

**NAME**

smbmount - mount an smbfs filesystem

**SYNOPSIS**

**smbmount** {service} {mount-point} [-o options]

**DESCRIPTION**

**smbmount** mounts a Linux SMB filesystem. It is usually invoked as **mount.smbfs** by the **mount(8)** command when using the "-t smbfs" option. This command only works in Linux, and the kernel must support the smbfs filesystem.

Options to **smbmount** are specified as a comma-separated list of key=value pairs. It is possible to send options other than those listed here, assuming that smbfs supports them. If you get mount failures, check your kernel log for errors on unknown options.

**smbmount** is a daemon. After mounting it keeps running until the mounted smbfs is unmounted. It will log things that happen when in daemon mode using the "machine name" smbmount, so typically this output will end up in *log.smbmount*. The

**smbmount** process may also be called mount.smbfs.

**Note**

**smbmount** calls **smbmnt(8)** to do the actual mount. You must make sure that **smbmnt** is in the path so that it can be found.

**OPTIONS**

username=<arg>

specifies the username to connect as. If this is not given, then the environment variable

**USER** is used. This option can also take the form "user%password" or "user/workgroup" or "user/workgroup%password" to allow the password and workgroup to be specified as part of the username.

password=<arg>

specifies the SMB password. If this option is not given then the environment variable **PASSWD** is used. If it can find no password **smbmount** will prompt for a password, unless the guest option is given.

Note that passwords which contain the argument delimiter character (i.e. a comma ',') will failed to be parsed correctly on the command line. However, the same password defined in the **PASSWD** environment variable or a credentials file (see below) will be read correctly.

credentials=<filename>

specifies a file that contains a username and/or password. The format of the file is:

```
username = <value>
```

```
password = <value>
```

This is preferred over having passwords in plaintext in a shared file, such as */etc/fstab*. Be sure to protect any credentials file properly.

krb

Use kerberos (Active Directory).

netbiosname=<arg>

sets the source NetBIOS name. It defaults to the local hostname.

uid=<arg>

sets the uid that will own all files on the mounted filesystem. It may be specified as either a username or a numeric uid.

- `gid=<arg>`  
sets the gid that will own all files on the mounted filesystem. It may be specified as either a groupname or a numeric gid.
- `port=<arg>`  
sets the remote SMB port number. The default is 445, fallback is 139.
- `fmask=<arg>`  
sets the file mask. This determines the permissions that remote files have in the local filesystem. This is not a umask, but the actual permissions for the files. The default is based on the current umask.
- `dmask=<arg>`  
Sets the directory mask. This determines the permissions that remote directories have in the local filesystem. This is not a umask, but the actual permissions for the directories. The default is based on the current umask.
- `debug=<arg>`  
Sets the debug level. This is useful for tracking down SMB connection problems. A suggested value to start with is 4. If set too high there will be a lot of output, possibly hiding the useful output.
- `ip=<arg>`  
Sets the destination host or IP address.
- `workgroup=<arg>`  
Sets the workgroup on the destination
- `sockopt=<arg>`  
Sets the TCP socket options. See the `smb.conf(5)` *socket options* option.
- `scope=<arg>`  
Sets the NetBIOS scope
- `guest`  
Don't prompt for a password
- `ro` mount read-only
- `rw`  
mount read-write
- `iocharset=<arg>`  
sets the charset used by the Linux side for codepage to charset translations (NLS). Argument should be the name of a charset, like iso8859-1. (Note: only kernel 2.4.0 or later)
- `codepage=<arg>`  
sets the codepage the server uses. See the `iocharset` option. Example value cp850. (Note: only kernel 2.4.0 or later)
- `ttl=<arg>`  
sets how long a directory listing is cached in milliseconds (also affects visibility of file size and date changes). A higher value means that changes on the server take longer to be noticed but it can give better performance on large directories, especially over long distances. Default is 1000ms but something like 10000ms (10 seconds) is probably more reasonable in many cases. (Note: only kernel 2.4.2 or later)

## ENVIRONMENT VARIABLES

The variable **USER** may contain the username of the person using the client. This information is used only if the protocol level is high enough to support session-level passwords. The variable can be used to set both username and password by using the format `username%password`.

The variable **PASSWD** may contain the password of the person using the client. This information is used only if the protocol level is high enough to support session-level passwords.

The variable **PASSWD\_FILE** may contain the pathname of a file to read the password from. A single line of input is read and used as the password.

## OTHER COMMANDS

File systems that have been mounted using the **smbmount** can be unmounted using the **smbumount** or the UNIX system **umount** command.

## BUGS

Passwords and other options containing `,` can not be handled. For passwords an alternative way of passing them is in a credentials file or in the `PASSWD` environment.

The credentials file does not handle usernames or passwords with leading space.

One smbfs bug is important enough to mention here, even if it is a bit misplaced:

&#8226;

Mounts sometimes stop working. This is usually caused by `smbmount` terminating. Since `smbfs` needs `smbmount` to reconnect when the server disconnects, the mount will eventually go dead. An `umount/mount` normally fixes this. At least 2 ways to trigger this bug are known.

Note that the typical response to a bug report is suggestion to try the latest version first. So please try doing that first, and always include which versions you use of relevant software when reporting bugs (minimum: `samba`, kernel, distribution)

## SEE ALSO

`Documentation/filesystems/smbfs.txt` in the linux kernel source tree may contain additional options and information.

FreeBSD also has a `smbfs`, but it is not related to `smbmount`

For Solaris, HP-UX and others you may want to look at **smbsh**(1) or at other solutions, such as Sharity or perhaps replacing the SMB server with a NFS server.

## AUTHOR

Volker Lendecke, Andrew Tridgell, Michael H. Warfield and others.

The current maintainer of `smbfs` and the userspace tools **smbmount**, **smbumount**, and **smbmnt** is Urban Widmark. The SAMBA Mailing list is the preferred place to ask questions regarding these programs.

The conversion of this manpage for Samba 2.2 was performed by Gerald Carter. The conversion to Doc-Book XML 4.2 for Samba 3.0 was done by Alexander Bokovoy.