NAME

rpcclient - tool for executing client side MS-RPC functions

SYNOPSIS

```
rpcclient [-c|--command=COMMANDS] [-I|--dest-ip=IP] [-p|--port=PORT] [-?|--help] [--usage]
  [-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] {BINDING-STRING|HOST}
```

DESCRIPTION

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

rpcclient is a utility initially developed to test MS-RPC functionality in Samba itself. It has undergone several stages of development and stability. Many system administrators have now written scripts around it to manage Windows NT clients from their UNIX workstation.

OPTIONS

BINDING-STRING|HOST

When connecting to a dcerpc service you need to specify a binding string.

The format is:

TRANSPORT:host[options]

where TRANSPORT is either ncacn_np (named pipes) for SMB or ncacn_ip_tcp for DCERPC over TCP/IP.

"host" is an IP or hostname or netbios name. If the binding string identifies the server side of an endpoint, "host" may be an empty string. See below for more details.

"options" can include a SMB pipe name if using the ncacn_np transport or a TCP port number if using the ncacn_ip_tcp transport, otherwise they will be auto-determined.

Examples:

⊕

⊕

⊕

⊕

⊕

⊕

⊕

⊕

The supported transports are:

⊕
- Connect using named pipes

⊕
- Connect over TCP/IP

⊕
- Connect over local RPC (unix sockets)

The supported options are:

⊕
- Use RPC integrity authentication level

⊕
- Enable RPC privacy (encryption) authentication level

⊕
- Use RPC connect level authentication (auth, but no sign or seal)

⊕
- Use RPC packet authentication level

•

- Use SPNEGO instead of NTLMSSP authentication

•

- Use plain NTLM instead of SPNEGO or NTLMSSP

•

- Use Kerberos instead of NTLMSSP authentication

•

- Create a schannel connection

•

- Use SMB1 for named pipes

•

- Use SMB2/3 for named pipes

•

- Enable the NDR validator

•

- Enable debug output of packets

•

- Check reply data for non-zero pad bytes

•

- Use big endian for RPC

•

- Use NDR64 for RPC

-c|--command=<command string>

Execute semicolon separated commands (listed below)

-I|--dest-ip IP-address

IP address is the address of the server to connect to. It should be specified in standard "a.b.c.d" notation.

Normally the client would attempt to locate a named SMB/CIFS server by looking it up via the NetBIOS name resolution mechanism described above in the *name resolve order* parameter above. Using this parameter will force the client to assume that the server is on the machine with the specified IP address and the NetBIOS name component of the resource being connected to will be ignored.

There is no default for this parameter. If not supplied, it will be determined automatically by the client as described above.

-p|--port port

This number is the TCP port number that will be used when making connections to the server. The standard (well-known) TCP port number for an SMB/CIFS server is 139, which is the default.

-?|--help

Print a summary of command line options.

--usage

Display brief usage message.

-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 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 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 ist 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:

- 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.

- 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.

- 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.

- 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 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 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 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 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 smb.conf file. However, a command line setting will take precedence over settings in 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 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 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 seperated 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 connnecting to a service.

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

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

COMMANDS

LSARPC

lsaquery

Query info policy

lookupsids

Convert SIDs to names

lookupsids3

Convert SIDs to names

lookupsids_level

Convert SIDs to names

lookupnames

Convert names to SIDs

lookupnames4
Convert names to SIDs

lookupnames_level
Convert names to SIDs

enumtrust
Enumerate trusted domains

enumprivs
Enumerate privileges

getdispname
Get the privilege name

lsaenumsid
Enumerate the LSA SIDS

lsacreateaccount
Create a new lsa account

lsaenumprivsaccount
Enumerate the privileges of an SID

lsaenumacctrights
Enumerate the rights of an SID

lsaaddpriv
Assign a privilege to a SID

lsadelpriv
Revoke a privilege from a SID

lsaaddacctrights
Add rights to an account

lsaremoveacctrights
Remove rights from an account

lsalookupprivvalue
Get a privilege value given its name

lsaquerysecobj
Query LSA security object

lsaquerytrustdominfo
Query LSA trusted domains info (given a SID)

lsaquerytrustdominfobyname
Query LSA trusted domains info (given a name), only works for Windows > 2k

lsaquerytrustdominfobysid
Query LSA trusted domains info (given a SID)

lsasettrustdominfo
Set LSA trusted domain info

getusername
Get username

createsecret
Create Secret

deletesecret
Delete Secret

querysecret
Query Secret

setsecret
Set Secret

retrieveprivatedata
Retrieve Private Data

storeprivatedata
Store Private Data

createtrustdom

Create Trusted Domain

deletetrustdom
Delete Trusted Domain

LSARPC-DS

dsroledominfo
Get Primary Domain Information

DFS

dfsversion
Query DFS support

dfsadd
Add a DFS share

dfsremove
Remove a DFS share

dfsgetinfo
Query DFS share info

dfsenum
Enumerate dfs shares

dfsenumex
Enumerate dfs shares

SHUTDOWN

shutdowninit
syntax: shutdown [-m message]

shutdownabort
syntax: shutdownabort

SRVSVC

srvinfo
Server query info

netshareenum

Enumerate shares

netshareenumall
: Enumerate all shares

netsharegetinfo
: Get Share Info

netsharesetinfo
: Set Share Info

netsharesetdfsflags
: Set DFS flags

netfileenum
: Enumerate open files

netremotetod
: Fetch remote time of day

netnamevalidate
: Validate sharename

netfilegetsec
: Get File security

netsessdel
: Delete Session

netsessenum
: Enumerate Sessions

netdiskenum
: Enumerate Disks

netconnenum
: Enumerate Connections

netshareadd
: Add share

netsharedel
Delete share

SAMR

queryuser
Query user info

querygroup
Query group info

queryusergroups
Query user groups

queryuseraliases
Query user aliases

querygroupmem
Query group membership

queryaliasmem
Query alias membership

queryaliasinfo
Query alias info

deletealias
Delete an alias

querydispinfo
Query display info

querydispinfo2
Query display info

querydispinfo3
Query display info

querydominfo
Query domain info

enumdomusers
: Enumerate domain users

enumdomgroups
: Enumerate domain groups

enumalsgroups
: Enumerate alias groups

enumdomains
: Enumerate domains

createdomuser
: Create domain user

createdomgroup
: Create domain group

createdomalias
: Create domain alias

samlookupnames
: Look up names

samlookuprids
: Look up names

deletedomgroup
: Delete domain group

deletedomuser
: Delete domain user

samquerysecobj
: Query SAMR security object

getdompwinfo
: Retrieve domain password info

getusrdompwinfo

Retrieve user domain password info

lookupdomain
Lookup Domain Name

chgpasswd
Change user password

chgpasswd2
Change user password

chgpasswd3
Change user password

getdispinfoidx
Get Display Information Index

setuserinfo
Set user info

setuserinfo2
Set user info2

SPOOLSS

adddriver <arch> <config> [<version>]
Execute an AddPrinterDriver() RPC to install the printer driver information on the server. Note that the driver files should already exist in the directory returned by getdriverdir. Possible values for *arch* are the same as those for the getdriverdir command. The *config* parameter is defined as follows:

```
Long Driver Name:\
Driver File Name:\
Data File Name:\
Config File Name:\
Help File Name:\
Language Monitor Name:\
Default Data Type:\
Comma Separated list of Files
```

Any empty fields should be enter as the string "NULL".

Samba does not need to support the concept of Print Monitors since these only apply to local printers whose driver can make use of a bi-directional link for communication. This field should be "NULL". On a remote NT print server, the Print Monitor for a driver must already be installed prior to adding the driver or else the RPC will fail.

The *version* parameter lets you specify the printer driver version number. If omitted, the default driver version for the specified architecture will be used. This option can be used to upload Windows 2000 (version 3) printer drivers.

addprinter <printername> <sharename> <drivername> <port>

Add a printer on the remote server. This printer will be automatically shared. Be aware that the printer driver must already be installed on the server (see adddriver) and the *port*must be a valid port name (see enumports.

deldriver <driver>

Delete the specified printer driver for all architectures. This does not delete the actual driver files from the server, only the entry from the server's list of drivers.

deldriverex <driver> [architecture] [version] [flags]

Delete the specified printer driver and optionally files associated with the driver. You can limit this action to a specific architecture and a specific version. If no architecture is given, all driver files of that driver will be deleted. *flags* correspond to numeric DPD_* values, i.e. a value of 3 requests (DPD_DELETE_UNUSED_FILES | DPD_DELETE_SPECIFIC_VERSION).

enumdata

Enumerate all printer setting data stored on the server. On Windows NT clients, these values are stored in the registry, while Samba servers store them in the printers TDB. This command corresponds to the MS Platform SDK GetPrinterData() function (* This command is currently unimplemented).

enumdataex

Enumerate printer data for a key

enumkey

Enumerate printer keys

enumjobs <printer>

List the jobs and status of a given printer. This command corresponds to the MS Platform SDK EnumJobs() function getjob

Get print job

setjob

Set print job

enumports [level]

Executes an EnumPorts() call using the specified info level. Currently only info levels 1 and 2 are supported.

enumdrivers [level]

Execute an EnumPrinterDrivers() call. This lists the various installed printer drivers for all architectures. Refer to the MS Platform SDK documentation for more details of the various flags and calling options. Currently supported info levels are 1, 2, and 3.

enumprinters [level]

Execute an EnumPrinters() call. This lists the various installed and share printers. Refer to the MS Platform SDK documentation for more details of the various flags and calling options. Currently supported info levels are 1, 2 and 5.

getdata <printername> <valuename;>

Retrieve the data for a given printer setting. See the enumdata command for more information. This command corresponds to the GetPrinterData() MS Platform SDK function.

getdataex

Get printer driver data with keyname

getdriver <printername>

Retrieve the printer driver information (such as driver file, config file, dependent files, etc...) for the given printer. This command corresponds to the GetPrinterDriver() MS Platform SDK function. Currently info level 1, 2, and 3 are supported.

getdriverdir <arch>

Execute a GetPrinterDriverDirectory() RPC to retrieve the SMB share name and subdirectory for storing printer driver files for a given architecture. Possible values for *arch* are "Windows 4.0" (for Windows 95/98), "Windows NT x86", "Windows NT PowerPC", "Windows Alpha_AXP", and "Windows NT R4000".

getdriverpackagepath

Get print driver package download directory getprinter <printername>

Retrieve the current printer information. This command corresponds to the GetPrinter() MS Platform SDK function.

openprinter <printername>

Execute an OpenPrinterEx() and ClosePrinter() RPC against a given printer.

openprinter_ex <printername>

Open printer handle

setdriver <printername> <drivername>

Execute a SetPrinter() command to update the printer driver associated with an installed printer. The printer driver must already be correctly installed on the print server.

See also the enumprinters and enumdrivers commands for obtaining a list of of installed printers and drivers.

getprintprocdir

Get print processor directory

addform

Add form

setform

Set form

getform

Get form

deleteform

Delete form

enumforms

Enumerate form

setprinter

Set printer comment

setprinterdata

Set REG_SZ printer data

setprintername <printername> <newprintername>
Set printer name

rffpcnex
Rffpcnex test

printercmp
Printer comparison test

enumprocs
Enumerate Print Processors

enumprocdatatypes
Enumerate Print Processor Data Types

enummonitors
Enumerate Print Monitors

createprinteric
Create Printer IC

playgdiscriptonprinteric
Create Printer IC

getcoreprinterdrivers
Get CorePrinterDriver

enumpermachineconnections
Enumerate Per Machine Connections

addpermachineconnection
Add Per Machine Connection

delpermachineconnection
Delete Per Machine Connection

NETLOGON

logonctrl2
Logon Control 2

getanydcname
: Get trusted DC name

getdcname
: Get trusted PDC name

dsr_getdcname
: Get trusted DC name

dsr_getdcnameex
: Get trusted DC name

dsr_getdcnameex2
: Get trusted DC name

dsr_getsitename
: Get sitename

dsr_getforesttrustinfo
: Get Forest Trust Info

logonctrl
: Logon Control

samlogon
: Sam Logon

change_trust_pw
: Change Trust Account Password

gettrustrid
: Get trust rid

dsr_enumtrustdom
: Enumerate trusted domains

dsenumdomtrusts
: Enumerate all trusted domains in an AD forest

deregisterdnsrecords

Deregister DNS records

netrenumtrusteddomains
Enumerate trusted domains

netrenumtrusteddomainsex
Enumerate trusted domains

getdcsitecoverage
Get the Site-Coverage from a DC

capabilities
Return Capabilities

logongetdomaininfo
Return LogonGetDomainInfo

FSRVP

fss_is_path_sup
Check whether a share supports shadow-copy

fss_get_sup_version
Get supported FSRVP version from server

fss_create_expose
Request shadow-copy creation and exposure

fss_delete
Request shadow-copy share deletion

fss_has_shadow_copy
Check for an associated share shadow-copy

fss_get_mapping
Get shadow-copy share mapping information

fss_recovery_complete
Flag read-write snapshot as recovery complete,

CLUSAPI

clusapi_open_cluster
: Open cluster

clusapi_get_cluster_name
: Get cluster name

clusapi_get_cluster_version
: Get cluster version

clusapi_get_quorum_resource
: Get quorum resource

clusapi_create_enum
: Create enum query

clusapi_create_enumex
: Create enumex query

clusapi_open_resource
: Open cluster resource

clusapi_online_resource
: Set cluster resource online

clusapi_offline_resource
: Set cluster resource offline

clusapi_get_resource_state
: Get cluster resource state

clusapi_get_cluster_version2
: Get cluster version2

clusapi_pause_node
: Pause cluster node

clusapi_resume_node
: Resume cluster node

DRSUAPI

dscracknames
: Crack Name

dsgetdcinfo
: Get Domain Controller Info

dsgetncchanges
: Get NC Changes

dswriteaccountspn
: Write Account SPN

ECHO

echoaddone
: Add one to a number

echodata
: Echo data

sinkdata
: Sink data

sourcedata
: Source data

EPMAPPER

epmmap
: Map a binding

epmlookup
: Lookup bindings

EVENTLOG

eventlog_readlog
: Read Eventlog

eventlog_numrecord
: Get number of records

eventlog_oldestrecord

Get oldest record

eventlog_reportevent
Report event

eventlog_reporteventsource
Report event and source

eventlog_registerevsource
Register event source

eventlog_backuplog
Backup Eventlog File

eventlog_loginfo
Get Eventlog Information

IRemoteWinspool

winspool_AsyncOpenPrinter
Open printer handle

winspool_AsyncCorePrinterDriverInstalled
Query Core Printer Driver Installed

NTSVCS

ntsvcs_getversion
Query NTSVCS version

ntsvcs_validatedevinst
Query NTSVCS device instance

ntsvcs_hwprofflags
Query NTSVCS HW prof flags

ntsvcs_hwprofinfo
Query NTSVCS HW prof info

ntsvcs_getdevregprop
Query NTSVCS device registry property

ntsvcs_getdevlistsize
Query NTSVCS device list size

ntsvcs_getdevlist
Query NTSVCS device list

MDSSVC

fetch_properties
Fetch connection properties

fetch_attributes
Fetch attributes for a CNID

WINREG

winreg_enumkey
Enumerate Keys

querymultiplevalues
Query multiple values

querymultiplevalues2
Query multiple values

WITNESS

GetInterfaceList
List the interfaces to which witness client connections can be made

Register
Register for resource state change notifications of a NetName and IPAddress

UnRegister
Unregister for notifications from the server

AsyncNotify
Request notification of registered resource changes from the server

RegisterEx
Register for resource state change notifications of a NetName, ShareName and multiple IPAddresses

WKSSVC

wkssvc_wkstagetinfo
    Query WKSSVC Workstation Information

wkssvc_getjoininformation
    Query WKSSVC Join Information

wkssvc_messagebuffersend
    Send WKSSVC message

wkssvc_enumeratecomputernames
    Enumerate WKSSVC computer names

wkssvc_enumerateusers
    Enumerate WKSSVC users

GENERAL OPTIONS

help
    Get help on commands

?
    Get help on commands

debuglevel
    Set debug level

debug
    Set debug level

list
    List available commands on pipe

exit
    Exit program

quit
    Exit program

sign
    Force RPC pipe connections to be signed

seal

Force RPC pipe connections to be sealed

packet

Force RPC pipe connections with packet authentication level

schannel

Force RPC pipe connections to be sealed with ’schannel’. Force RPC pipe connections to be sealed with ’schannel’. Assumes valid machine account to this domain controller.

schannelsign

Force RPC pipe connections to be signed (not sealed) with ’schannel’. Assumes valid machine account to this domain controller.

timeout

Set timeout (in milliseconds) for RPC operations

transport

Choose ncacn transport for RPC operations

none

Force RPC pipe connections to have no special properties

BUGS

rpcclient is designed as a developer testing tool and may not be robust in certain areas (such as command line parsing). It has been known to generate a core dump upon failures when invalid parameters where passed to the interpreter.

From Luke Leighton’s original rpcclient man page:

WARNING! The MSRPC over SMB code has been developed from examining Network traces. No documentation is available from the original creators (Microsoft) on how MSRPC over SMB works, or how the individual MSRPC services work. Microsoft’s implementation of these services has been demonstrated (and reported) to be... a bit flaky in places.

The development of Samba’s implementation is also a bit rough, and as more of the services are understood, it can even result in versions of **smbd**(8) and **rpcclient**(1) that are incompatible for some commands or services. Additionally, the developers are sending reports to Microsoft, and problems found or reported to Microsoft are fixed in Service Packs, which may result in incompatibilities.

VERSION

This man page is part of version 4.16.11 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 original rpcclient man page was written by Matthew Geddes, Luke Kenneth Casson Leighton, and rewritten by Gerald Carter. 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.