

NAME

smbd - server to provide SMB/CIFS services to clients

SYNOPSIS

```
smbd [-D] [-F] [-S] [-i] [-h] [-V] [-b] [-d <debug level>] [-l <log directory>] [-p <port number(s)>] [-O
<socket option>] [-s <configuration file>]
```

DESCRIPTION

This program is part of the **samba**(7) suite.

smbd is the server daemon that provides filesharing and printing services to Windows clients. The server provides filespace and printer services to clients using the SMB (or CIFS) protocol. This is compatible with the LanManager protocol, and can service LanManager clients. These include MSCLIENT 3.0 for DOS, Windows for Workgroups, Windows 95/98/ME, Windows NT, Windows 2000, OS/2, DAVE for Macintosh, and smbfs for Linux.

An extensive description of the services that the server can provide is given in the man page for the configuration file controlling the attributes of those services (see **smb.conf**(5)). This man page will not describe the services, but will concentrate on the administrative aspects of running the server.

Please note that there are significant security implications to running this server, and the **smb.conf**(5) manual page should be regarded as mandatory reading before proceeding with installation.

A session is created whenever a client requests one. Each client gets a copy of the server for each session. This copy then services all connections made by the client during that session. When all connections from its client are closed, the copy of the server for that client terminates.

The configuration file, and any files that it includes, are automatically reloaded every minute, if they change. You can force a reload by sending a SIGHUP to the server. Reloading the configuration file will not affect connections to any service that is already established. Either the user will have to disconnect from the service, or **smbd** killed and restarted.

OPTIONS

-D

If specified, this parameter causes the server to operate as a daemon. That is, it detaches itself and runs in the background, fielding requests on the appropriate port. Operating the server as a daemon is the recommended way of running **smbd** for servers that provide more than casual use file and print services. This switch is assumed if **smbd** is executed on the command line of a shell.

-F If specified, this parameter causes the main **smbd** process to not daemonize, i.e. double-fork and disassociate with the terminal. Child processes are still created as normal to service each connection request, but the main process does not exit. This operation mode is suitable for running **smbd** under process supervisors such as **supervise** and **svscan** from Daniel J. Bernstein's **daemontools** package, or the AIX process monitor.

-S If specified, this parameter causes **smbd** to log to standard output rather than a file.

-i If this parameter is specified it causes the server to run "interactively", not as a daemon, even if the server is executed on the command line of a shell. Setting this parameter negates the implicit daemon mode when run from the command line. **smbd** also logs to standard output, as if the **-S** parameter had been given.

-V

Prints the program version number.

-s <configuration file>

The file specified contains the configuration details required by the server. The information in this file includes server-specific information such as what printcap file to use, as well as descriptions of all the services that the server is to provide. See *smb.conf* for more information. The default configuration file name is determined at compile time.

-d|--debuglevel=level

level is an integer from 0 to 10. The default value if this parameter is not specified is zero.

The higher this value, the more detail will be logged to the log files about the activities of the server. At level 0, only critical errors and serious warnings will be logged. Level 1 is a reasonable level for day-to-day running - it generates a small amount of information about operations carried out.

Levels above 1 will generate considerable amounts of log data, and should only be used when investigating a problem. Levels above 3 are designed for use only by developers and generate HUGE amounts of log data, most of which is extremely cryptic.

Note that specifying this parameter here will override the

parameter in the *smb.conf* file.

-l|--logfile=logdirectory

Base directory name for log/debug files. The extension "**.progname**" will be appended (e.g. log.smb-client, log.smbd, etc...). The log file is never removed by the client.

-h|--help

Print a summary of command line options.

-b Prints information about how Samba was built.

-p <port number(s)>

port number(s) is a space or comma-separated list of TCP ports *smbd* should listen on. The default value is taken from the *ports* parameter in *smb.conf*

The default ports are 139 (used for SMB over NetBIOS over TCP) and port 445 (used for plain SMB over TCP).

FILES

/etc/inetd.conf

If the server is to be run by the **inetd** meta-daemon, this file must contain suitable startup information for the meta-daemon.

/etc/rc

or whatever initialization script your system uses).

If running the server as a daemon at startup, this file will need to contain an appropriate startup sequence for the server.

/etc/services

If running the server via the meta-daemon **inetd**, this file must contain a mapping of service name (e.g., netbios-ssn) to service port (e.g., 139) and protocol type (e.g., tcp).

/etc/samba/smb.conf

This is the default location of the **smb.conf(5)** server configuration file.

This file describes all the services the server is to make available to clients. See **smb.conf(5)** for more information.

LIMITATIONS

On some systems **smbd** cannot change uid back to root after a `setuid()` call. Such systems are called trap-door uid systems. If you have such a system, you will be unable to connect from a client (such as a PC) as two different users at once. Attempts to connect the second user will result in access denied or similar.

ENVIRONMENT VARIABLES

PRINTER

If no printer name is specified to printable services, most systems will use the value of this variable (or **lp** if this variable is not defined) as the name of the printer to use. This is not specific to the server, however.

PAM INTERACTION

Samba uses PAM for authentication (when presented with a plaintext password), for account checking (is this account disabled?) and for session management. The degree to which samba supports PAM is restricted by the limitations of the SMB protocol and the obey pam restrictions **smb.conf(5)** parameter. When this is set, the following restrictions apply:

•

Account Validation: All accesses to a samba server are checked against PAM to see if the account is valid, not disabled and is permitted to login at this time. This also applies to encrypted logins.

•

Session Management: When not using share level security, users must pass PAM's session checks before access is granted. Note however, that this is bypassed in share level security. Note also that some older pam configuration files may need a line added for session support.

VERSION

This man page is correct for version 3.0 of the Samba suite.

DIAGNOSTICS

Most diagnostics issued by the server are logged in a specified log file. The log file name is specified at compile time, but may be overridden on the command line.

The number and nature of diagnostics available depends on the debug level used by the server. If you have problems, set the debug level to 3 and peruse the log files.

Most messages are reasonably self-explanatory. Unfortunately, at the time this man page was created, there are too many diagnostics available in the source code to warrant describing each and every diagnostic. At this stage your best bet is still to grep the source code and inspect the conditions that gave rise to the diagnostics you are seeing.

TDB FILES

Samba stores its data in several TDB (Trivial Database) files, usually located in */var/lib/samba*.

(*) information persistent across restarts (but not necessarily important to backup).

account_policy.tdb*

NT account policy settings such as pw expiration, etc...

brlock.tdb

byte range locks

browse.dat

browse lists

connections.tdb

share connections (used to enforce max connections, etc...)

gencache.tdb

generic caching db

group_mapping.tdb*

group mapping information

locking.tdb

share modes & oplocks

login_cache.tdb*

bad pw attempts

messages.tdb
 Samba messaging system

netsamlogon_cache.tdb*
 cache of user net_info_3 struct from net_samlogon() request (as a domain member)

ntdrivers.tdb*
 installed printer drivers

ntforms.tdb*
 installed printer forms

ntprinters.tdb*
 installed printer information

printing/
 directory containing tdb per print queue of cached lpq output

registry.tdb
 Windows registry skeleton (connect via regedit.exe)

sessionid.tdb
 session information (e.g. support for 'utmp = yes')

share_info.tdb*
 share acls

winbindd_cache.tdb
 winbindd's cache of user lists, etc...

winbindd_idmap.tdb*
 winbindd's local idmap db

wins.dat*
 wins database when 'wins support = yes'

SIGNALS

Sending the **smbd** a SIGHUP will cause it to reload its *smb.conf* configuration file within a short period of time.

To shut down a user's **smbd** process it is recommended that **SIGKILL (-9) NOT** be used, except as a last resort, as this may leave the shared memory area in an inconsistent state. The safe way to terminate an **smbd** is to send it a SIGTERM (-15) signal and wait for it to die on its own.

The debug log level of **smbd** may be raised or lowered using **smbcontrol(1)** program (SIGUSR[1|2] signals are no longer used since Samba 2.2). This is to allow transient problems to be diagnosed, whilst still running at a normally low log level.

Note that as the signal handlers send a debug write, they are not re-entrant in **smbd**. This you should wait until **smbd** is in a state of waiting for an incoming SMB before issuing them. It is possible to make the signal handlers safe by un-blocking the signals before the select call and re-blocking them after, however this would affect performance.

SEE ALSO

hosts_access(5), **inetd(8)**, **nmbd(8)**, **smb.conf(5)**, **smbclient(1)**, **testparm(1)**, **testprns(1)**, and the Internet RFC's *rfc1001.txt*, *rfc1002.txt*. In addition the CIFS (formerly SMB) specification is available as a link from the Web page <http://samba.org/cifs/>.

AUTHOR

The original Samba software and related utilities were created by Andrew Tridgell. Samba is now developed by the Samba Team as an Open Source project similar to the way the Linux kernel is developed.

The original Samba man pages were written by Karl Auer. The man page sources were converted to YODL format (another excellent piece of Open Source software, available at <ftp://ftp.icce.rug.nl/pub/unix/>) and updated for the Samba 2.0 release by Jeremy Allison. The conversion to DocBook for Samba 2.2 was done

by Gerald Carter. The conversion to DocBook XML 4.2 for Samba 3.0 was done by Alexander Bokovoy.