $NUMCHARS -gt 8 ]
then
        # Windows group name includes invalid characters or is too long.
        exit 99
fi

# If the UNIX group does not exists, add the group
groupls -l $UNIXGROUP > /dev/null 2>&1
if [ $? -ne 0 ]
then
        groupadd $UNIXGROUP
fi

# Echo the GID
GID=`cat /etc/group | grep $UNIXGROUP | cut -d ":" -f3`
echo $GID

exit 0
```

NOTE: Windows group names that do not conform to the restrictions on UNIX group names will not migrate correctly. UNIX group names are limited to 8 alpha or numeric characters; other characters allowed in Windows group names, such as space and underscore, are not permitted in UNIX group account names

The above scripts can be downloaded from: http://www.sco.com/support/docs/openserver/600/migration/afps-samba-scripts.zip. You may need to customize these scripts as appropriate to your requirements. For example, depending on your locale, you may need to add a line with: “`LANG=C;export LANG`”.

Ensure that each script can be executed by Samba using the command:

```
# chmod 755 scriptname
```

- Reference the scripts in the `[global]` section of the Samba configuration file, *smb.conf*, as follows:

```
[global]
      add user script = /etc/samba/scripts/adduser.sh "%u"
      add group script = /etc/samba/scripts/addgroup.sh "%g"
```

The following sequence of commands has proven to be effective for migrating former AFPS users and groups to a Samba server via a Windows NT4 or Windows 2000 server:

```
# net rpc getsid –SPDC_name -UAdministrator

# net rpc join –SPDC_name –wDomain_name -UAdministrator

# /etc/samba/scripts/initGroups.sh

# net rpc vampire –SPDC_name -UAdministrator
```

where *PDC_name* and *Domain_name* must be replaced with values appropriate to your installation.

NOTE: On OpenServer 6, use **smbnet** in place of **net** when typing the above commands.

For a more detailed explanation of these commands, please refer to Chapter 33, "Migration from NT4 PDC to Samba-3 PDC", in *The Official Samba-3 HOWTO and Reference Guide.*

Also, you need to add a user account for the *root* user, or another UNIX user with *root* privileges. You may later be asked to specify this user account when adding a workstation to the domain.

Finally, if you have users connecting from Windows NT4, 200x, or XP Professional workstations, you need to create machine accounts for each system. Each machine account requires a UNIX user account with the name of the machine followed by '$'. For example, if the workstation is named *wkstat*, you need to create a UNIX user account with the name *wkstat$* using the command:

```
# useradd -g machines -d /var/nobody -c "comment" -s /bin/false wkstat$
```

where ***machines*** is the machine group created by the initGroups.sh script.

Add the machine account to the Samba user account database using the command:

```
# pdbedit -a -m –u wkstat$
```

Once the user accounts and machine accounts have been created, the users should be able to log in to the Samba server from their workstations using the passwords originally assigned in AFPS. However, if they encounter a message similar to the following:

```
Windows cannot connect to the domain, either because the domain
controller is down or otherwise unavailable, or because your
computer account was not found
```

then it is necessary for someone to log in to the workstation using the local administrator account and repeat the process of adding the workstation to the domain. This is necessary to establish the machine trust relationship with the Samba server.

Once the user has successfully logged in to their workstation, they can change their password using the normal Windows password management tools. (In the case of Windows XP, the **Change Password** option is provided in the popup window that appears when the user presses <Ctrl><Alt><Delete>.)

Chapter 6, ``Domain Membership'', in *The Official Samba-3 HOWTO and Reference Guide* includes more detailed instructions for adding workstations to domains.

4 Migrating directory shares

4.1 Access control information (permissions)

AFPS implements the NT-style Access Control Lists (ACLs) for shares, files, directories, and printers. The ACLs are maintained in a database and used to determine a user's access rights to the resource. The ACLs override the UNIX filesystem permissions unless AFPS is configured to enforce UNIX permissions.

In contrast, Samba honors UNIX filesystem access controls. Windows users who log in to a Samba server are mapped to a UNIX user. The UNIX filesystem then determines whether that user should be given access to selected files and directories. Unless the UNIX platform supports POSIX ACLs, Samba itself does not provide ACL support.

When creating and configuring shares, the administrator can employ options that override the native filesystem permissions and behaviors. For example, read and write access can be limited to specific users or groups of users. These options can be used to provide behavior that is close to that expected of a Windows NT server.

When migrating from AFPS to Samba, the ACL information is lost and this can result in Windows users having access to files and directories that were previously restricted. It is necessary, therefore, to carefully review the access permissions on all files and directories previously shared by AFPS before they are made available as Samba shares.

Refer to Chapter 14, ``File, Directory, and Share Access Controls'', in *The Official Samba-3 HOWTO and Reference Guide* for further information.

4.2 Replacing DOS---- group ownership

When creating directory shares under Samba, if the directory was previously shared by AFPS, you must ensure that files with *DOS----* group ownerships are updated with the correct UNIX group ownership so that Samba applies the appropriate access control for the share. There are eight of these AFPS groups: *DOS----*, *DOS-a--*, *DOS--s-*, *DOS---h,* *DOS-as-*, *DOS-a-h, DOS—sh,* and *DOS-ash*.

If the shared directory contains files owned by a single user – for example, the contents of a user's home directory– changing the group ownership of the files can be achieved simply by using the commands:

```
# cd directory
# chgrp –R groupname *
```

If the shared directory contains read-only files shared by multiple users – for example, shared applications – you need to select a group to which all relevant users belong (you may need to create a group and add users to it as needed), then use the above commands to assign group access to the shared files in that directory. You may also need to adjust the group file ownership and permissions to allow different groups of users to access specific files and directories.

If the directory contains files created and shared by multiple users, the problem becomes more complex. The simplest option is to proceed as described for the read-only share. (Also, consider using the **force group** parameter when defining the share, so that the same group ownership is used for future files and directories.) Alternatively, you can set the group ownership of each file or directory to be the primary group of the user owning that file or directory. This second option takes more effort to implement and could result is users being denied access to a file or directory where an AFPS ACL would previously have allowed access.

Once you have reassigned the group ownership of all files and directories with *DOS*---- group ownership, remove the AFPS *DOS*---- groups using the following commands:

```
# groupdel DOS----
# groupdel DOS-a--
# groupdel DOS--s-
# groupdel DOS---h
# groupdel DOS-as-
# groupdel DOS-a-h
# groupdel DOS--sh
# groupdel DOS-ash
```

4.3 User home directory shares

With the `[homes]` share defined in *smb.conf*, the behavior of Samba is to automatically make each user's home directory available as a share. If this is the desired behavior, then there is no need to create those shares. Note that a user’s home directory appears as a share only to the user who owns that directory. In other words, users logged on to the server do not see the shares relating to home directories of other users.

The following entry in the *smb.conf* file enables the automatic sharing of all user home directories:

```
[homes]
     comment = Home Directories
     browseable = no
     writable = yes
```

If this is not the desired behavior, you can disable this feature by removing the `[homes]` entry from *smb.conf*. You can then manually create specific shares for each user's home directory, using entries similar to the following example:

```
[fred]
     comment = Fred's share
     path = /home/fred
     valid users = fred
     writable = yes
```

The easiest way to create and administer Samba shares is using the SWAT utility.

4.4 Other shared directories

Directories shared by groups of users need to be created manually. This can be done by adding the share details to the *smb.conf* file, similar to the following examples:

```
[Public]
      comment = Public Share
      path = /home/Public
      force group = users
      read only = No

[Apps]
      comment = Shared Applications
      path = /home/Apps
      read only = yes
```

Alternately, you can use the Samba web-based administration tool, SWAT, to create the shares.

Samba provides a very rich selection of parameters that can be used to control and manage shares. The **Shares** section of the SWAT utility provides basic and advanced views of the available parameters, along with detailed help. Further information on configuring shares can be found in Chapter 8, ``Advanced Disk Shares'', in *Using Samba, 2nd Edition*.

5 Migrating printer shares

With the `[printers]` share defined in smb.conf, the expected behavior of Samba is to automatically share printers that are available in the UNIX System V spooler sub-system. In general, if you wish to have all of the UNIX printers appear as Samba shares, you should not need to create printer-specific shares.

For Samba on SCO OpenServer Release 6, the UNIX printers automatically become visible to Windows users connecting to the server. With Samba version 3.0 on UnixWare 7, however, the printers are recognized only if they have been defined in */etc/printcap*. On UnixWare, for each printer that you want Samba to make available as a shared printer for Windows clients, you need to add an entry of the following format to the file */etc/printcap*:

{***printname***}|{***Printer_description***}:

For example:

laserjet|HP Laserjet Printer:

Also, you need to add the following parameters to the `[global]` section of the *smb.conf* file:

```
printcap = /etc/printcap
printing = sysv
```

This tells Samba to use UNIX System V style-print commands but to look for the list of printers in the BSD */etc/printcap* file.

If you have installed and activated the optional CUPS-style of printing on your UnixWare system, you need to define the following parameters in the `[global]` section of the *smb.conf* file:

```
printcap = /etc/printcap
printing = cups
```

Activating CUPS on UnixWare automatically creates entries for each printer in */etc/printcap*, so Samba is directed to look in this file for the printer definitions. Also, Samba must be instructed to use the CUPS-style print commands to access the printers.

NOTE: Once you have installed and activated CUPS on UnixWare, you also have the option of printing directly using CUPS protocol rather than accessing them as Samba shares. This requires that you install additional CUPS software on the Windows clients.

If you prefer not to share all of the UNIX printers, or prefer to restrict access on a per-user basis, you can manually create entries for each printer in the *smb.conf* file.

NOTE: If you encounter problems with permissions when accessing printers, check that the directory */usr/spool/samba* exists and that it has read, write, and execute permissions for all users by running the following:

```
# mkdir /usr/spool/samba
# chmod ugo+rwx /usr/spool/samba
```

Refer to *The Official Samba-3 HOWTO and Reference Guide* for further details on configuring printers.

Appendix A. Installing Samba 3 on SCO OpenServer Release 6

With SCO OpenServer Release 6, if you are using a Maintenance Pack prior to Maintenance Pack 2, you must install SLS OSS701A, The Account Management Supplement, in order to create machine accounts as described in this document. OSS701A is available at: http://www.sco.com/support/update/download/release.php?rid=131.

Samba is installed by default as part of SCO OpenServer Release 6, but it is not activated. After installation, you must configure and activate it. To do so, run the command **mkdev samba**. When prompted, enter the appropriate information as follows:

Workgroup Name/NT-Domain

> Default: ``WORKGROUP''

Machine name

> Default: your system's name, in capital letters, as reported by the **uname -n** command – for example, ``MYSYSTEM''.

Windows Internet Naming Service (WINS) usage

> Specify whether or not your network uses the Windows Internet Naming Service (WINS). The use of WINS is recommended; failure to do so may significantly increase your network traffic. However, some sites are unable to use it because of security policies or other considerations.
>
> If you are using WINS, you are asked whether your machine is intended to be the WINS server for your network. If another machine will be the WINS server, you are asked for its IP address.
>
> **NOTE:** A network may not have more than one WINS server. If any other machine is acting as the WINS server, your machine cannot do so.

Network interface(s)

> Specify the network interface(s) over which Samba will run. If your system has only one Network Interface Card, then this is configured automatically. If you have multiple NICs, then you need to enter the interfaces – for example, ``net0'', ``net1''. The loopback device (``lo0'') is added automatically.

Microsoft Security Domain or Active Directory

Specify whether your system is being installed into an already-existing Microsoft Security Domain or Active Directory.

If **Yes**, you are asked for the name of the Primary Domain Controller.

If **No**, you are asked whether your system is to be the Primary Domain Controller in a new domain.

Based on the answers to these questions, an initial *smb.conf* file is created. This file should work in most circumstances. Additional configuration should be completed using the SWAT utility.

Once this initial configuration is complete, Samba is automatically launched at boot-time.

Check the SCO Support Download web site for the latest available version of Samba for OpenServer 6.

Appendix B. Installing Samba 3 on UnixWare 7

Samba is available as part of the base UnixWare 7.1.4 operating system installation. Updated versions are available on the UnixWare 7.1.4 Maintenance Pack CD's. Follow the instructions in the Maintenance Pack *Release and Installation Notes* to install the Samba package.

Note the following when installing Samba on UnixWare 7:

- If you are upgrading from a previous release of Samba on UnixWare 7, save a copy of your existing */usr/lib/samba/lib/smb.conf* file before you begin the installation, so that you can restore any settings that might be affected by the upgrade.

- By default, */tmp* is automatically shared. This can be a security concern, since various system utilities keep temporary data in */tmp*. To remove the */tmp* share, log in to SWAT and select the **Shares** icon. On the next screen, highlight the **tmp** share in the list box and select the **Delete Share** button.

- Samba cannot run together with AFPS, or with the NetBIOS driver enabled, so you must first determine if either AFPS or NetBIOS is running and disable each as necessary. To determine if AFPS or NetBIOS are running, log in as the *root* user and run:

  ```
  # cd /etc/rc2.d
  ```

  Determine if either of the following files exist in this directory

  S74netbios
  S99ms_srv

  If one or both of these files exist, enter the appropriate command or commands shown below:

  ```
  # mv S74netbios s74netbios
  # mv S99ms_srv s99ms_srv
  ```

  Then reboot the system:

  ```
  # shutdown -i6 -g0 –y
  ```

To start Samba, enter:

```
# /etc/init.d/samba start
```

To stop Samba, enter:

```
# /etc/init.d/samba stop
```

To enable Samba to automatically start whenever the system boots, type:

```
# /etc/init.d/samba enable
```

To disable the automatic startup, type:

```
# /etc/init.d/samba disable
```

Samba is configured with the SWAT (Samba Web Administration Tool) utility using a web browser. To start SWAT:

1. Enter:

   ```
   # /usr/lib/samba/sbin/swat
   ```

2. Launch a web browser and enter the URL **http://*hostname*:901,** where ***hostname*** is either the host name of the UnixWare system, or ``localhost'' if the browser is running on the same system.

3. When prompted to login, enter the *root* user name and password.

4. Select the **Status** icon to start the Samba daemons.

Localization settings are accessed from the SWAT home page by clicking on the **Globals** tab, and then selecting **Advanced View**. Set appropriate values for your locale for the client code page, the character encoding system, and the other options. (Each option has context-sensitive help.)

Samba documentation and manual pages are available from the *DOS and Windows* category in the online documentation. Note that the SWAT home page also provides direct links to the Samba documentation that is included on the UnixWare 7 system.

Appendix C. Comparing AFPS 4.0 and Samba 3.0

Function	AFPS 4.0	Samba 3.0
Installation	AFPS is available as a **custom**-installable package for SCO OpenServer and as a **pkgadd**-installable package for UnixWare 7. During installation, the software prompts the user for basic configuration information, such as the server and domain names, and the administrator's password. It also asks for the server role (PDC or BDC) and automatically joins an existing domain if BDC is selected.	Samba is available as a **custom**-installable package for SCO OpenServer, and as a **pkgadd**-installable package for UnixWare 7. Configuration information is entered post-installation. The default configuration assumes that the server name is the same as the UNIX host name, the domain name is set to ``MYGROUP'' and the server role is that of a stand-alone server.
Configuration	In general, it is not necessary to change any AFPS configuration parameters since the default configuration is suitable to most operations as a domain controller. If the configuration does need to be changed, AFPS maintains an extensive range of configuration options in an NT-style registry that can be viewed and modified using the Windows NT4/2000/2003/XP **regedit** tool or the AFPS **regadm** UNIX command line utility.	Samba configuration information is held in a text file (*smb.conf*) that can be edited using a web-based configuration tool called SWAT (Samba Web Administration Tool), or using a text editor. The default configuration file includes a small subset of the possible configuration options. There is an extensive range of configuration options that can be added as needed for tasks such as changing the server name, controlling resource access, or joining an NT4-style domain. All available options are shown when you select the advanced view in SWAT.

System Requirements	AFPS requires that the host UNIX system provides TCP/IP networking and a TCP NetBIOS.	Samba requires that the host UNIX system provides TCP/IP networking. Samba provides its own TCP NetBIOS implementation.
Software Licensing	Purchasers of the AFPS product receive a Certificate of License and Authenticity (COLA) with printed license details that allow the server to run with a 5-user base license. Additional licenses are available in increments of 10-user, 25-user, and unlimited-user. You must install a 5-user base license before installing these additional licenses.	Samba is an open source product and is available at no cost.
NT Domain Support	AFPS can function fully as a Microsoft NT4-style Primary Domain Controller (PDC) or Backup Domain Controller (PDC).	Samba has the ability to function as a Microsoft Windows NT4-style Domain Controller. It can establish a secure connection with a PDC running Microsoft Windows and obtain a copy of the SAM (user accounts and groups) database, but does not provide the SAM replication capabilities of a true Domain Controller. This means that Samba cannot operate as a BDC when the PDC is either a Microsoft Windows NT4 or SCO AFPS 4.0 server, nor can it replicate account data to NT4 or AFPS BDCs. Samba can, however, function as a PDC to multiple Samba BDCs.
Client Support	AFPS 4.0 supports the following clients: Windows 3.1.1, 95, 98, 98 SE, Me, and NT4 Workstation	Samba 3 supports Windows 9x/Me and Windows NT/200x/XP.

Active Directory Support	AFPS does not support Active Directory, but it can exist as an NT4-style BDC in an Active Directory-based network where a Windows 2000 or 2003 server is running in mixed or native mode and acting as a NT4-style PDC.	Samba has the ability to join an Active Directory Domain and use Kerberos protocols to authenticate users. This enables the Samba server to become a Domain Member server in an Active Directory Domain. Samba cannot, however, be a Domain Controller within an Active Directory tree, nor can it be an Active Directory server. This means that Samba cannot act as a Backup Domain Controller to an Active Directory Domain Controller.
User Administration	AFPS user accounts can be created using the NT4 User Manager for Domains or the AFPS **net user** command line utility. An equivalent UNIX user is automatically created when an AFPS user account is created. Also, AFPS user accounts can be created automatically when using the SCOadmin **User Manager** to create UNIX users. AFPS stores user account information in an NT-style SAM (Security Account Management) database.	Samba provides the **smbpasswd** and **pdbedit** command line tools for creating and managing user accounts. The Samba web-based SWAT tool also provides functionality for creating and managing users. However, a UNIX user needs to have been created before the equivalent Samba user account can be created. A UNIX user does not become a Samba user until that user has been added to the Samba *passdb* backend database, using one of these tools. The Samba *passdb* backend database can be a text file, a SAM-like database, an LDAP directory, or an SQL database. LDAP offers the best solution for central administration of multiple Samba servers and is the most suitable solution for heterogeneous environments. However, the SAM-like database (tdbsam) is the closest in functionality to AFPS and is the suggested default for migrations.

Server Administration	AFPS is designed to be administered using the Windows NT 4.0 Server Manager, and the Windows 2000 Microsoft Management Console. These tools can be used to manage shares, view the server status (sessions and open files), stop and start services, and view the event logs. This gives a Windows administrator a consistent view of all servers in a domain that includes a mix of AFPS and NT4 servers. AFPS also provides a **net** utility, with a rich command set, that should be familiar to an NT server administrator accustomed to administering the server from the command line. All server administration functions can be performed using this utility. AFPS is integrated into the SCOadmin **Filesystem Manager** for the purpose of creating and managing shares.	Samba provides a sophisticated web-based server administration tool called SWAT. This tool can be used to administer most aspects of the Samba server including configuring servers, managing filesystem and printer shares, managing users, starting and stopping the server, and monitoring status of sessions and open files. Samba also provides a **net** utility, similar to the **net** utility provided on Windows systems. The **net** utility enables a number of server administration tasks to be performed from the UNIX command line. The Samba server can also be administered using the Windows NT 4.0 Server Manager and the Windows 2000 Microsoft Management Console. Using these tools, administrative tasks are limited to creating and managing shares and viewing lists of sessions and open files.
User Authentication	Windows clients may use clear-text strings for simple password-based authentication or encrypted passwords as part of a challenge/response authentication model (NTLMv1 and NTLMv2). Users are authenticated against encrypted password information stored in an NT4-style SAM database. AFPS does not support Active Directory and Kerberos authentication.	Windows clients may use clear-text strings for simple password-based authentication or encrypted passwords as part of a challenge/response authentication model (NTLMv1 and NTLMv2). Users are authenticated against password information stored in the Samba *passdb* backend database, which can be a text file, a SAM-like database, an LDAP directory, or an SQL database. Samba can also authenticate users against Active Directory and Kerberos.

File and Directory Access Control	AFPS implements NT-style Access Control Lists (ACLs) for shares, files, directories, and printers. For the Windows user and server administrator, the ACL behavior is identical to that of an NT server. ACLs are maintained in a database and used to determine a user's access rights to the resource. The ACLs override the UNIX filesystem permissions unless AFPS is configured to enforce UNIX permissions.	Samba honors UNIX filesystem access controls. Windows users who log in to a Samba server are mapped to a UNIX user. The UNIX user identity is then used to determine whether or not the user should be given access to selected files and directories. Samba does not implement fine-grained access control, as provided by NT servers. However, on UNIX platforms that have native support for POSIX ACLs, the Windows ACLs can be mapped to their equivalent POSIX ACLs. In the absence of POSIX ACLs, access control is limited to the capabilities of the native UNIX filesystem. When creating and configuring shares, the administrator can employ options that override the native filesystem permissions and behaviors. For example, read and write access can be limited to specific users or groups of users. These options can be used to provide behavior that is close to that expected of a Windows NT server.
UNIX Filesystem Integration	AFPS provides an extensive range of features and options to support the integration of UNIX and Windows filesystems, including hidden files, UNIX links, file permissions, name mangling, case sensitivity of filenames, file locking, and opportunistic locking (oplocks).	Samba provides an extensive range of features and options to support the integration of UNIX and Windows filesystems, including hidden files, UNIX links, file permissions, name mangling, case sensitivity of filenames, file locking, and opportunistic locking (oplocks).

Printing Support	The AFPS administrator can make any printer in the System V printer subsystem available as a shared resource to Windows clients. AFPS printer shares can be created and managed using the **net** command line utility or the SCOadmin **Printer Manager**. AFPS includes support for printing from UNIX to printers attached to remote Windows clients. AFPS supports the native Windows NT RPC-based printer interface (SPOOLSS), including the mechanism for Windows clients to automatically download the printer driver from the server when adding a new printer.	Samba can automatically offer any configured UNIX printer as a shared resource available to Windows clients. Samba supports a variety of UNIX printer subsystems including BSD, System V, and CUPS. Samba includes support for printing from UNIX to printers attached to remote Windows clients. Samba supports the native Windows NT RPC-based printer interface (SPOOLSS), including the mechanism for Windows clients to automatically download the printer driver from the server when adding a new printer.
Name Resolution	On UnixWare 7, AFPS can perform host name resolution using all of the following methods: • Querying DNS servers • Using the */etc/hosts* file • Querying WINS servers • Using an LMHOSTS file • Performing broadcast name resolution On UW7, AFPS also provides a fully NT4-compatible WINS server, including the ability to synchronize WINS databases. On SCO OpenServer, AFPS can perform host name resolution using only the broadcast and LMHOSTS methods.	Samba can perform host name resolution using all of the following methods: • Querying DNS servers • Using the */etc/hosts* file • Querying WINS servers • Using an LMHOSTS file • Performing broadcast name resolution Samba can also be set up as a WINS server, but it cannot communicate with Windows WINS servers in order to synchronize with their WINS databases.

Browsing	AFPS has full support for browsing and can participate as a master browser, a backup browser, a domain master browser, a potential browser, or a server that doesn't participate in browsing elections.	Samba has full support for browsing and can participate as a master browser, a backup browser, a domain master browser, a potential browser, or a server that doesn't participate in browsing elections.
Inter-domain Trust Relationships	AFPS can participate in NT4-style trust relationships, which means that users from one domain may be given access rights and privileges in another domain.	Samba can participate in NT4-style trust relationships, which means that users from one domain may be given access rights and privileges in another domain. However, trust relationship support in Samba is at an early stage of development, so it may not always function correctly.
Distributed Filesystem Support	AFPS does not support Microsoft DFS.	A Samba server can host a Microsoft Distributed Filesystem (DFS) tree. DFS provides a means of separating the logical view of files and directories that users see from the actual physical locations of these resources on the network. It allows for higher availability, smoother storage expansion, and load-balancing. DFS trees on Samba work with all DFS-aware Windows clients.
National Language Support	AFPS on SCO OpenServer is available in English, French, and German. The NT Server Tools are also available in these languages. The *Administrator's Handbook* and the manual pages have been translated into French and German. AFPS on UnixWare 7 only provides documentation in English, but the NT Server Tools are available in English, French, and German.	SWAT can be configured to display messages to match the settings of the language configurations of the Web browser. Languages supported include English, French, German, Italian, Polish, Dutch, Japanese, and Turkish. Some portions of the Samba HOWTO documentation are available on the Samba web site in French and German.

	AFPS supports the UNICODE multi-byte character set encoding scheme ``on the wire’’. Options are available for converting filenames through a DOS code page to equivalents that can be represented by UNIX character sets other than US English.	Samba supports the UNICODE multi-byte character set encoding scheme ``on the wire’’. Options are available for converting filenames through a DOS code page to equivalents that can be represented by UNIX character sets other than US English.
Documentation	The following online documentation is included with AFPS: • *Installation and Release Notes* • *Quick Start Guide* (UnixWare 7) • *Administration Guide* (UnixWare 7) • *Administrator's Handbook* (SCO OpenServer) • *System Guide* (SCO OpenServer) • *Concepts and Planning Guide* • *Programmers API Reference* • Manual pages for UNIX commands • Reference pages readable with **net help** command	The following online documentation is included with Samba: • *Using Samba, 2ed*. - by Jay Ts, Robert Eckstein, and David Collier-Brown • *The Samba HOWTO Collection* • Manual pages for UNIX