NAME

net − Tool for administration of Samba and remote CIFS servers.

SYNOPSIS

net {<ads|rap|rpc>} [−h|−−help] [−d|−−debuglevel=DEBUGLEVEL] [−−debug−stdout]
[−−configfile=CONFIGFILE] [−−option=name=value] [−l|−−log−basename=LOGFILEBASE]
[−−leak−report] [−−leak−report−full] [−R|−−name−resolve=NAME−RESOLVE−ORDER]
[−O|−−socket−options=SOCKETOPTIONS] [−m|−−max−protocol=MAXPROTOCOL]
[−n|−−netbiosname=NETBIOSNAME] [−−netbios−scope=SCOPE] [−W|−−workgroup=WORKGROUP]
[−−realm=REALM] [−U|−−user=[DOMAIN/]USERNAME[%PASSWORD]] [−N|−−no−pass]
[−−password=STRING] [−−pw−nt−hash] [−A|−−authentication−file=FILE] [−P|−−machine−pass]
[−−simple−bind−dn=DN] [−−use−kerberos=desired|required|off] [−−use−krb5−ccache=CCACHE]
[−−use−winbind−ccache] [−−client−protection=sign|encrypt|off] [−V|−−version]
[−w|−−target−workgroup workgroup] [−I|−−ipaddress ip−address] [−p|−−port port] [−−myname]
[−S|−−server server] [−−long] [−v|−−verbose] [−f|−−force] [−−request−timeout seconds]
[−t|−−timeout seconds] [−−dns−ttl TTL−IN−SECONDS] [−i|−−stdin]

DESCRIPTION

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

The Samba net utility is meant to work just like the net utility available for windows and DOS. The first argument should be used to specify the protocol to use when executing a certain command. ADS is used for ActiveDirectory, RAP is using for old (Win9x/NT3) clients and RPC can be used for NT4 and Windows 2000. If this argument is omitted, net will try to determine it automatically. Not all commands are available on all protocols.

OPTIONS

−w|−−target−workgroup target−workgroup
: Sets target workgroup or domain. You have to specify either this option or the IP address or the name of a server.

−I|−−ipaddress ip−address
: IP address of target server to use. You have to specify either this option or a target workgroup or a target server.

−p|−−port port
: Port on the target server to connect to (usually 139 or 445). Defaults to trying 445 first, then 139.

−S|−−server server
: Name of target server. You should specify either this option or a target workgroup or a target IP address.

−−long
: When listing data, give more information on each item.

−v|−−verbose
: When listing data, give more verbose information on each item.

−f|−−force
: Enforcing a net command.

−−request−timeout 30
: Let client requests timeout after 30 seconds the default is 10 seconds.

−t|−−timeout 30
: Set timeout for client operations to 30 seconds.

−i|−−stdin
: Take input for net commands from standard input.

−T|−−test

Only test command sequence, dry−run.

−F|−−flags FLAGS
Pass down integer flags to a net subcommand.

−C|−−comment COMMENT
Pass down a comment string to a net subcommand.

−−myname MYNAME
Use MYNAME as a requester name for a net subcommand.

−c|−−container CONTAINER
Use a specific AD container for net ads operations.

−M|−−maxusers MAXUSERS
Fill in the maxusers field in net rpc share operations.

−r|−−reboot
Reboot a remote machine after a command has been successfully executed (e.g. in remote join operations).

−−force−full−repl
When calling "net rpc vampire keytab" this option enforces a full re−creation of the generated keytab file.

−−single−obj−repl
When calling "net rpc vampire keytab" this option allows one to replicate just a single object to the generated keytab file.

−−clean−old−entries
When calling "net rpc vampire keytab" this option allows one to cleanup old entries from the generated keytab file.

−−db
Define dbfile for "net idmap" commands.

−−lock
Activates locking of the dbfile for "net idmap check" command.

−a|−−auto
Activates noninteractive mode in "net idmap check".

−−repair
Activates repair mode in "net idmap check".

−−acls
Includes ACLs to be copied in "net rpc share migrate".

−−attrs
Includes file attributes to be copied in "net rpc share migrate".

−−timestamps
Includes timestamps to be copied in "net rpc share migrate".

−X|−−exclude DIRECTORY
Allows one to exclude directories when copying with "net rpc share migrate".

−−destination SERVERNAME
Defines the target servername of migration process (defaults to localhost).

−L|−−local
Sets the type of group mapping to local (used in "net groupmap set").

−D|−−domain
Sets the type of group mapping to domain (used in "net groupmap set").

−N|−−ntname NTNAME

Sets the ntname of a group mapping (used in "net groupmap set").

−−rid RID
Sets the rid of a group mapping (used in "net groupmap set").

−−reg−version REG_VERSION
Assume database version {n|1,2,3} (used in "net registry check").

−o|−−output FILENAME
Output database file (used in "net registry check").

−−wipe
Create a new database from scratch (used in "net registry check").

−−precheck PRECHECK_DB_FILENAME
Defines filename for database prechecking (used in "net registry import").

−−no−dns−updates
Do not perform DNS updates as part of "net ads join".

−−keep−account
Prevent the machine account removal as part of "net ads leave".

−−json
Report results in JSON format for "net ads info" and "net ads lookup".

−−recursive
Traverse a directory hierarchy.

−−continue
Continue traversing a directory hierarchy in case conversion of one file fails.

−−follow−symlinks
Follow symlinks encountered while traversing a directory.

−−dns−ttl TTL−IN−SECONDS
Specify the Time to Live (TTL) of DNS records. DNS records will be created or updated with the given TTL. The TTL is specified in seconds. Can be used with "net ads dns register" and "net ads join". The default is 3600 seconds.

−d|−−debuglevel=DEBUGLEVEL
level is an integer from 0 to 10. The default value if this parameter is not specified is 1 for client applications.

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 **log level** parameter in the /etc/samba/smb.conf file.

−−debug−stdout
This will redirect debug output to STDOUT. By default all clients are logging to STDERR.

−−configfile=<configuration file>
The file specified contains the configuration details required by the client. The information in this file can be general for client and server or only provide client specific like options such as **client smb encrypt**. See /etc/samba/smb.conf for more information. The default configuration file name is determined at compile time.

−−option=<name>=<value>
Set the **smb.conf**(5) option "<name>" to value "<value>" from the command line. This overrides compiled−in defaults and options read from the configuration file. If a name or a value includes a space, wrap whole −−option=name=value into quotes.

−l|−−log−basename=logdirectory
Base directory name for log/debug files. The extension **".progname"** will be appended (e.g. log.smbclient, log.smbd, etc...). The log file is never removed by the client.

−−leak−report
Enable talloc leak reporting on exit.

−−leak−report−full
Enable full talloc leak reporting on exit.

−V|−−version
Prints the program version number.

−R|−−name−resolve=NAME−RESOLVE−ORDER
This option is used to determine what naming services and in what order to resolve host names to IP addresses. The option takes a space−separated string of different name resolution options. The best is to wrap the whole −−name−resolve=NAME−RESOLVE−ORDER into quotes.

The options are: "lmhosts", "host", "wins" and "bcast". They cause names to be resolved as follows:

- **lmhosts**: Lookup an IP address in the Samba lmhosts file. If the line in lmhosts has no name type attached to the NetBIOS name (see the **lmhosts**(5) for details) then any name type matches for lookup.
- **host**: Do a standard host name to IP address resolution, using the system /etc/hosts, NIS, or DNS lookups. This method of name resolution is operating system dependent, for instance on IRIX or Solaris this may be controlled by the /etc/nsswitch.conf file). Note that this method is only used if the NetBIOS name type being queried is the 0x20 (server) name type, otherwise it is ignored.
- **wins**: Query a name with the IP address listed in the *wins server* parameter. If no WINS server has been specified this method will be ignored.
- **bcast**: Do a broadcast on each of the known local interfaces listed in the *interfaces* parameter. This is the least reliable of the name resolution methods as it depends on the target host being on a locally connected subnet.

If this parameter is not set then the name resolve order defined in the /etc/samba/smb.conf file parameter (**name resolve order**) will be used.

The default order is lmhosts, host, wins, bcast. Without this parameter or any entry in the **name resolve order** parameter of the /etc/samba/smb.conf file, the name resolution methods will be attempted in this order.

−O|−−socket−options=SOCKETOPTIONS
TCP socket options to set on the client socket. See the socket options parameter in the /etc/samba/smb.conf manual page for the list of valid options.

−m|−−max−protocol=MAXPROTOCOL
The value of the parameter (a string) is the highest protocol level that will be supported by the client.

Note that specifying this parameter here will override the **client max protocol** parameter in the /etc/samba/smb.conf file.

−n|−−netbiosname=NETBIOSNAME
This option allows you to override the NetBIOS name that Samba uses for itself. This is identical to setting the **netbios name** parameter in the /etc/samba/smb.conf file. However, a command line setting

will take precedence over settings in /etc/samba/smb.conf.

−−netbios−scope=SCOPE
This specifies a NetBIOS scope that nmblookup will use to communicate with when generating NetBIOS names. For details on the use of NetBIOS scopes, see rfc1001.txt and rfc1002.txt. NetBIOS scopes are *very* rarely used, only set this parameter if you are the system administrator in charge of all the NetBIOS systems you communicate with.

−W|−−workgroup=WORKGROUP
Set the SMB domain of the username. This overrides the default domain which is the domain defined in smb.conf. If the domain specified is the same as the servers NetBIOS name, it causes the client to log on using the servers local SAM (as opposed to the Domain SAM).

Note that specifying this parameter here will override the **workgroup** parameter in the /etc/samba/smb.conf file.

−r|−−realm=REALM
Set the realm for the domain.

Note that specifying this parameter here will override the **realm** parameter in the /etc/samba/smb.conf file.

−U|−−user=[DOMAIN\]USERNAME[%PASSWORD]
Sets the SMB username or username and password.

If %PASSWORD is not specified, the user will be prompted. The client will first check the **USER** environment variable (which is also permitted to also contain the password separated by a %), then the **LOGNAME** variable (which is not permitted to contain a password) and if either exists, the value is used. If these environmental variables are not found, the username found in a Kerberos Credentials cache may be used.

A third option is to use a credentials file which contains the plaintext of the username and password. This option is mainly provided for scripts where the admin does not wish to pass the credentials on the command line or via environment variables. If this method is used, make certain that the permissions on the file restrict access from unwanted users. See the *−A* for more details.

Be cautious about including passwords in scripts or passing user−supplied values onto the command line. For security it is better to let the Samba client tool ask for the password if needed, or obtain the password once with kinit.

While Samba will attempt to scrub the password from the process title (as seen in ps), this is after startup and so is subject to a race.

−N|−−no−pass
If specified, this parameter suppresses the normal password prompt from the client to the user. This is useful when accessing a service that does not require a password.

Unless a password is specified on the command line or this parameter is specified, the client will request a password.

If a password is specified on the command line and this option is also defined the password on the command line will be silently ignored and no password will be used.

−−password
Specify the password on the commandline.

Be cautious about including passwords in scripts or passing user−supplied values onto the command line. For security it is better to let the Samba client tool ask for the password if needed, or obtain the

password once with kinit.

If −−password is not specified, the tool will check the **PASSWD** environment variable, followed by **PASSWD_FD** which is expected to contain an open file descriptor (FD) number.

Finally it will check **PASSWD_FILE** (containing a file path to be opened). The file should only contain the password. Make certain that the permissions on the file restrict access from unwanted users!

While Samba will attempt to scrub the password from the process title (as seen in ps), this is after startup and so is subject to a race.

−−pw−nt−hash
The supplied password is the NT hash.

−A|−−authentication−file=filename
This option allows you to specify a file from which to read the username and password used in the connection. The format of the file is:

```
username = <value>
password = <value>
domain   = <value>
```

Make certain that the permissions on the file restrict access from unwanted users!

−P|−−machine−pass
Use stored machine account password.

−−simple−bind−dn=DN
DN to use for a simple bind.

−−use−kerberos=desired|required|off
This parameter determines whether Samba client tools will try to authenticate using Kerberos. For Kerberos authentication you need to use dns names instead of IP addresses when connecting to a service.

Note that specifying this parameter here will override the **client use kerberos** parameter in the /etc/samba/smb.conf file.

−−use−krb5−ccache=CCACHE
Specifies the credential cache location for Kerberos authentication.

This will set −−use−kerberos=required too.

−−use−winbind−ccache
Try to use the credential cache by winbind.

−−client−protection=sign|encrypt|off
Sets the connection protection the client tool should use.

Note that specifying this parameter here will override the **client protection** parameter in the /etc/samba/smb.conf file.

In case you need more fine grained control you can use: −−option=clientsmbencrypt=OPTION, −−option=clientipcsigning=OPTION, −−option=clientsigning=OPTION.

COMMANDS

CHANGESECRETPW

This command allows the Samba machine account password to be set from an external application to a machine account password that has already been stored in Active Directory. DO NOT USE this command unless you know exactly what you are doing. The use of this command requires that the force flag (−f) be used also. There will be NO command prompt. Whatever information is piped into stdin, either by typing at the command line or otherwise, will be stored as the literal machine password. Do NOT use this without care and attention as it will overwrite a legitimate machine password without warning. YOU HAVE BEEN WARNED.

TIME

The NET TIME command allows you to view the time on a remote server or synchronise the time on the local server with the time on the remote server.

TIME

Without any options, the NET TIME command displays the time on the remote server. The remote server must be specified with the −S option.

TIME SYSTEM

Displays the time on the remote server in a format ready for /bin/date. The remote server must be specified with the −S option.

TIME SET

Tries to set the date and time of the local server to that on the remote server using /bin/date. The remote server must be specified with the −S option.

TIME ZONE

Displays the timezone in hours from GMT on the remote server. The remote server must be specified with the −S option.

[RPC|ADS] JOIN [TYPE] [−−no−dns−updates] [−U username[%password]] [dnshostname=FQDN] [createupn=UPN] [createcomputer=OU] [machinepass=PASS] [osName=string osVer=string] [options]

Join a domain. If the account already exists on the server, and [TYPE] is MEMBER, the machine will attempt to join automatically. (Assuming that the machine has been created in server manager) Otherwise, a password will be prompted for, and a new account may be created.

[TYPE] may be PDC, BDC or MEMBER to specify the type of server joining the domain.

[FQDN] (ADS only) set the dnsHostName attribute during the join. The default format is netbiosname.dnsdomain.

[UPN] (ADS only) set the principalname attribute during the join. The default format is host/netbiosname@REALM.

[OU] (ADS only) Precreate the computer account in a specific OU. The OU string reads from top to bottom without RDNs, and is delimited by a '/'. Please note that '\' is used for escape by both the shell and ldap, so it may need to be doubled or quadrupled to pass through, and it is not used as a delimiter.

[PASS] (ADS only) Set a specific password on the computer account being created by the join.

[osName=string osVer=String] (ADS only) Set the operatingSystem and operatingSystemVersion attribute during the join. Both parameters must be specified for either to take effect.

[RPC] OLDJOIN [options]

Join a domain. Use the OLDJOIN option to join the domain using the old style of domain joining − you need to create a trust account in server manager first.

[RPC|ADS] USER
[RPC|ADS] USER

List all users

[RPC|ADS] USER DELETE target
Delete specified user

[RPC|ADS] USER INFO target
List the domain groups of the specified user.

[RPC|ADS] USER RENAME oldname newname
Rename specified user.

[RPC|ADS] USER ADD name [password] [-F user flags] [-C comment]
Add specified user.

[RPC|ADS] GROUP
[RPC|ADS] GROUP [misc options] [targets]
List user groups.

[RPC|ADS] GROUP DELETE name [misc. options]
Delete specified group.

[RPC|ADS] GROUP ADD name [-C comment]
Create specified group.

[ADS] LOOKUP
Lookup the closest Domain Controller in our domain and retrieve server information about it.

[RAP|RPC] SHARE
[RAP|RPC] SHARE [misc. options] [targets]
Enumerates all exported resources (network shares) on target server.

[RAP|RPC] SHARE ADD name=serverpath [-C comment] [-M maxusers] [targets]
Adds a share from a server (makes the export active). Maxusers specifies the number of users that can be connected to the share simultaneously.

SHARE DELETE sharename
Delete specified share.

[RPC|RAP] FILE
[RPC|RAP] FILE
List all open files on remote server.

[RPC|RAP] FILE CLOSE fileid
Close file with specified *fileid* on remote server.

[RPC|RAP] FILE INFO fileid
Print information on specified *fileid*. Currently listed are: file−id, username, locks, path, permissions.

[RAP|RPC] FILE USER user
List files opened by specified *user*. Please note that net rap file user does not work against Samba servers.

SESSION
RAP SESSION
Without any other options, SESSION enumerates all active SMB/CIFS sessions on the target server.

RAP SESSION DELETE|CLOSE CLIENT_NAME
Close the specified sessions.

RAP SESSION INFO CLIENT_NAME
Give a list with all the open files in specified session.

RAP SERVER *DOMAIN*
List all servers in specified domain or workgroup. Defaults to local domain.

RAP DOMAIN
Lists all domains and workgroups visible on the current network.

RAP PRINTQ
RAP PRINTQ INFO QUEUE_NAME
Lists the specified print queue and print jobs on the server. If the *QUEUE_NAME* is omitted, all queues are listed.

RAP PRINTQ DELETE JOBID
Delete job with specified id.

RAP VALIDATE *user* [*password*]
Validate whether the specified user can log in to the remote server. If the password is not specified on the commandline, it will be prompted.

Note

Currently NOT implemented.

RAP GROUPMEMBER
RAP GROUPMEMBER LIST GROUP
List all members of the specified group.

RAP GROUPMEMBER DELETE GROUP USER
Delete member from group.

RAP GROUPMEMBER ADD GROUP USER
Add member to group.

RAP ADMIN *command*
Execute the specified *command* on the remote server. Only works with OS/2 servers.

Note

Currently NOT implemented.

RAP SERVICE
RAP SERVICE START NAME [arguments...]
Start the specified service on the remote server. Not implemented yet.

Note

Currently NOT implemented.

RAP SERVICE STOP
Stop the specified service on the remote server.

Note

Currently NOT implemented.

RAP PASSWORD *USER OLDPASS NEWPASS*
Change password of *USER* from *OLDPASS* to *NEWPASS*.

LOOKUP
LOOKUP HOST HOSTNAME [TYPE]
Lookup the IP address of the given host with the specified type (netbios suffix). The type defaults to 0x20 (workstation).

LOOKUP LDAP [DOMAIN]
Give IP address of LDAP server of specified *DOMAIN*. Defaults to local domain.

LOOKUP KDC [REALM]
Give IP address of KDC for the specified *REALM*. Defaults to local realm.

LOOKUP DC [DOMAIN]
Give IP's of Domain Controllers for specified *DOMAIN*. Defaults to local domain.

LOOKUP MASTER DOMAIN
Give IP of master browser for specified *DOMAIN* or workgroup. Defaults to local domain.

LOOKUP NAME [NAME]
Lookup username's sid and type for specified *NAME*

LOOKUP SID [SID]
Give sid's name and type for specified *SID*

LOOKUP DSGETDCNAME [NAME] [FLAGS] [SITENAME]
Give Domain Controller information for specified domain *NAME*

CACHE
Samba uses a general caching interface called 'gencache'. It can be controlled using 'NET CACHE'.

All the timeout parameters support the suffixes:
s − Seconds
m − Minutes
h − Hours
d − Days
w − Weeks

CACHE ADD key data time-out
Add specified key+data to the cache with the given timeout.

CACHE DEL key
Delete key from the cache.

CACHE SET key data time-out
Update data of existing cache entry.

CACHE SEARCH PATTERN
Search for the specified pattern in the cache data.

CACHE LIST
List all current items in the cache.

CACHE FLUSH
Remove all the current items from the cache.

GETLOCALSID [DOMAIN]
Prints the SID of the specified domain, or if the parameter is omitted, the SID of the local server.

SETLOCALSID S−1−5−21−x−y−z
Sets SID for the local server to the specified SID.

GETDOMAINSID
Prints the local machine SID and the SID of the current domain.

SETDOMAINSID
Sets the SID of the current domain.

GROUPMAP
Manage the mappings between Windows group SIDs and UNIX groups. Common options include:

- unixgroup − Name of the UNIX group
- ntgroup − Name of the Windows NT group (must be resolvable to a SID
- rid − Unsigned 32−bit integer
- sid − Full SID in the form of "S−1−..."
- type − Type of the group; either 'domain', 'local', or 'builtin'
- comment − Freeform text description of the group

GROUPMAP ADD

Add a new group mapping entry:

net groupmap add {rid=int|sid=string} unixgroup=string \
[type={domain|local}] [ntgroup=string] [comment=string]

GROUPMAP DELETE

Delete a group mapping entry. If more than one group name matches, the first entry found is deleted.

net groupmap delete {ntgroup=string|sid=SID}

GROUPMAP MODIFY

Update an existing group entry.

net groupmap modify {ntgroup=string|sid=SID} [unixgroup=string] \
[comment=string] [type={domain|local}]

GROUPMAP LIST

List existing group mapping entries.

net groupmap list [verbose] [ntgroup=string] [sid=SID]

MAXRID

Prints out the highest RID currently in use on the local server (by the active 'passdb backend').

RPC INFO

Print information about the domain of the remote server, such as domain name, domain sid and number of users and groups.

[RPC|ADS] TESTJOIN

Check whether participation in a domain is still valid.

[RPC|ADS] CHANGETRUSTPW

Force change of domain trust password.

RPC TRUSTDOM

RPC TRUSTDOM ADD DOMAIN

Add a interdomain trust account for *DOMAIN*. This is in fact a Samba account named *DOMAIN$* with the account flag **'I'** (interdomain trust account). This is required for incoming trusts to work. It makes Samba be a trusted domain of the foreign (trusting) domain. Users of the Samba domain will be made available in the foreign domain. If the command is used against localhost it has the same effect as smbpasswd −a −i DOMAIN. Please note that both commands expect a appropriate UNIX account.

RPC TRUSTDOM DEL DOMAIN

Remove interdomain trust account for *DOMAIN*. If it is used against localhost it has the same effect as smbpasswd −x DOMAIN$.

RPC TRUSTDOM ESTABLISH DOMAIN

Establish a trust relationship to a trusted domain. Interdomain account must already be created on the remote PDC. This is required for outgoing trusts to work. It makes Samba be a trusting domain of a foreign (trusted) domain. Users of the foreign domain will be made available in our domain. You'll need winbind and a working idmap config to make them appear in your system.

RPC TRUSTDOM REVOKE DOMAIN

Abandon relationship to trusted domain

RPC TRUSTDOM LIST

List all interdomain trust relationships.

RPC TRUST

RPC TRUST CREATE

Create a trust object by calling lsaCreateTrustedDomainEx2. The can be done on a single server or on two servers at once with the possibility to use a random trust password.

Options:

otherserver
: Domain controller of the second domain

otheruser
: Admin user in the second domain

otherdomainsid
: SID of the second domain

other_netbios_domain
: NetBIOS (short) name of the second domain

otherdomain
: DNS (full) name of the second domain

trustpw
: Trust password

Examples:

Create a trust object on srv1.dom1.dom for the domain dom2

```
net rpc trust create \
  otherdomainsid=S-x-x-xx-xxxxxxxxxx-xxxxxxxxxx-xxxxxxxxx \
  other_netbios_domain=dom2 \
  otherdomain=dom2.dom \
  trustpw=12345678 \
  -S srv1.dom1.dom
```

Create a trust relationship between dom1 and dom2

```
net rpc trust create \
  otherserver=srv2.dom2.test \
  otheruser=dom2adm \
  -S srv1.dom1.dom
```

RPC TRUST DELETE

Delete a trust object by calling lsaDeleteTrustedDomain. The can be done on a single server or on two servers at once.

Options:

otherserver
: Domain controller of the second domain

otheruser
: Admin user in the second domain

otherdomainsid
: SID of the second domain

Examples:

Delete a trust object on srv1.dom1.dom for the domain dom2

```
net rpc trust delete \
  otherdomainsid=S-x-x-xx-xxxxxxxxxx-xxxxxxxxxx-xxxxxxxxx \
  -S srv1.dom1.dom
```

Delete a trust relationship between dom1 and dom2

```
net rpc trust delete \
  otherserver=srv2.dom2.test \
  otheruser=dom2adm \
  -S srv1.dom1.dom
```

RPC RIGHTS

This subcommand is used to view and manage Samba's rights assignments (also referred to as privileges). There are three options currently available: *list*, *grant*, and *revoke*. More details on Samba's privilege model and its use can be found in the Samba-HOWTO-Collection.

RPC ABORTSHUTDOWN

Abort the shutdown of a remote server.

RPC SHUTDOWN [-t timeout] [-r] [-f] [-C message]

Shut down the remote server.

-r
: Reboot after shutdown.

-f
: Force shutting down all applications.

-t timeout
: Timeout before system will be shut down. An interactive user of the system can use this time to cancel the shutdown.

-C message
: Display the specified message on the screen to announce the shutdown.

RPC SAMDUMP

Print out sam database of remote server. You need to run this against the PDC, from a Samba machine joined as a BDC.

RPC VAMPIRE

Export users, aliases and groups from remote server to local server. You need to run this against the PDC, from a Samba machine joined as a BDC. This vampire command cannot be used against an Active Directory, only against an NT4 Domain Controller.

RPC VAMPIRE KEYTAB

Dump remote SAM database to local Kerberos keytab file.

RPC VAMPIRE LDIF

Dump remote SAM database to local LDIF file or standard output.

RPC GETSID

Fetch domain SID and store it in the local secrets.tdb.

ADS GPO

ADS GPO APPLY <USERNAME|MACHINENAME>

Apply GPOs for a username or machine name. Either username or machine name should be provided to the command, not both.

ADS GPO GETGPO [GPO]

List specified GPO.

ADS GPO LINKADD [LINKDN] [GPODN]

Link a container to a GPO. *LINKDN* Container to link to a GPO. *GPODN* GPO to link container to. DNs must be provided properly escaped. See RFC 4514 for details.

ADS GPO LINKGET [CONTAINER]
Lists gPLink of a container.

ADS GPO LIST <USERNAME|MACHINENAME>
Lists all GPOs for a username or machine name. Either username or machine name should be provided to the command, not both.

ADS GPO LISTALL
Lists all GPOs on a DC.

ADS GPO REFRESH [USERNAME] [MACHINENAME]
Lists all GPOs assigned to an account and download them. *USERNAME* User to refresh GPOs for. *MACHINENAME* Machine to refresh GPOs for.

ADS DNS
ADS DNS REGISTER [HOSTNAME [IP [IP.....]]]
Add host dns entry to Active Directory.

ADS DNS UNREGISTER <HOSTNAME>
Remove host dns entry from Active Directory.

ADS LEAVE [−−keep−account]
Make the remote host leave the domain it is part of.

ADS STATUS
Print out status of machine account of the local machine in ADS. Prints out quite some debug info. Aimed at developers, regular users should use NET ADS TESTJOIN.

ADS PRINTER
ADS PRINTER INFO [PRINTER] [SERVER]
Lookup info for *PRINTER* on *SERVER*. The printer name defaults to "*", the server name defaults to the local host.

ADS PRINTER PUBLISH PRINTER
Publish specified printer using ADS.

ADS PRINTER REMOVE PRINTER
Remove specified printer from ADS directory.

ADS SEARCH *EXPRESSION ATTRIBUTES...*
Perform a raw LDAP search on a ADS server and dump the results. The expression is a standard LDAP search expression, and the attributes are a list of LDAP fields to show in the results.

Example: **net ads search '(objectCategory=group)' sAMAccountName**

ADS DN *DN (attributes)*
Perform a raw LDAP search on a ADS server and dump the results. The DN standard LDAP DN, and the attributes are a list of LDAP fields to show in the result.

Example: **net ads dn 'CN=administrator,CN=Users,DC=my,DC=domain' SAMAccountName**

ADS KEYTAB *CREATE*
Creates a new keytab file if one doesn't exist with default entries. Default entries are kerberos principals created from the machinename of the client, the UPN (if it exists) and any Windows SPN(s) associated with the computer AD account for the client. If a keytab file already exists then only missing kerberos principals from the default entries are added. No changes are made to the computer AD account.

ADS KEYTAB *ADD (principal | machine | serviceclass | windows SPN*
Adds a new keytab entry, the entry can be either;

kerberos principal
A kerberos principal (identified by the presence of '@') is just added to the keytab file.

machinename
A machinename (identified by the trailing '$') is used to create a a kerberos principal

'machinename@realm' which is added to the keytab file.

serviceclass
: A serviceclass (such as 'cifs', 'html' etc.) is used to create a pair of kerberos principals 'serviceclass/fully_qualified_dns_name@realm' & 'serviceclass/netbios_name@realm' which are added to the keytab file.

Windows SPN
: A Windows SPN is of the format 'serviceclass/host:port', it is used to create a kerberos principal 'serviceclass/host@realm' which will be written to the keytab file.

Unlike old versions no computer AD objects are modified by this command. To preserve the bevhaviour of older clients 'net ads keytab ad_update_ads' is available.

ADS KEYTAB *ADD_UPDATE_ADS (principal | machine | serviceclass | windows SPN*
Adds a new keytab entry (see section for net ads keytab add). In addition to adding entries to the keytab file corrosponding Windows SPNs are created from the entry passed to this command. These SPN(s) added to the AD computer account object associated with the client machine running this command for the following entry types;

serviceclass
: A serviceclass (such as 'cifs', 'html' etc.) is used to create a pair of Windows SPN(s) 'param/full_qualified_dns' & 'param/netbios_name' which are added to the AD computer account object for this client.

Windows SPN
: A Windows SPN is of the format 'serviceclass/host:port', it is added as passed to the AD computer account object for this client.

ADS setspn *SETSPN LIST [machine]*
Lists the Windows SPNs stored in the 'machine' Windows AD Computer object. If 'machine' is not specified then computer account for this client is used instead.

ADS setspn *SETSPN ADD SPN [machine]*
Adds the specified Windows SPN to the 'machine' Windows AD Computer object. If 'machine' is not specified then computer account for this client is used instead.

ADS setspn *SETSPN DELETE SPN [machine]*
DELETE the specified Window SPN from the 'machine' Windows AD Computer object. If 'machine' is not specified then computer account for this client is used instead.

ADS WORKGROUP
Print out workgroup name for specified kerberos realm.

ADS ENCTYPES
List, modify or delete the value of the "msDS−SupportedEncryptionTypes" attribute of an account in AD.

This attribute allows one to control which Kerberos encryption types are used for the generation of initial and service tickets. The value consists of an integer bitmask with the following values:

0x00000001 DES−CBC−CRC

0x00000002 DES−CBC−MD5

0x00000004 RC4−HMAC

0x00000008 AES128−CTS−HMAC−SHA1−96

0x00000010 AES256−CTS−HMAC−SHA1−96

ADS ENCTYPES LIST *<ACCOUNTNAME>*
List the value of the "msDS−SupportedEncryptionTypes" attribute of a given account.

Example: **net ads enctypes list Computername**

ADS ENCTYPES SET *<ACCOUNTNAME> [enctypes]*

Set the value of the "msDS−SupportedEncryptionTypes" attribute of the LDAP object of ACCOUNTNAME to a given value. If the value is omitted, the value is set to 31 which enables all the currently supported encryption types.

Example: **net ads enctypes set Computername 24**

ADS ENCTYPES DELETE *<ACCOUNTNAME>*

Deletes the "msDS−SupportedEncryptionTypes" attribute of the LDAP object of ACCOUNTNAME.

Example: **net ads enctypes set Computername 24**

SAM CREATEBUILTINGROUP <NAME>

(Re)Create a BUILTIN group. Only a wellknown set of BUILTIN groups can be created with this command. This is the list of currently recognized group names: Administrators, Users, Guests, Power Users, Account Operators, Server Operators, Print Operators, Backup Operators, Replicator, RAS Servers, Pre−Windows 2000 compatible Access. This command requires a running Winbindd with idmap allocation properly configured. The group gid will be allocated out of the winbindd range.

SAM CREATELOCALGROUP <NAME>

Create a LOCAL group (also known as Alias). This command requires a running Winbindd with idmap allocation properly configured. The group gid will be allocated out of the winbindd range.

SAM DELETELOCALGROUP <NAME>

Delete an existing LOCAL group (also known as Alias).

SAM MAPUNIXGROUP <NAME>

Map an existing Unix group and make it a Domain Group, the domain group will have the same name.

SAM UNMAPUNIXGROUP <NAME>

Remove an existing group mapping entry.

SAM ADDMEM <GROUP> <MEMBER>

Add a member to a Local group. The group can be specified only by name, the member can be specified by name or SID.

SAM DELMEM <GROUP> <MEMBER>

Remove a member from a Local group. The group and the member must be specified by name.

SAM LISTMEM <GROUP>

List Local group members. The group must be specified by name.

SAM LIST <users|groups|localgroups|builtin|workstations> [verbose]

List the specified set of accounts by name. If verbose is specified, the rid and description is also provided for each account.

SAM RIGHTS LIST

List all available privileges.

SAM RIGHTS GRANT <NAME> <PRIVILEGE>

Grant one or more privileges to a user.

SAM RIGHTS REVOKE <NAME> <PRIVILEGE>

Revoke one or more privileges from a user.

SAM SHOW <NAME>

Show the full DOMAIN\\NAME the SID and the type for the corresponding account.

SAM SET HOMEDIR <NAME> <DIRECTORY>

Set the home directory for a user account.

SAM SET PROFILEPATH <NAME> <PATH>

Set the profile path for a user account.

SAM SET COMMENT <NAME> <COMMENT>
Set the comment for a user or group account.

SAM SET FULLNAME <NAME> <FULL NAME>
Set the full name for a user account.

SAM SET LOGONSCRIPT <NAME> <SCRIPT>
Set the logon script for a user account.

SAM SET HOMEDRIVE <NAME> <DRIVE>
Set the home drive for a user account.

SAM SET WORKSTATIONS <NAME> <WORKSTATIONS>
Set the workstations a user account is allowed to log in from.

SAM SET DISABLE <NAME>
Set the "disabled" flag for a user account.

SAM SET PWNOTREQ <NAME>
Set the "password not required" flag for a user account.

SAM SET AUTOLOCK <NAME>
Set the "autolock" flag for a user account.

SAM SET PWNOEXP <NAME>
Set the "password do not expire" flag for a user account.

SAM SET PWDMUSTCHANGENOW <NAME> [yes|no]
Set or unset the "password must change" flag for a user account.

SAM POLICY LIST
List the available account policies.

SAM POLICY SHOW <account policy>
Show the account policy value.

SAM POLICY SET <account policy> <value>
Set a value for the account policy. Valid values can be: "forever", "never", "off", or a number.

SAM PROVISION
Only available if ldapsam:editposix is set and winbindd is running. Properly populates the ldap tree with the basic accounts (Administrator) and groups (Domain Users, Domain Admins, Domain Guests) on the ldap tree.

IDMAP DUMP <local tdb file name>
Dumps the mappings contained in the local tdb file specified. This command is useful to dump only the mappings produced by the idmap_tdb backend.

IDMAP RESTORE [input file]
Restore the mappings from the specified file or stdin.

IDMAP SET SECRET <DOMAIN> <secret>
Store a secret for the specified domain, used primarily for domains that use idmap_ldap as a backend. In this case the secret is used as the password for the user DN used to bind to the ldap server.

IDMAP SET RANGE <RANGE> <SID> [index] [−−db=<DB>]
Store a domain−range mapping for a given domain (and index) in autorid database.

IDMAP SET CONFIG <config> [−−db=<DB>]
Update CONFIG entry in autorid database.

IDMAP GET RANGE <SID> [index] [−−db=<DB>]
Get the range for a given domain and index from autorid database.

IDMAP GET RANGES [<SID>] [−−db=<DB>]
Get ranges for all domains or for one identified by given SID.

IDMAP GET CONFIG [−−db=<DB>]
Get CONFIG entry from autorid database.

IDMAP DELETE MAPPING [−f] [−−db=<DB>] <ID>
Delete a mapping sid <−> gid or sid <−> uid from the IDMAP database. The mapping is given by <ID> which may either be a sid: S−x−..., a gid: "GID number" or a uid: "UID number". Use −f to delete an invalid partial mapping <ID> −> xx

Use "smbcontrol all idmap ..." to notify running smbd instances. See the **smbcontrol**(1) manpage for details.

IDMAP DELETE RANGE [−f] [−−db=<TDB>] <RANGE>|(<SID> [<INDEX>])
Delete a domain range mapping identified by 'RANGE' or "domain SID and INDEX" from autorid database. Use −f to delete invalid mappings.

IDMAP DELETE RANGES [−f] [−−db=<TDB>] <SID>
Delete all domain range mappings for a domain identified by SID. Use −f to delete invalid mappings.

IDMAP CHECK [−v] [−r] [−a] [−T] [−f] [−l] [−−db=<DB>]
Check and repair the IDMAP database. If no option is given a read only check of the database is done. Among others an interactive or automatic repair mode may be chosen with one of the following options:

−r|−−repair
Interactive repair mode, ask a lot of questions.

−a|−−auto
Noninteractive repair mode, use default answers.

−v|−−verbose
Produce more output.

−f|−−force
Try to apply changes, even if they do not apply cleanly.

−T|−−test
Dry run, show what changes would be made but don't touch anything.

−l|−−lock
Lock the database while doing the check.

−−db <DB>
Check the specified database.

It reports about the finding of the following errors:

Missing reverse mapping:
A record with mapping A−>B where there is no B−>A. Default action in repair mode is to "fix" this by adding the reverse mapping.

Invalid mapping:
A record with mapping A−>B where B−>C. Default action is to "delete" this record.

Missing or invalid HWM:
A high water mark is not at least equal to the largest ID in the database. Default action is to "fix" this by setting it to the largest ID found +1.

Invalid record:
Something we failed to parse. Default action is to "edit" it in interactive and "delete" it in automatic mode.

USERSHARE
Starting with version 3.0.23, a Samba server now supports the ability for non−root users to add user defined shares to be exported using the "net usershare" commands.

To set this up, first set up your /etc/samba/smb.conf by adding to the [global] section: usershare path = /usr/local/samba/lib/usershares Next create the directory /usr/local/samba/lib/usershares, change the owner to root and set the group owner to the UNIX group who should have the ability to create usershares, for example a group called "serverops". Set the permissions on /usr/local/samba/lib/usershares to 01770. (Owner and group all access, no access for others, plus the sticky bit, which means that a file in that directory can be renamed or deleted only by the owner of the file). Finally, tell smbd how many usershares you will allow by adding to the [global] section of /etc/samba/smb.conf a line such as : usershare max shares = 100. To allow 100 usershare definitions. Now, members of the UNIX group "serverops" can create user defined shares on demand using the commands below.

The usershare commands are:

net usershare add sharename path [comment [acl] [guest_ok=[y|n]]] − to add or change a user defined share.

net usershare delete sharename − to delete a user defined share.

net usershare info [−−long] [wildcard sharename] − to print info about a user defined share.

net usershare list [−−long] [wildcard sharename] − to list user defined shares.

USERSHARE ADD sharename path [comment] [acl] [guest_ok=[y|n]]

Add or replace a new user defined share, with name "sharename".

"path" specifies the absolute pathname on the system to be exported. Restrictions may be put on this, see the global /etc/samba/smb.conf parameters: "usershare owner only", "usershare prefix allow list", and "usershare prefix deny list".

The optional "comment" parameter is the comment that will appear on the share when browsed to by a client.

The optional "acl" field specifies which users have read and write access to the entire share. Note that guest connections are not allowed unless the /etc/samba/smb.conf parameter "usershare allow guests" has been set. The definition of a user defined share acl is: "user:permission", where user is a valid username on the system and permission can be "F", "R", or "D". "F" stands for "full permissions", ie. read and write permissions. "D" stands for "deny" for a user, ie. prevent this user from accessing this share. "R" stands for "read only", ie. only allow read access to this share (no creation of new files or directories or writing to files).

The default if no "acl" is given is "Everyone:R", which means any authenticated user has read−only access.

The optional "guest_ok" has the same effect as the parameter of the same name in /etc/samba/smb.conf, in that it allows guest access to this user defined share. This parameter is only allowed if the global parameter "usershare allow guests" has been set to true in the /etc/samba/smb.conf.

There is no separate command to modify an existing user defined share, just use the "net usershare add [sharename]" command using the same sharename as the one you wish to modify and specify the new options you wish. The Samba smbd daemon notices user defined share modifications at connect time so will see the change immediately, there is no need to restart smbd on adding, deleting or changing a user defined share.

USERSHARE DELETE sharename

Deletes the user defined share by name. The Samba smbd daemon immediately notices this change, although it will not disconnect any users currently connected to the deleted share.

USERSHARE INFO [--long] [wildcard sharename]

Get info on user defined shares owned by the current user matching the given pattern, or all users.

net usershare info on its own dumps out info on the user defined shares that were created by the current user, or restricts them to share names that match the given wildcard pattern ('*' matches one or more characters, '?' matches only one character). If the '−−long' option is also given, it prints out info on user defined shares created by other users.

The information given about a share looks like: [foobar] path=/home/jeremy comment=testme

usershare_acl=Everyone:F guest_ok=n And is a list of the current settings of the user defined share that can be modified by the "net usershare add" command.

USERSHARE LIST [--long] wildcard sharename

List all the user defined shares owned by the current user matching the given pattern, or all users.

net usershare list on its own list out the names of the user defined shares that were created by the current user, or restricts the list to share names that match the given wildcard pattern ('*' matches one or more characters, '?' matches only one character). If the '−−long' option is also given, it includes the names of user defined shares created by other users.

[RPC] CONF

Starting with version 3.2.0, a Samba server can be configured by data stored in registry. This configuration data can be edited with the new "net conf" commands. There is also the possibility to configure a remote Samba server by enabling the RPC conf mode and specifying the address of the remote server.

The deployment of this configuration data can be activated in two levels from the */etc/samba/smb.conf* file: Share definitions from registry are activated by setting *registry shares* to “yes” in the [global] section and global configuration options are activated by setting **include = registry** in the [global] section for a mixed configuration or by setting **config backend = registry** in the [global] section for a registry−only configuration. See the **smb.conf**(5) manpage for details.

The conf commands are:

net [rpc] conf list − Dump the complete configuration in smb.conf like format.
net [rpc] conf import − Import configuration from file in smb.conf format.
net [rpc] conf listshares − List the registry shares.
net [rpc] conf drop − Delete the complete configuration from registry.
net [rpc] conf showshare − Show the definition of a registry share.
net [rpc] conf addshare − Create a new registry share.
net [rpc] conf delshare − Delete a registry share.
net [rpc] conf setparm − Store a parameter.
net [rpc] conf getparm − Retrieve the value of a parameter.
net [rpc] conf delparm − Delete a parameter.
net [rpc] conf getincludes − Show the includes of a share definition.
net [rpc] conf setincludes − Set includes for a share.
net [rpc] conf delincludes − Delete includes from a share definition.

[RPC] CONF LIST

Print the configuration data stored in the registry in a smb.conf−like format to standard output.

[RPC] CONF IMPORT [--test|-T] filename [section]

This command imports configuration from a file in smb.conf format. If a section encountered in the input file is present in registry, its contents is replaced. Sections of registry configuration that have no counterpart in the input file are not affected. If you want to delete these, you will have to use the "net conf drop" or "net conf delshare" commands. Optionally, a section may be specified to restrict the effect of the import command to that specific section. A test mode is enabled by specifying the parameter "−T" on the commandline. In test mode, no changes are made to the registry, and the resulting configuration is printed to standard output instead.

[RPC] CONF LISTSHARES

List the names of the shares defined in registry.

[RPC] CONF DROP

Delete the complete configuration data from registry.

[RPC] CONF SHOWSHARE sharename

Show the definition of the share or section specified. It is valid to specify "global" as sharename to retrieve the global configuration options from registry.

[RPC] CONF ADDSHARE sharename path [writeable={y|N} [guest_ok={y|N} [comment]]]
Create a new share definition in registry. The sharename and path have to be given. The share name may *not* be "global". Optionally, values for the very common options "writeable", "guest ok" and a "comment" may be specified. The same result may be obtained by a sequence of "net conf setparm" commands.

[RPC] CONF DELSHARE sharename
Delete a share definition from registry.

[RPC] CONF SETPARM section parameter value
Store a parameter in registry. The section may be global or a sharename. The section is created if it does not exist yet.

[RPC] CONF GETPARM section parameter
Show a parameter stored in registry.

[RPC] CONF DELPARM section parameter
Delete a parameter stored in registry.

[RPC] CONF GETINCLUDES section
Get the list of includes for the provided section (global or share).

Note that due to the nature of the registry database and the nature of include directives, the includes need special treatment: Parameters are stored in registry by the parameter name as valuename, so there is only ever one instance of a parameter per share. Also, a specific order like in a text file is not guaranteed. For all real parameters, this is perfectly ok, but the include directive is rather a meta parameter, for which, in the smb.conf text file, the place where it is specified between the other parameters is very important. This can not be achieved by the simple registry smbconf data model, so there is one ordered list of includes per share, and this list is evaluated after all the parameters of the share.

Further note that currently, only files can be included from registry configuration. In the future, there will be the ability to include configuration data from other registry keys.

[RPC] CONF SETINCLUDES section [filename]+
Set the list of includes for the provided section (global or share) to the given list of one or more filenames. The filenames may contain the usual smb.conf macros like %I.

[RPC] CONF DELINCLUDES section
Delete the list of includes from the provided section (global or share).

REGISTRY
Manipulate Samba's registry.

The registry commands are:
net registry enumerate – Enumerate registry keys and values.
net registry enumerate_recursive – Enumerate registry key and its subkeys.
net registry createkey – Create a new registry key.
net registry deletekey – Delete a registry key.
net registry deletekey_recursive – Delete a registry key with subkeys.
net registry getvalue – Print a registry value.
net registry getvalueraw – Print a registry value (raw format).
net registry setvalue – Set a new registry value.
net registry increment – Increment a DWORD registry value under a lock.
net registry deletevalue – Delete a registry value.
net registry getsd – Get security descriptor.
net registry getsd_sdd1 – Get security descriptor in sddl format.
net registry setsd_sdd1 – Set security descriptor from sddl format string.
net registry import – Import a registration entries (.reg) file.
net registry export – Export a registration entries (.reg) file.
net registry convert – Convert a registration entries (.reg) file.
net registry check – Check and repair a registry database.

REGISTRY ENUMERATE key
Enumerate subkeys and values of *key*.

REGISTRY ENUMERATE_RECURSIVE key
Enumerate values of *key* and its subkeys.

REGISTRY CREATEKEY key
Create a new *key* if not yet existing.

REGISTRY DELETEKEY key
Delete the given *key* and its values from the registry, if it has no subkeys.

REGISTRY DELETEKEY_RECURSIVE key
Delete the given *key* and all of its subkeys and values from the registry.

REGISTRY GETVALUE key name
Output type and actual value of the value *name* of the given *key*.

REGISTRY GETVALUERAW key name
Output the actual value of the value *name* of the given *key*.

REGISTRY SETVALUE key name type value ...
Set the value *name* of an existing *key*. *type* may be one of *sz*, *multi_sz* or *dword*. In case of *multi_sz value* may be given multiple times.

REGISTRY INCREMENT key name [inc]
Increment the DWORD value *name* of *key* by *inc* while holding a g_lock. *inc* defaults to 1.

REGISTRY DELETEVALUE key name
Delete the value *name* of the given *key*.

REGISTRY GETSD key
Get the security descriptor of the given *key*.

REGISTRY GETSD_SDDL key
Get the security descriptor of the given *key* as a Security Descriptor Definition Language (SDDL) string.

REGISTRY SETSD_SDDL keysd
Set the security descriptor of the given *key* from a Security Descriptor Definition Language (SDDL) string *sd*.

REGISTRY IMPORT file [--precheck <check-file>] [opt]
Import a registration entries (.reg) *file*.

The following options are available:

−−precheck *check−file*
This is a mechanism to check the existence or non−existence of certain keys or values specified in a precheck file before applying the import file. The import file will only be applied if the precheck succeeds.

The check−file follows the normal registry file syntax with the following semantics:

- <value name>=<value> checks whether the value exists and has the given value.
- <value name>=− checks whether the value does not exist.
- [key] checks whether the key exists.
- [−key] checks whether the key does not exist.

REGISTRY EXPORT keyfile[opt]
Export a *key* to a registration entries (.reg) *file*.

REGISTRY CONVERT in out [[inopt] outopt]
Convert a registration entries (.reg) file *in*.

REGISTRY CHECK [-ravTl] [-o <ODB>] [--wipe] [<DB>]
Check and repair the registry database. If no option is given a read only check of the database is done. Among others an interactive or automatic repair mode may be chosen with one of the following options

−r|−−repair
Interactive repair mode, ask a lot of questions.

−a|−−auto
Noninteractive repair mode, use default answers.

−v|−−verbose
Produce more output.

−T|−−test
Dry run, show what changes would be made but don't touch anything.

−l|−−lock
Lock the database while doing the check.

−−reg−version={1,2,3}
Specify the format of the registry database. If not given it defaults to the value of the binary or, if an registry.tdb is explicitly stated at the commandline, to the value found in the INFO/version record.

[−−db] <DB>
Check the specified database.

−o|−−output <ODB>
Create a new registry database <ODB> instead of modifying the input. If <ODB> is already existing −−wipe may be used to overwrite it.

−−wipe
Replace the registry database instead of modifying the input or overwrite an existing output database.

EVENTLOG
Starting with version 3.4.0 net can read, dump, import and export native win32 eventlog files (usually *.evt). evt files are used by the native Windows eventviewer tools.

The import and export of evt files can only succeed when *eventlog list* is used in */etc/samba/smb.conf* file. See the **smb.conf**(5) manpage for details.

The eventlog commands are:
net eventlog dump − Dump a eventlog *.evt file on the screen.
net eventlog import − Import a eventlog *.evt into the samba internal tdb based representation of eventlogs.
net eventlog export − Export the samba internal tdb based representation of eventlogs into an eventlog *.evt file.

EVENTLOG DUMP filename
Prints a eventlog *.evt file to standard output.

EVENTLOG IMPORT filename eventlog
Imports a eventlog *.evt file defined by *filename* into the samba internal tdb representation of eventlog defined by *eventlog*. *eventlog* needs to part of the *eventlog list* defined in /etc/samba/smb.conf. See the **smb.conf**(5) manpage for details.

EVENTLOG EXPORT filename eventlog
Exports the samba internal tdb representation of eventlog defined by *eventlog* to a eventlog *.evt file defined by *filename*. *eventlog* needs to part of the *eventlog list* defined in /etc/samba/smb.conf. See the **smb.conf**(5) manpage for details.

DOM

Starting with version 3.2.0 Samba has support for remote join and unjoin APIs, both client and server−side. Windows supports remote join capabilities since Windows 2000.

In order for Samba to be joined or unjoined remotely an account must be used that is either member of the Domain Admins group, a member of the local Administrators group or a user that is granted the SeMachineAccountPrivilege privilege.

The client side support for remote join is implemented in the net dom commands which are:

net dom join − Join a remote computer into a domain.
net dom unjoin − Unjoin a remote computer from a domain.
net dom renamecomputer − Renames a remote computer joined to a domain.

DOM JOIN domain=DOMAIN ou=OU account=ACCOUNT password=PASSWORD reboot

Joins a computer into a domain. This command supports the following additional parameters:

- *DOMAIN* can be a NetBIOS domain name (also known as short domain name) or a DNS domain name for Active Directory Domains. As in Windows, it is also possible to control which Domain Controller to use. This can be achieved by appending the DC name using the \ separator character. Example: MYDOM\MYDC. The *DOMAIN* parameter cannot be NULL.
- *OU* can be set to a RFC 1779 LDAP DN, like *ou=mymachines,cn=Users,dc=example,dc=com* in order to create the machine account in a non−default LDAP container. This optional parameter is only supported when joining Active Directory Domains.
- *ACCOUNT* defines a domain account that will be used to join the machine to the domain. This domain account needs to have sufficient privileges to join machines.
- *PASSWORD* defines the password for the domain account defined with *ACCOUNT*.
- *REBOOT* is an optional parameter that can be set to reboot the remote machine after successful join to the domain.

Note that you also need to use standard net parameters to connect and authenticate to the remote machine that you want to join. These additional parameters include: −S computer and −U user.

Example: net dom join −S xp −U XP\\administrator%secret domain=MYDOM account=MYDOM\\administrator password=topsecret reboot.

This example would connect to a computer named XP as the local administrator using password secret, and join the computer into a domain called MYDOM using the MYDOM domain administrator account and password topsecret. After successful join, the computer would reboot.

DOM UNJOIN account=ACCOUNT password=PASSWORD reboot

Unjoins a computer from a domain. This command supports the following additional parameters:

- *ACCOUNT* defines a domain account that will be used to unjoin the machine from the domain. This domain account needs to have sufficient privileges to unjoin machines.
- *PASSWORD* defines the password for the domain account defined with *ACCOUNT*.
- *REBOOT* is an optional parameter that can be set to reboot the remote machine after successful unjoin from the domain.

Note that you also need to use standard net parameters to connect and authenticate to the remote machine that you want to unjoin. These additional parameters include: −S computer and −U user.

Example: net dom unjoin −S xp −U XP\\administrator%secret account=MYDOM\\administrator password=topsecret reboot.

This example would connect to a computer named XP as the local administrator using password secret, and

unjoin the computer from the domain using the MYDOM domain administrator account and password topsecret. After successful unjoin, the computer would reboot.

DOM RENAMECOMPUTER newname=NEWNAME account=ACCOUNT password=PASSWORD reboot
Renames a computer that is joined to a domain. This command supports the following additional parameters:

- *NEWNAME* defines the new name of the machine in the domain.
- *ACCOUNT* defines a domain account that will be used to rename the machine in the domain. This domain account needs to have sufficient privileges to rename machines.
- *PASSWORD* defines the password for the domain account defined with *ACCOUNT*.
- *REBOOT* is an optional parameter that can be set to reboot the remote machine after successful rename in the domain.

Note that you also need to use standard net parameters to connect and authenticate to the remote machine that you want to rename in the domain. These additional parameters include: −S computer and −U user.

Example: net dom renamecomputer −S xp −U XP\\administrator%secret newname=XPNEW account=MYDOM\\administrator password=topsecret reboot.

This example would connect to a computer named XP as the local administrator using password secret, and rename the joined computer to XPNEW using the MYDOM domain administrator account and password topsecret. After successful rename, the computer would reboot.

G_LOCK
Manage global locks.

G_LOCK DO lockname timeout command
Execute a shell command under a global lock. This might be useful to define the order in which several shell commands will be executed. The locking information is stored in a file called g_lock.tdb. In setups with CTDB running, the locking information will be available on all cluster nodes.

- *LOCKNAME* defines the name of the global lock.
- *TIMEOUT* defines the timeout.
- *COMMAND* defines the shell command to execute.

G_LOCK LOCKS
Print a list of all currently existing locknames.

G_LOCK DUMP lockname
Dump the locking table of a certain global lock.

TDB
Print information from tdb records.

TDB LOCKING key [DUMP]
List sharename, filename and number of share modes for a record from locking.tdb. With the optional DUMP options, dump the complete record.

- *KEY* Key of the tdb record as hex string.

vfs
Access shared filesystem through the VFS.

vfs stream2abouble [--recursive] [--verbose] [--continue] [--follow-symlinks] share path
Convert file streams to AppleDouble files.

- *share* A Samba share.

- *path* A relative path of something in the Samba share. "." can be used for the root directory of the share.

Options:

−−recursive
Traverse a directory hierarchy.

−−verbose
Verbose output.

−−continue
Continue traversing a directory hierarchy if a single conversion fails.

−−follow−symlinks
Follow symlinks encountered while traversing a directory.

vfs getntacl share path
Display the security descriptor of a file or directory.

- *share* A Samba share.

- *path* A relative path of something in the Samba share. "." can be used for the root directory of the share.

OFFLINEJOIN
Starting with version 4.15 Samba has support for offline join APIs. Windows supports offline join capabilities since Windows 7 and Windows 2008 R2.

The following offline commands are implemented:
net offlinejoin provision − Provisions a machine account in AD.
net offlinejoin requestodj − Requests a domain offline join.

OFFLINEJOIN PROVISION domain=DOMAIN machine_name=MACHINE_NAME machine_account_ou=MACHINE_ACCOUNT_OU dcname=DCNAME defpwd reuse savefile=FILENAME printblob
Provisions a machine account in AD. This command needs network connectivity to the domain controller to succeed. This command supports the following additional parameters:

- *DOMAIN* can be a NetBIOS domain name (also known as short domain name) or a DNS domain name for Active Directory Domains. The *DOMAIN* parameter cannot be NULL.
- *MACHINE_NAME* defines the machine account name that will be provisioned in AD. The *MACHINE_NAME* parameter cannot be NULL.
- *MACHINE_ACCOUNT_OU* can be set to a RFC 1779 LDAP DN, like *ou=mymachines,cn=Users,dc=example,dc=com* in order to create the machine account in a non−default LDAP container. This optional parameter is only supported when joining Active Directory Domains.
- *DCNAME* defines a specific domain controller for creating the machine account in AD.
- *DEFPWD* is an optional parameter that can be set to enforce using the default machine account password. The use of this parameter is not recommended as the default machine account password can be easily guessed.
- *REUSE* is an optional parameter that can be set to enforce reusing an existing machine account in AD.
- *SAVEFILE* is an optional parameter to store the generated provisioning data on disk.
- *PRINTBLOB* is an optional parameter to print the generated provisioning data on stdout.

Example: net offlinejoin provision −U administrator%secret domain=MYDOM machine_name=MYHOST savefile=provisioning.txt

OFFLINEJOIN REQUESTODJ loadfile=FILENAME

Requests an offline domain join by providing file−based provisioning data. This command supports the following additional parameters:

- *LOADFILE* is a required parameter to load the provisioning from a file.

Example: net offlinejoin requestodj −U administrator%secret loadfile=provisioning.txt

HELP [COMMAND]

Gives usage information for the specified command.

VERSION

This man page is complete for version 3 of the Samba suite.

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 net manpage was written by Jelmer Vernooij.