IT資格道場

LPIC-3 300試験 学習用テキスト

予備知識ゼロから合格ラインへ —— 読んで理解する1冊目

無料サンプル(第1章まで)

LPI公式 LPIC-3 Exam 300 Objectives Version 3.0(300-300・2026-11-24 取得)準拠
2026年11月版

SAMPLE

ご利用にあたって

- このテキストは「IT資格道場」(www.shikaku-doujou.com)の購入者限定テキストのうち、学習用テキストの第1章までを収録した**無料サンプル**です。どなたでもご覧いただけます。
- 無料サンプルも著作物です。転載・改変しての再配布・二次販売はできません。
- AI モデルの学習データとしてのご利用(学習目的の入力・データセット化を含む)はできません。
- 製品版(学習用テキスト・頻出度別ノート・暗記用ノートの3点)は、ご購入者を識別できる透かし入りのPDFとして提供されます。
- 本PDFはスマートフォン・タブレットでの閲覧に合わせたA5版面・文字サイズで組版しています。

SAMPLE

目次

この教材群の使い方(3点セット)

第1章 Samba Basics [301.1/301.2/301.3/301.4]

- Samba の部品とデーモンの役割 [301.1]
- SMB/CIFS と AD が使うポート [301.1]
- SMB のバージョンと混在ネットワークの課題 [301.1]
- Samba 3 と Samba 4、VFS モジュール、CTDB [301.1]
- smb.conf の形、include、変数 [301.2]
- server role と security [301.2]
- server string、server services、vfs objects [301.2]
- Samba の TLS 設定 [301.2]
- レジストリに置く設定 [301.2]
- net registry と net conf [301.2]
- testparm と設定のトラブルの切り分け [301.2]
- Samba を設定する Windows の道具 [301.2]
- Samba サービスの起動と停止 [301.3]
- smbcontrol と smbstatus [301.3]
- TDB ファイルのバックアップと戻し [301.3]
- AD DC のバックアップと戻し [301.3]
- 仮想化が DC に与える影響 [301.3]
- ログの設定 [301.4]
- パスワードデータベースの照会と変更 [301.4]
- 主な TDB ファイルと壊れの見分け方 [301.4]
- sam.ldb の照会と変更（ldb ツール） [301.4]
- ごみ箱と dbcheck [301.4]
- 名前を変えた DC の複製とイベントログ [301.4]

SAMPLE

rpcclient によるサーバーの照会 [301.4]

サンプルはここまでです

SAMPLE

この教材群の使い方(3点セット)

種類	役割
① 学習用テキスト	これだけで問題が解けるようになる本編
② 頻出度別ノート	頻出順に絞った問題と解説で「解ける」に橋渡し。試験直前の最終チェックにも
③ 暗記用ノート	覚えるべき要点を一問一答に凝縮。空き時間の反復と用語の再確認用

使い方: ① 学習用を読む → ② 頻出度別で解き方を掴む → ③ 問題演習で腕試し → ④ 頻出度別に戻って再確認(試験直前もここ)。暗記用は空き時間や用語の再確認に使います。

このテキストの位置づけ: ①の学習用テキストです。まず通読します。これだけで問題が解ける状態が目標です。

- 対象試験: LPIC-3 Mixed Environments（Exam 300）（300-300、出題範囲 Version 3.0）
- 形式: 60問・90分。多肢選択と穴埋め。合格は 200〜800 のスケールで 500点
- 構成: 全5章。各章の見出しに細目の番号つき
- 細目: Samba の基礎 301・Samba と AD ドメイン 302・共有の設定 303・クライアントの設定 304・Linux の ID 管理とファイル共有 305
- 脚注: 巻末の「商標と出典」に集約

SAMPLE

第1章 Samba Basics [301.1/301.2/301.3/301.4]

この章は4つの細目からなります。

- Samba の概念と構造（301.1）
- Samba の設定（301.2）
- 通常のメンテナンス（301.3）
- トラブルシューティング（301.4）

試験は Samba 4.8 以上が対象です。道具の名前、設定項目の綴り、既定値、ポート番号が問われます。

Samba の部品とデーモンの役割 [301.1]

Samba は、UNIX 系の OS に SMB プロトコルを話させ、Active Directory の機能も提供するソフトウェアの集まりです。部品の中心は4つのデーモンで、役割は次のとおりです：

デーモン	役割	使う場面
smbd	ファイル共有と印刷のサービスを Windows クライアントに提供する	ファイルサーバー、ドメインメンバー
nmbd	NetBIOS over IP の名前サービスに応答し、ブラウジングに参加する。WINS サーバーにもなれる	NetBIOS の名前解決が要る環境

SAMPLE

デーモン	役割	使う場面
samba	Active Directory、ファイル共有、印刷を提供する。SMB のほか DCE/RPC、LDAP、Kerberos も話す	AD ドメインコントローラー（AD DC）
winbindd	NSS（Name Service Switch）、PAM、ntlm_auth、Samba 自身に、ドメインのユーザーとグループの情報を渡す	ドメインメンバー

ここで取り違えやすいのが smbd と samba の違いです。smbd はファイルと印刷だけを受け持ちます。samba は Samba 4 で加わった AD DC 用のデーモンで、AD の機能に加えてファイルと印刷も受け持ちます。AD DC では samba が、smbd と winbindd をサブプロセスとして自分で起動するので、別に動かす必要はありません。[1]

NetBIOS は nmbd が実装し、UDP 137 で名前の問い合わせに答えます。WINS サーバーとして動くこともでき、WINS プロキシとしてブロードキャストの問い合わせを中継することもできます。

winbindd には、NSS を使わない動かし方もあります。smbd、ntlm_auth、pam_winbind.so のために DC への接続だけを管理させる形で、`idmap config * : range` が要りません。man ページはこれを netlogon proxy only mode と呼びます。

役割ごとに動かすデーモンを並べると、次の組み合わせになります。

構成	動かすもの
スタンドアロンのファイルサーバー	smbd（名前解決が要るなら nmbd も）

SAMPLE

構成	動かすもの
AD ドメインメンバー	smbd と winbindd
AD DC	samba だけ（smbd、nmbd、winbindd は別に起動しない）

SMB/CIFS と AD が使うポート [301.1]

AD DC は多くのサービスを同時に開きます。ファイアウォールの設定と、どのサービスがどのポートかを結びつける問いが出ます。[2]

サービス	ポート	補足
DNS	53（tcp/udp）	Samba 内蔵の DNS か BIND 9 が提供する
Kerberos	88（tcp/udp）	
kpasswd（パスワード変更）	464（tcp/udp）	
RPC エンドポイントマッパー	135（tcp）	DCE/RPC Locator Service
LDAP	389（tcp/udp）	
LDAPS	636（tcp）	`tls enabled = yes`（既定）の場合に開く
Global Catalog	3268（tcp）	SSL 版は 3269（tcp）
NetBIOS Name Service	137（udp）	nmbd
NetBIOS Datagram	138（udp）	
NetBIOS Session	139（tcp）	NetBIOS 上の SMB
SMB over TCP	445（tcp）	NetBIOS を介さない SMB

SAMPLE

サービス	ポート	補足
動的 RPC	49152〜65535（tcp）	Windows Server 2008 以降の範囲と同じ

smbd の既定の待ち受けは 139 と 445 です。`smb ports`（Samba 4.23 以降は `server smb transports` が本来の名前で、`smb ports` はその別名）の既定は、tcp トランスポートが 445、nbt が 139 です。

動的 RPC のポートは、Samba 4.7 より前は TCP 1024〜1300 でした。4.7 以降は `rpc server port` で手動指定もできます。既定は 0 で、`rpc server dynamic port range` の最初の空きポートを選び、この設定が効くのは AD DC として動くときだけです。

LDAPS（636）が開くかどうかは `tls enabled` で決まり、no にすると開きません。

ファイアウォールの内側に置いた AD DC を使う場面では、認証に 88、パスワード変更に 464、ディレクトリの検索に 389、ファイル共有に 445、名前解決に 53 が要ります。LDAPS で DC にバインドするアプリがあれば 636 を、Global Catalog を引くアプリがあれば 3268（SSL なら 3269）を足します。DC 側では、動的 RPC の範囲 49152〜65535 も開けておきます。

SMB のバージョンと混在ネットワークの課題 [301.1]

SMB には世代があり、世代ごとに Windows の対応 OS と機能が違い、Samba の設定では方言名で最小と最大を指定します。[3]

世代	Samba の方言名	登場した Windows	備考
SMB 1.0	NT1	Windows NT	CIFS と呼ばれる

SAMPLE

世代	Samba の方言名	登場した Windows	備考
SMB 2.0	SMB2_02	Vista、Server 2008	最初の SMB2
SMB 2.1	SMB2_10	Windows 7	
SMB 3.0	SMB3_00	Windows 8、Server 2012	暗号化、マルチチャネルが加わる
SMB 3.02	SMB3_02	Windows 8.1	
SMB 3.11	SMB3_11	Windows 10	SMB3 と書くと既定でこの版が選ばれる

`SMB2` と書くと SMB2_10 が、`SMB3` と書くと SMB3_11 が選ばれます。

最小と最大の既定は次のとおりです：

設定	既定	意味
server min protocol	SMB2_02	サーバーが許す最も古い版。4.11 からこの既定になり、SMB1（NT1）は既定では許可されない
client min protocol	SMB2_02	クライアント側の最小（4.11 からこの既定）
server max protocol	SMB3	サーバーが話す最も新しい版
ntlm auth	ntlmv2-only	NTLMv2 のログオンだけを許す

SMB1 しか話せない古いクライアントがあるときに限り、`server min protocol = NT1` を設定します。コマンドラインのツールは `--option='client min`

SAMPLE

`protocol=NT1'` で、その実行だけ SMB1 を許せます。

SMB2 以降の機能のうち、問われるのは次の4つです：

機能	設定項目と既定	条件
SMB2 リース	`smb2 leases` = yes	SMB1 の oplock よりも積極的にクライアントがキャッシュできる。`oplocks = yes` と `kernel oplocks = no` が必要
durable handle	`durable handles` = yes	`kernel oplocks`、`kernel share modes`、`posix locking` がすべて no のときだけ有効
マルチチャネル	`server multi channel support`	4.4 で追加。4.14 まで実験的。4.15 以降は Linux と FreeBSD で既定が yes
SMB レベルの暗号化	`client smb encrypt`、`server smb encrypt`	SMB 3.0 以降（Windows 8 以降）。値は off、if_required、desired、required、default（if_required と同じ）

異種混在のネットワークでは、Windows と UNIX の違いが NetBIOS、ID、認証の3つの設定に現れます。

NetBIOS では、`disable netbios` の既定は no ですが、NetBIOS は Windows 2000 より前の Windows で唯一のブラウズ方式なので、yes にすると、NetBIOS だけを話す古いクライアントからサーバーが見えなくなります。

SAMPLE

ID の違いは、Windows が SID でユーザーを識別し、UNIX が UID と GID で識別する点にあります。メンバーサーバーでは、DC 側のアカウントに加えて UNIX 側にも対応するユーザーが要り、それを供給するのが winbind です。

認証では、既定で NTLMv2 のログオンだけを許し、それ以外は受け付けません。`security = ads` は Kerberos と、`net` によるドメイン参加が前提です。

Windows 7 以降のクライアントだけが接続する新しいファイルサーバーで、SMB1 を無効のまま保ちたいとします。設定は何も足さなくて済みます。`server min protocol` の既定が SMB2_02 なので、SMB1 は初めから許可されていません。Windows XP 専用機が1台だけ接続できず、その機械の入れ替えを待つ間は、`server min protocol = NT1` を書いて SMB1 を許可します。この設定は全クライアントに SMB1 を開くので、入れ替えが済んだら削除します。

Samba 3 と Samba 4、VFS モジュール、CTDB [301.1]

Samba 3 系の中心は、NT4 風のドメインとファイル・印刷サービスの実装でした。この系統の役割は今も smbd と nmbd に残っています。smb.conf では `server role = classic primary domain controller` と `classic backup domain controller` が、NT4 風ドメインのログオンサービスを提供する役割です。Samba 4 の違いは、AD DC になれることです。`server role = active directory domain controller` を指定して samba デーモンで動かします。`security = ads` はメンバーサーバーの設定で、AD DC にはなりません。[4]

比べる点	Samba 3 系の使い方	Samba 4 で加わったもの
ドメイン	NT4 風（classic PDC・BDC）	Active Directory（AD DC）
中心のデーモン	smbd・nmbd	samba（AD DC）。ファイルサーバーは smbd

SAMPLE

比べる点	Samba 3 系の使い方	Samba 4 で加わったもの
話すプロトコル	SMB・NetBIOS	DCE/RPC、LDAP、Kerberos、DNS も

VFS（Virtual File System）モジュールは、smbd の I/O の層に差し込んで、Samba の機能を拡張する部品です。`vfs objects` に名前を並べて使います。試験で知っておく名前は次の4つです：

モジュール	働き
acl_xattr	細かい粒度の Windows ACL を使えるようにする
recycle	削除したファイルをすぐ消さず、一時ディレクトリへ移す
full_audit	選んだクライアント操作を記録する
shadow_copy2	スナップショットを Windows クライアントにシャドウコピーとして見せる（shadow_copy の強化版）

CTDB は、クラスタ化した Samba のためのクラスタ DB コンポーネントです。高可用性と負荷分散を備えた CIFS サーバーのクラスタを作ります。主な機能は3つです。

- TDB のクラスタ版（ノードが落ちても自動で再構築する）
- ノードとサービスの監視
- 公開 IP アドレスのプールの管理

クラスタ内の全ノードから1台のリーダーが選ばれ、DB のリカバリや IP のフェイルオーバーなど、特権の要る操作をまとめます（旧称は recovery master）。

SAMPLE

CTDB は Samba 4.2.0 以降に含まれ、クラスタの全ノードに Samba と CTDB が要ります。

smb.conf の形、include、変数 [301.2]

smb.conf は Samba のサーバー設定ファイルです。既定の場所はビルド時に決まり、文書によって書かれた場所が違います（smbd(8) は `/usr/local/samba/lib/smb.conf`、testparm(1) などは `${prefix}/etc/smb.conf`）。ビルド時に決まった設定ファイルの場所や `private dir` は、`smbd -b` の出力（CONFIGFILE、LOCKDIR、STATEDIR、CACHEDIR、PRIVATE_DIR）で確かめられます。ディストリビューションのパッケージでは `/etc/samba/smb.conf` に置かれることが多くなります。[5]

ファイルは `[global]` の節と、共有ごとの節からなります。`[global]` はサーバー全体の設定で、共有の節は共有名がそのまま節の名前になります。項目は `名前 = 値` の形で書きます。

```
[global]
   workgroup = EXAMPLE
   server role = standalone
   server string = Samba %v on %h
   log file = /var/log/samba/log.%m

[docs]
   path = /srv/docs
   read only = no
```

設定を変えたあとの反映は3通りです。

SAMPLE

方法	動き
何もしない	smbd は smb.conf と include 先を3分ごとに見て、変更があれば再読込する
`smbcontrol all reload-config`	smbd、nmbd、winbindd に再読込させる
SIGHUP を送る	smbd が再読込する

再読込は、すでに確立した接続には影響しません。新しい接続から新しい設定が効きます。

`include` は別のファイルを、その行の位置に文字どおり差し込みます（既定値は空）。`include = /etc/samba/smb.conf.%m` のように、クライアントごとに別ファイルを読ませる使い方があります。`include` の値には、`%u`、`%P`、`%S` を使えません。

変数は、値の中の `%` で始まる記号が実行時に置き換えられる仕組みです。

変数	置き換わる値
%u	接続しているユーザー名
%m	クライアントの NetBIOS 名
%h	サーバーのホスト名
%v	Samba のバージョン

`path = /tmp/%u` と書いた共有に john が接続すると、パスは `/tmp/john` と解釈されます。`log file = /var/log/samba/%m.log` なら、クライアントごとに別のログが作られます。

`%m` には落とし穴があります。クライアントが 445 番ポートで接続すると、NetBIOS 名をサーバーに送ってきません。このため、445 を待ち受けている間

SAMPLE

は `%m` が使えません。`smb ports = 139` にして 139 だけで待ち受けると、`%m` が使えるようになります。

server role と security [301.2]

`server role` は Samba の基本の動作モードを決め、`security` は認証のやり方を決めます。どちらも既定は auto で、互いを参照し合い、片方が書かれていれば、もう片方を推定します。どちらも書かないと、ドメインに参加していない単純なファイルサーバーとして動きます。[6]

server role の値	動き
auto（既定）	`security` の設定に従う
standalone	ドメインに参加せず、このマシンのアカウントで認証する
member server	`net` でドメインに参加し、認証を DC に渡す
classic primary domain controller	NT4 風ドメインの PDC としてログオンサービスを提供する
classic backup domain controller	同じく BDC
active directory domain controller	AD DC として動く（`domain controller` と書いても同じ）
IPA primary domain controller	FreeIPA の導入ツールが設定する値で、管理者が手で書くものではない

security の値	動き
auto（既定）	`server role` の設定に従う

SAMPLE

security の値	動き
user	クライアントがユーザー名とパスワードでログオンする。`server role` を書かなければ実質この値。スタンドアロンのファイルサーバーや DC で使う
ads	AD のドメインメンバーとして動く。Kerberos の設定と、`net` によるドメイン参加が要る。この値は `require strong key` と `client schannel` を yes に強制する
domain	Windows NT ドメインのメンバーとして動く。`net` で参加し、NT の PDC か BDC に認証を渡す。NT4 時代の方式

覚える組み合わせは3つです。

- スタンドアロン: `server role = standalone` または `security = user`
- AD ドメインメンバー: `security = ads`（`server role = member server` を書いても、メンバーとして動く）
- AD DC: `server role = active directory domain controller`

`security = ads` は、AD DC になる設定ではありません。ドメインメンバーのための設定です。

メンバーサーバーには追加の条件があります。DC 側にアカウントがあるだけでは足りず、UNIX 側にも対応するユーザーが要り、それを winbind が供給できます。

ゲスト共有（パスワードなしの共有）を主に作るなら、`security = user` と `map to guest` の組み合わせを使います。共有の名前は認証が済んだあとにサーバ

SAMPLE

ーへ届くので、未知のユーザーを guest に割り当てる設定がなければ、`security = user` でゲスト共有は成り立ちません。

ドメインに参加せず、ローカルのユーザーだけで使う小さな共有なら、`server role = standalone` です。すでにある AD に参加させるファイルサーバーなら、メンバーサーバー（`security = ads`）です。ドメインコントローラーそのものを作るなら、`server role = active directory domain controller` を指定して samba デーモンを動かします。

server string、server services、vfs objects [301.2]

3つの項目は、名前が似ていても働きが異なります。

項目	働き	既定
server string	印刷マネージャのコメント欄や `net view` の IPC 接続の横に出る文字列。ブラウズリストでマシン名の横にも出る	`Samba %v`（%v がバージョン、%h がホスト名に置き換わる）
server services	samba デーモンが起動するサービスの一覧	下の表
vfs objects	共有の I/O を担うバックエンド（VFS モジュール）の名前を並べる	空（通常のディスク I/O）

`server services` の既定の値は次のとおりです：

```
server services = s3fs, rpc, nbt, wrepl, ldap, cldap, kdc, drepl,
ft_scanner, winbindd, ntp_signd, kcc, dnsupdate, dns
```

SAMPLE

書き方は2通りです。値を書くと、既定の一覧を丸ごと置き換えます。`+` や `-` を付けると、既定の一覧に足したり、そこから引いたりできます。例は `server services = -s3fs, +smb` で、s3fs を外して smb を足します。[7]

`vfs objects` は、書き直すと以前の定義を上書きします。複数のモジュールは空白で並べ、`vfs objects = extd_audit recycle` のように書きます。

Samba の TLS 設定 [301.2]

LDAPS（636）などの暗号化通信は、`tls` で始まる項目で設定します。5つの項目を、既定値と合わせて覚えます。[8]

項目	指すもの	既定
tls enabled	可能なときに TLS を使うか	yes
tls keyfile	RSA 秘密鍵（PEM）。パスフレーズなしで読めること	`tls/key.pem`
tls certfile	サーバー証明書（PEM）。中間証明書も含められる	`tls/cert.pem`
tls cafile	信頼する root CA の証明書（PEM）	`tls/ca.pem`
tls dh params file	DH 暗号が使う Diffie-Hellman パラメーターのファイル	空（未設定）

パスが `/` で始まらないと、`private dir` からの相対パスと見なされます。AD DC を作ると、証明書と秘密鍵は `private/tls/` の下に自動で作られます。

`tls priority` は、GnuTLS を使う部分（AD DC）で有効にする TLS プロトコルの指定で、既定は `NORMAL:-VERS-SSL3.0` です（SSL 3.0 は無効）。

SAMPLE

AD DC の証明書を取り替えたあとは、LDAP サーバーに `smbcontrol ldap_server reload-certs` を送ります（Samba 4.19 以降）。再起動は要りません。4.18 までは証明書を起動時にしか読まないので、取り替えたら samba を再起動します。

```
[global]
   tls enabled  = yes
   tls keyfile  = tls/key.pem
   tls certfile = tls/cert.pem
   tls cafile   = tls/ca.pem
```

`ldaps://` で DC に接続できないときは、まず `tls enabled` が yes か（no なら 636 は開きません）、次に `tls keyfile`、`tls certfile`、`tls cafile` が指すファイルが `private dir` の下に実在し、鍵が読めるかを確かめます。証明書を新しくしたあとは、Samba 4.19 以降なら `reload-certs` を送れば足り、再起動は要りません。

レジストリに置く設定 [301.2]

Samba は、設定をファイルの代わりにレジストリへ置けます。レジストリ設定は、Samba 3.2.0 から使えます。保存先は `HKLM\Software\Samba\smbconf` のキーです。[9]

使い方には2つの段階があります。

段階	設定	読み込まれるもの
共有だけ	`registry shares = yes`	共有の定義がレジストリから、要求に応じて読み込まれる

SAMPLE

段階	設定	読み込まれるもの
グローバル設定も	`config backend = registry`（推奨）、または `include = registry`（混在）	グローバル設定もレジストリから読み込まれる

- `config backend` の値は file（既定）か registry です。registry にすると、それまでに読んだ設定は捨てられ、グローバル設定がレジストリから読まれます。レジストリの中で `config backend` を設定することはできません。
- `registry shares` の既定は no ですが、`config backend = registry` のときは yes になります。同じ名前の共有が smb.conf とレジストリにあると、smb.conf の共有が優先されます。
- `include = registry` は、グローバル設定をレジストリから読み込む特別な意味を持ち、registry という名前のファイルは読みません。`registry shares` も自動で有効になります。

型は Windows のレジストリと同じで、REG_SZ は NULL で終わる文字列、REG_MULTI_SZ は、NULL で終わる文字列の並びを空文字列で終えたものです。`net registry setvalue` の type には、`sz`、`multi_sz`、`dword` のどれかを指定し、`multi_sz` 以外は value を複数回指定できません。

`samba-regedit` は ncurses で作られた端末用のツールで、キー、サブキー、値を表示し、編集できます。

共有を足すたびに smb.conf を書き換えたくないときは、`registry shares = yes` にして、共有の定義だけをレジストリに置きます。`net conf addshare` で作った共有は、要求に応じて読み込まれます。グローバル設定は smb.conf のままで、smb.conf に同じ名前の共有があれば、そちらが優先され、レジストリ側は使われません。

SAMPLE

net registry と net conf [301.2]

レジストリを操作する道具は2系統あり、`net registry` は、キーと値を直接扱う汎用の入り口です。`net conf` は、smb.conf の形でレジストリの設定を読み書きするための専用の入り口で、`net rpc conf` にすればリモートの Samba の設定も扱えます。[10]

`net registry` の副コマンドは次のとおりです：

副コマンド	働き
enumerate、enumerate_recursive	キーの下のサブキーと値を一覧する（後者はサブキーの中まで）
createkey	新しいキーを作る
deletekey、deletekey_recursive	キーを消す（後者はサブキーごと）
getvalue、getvalueraw	値の型と中身を表示する（後者は中身だけ）
setvalue	既存のキーに値を設定する（型は sz、multi_sz、dword）
increment	DWORD の値をロックを取って増やす（既定は 1）
deletevalue	値を消す
getsd、getsd_sddl、setsd_sddl	セキュリティ記述子を取得、設定する
import、export	`.reg` ファイルの取り込みと書き出し
convert	`.reg` ファイルを変換する
check	レジストリの DB を検査、修復する

SAMPLE

`import` に `--precheck <check-file>` を付けると、チェックファイルに書いたキーや値の有無を先に調べ、成功したときだけ取り込みます。`[key]` はキーがあること、`[-key]` はないことを、`name=value` は値があること、`name=-` はないことを確かめます。

`net registry check [-ravTl] [-o ODB] [--wipe] [DB]` は、オプションなしなら読み取り専用の検査で、`-r` が対話での修復、`-a` が自動での修復、`-T` がドライランです。

`net conf` の副コマンドは次のとおりです：

副コマンド	働き
list	レジストリの設定を smb.conf の形で出力する
import	smb.conf 形式のファイルを取り込む。節を指定しなければ、既存のレジストリ設定を消してから入れ直す。`-T` は試験表示だけ
listshares	レジストリの共有名の一覧
showshare	共有の定義を表示する。`global` を指定すればグローバル設定が出る
addshare	共有を作る（共有名は global にできない）
delshare、drop	共有を消す、設定全体を消す
setparm、getparm、delparm	項目を保存する、取得する、消す
getincludes、setincludes、delincludes	共有の include を扱う

smb.conf を、レジストリを使う構成へ切り替える流れは次のとおりです：

SAMPLE

```
$ net conf import /etc/samba/smb.conf
$ net conf setparm global "server string" "Registry Samba"
$ net conf addshare data /srv/data writeable=y guest_ok=n "data share"
$ net conf list
```

そのあと smb.conf の `[global]` に `config backend = registry` を書くと、それまでに読んだ設定は捨てられ、グローバル設定がレジストリから読まれます。

testparm と設定のトラブルの切り分け [301.2]

testparm は、smb.conf の内部的な正しさを検査する構文チェッカーです。エラーがあれば終了コード 1 を、なければ 0 を返します。ただし、サービスが実際に使える保証にはなりません。パスが存在するか、ポートが空いているかまでは見ません。AD DC として動かしているときは、`samba-tool testparm` を使います。[11]

```
$ testparm -s /etc/samba/smb.conf
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Server role: ROLE_STANDALONE
```

設定ファイル名を省くと、既定の smb.conf を検査します。ホスト名を渡すなら、IP アドレスも省けません。片方だけだと `You must specify both a machine name and an IP address.` で止まります。

`Loaded services file OK.` が構文の検査を通った印です。`Server role:` の行は、設定から決まった役割を示します。項目名を書き間違えると、`Unknown parameter encountered` の警告が出ます。

SAMPLE

オプション	働き
-s、--suppress-prompt	Enter の待ちを省いて、設定をすぐ出力する
-v、--verbose	smb.conf に書いていない、既定値のままの項目も出力する
--parameter-name=名前	その項目だけを出力する（節を指定しなければ global に限る）
--section-name=名前	その節だけを出力する
--show-all-parameters	型や取りうる値も表示する
設定ファイル名のあとに hostname と hostIP	`hosts allow` と `hosts deny` により、そのホストが各サービスへ接続できるかを確かめる

切り分けは、構文、反映、ログの順に進めます：

1. 構文の検査（`testparm`）
2. 変更の反映（`smbcontrol all reload-config`）
3. ログのレベルを上げた、動きの確認（301.4 で扱う `log level`）

共有に接続できないときは、`testparm` が `Loaded services file OK.` を返しても、原因が設定の外にあるとは限りません。`hosts allow` や `hosts deny` が怪しいときは、`testparm /etc/samba/smb.conf client01 192.168.1.50` のように設定ファイル名、ホスト名、IP アドレスを渡すと、そのホストが各サービスに接続できるかが表示されます。

Samba を設定する Windows の道具 [301.2]

Samba の AD DC は、Windows の管理ツールから管理できます。道具は5つです。[12]

SAMPLE

道具	働き
RSAT	Remote Server Administration Tools。Windows から AD を管理するツールの集まり。Windows Server に含まれ、Windows 10 の 1809 以降はオプション機能として入れる
MMC	管理ツールを載せる管理コンソール。RSAT の AD DS ツールは、ユーザーとコンピューター（ADUC）と、サイトとサービスの MMC スナップインを提供する。GPO の管理も MMC のスナップインで行う
ADSI Edit	Active Directory Services Interfaces Editor。ドメイン管理者でログインし、Connect To で接続して、AD のツリーを属性単位で参照、編集する
LDP	LDAP 互換のディレクトリ（AD DS など）に、接続、バインド、検索、変更、追加、削除を行う GUI の LDAP クライアント。セキュリティ記述子やレプリケーションのメタデータも見られる
Regedit	Windows のレジストリエディター。smb.conf のレジストリ設定にも、regedit や `net (rpc) registry` でアクセスできる

ユーザーやグループの日常の管理は ADUC です。属性を1つずつ直接見たいときは ADSI Edit か LDP です。Samba の設定そのものをレジストリに置いたときは、Regedit か `net registry` で編集します。

SAMPLE

Samba サービスの起動と停止 [301.3]

起動と停止の作法は、サーバーの役割で変わります。[13]

ファイルサーバーやドメインメンバーでは、役割ごとのサービスを個別に管理します。Debian 系のパッケージでは、ユニット名は `smbd`、`nmbd`、`winbind` です。

AD DC では、`samba` デーモンが1つのサービスとして全体を動かします。samba が smbd と winbindd をサブプロセスとして起動するので、それらを別に起動すると正しく動きません。DC では次のようにして、3つを無効にします。

```
# systemctl mask smbd nmbd winbind
# systemctl disable smbd nmbd winbind
# systemctl enable samba-ad-dc
# systemctl start samba-ad-dc
```

`mask` はユニットを起動できない状態にし、ほかのユニットから有効にされることも、手で起動することも防ぎます。`disable` は自動起動を外すだけで、起動そのものは止めません。

停止は `systemctl stop samba-ad-dc` です。起動時の自動起動の解除は `systemctl disable samba-ad-dc` です。

`samba-ad-dc` は、ディストリビューションのパッケージが用意するユニット名です。Samba 自体は systemd のサービスファイルを提供しません。ソースからビルドしたときは、`/etc/systemd/system/samba-ad-dc.service` を自分で作り、`systemctl daemon-reload` で読み込ませます。ユニットの中身の要点は次のとおりです：

SAMPLE

```
[Service]
Type=forking
ExecStart=/usr/local/samba/sbin/samba -D
PIDFile=/usr/local/samba/var/run/samba.pid
ExecReload=/bin/kill -HUP $MAINPID
```

systemd を介さずに smbd へ直接シグナルを送る場合は、次の使い分けになります。

シグナル	動き
SIGHUP	smb.conf を再読込する
SIGTERM（-15）	安全に終了する
SIGKILL（-9）	共有メモリを不整合にするので、最後の手段にする

smbcontrol と smbstatus [301.3]

`smbcontrol` は、動いている smbd、nmbd、winbindd にメッセージを送る小さなプログラムです。書式は `smbcontrol [destination] [message-type] [parameter]` です。[14]

destination（宛先）は、nmbd、smbd、winbindd、プロセス ID のどれかです。`all` を宛先にすると、動いているすべてのデーモン（nmbd と winbind も含む）に送ります。`all` の意味は Samba 3.3 で変わりました。それより前は、smbd がこの役目でした。プロセス ID を1つ指定すると、そのプロセスにだけ送ります。

主なメッセージの種類と、送れる先は次のとおりです：

SAMPLE

message-type	働き	送れる先
reload-config	smb.conf を再読込させる	smbd、nmbd、winbindd
debug	デバッグレベルを設定する。smbd と winbindd では親が子へ再配信する	どの宛先でも
debuglevel	現在のデバッグレベルを標準出力に出す	どの宛先でも
shutdown	指定したデーモンを停止させる	smbd、nmbd
ping	指定した回数の ping を送り、同数の pong を待つ	どの宛先でも
close-share	指定した共有へのクライアント接続を閉じる。`*` で全共有	smbd のみ
close-denied-share	今アクセスを拒否されるようになったユーザーだけを切る。アクセス制御の変更後に、全 smbd へ安全に送れる	smbd のみ
kill-client-ip	指定した IP アドレスからの接続を閉じさせる	smbd のみ
force-election	新しいブラウズマスターの選出を強制する	nmbd
profile	`on`、`off`、`count`、`flush` でプロファイル収集を切り替える	smbd、nmbd
pool-usage	talloc プールのメモリ使用を表示する	smbd、nmbd

SAMPLE

message-type	働き	送れる先
num-children	smbd の子プロセスの数を調べる	smbd のみ
reload-certs	AD DC の LDAP サーバーに、ldaps:// の TLS 証明書を読み直させる	ldap_server のみ

共有のアクセス権を絞ったあとに `close-share` で全員を切ると、業務に響きます。`close-denied-share` なら、今は許されなくなったユーザーだけが切られます。

```
# smbcontrol all reload-config
# smbcontrol smbd debug 3
# smbcontrol smbd ping 2
# smbcontrol smbd close-denied-share data
```

`smbstatus` は、smbd への現在の接続、開いている共有、かかっているロックを表示します。

オプション	働き
-b	簡易な出力
-v	詳細な出力
-L	ロックだけを表示する
-S	共有だけを表示する
-p	smbd のプロセスの一覧を出す
-u ユーザー	そのユーザーの情報だけ
-n	UID と GID を数値のまま出す

SAMPLE

オプション	働き
-j	JSON で出す
-f	プロセスの生存確認を省いて速くする
-B	byte range lock も含める

`-f` は速くなる代わりに、異常終了したプロセスの古い情報が出ることがあり、確実な状態は見られません。

TDB ファイルのバックアップと戻し [301.3]

TDB（Trivial Database）は、Samba が内部の状態を持つための小さなデータベースです。稼働中の TDB を `cp` でコピーしてはいけません。書き込みの途中を写し取ると壊れたコピーができます。専用の道具を使います。[15]

道具	働き
tdbbackup	TDB を複製してバックアップし、整合性も検査する。稼働中でも安全に実行できる
tdbdump	TDB の中身を人が読める形で標準出力に出す
tdbrestore	`tdbdump` の出力を標準入力から読み、指定した TDB ファイルとして書き出す

`tdbbackup` の書式は `tdbbackup [-s suffix] *.tdb` です。

オプション	働き
-s suffix	バックアップに付ける拡張子（`.bak` など）
-v	破損を検査する。破損していれば、以前のバックアップを復元する

SAMPLE

オプション	働き
-n hash size	ハッシュのサイズを指定する
-l	ロックを無効にする

守るべき重要なファイルは、`secrets.tdb` と `passdb.tdb` です。通常は `/usr/local/samba/private`（システムによっては `/etc/samba`）にあり、ほかの `*.tdb` は `/usr/local/samba/var`、`/var/cache`、`/var/lib/samba` にあります。

戻すときは、`tdbbackup` が作るバックアップ（そのまま使える完全な TDB ファイル）を使うか、`tdbdump` でテキストに書き出したものを `tdbrestore` で TDB へ作り直します。

```
# tdbbackup -s .bak /var/lib/samba/private/secrets.tdb
# tdbdump /var/lib/samba/private/secrets.tdb > secrets.dump
# tdbrestore /var/lib/samba/private/secrets.tdb.new < secrets.dump
```

AD DC のバックアップと戻し [301.3]

Samba 4.9 以降は、`samba-tool domain backup` で AD ドメインのバックアップを取ります（offline は 4.10 で加わりました）。副コマンドは4つです。[16]

副コマンド	働き	向く目的
online	稼働中の DC のデータベースを、ネットワーク越しに複製する	ドメインの長期の置き換え

SAMPLE

副コマンド	働き	向く目的
offline	ローカルのドメインのディレクトリを、ロックしながら tar にまとめる。DC を止める必要はない。レプリケーションされないローカルの追加データも含む	調査（フォレンジック）
rename	稼働中の DC のデータベースを、ドメイン名を変えて複製する	一時的な置き換え。元のドメインと並行稼働できる
restore	バックアップファイルからドメインを復元する	

書式は次のとおりです：

```
# samba-tool domain backup online  --targetdir=/backup --
server=dc1.example.com -UAdministrator
# samba-tool domain backup offline --targetdir=/backup
# samba-tool domain backup rename NEWDOMAIN new.renamed.com --
server=dc1.example.com --targetdir=/backup -UAdministrator
# samba-tool domain backup restore --backup-file=/backup/samba-
backup.tar.bz2 --newservername=dc9 --targetdir=/usr/local/samba-restore
```

バックアップファイルは `.tar.bz2` で、ドメイン全体が入り、内容は中の `backup.txt` に書かれます。offline の実行には `lmdb-utils` パッケージ（`mdb_copy`）が要ります。offline と restore はローカルのファイルを読み書きするので root の権限が要り、restore は sysvol のファイルの NTACL を保って戻すためにも root で実行します。online は管理者権限の AD ユーザーでも実行できますが、ファイルにドメインの全情報が入るので、root で実行して読める人を絞るほうが安全です。

SAMPLE

バックアップはドメイン単位で、DC 単位ではありません。復旧の方針は次の2つです：

- DC が1台だけ壊れ、ドメインが健在なら、`samba-tool domain join` で入れ直します（`backup restore` で個別の DC を直すのは誤り）
- ドメイン全体が壊れたときだけ、`backup restore` を使います。復元すると新しい DC が1台でき、残りの DC は、その DC へ新規に join し直します

復元の手順は4つです。

1. 旧 DC の samba をすべて止めます（rename の場合は不要）。
2. 新しい DC 1台で `samba-tool domain backup restore` を実行し、`--targetdir` には空か未作成のディレクトリ、`--newservername` には旧ネットワークに存在しなかったサーバー名を指定します。
3. その DC で `samba -s <targetdir>/etc/smb.conf` として起動し、所属サイトは `Default-First-Site-Name`、`--site` で変えられます。
4. 旧 DC を `samba-tool domain join` で再参加させます。

復元後は、`private/tls` と `bind-dns` が空です。バックアップはレプリケートされるドメインのデータで、DC ごとの証明書は入りません。DNS は内蔵の DNS に戻ります。BIND9_DLZ にしたいなら `samba_upgradedns` を使います。

バックアップの前に、`samba-tool dbcheck` を回すのが望ましい手順です。バックアップは不整合も含めて、そのまま写し取るためです。

古い方式の `samba_backup` スクリプト（Samba 4.0 から 4.8 向け）は非推奨の DRAFT で、4.9.0 以降はバックアップが大きく改善されたため、今は使いません。この古い方式でも、複数の DC があるのに1台をバックアップから復元してはならない、という決まりは同じで、レプリケーションの記録がずれ、ディレクトリが壊れるためです。

SAMPLE

3台の Samba DC のうち1台のディスクが壊れ、残る2台が健在なら、壊れた DC を `samba-tool domain join` で入れ直します。3台とも使えなくなったときは、最新の online か offline のバックアップから、新しい名前の DC 1台に `backup restore` します。旧 DC は join し直すまで、復元したドメインに参加していません。

仮想化が DC に与える影響 [301.3]

DC を仮想マシンで動かすと、スナップショットや複製の扱いが危険になります。AD のデータベースには、更新の順序を示す USN という論理時計があります。スナップショットを適用すると、この USN が誰にも気づかれないまま巻き戻ります。[17]

USN が巻き戻ると、すでにその USN を見た相手の DC は、巻き戻った DC からの変更を取り込みません。DC 同士の内容がずれ、これを USN rollback と呼びます。

これを防ぐ仕組みが VM-Generation ID（Virtual Machine Generation Identifier）です。ハイパーバイザーが、ゲスト VM のアドレス空間に公開する識別子で、仮想 DC が VM の巻き戻しを検出するために使います（Windows Server 2012 以降）。

動きは次の順です：

1. AD DS は DC の昇格時に、VM-Generation ID の値を NTDS.DIT に保存します。
2. VM を起動するたびに、保存した値とハイパーバイザーが渡す現在の値を比べます。
3. 値が違えば、USN の再利用と SID の重複を防ぐために、Invocation ID をリセットし、RID プールを捨てます。

SAMPLE

4. スナップショットから戻したときは、SYSVOL を非権威的に同期し直します。

VM-Generation ID に対応していないハイパーバイザーでは、保護が働きません。DCCloneConfig.xml があるのに値が来ないクローンの DC は、DSRM で起動して環境を守り、対応するハイパーバイザーに移せばやり直せます。

Samba のバックアップには、スナップショットではなく、`samba-tool domain backup` によるドメイン単位のバックアップを使います。

複数台の Samba DC がある環境で、1台の仮想マシンのスナップショットを復元して直そうとしている場面で考えます。この方法では、巻き戻った DC のデータがほかの DC とずれ、ディレクトリが壊れます。正しいのは、その DC を新しく `samba-tool domain join` で入れ直し、ほかの DC から最新のデータをもらう方法で、スナップショットの復元は使いません。

ログの設定 [301.4]

Samba のログは、既定では無効に近い設定です。`log level` の既定は 0 で、重大なエラーしか残りません。設定すると有効になり、高いレベルは低いレベルのイベントも含みます。レベルは 1 が最小で、数字が大きいほど詳しくなります。3 を超えるレベルは開発者向けで、大量の出力が出ます。[18]

`log level`（別名 `debuglevel`）は、先頭に全体のレベルを置き、そのあとに `クラス:レベル` を並べると、デバッグクラスごとにレベルを指定できます。

```
log level = 3 passdb:5 auth:10 winbind:2
```

主なクラスは、all、tdb、printdrivers、lanman、smb、rpc_parse、rpc_srv、rpc_cli、passdb、sam、auth、winbind です。VFS モジュールは、full_audit や

SAMPLE

recycle のような動的なクラスを登録します。

クラスごとに別のログファイルへ出すには、`@` を使います：

```
log level = 1 full_audit:1@/var/log/audit.log winbind:2
```

クライアントごとにログを分けるには、`log file` に変数を使います：

```
log file = /var/log/samba/%m.log
```

`%m` はクライアントの NetBIOS 名なので、接続元ごとに別のログになります。ログの置き場は `/var/log/samba/` です。`log file` に既定値はありません。

項目	内容
max log size	KB 単位の整数で、既定は 5000。超えると `.old` を付けてリネームする。0 は無制限。ローテート時に以前の `.old` は上書きされる

設定を変えたら、`smbcontrol all reload-config` で反映します。動いているデーモンのレベルだけを一時的に上げるなら、`smbcontrol smbd debug 5` を使い、確認は `smbcontrol smbd debuglevel` です。再起動は要りません。

コマンドラインのツールの `-d`（`--debuglevel`）は 0 から 10 の整数で、クライアントの既定は 1、0 は重大なエラーだけ、3 を超えると開発者向けです。`-d` を指定すると、smb.conf の `log level` より優先されます。

調査では、まず `log level = 1 auth:3`（認証だけ詳しく）のようにクラスを絞り、全体を 10 にするのは避けます。

SAMPLE

パスワードデータベースの照会と変更 [301.4]

ユーザーのパスワード情報は、`passdb backend` が指す場所に置かれます。`smbpasswd`（旧方式のファイル）、`tdbsam`（TDB。`private dir` の `passdb.tdb`）、`ldapsam`（LDAP）などがあります。`pdbedit` は、バックエンドに依存せず使えます。[19]

`smbpasswd` は実行する人で働きが変わり、root はほかのユーザーを操作でき、一般ユーザーは自分のパスワードだけを変えられます。

オプション	働き
-a	ユーザーを追加する（root のみ。システムにすでにあるユーザーが必要）
-x	ユーザーを削除する
-d	アカウントを無効にする
-e	アカウントを有効にする
-n	パスワードなしにする
-s	サイレントモード。パスワードを標準入力から読む
-c	使う smb.conf を指定する

`pdbedit` は root だけが実行でき、バックエンドを選びません。

オプション	働き
-L	ユーザーの一覧。出力は `ユーザー名:uid:フルネーム` の形
-v	詳細表示（`-L` と併用）

SAMPLE

オプション	働き
-w	smbpasswd 形式で出す（`-v` と併用でハッシュも出る）
-a -u 名前	ユーザーを追加する（パスワードの入力がある）
-r	ユーザーを変更する
-x -u 名前	ユーザーを削除する
-m	`-a` と併用して、マシントラストアカウントを追加する
--set-nt-hash	NT ハッシュ（16進文字列）でパスワードを設定する。パスワードの同期に使う

```
# pdbedit -L
# pdbedit -L -v -u alice
```

`smbpasswd -a` も `pdbedit -a` も、システムの UNIX ユーザーを前提にします。pdbedit(8) は追加・削除・変更に `-u` が要ると書きますが、ユーザー名をコマンドの最後に置いても動きます（`pdbedit -x bob` と `pdbedit -x -u bob` は同じ）。この2つはスタンドアロンのサーバーやメンバーサーバー向けで、AD DC のユーザーは `samba-tool user` で扱います。

主な TDB ファイルと壊れの見分け方 [301.4]

TDB ファイルは、Samba が持つ状態の置き場です。[20]

SAMPLE

ファイル	置き場	中身
passdb.tdb	PRIVATE_DIR	tdbsam を使うときだけ存在する。SambaSAMAccount の情報（ユーザーのパスワード情報）を持つ。UNIX のアカウントが `/etc/passwd` などにも要る
secrets.tdb	PRIVATE_DIR	ワークグループ、ドメイン、マシンの SID、LDAP ディレクトリの更新パスワードなどを持つ。非常に機密性が高い
registry.tdb	STATEDIR	Windows のレジストリのスケルトン。読み取り専用で、winreg RPC を介して見える
account_policy.tdb	STATEDIR	パスワードの期限などのアカウントポリシー
group_mapping.tdb	STATEDIR	Windows のグループ（SID）と UNIX のグループの対応
share_info.tdb	STATEDIR	共有ごとの ACL
winbindd_idmap.tdb	STATEDIR	winbindd のローカルな IDMAP

置き場の名前は `smbd -b` の出力に出るもので、PRIVATE_DIR、STATEDIR、LOCKDIR、CACHEDIR の 4 種類があります。brlock.tdb、locking.tdb、sessionid.tdb、connections.tdb、g_lock.tdb のようなロックの情報は LOCKDIR に置かれ、`TDB_CLEAR_IF_FIRST` の属性が付いています。

SAMPLE

`ntdrivers.tdb`、`ntforms.tdb`、`ntprinters.tdb`、`unexpected.tdb` は、Samba 3.6 で削除済みで、今の Samba では使われません。

`TDB_CLEAR_IF_FIRST` の属性が付いた TDB は、最初に開くとき（再起動後など）に消去されます。`gencache.tdb` のように `*` が付いたものは、破損していたときだけ消去されます。

`tdbdump` は、TDB の中身を人が読める形で標準出力へ出します。

オプション	働き
-k キー	1つのキーだけ出す
-e	破損した DB から救出する（信頼性は低い）
-x	16進数でダンプする
-h	ヘルプ

`tdbtool` は、対話でも引数でも使える、表示と編集のツールです。コマンドは次のとおりです：

create、open、erase、dump、keys、hexkeys、info、insert、store、show、delete、list、first、next、check、repack、quit

`-l` でロックを無効にできます。

Samba の TDB の中身は実装の内部です。`tdbtool` で変更すべきではありません。

壊れの見分け方は3つです：

方法	内容
`tdbbackup -v *.tdb`	破損を検査する

SAMPLE

方法	内容
`tdbtool` の `check`	TDB の整合性を確認する
`net registry check`	registry.tdb の検査と修復をする

```
# tdbdump /var/lib/samba/private/secrets.tdb | head
# tdbtool registry.tdb
tdb> keys
tdb> check
```

sam.ldb の照会と変更（ldb ツール）[301.4]

AD DC のユーザーやグループなどのデータは、`sam.ldb` という LDB のデータベースにあります。LDB は LDAP に似た構造を持つデータベースで、ldb のツールでファイルを直接読み書きできます。ツールは `-H` でデータベースのファイルを指して使います。[21]

ツール	働き
ldbsearch	式で検索して、属性を出力する
ldbmodify	LDIF の内容で、レコードの追加、変更、削除をする。ファイル名が `-` なら標準入力
ldbedit	問い合わせの結果を LDIF にして好みのエディターで編集し、結果を LDB に戻す
ldbadd	LDIF のレコードを追加する。DN は DB 内で一意で、LDIF はスキーマに合わせる
ldbdel	DN を指定して、レコードを削除する

主なオプションです：

SAMPLE

オプション	働き
-H URL	対象の LDB（環境変数 `LDB_URL` でも指定できる）
-s one、sub、base	検索の範囲（ldbsearch）。base は起点だけ、one は起点の 1 階層下、sub は起点以下すべて
-b basedn	ベース DN（ldbsearch、ldbedit）
-i	検索式を標準入力から読む（ldbsearch）
-e エディター	ldbedit が使うエディター。`VISUAL` と `EDITOR` より優先され、なければ vi
-a	全レコードを編集する（ldbedit）

`-H` に書ける URL の形は次のとおりです：

形	意味
tdb://ファイル、mdb://ファイル	バックエンドを指定する
ldb://ファイル	自動判別
ldapi://ソケット	UNIX ドメインソケットで LDAP に接続する
ldap://ホスト	LDAP で接続する
ldaps://ホスト	TLS つきの LDAP

LDIF は、`ldbmodify`、`ldbadd`、`ldbedit` が入出力に使うテキスト形式で、DN の行で対象のレコードが決まります。

SAMPLE

```
$ ldbsearch -H /var/lib/samba/private/sam.ldb -b 'DC=example,DC=com'
'(sAMAccountName=alice)' dn sAMAccountName

$ cat mod.ldif
dn: CN=alice,CN=Users,DC=example,DC=com
changetype: modify
replace: description
description: finance
-
add: telephoneNumber
telephoneNumber: 03-0000-0000

$ ldbmodify -H /var/lib/samba/private/sam.ldb mod.ldif
```

LDIF では、`-` の行が1つの変更の区切りで、行頭が空白の行は前の行の続きで、`#` で始まる行はコメントです。`changetype` を書かない LDIF は `ldbadd` で追加するレコードになり、`changetype: modify` のように書けば `ldbmodify` で変更や削除ができます。

AD のデータは、`private dir` の中の2つの LDB にあり、`sam.ldb` がユーザー、グループ、コンピューターなどのディレクトリ、`secrets.ldb` がマシンの秘密情報です。どちらも `-H` にファイルを渡して読めますが、稼働中の DB を通常のコピーで取ってはいけません。

検索には、AD 向けのオプションもあります：

オプション	働き
--show-deleted	削除済みのオブジェクトも出す
--show-recycled	recycled のオブジェクトも出す
--show-deactivated-link	無効化されたリンクを出す

SAMPLE

オプション	働き
--show-binary	バイナリの属性を LDIF で出す
--cross-ncs	名前コンテキストの境界を越えて検索する
--extended-dn	拡張 DN（GUID と SID つき）を出す
--relax	relax コントロールを渡す
--controls	コントロールを指定する

通常の管理は `samba-tool` で行い、ldb ツールは調査や復旧に使います。

ごみ箱と dbcheck [301.4]

AD で削除したオブジェクトは、すぐには消えず、deleted objects container に tombstone（削除済みオブジェクト）として残ります。`samba-tool domain tombstones expunge NC` は、この tombstone を消去し、`--tombstone-lifetime=DAYS` が保持日数、`--current-time=YYYY-MM-DD` が保持期間を数える基準の日付です。tombstone や recycled の状態になったオブジェクトはリンク属性を持たず、Samba 4.5 以降は、そうなったオブジェクトのリンクを dbcheck が除去する提案をします。[22]

ごみ箱（Recycle Bin）は、削除したオブジェクトを復元できるようにする AD の機能です。Windows 側では、Active Directory Administrative Center（ADAC）でごみ箱を有効にし、削除したオブジェクトを復元しますが、ADAC が復元できるのは domain パーティションのオブジェクトだけです。Samba の ldb ツールは、`--show-deleted` と `--show-recycled` を付けると、削除済みと recycled のオブジェクトも検索結果に含めます。

`samba-tool dbcheck` は、ローカルの AD データベースのエラーを検査し、LDAP 越しでは実行できません。各 AD DC のローカルで実行します。

SAMPLE

オプション	働き
--cross-ncs	すべてのパーティションを検査する（なければメインのドメインだけ）
--fix	修復する（エラーごとに確認がある）
--yes	すべて自動で承諾する。付けないとオブジェクトごとに 3 回 fsync が走る

dbcheck は全オブジェクトを調べ、たとえば、双方向リンクのバックリンクの欠落、`replPropertyMetaData` の不整合、deleted objects container、RootDSE の `dsServiceName` を確かめます。

`--fix` の前には、online と offline の両方でドメインバックアップを取ります。そのあと問題が出たら、次の順に戻します。

1. レプリケートのやり直し（`samba-tool drs replicate --full-sync --sync-forced`）
2. DC の再 join
3. 最後の手段としての、バックアップからの復元

dbcheck は、バックアップの前に回す検査でもあり、バックアップは不整合も含めて写し取るためです。

名前を変えた DC の複製とイベントログ [301.4]

`samba-tool domain backup rename` は、稼働中の DC のデータベースを、ドメイン名を変えて複製したバックアップを作ります。一時的な代替ドメインや、検証用の環境を作るときに使います。元のドメインは変更されず、複製だけが書き換えられます。[23]

SAMPLE

```
# samba-tool domain backup rename NEWDOMAIN new.renamed.com --
server=dc1.example.com --targetdir=/backup -UAdministrator
# samba-tool domain backup restore --newservername=testdc --
targetdir=/usr/local/samba-test --backup-file=/backup/samba-
backup.tar.bz2
```

第1引数は新しい NetBIOS 名、第2引数は新しい DNS のレルムです。変わるものは次のとおりです：

- すべての DN
- ドメインのパーティションオブジェクトと、その NetBIOS 名
- DNS ゾーンのオブジェクトと、`dnsRoot` 属性

変わらないものもあります。`userPrincipalName` は更新されません。`--keep-dns-realm` を付けて改名すると、完全な `userPrincipalName` のままでログインできます。GPO の属性（`gPCFileSysPath` と `gPLink`）も更新されません。

改名した複製は、一時的な置き換え用です。本番のドメインの長期の改名には、今のところ対応していません。この機能は Samba 4.9 からのものです。

Windows の `rendom` とは考え方が違います。`rendom` は DC を1台ずつ、その場で改名します。Samba は、改名した複製から新しい DC を1台作り、そこから DC をゼロから足し直します。重なる期間がありません。

イベントログ転送（eventlog shipping）は、Samba のイベントログに、UNIX 側のログを流し込む仕組みです。Samba は UNIX のログを自動では取り込みません。外部のプロセスで `/var/log/messages` などを解析し、`eventlogadm` で Samba のイベントログに書くと、Windows のイベントビューアーからそのログが読めるようになります。

SAMPLE

項目	内容
eventlog list	イベントビューアーに見せるログ名の一覧。既定は空。例は `eventlog list = Application System Security SyslogLinux`。ログは `$(statedir)/eventlog` の TDB に対応する
eventlogadm -o write ログ名	標準入力のレコードをイベントログに書く
eventlogadm -o addsource ログ名 ソース名 メッセージファイル	ログのソースを登録する（メッセージ DLL をレジストリに登録する）
eventlogadm -o dump ログ名 レコード番号	TDB のレコードを画面に出す

レコードは、`LEN`、`RS1`、`TMG`、`EID`、`ETP`、`SRC`、`STR` といった項目を持つ行で渡し、`RS1` は決まった値 1699505740 で、`LEN` は 0 にします。

rpcclient によるサーバーの照会 [301.4]

`rpcclient` は、クライアント側から MS-RPC の関数を実行するツールです。もとは Samba 自身の MS-RPC 機能のテスト用で、UNIX から Windows NT のクライアントを管理するスクリプトにも使われます。[24]

接続先は `TRANSPORT:host[options]` の形で、TRANSPORT は、SMB の名前付きパイプの `ncacn_np` か、DCE/RPC over TCP の `ncacn_ip_tcp` です。

オプション	働き
-c "コマンド"	コマンドを実行して終わる
-U [DOMAIN/]USER[%PASSWORD]	ユーザーを指定する
-N	パスワードなし

SAMPLE

オプション	働き
-I	宛先の IP アドレス
-p	ポート
-A	認証ファイル
-k	Kerberos を使う（4.15 から非推奨。今の書き方は `--use-kerberos=required`、`desired`、`off`）

コマンド	働き	インターフェース
enumdomusers	ドメインのユーザー一覧	SAMR
querydominfo	ドメイン情報の照会	SAMR
queryuser	ユーザー情報	SAMR
enumdomgroups	グループ一覧	SAMR
querygroupmem	グループのメンバー	SAMR
srvinfo	サーバー情報	SRVSVC
netshareenum	共有の一覧	SRVSVC
lookupnames	名前から SID	LSARPC
lookupsids	SID から名前	LSARPC
enumprivs	権限の一覧	LSARPC

```
$ rpcclient -U 'EXAMPLE\administrator' dc1 -c 'enumdomusers'
$ rpcclient -U 'EXAMPLE\administrator' dc1 -c 'lookupnames alice'
$ rpcclient -U 'EXAMPLE\administrator' ncacn_np:dc1 -c 'netshareenum'
```

SAMPLE

`lookupnames` は名前から SID、`lookupsids` は SID から名前への変換で、向きを取り違えると答えが逆になります。

SAMPLE

サンプルはここまでです

続き(第2章〜巻末)は、LPIC-3 300 問題集のご購入で、頻出度別ノート・暗記用ノートと合わせて3点すべてダウンロードできます。

Footnotes

1. https://www.samba.org/samba/docs/current/man-html/samba.7.html ／ https://www.samba.org/samba/docs/current/man-html/smbd.8.html ／ https://www.samba.org/samba/docs/current/man-html/nmbd.8.html ／ https://www.samba.org/samba/docs/current/man-html/samba.8.html ／ https://www.samba.org/samba/docs/current/man-html/winbindd.8.html ↩

2. https://wiki.samba.org/index.php/Samba_AD_DC_Port_Usage ／ https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（rpc server port・server smb transports） ↩

3. https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（server min protocol・server max protocol・smb2 leases・durable handles・server multi channel support・client smb encrypt・server smb encrypt・disable netbios・ntlm auth）／ https://learn.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3 ↩

SAMPLE

4. https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（server role・security・vfs objects）／https://wiki.samba.org/index.php/Virtual_File_System_Modules ／https://wiki.samba.org/index.php/CTDB_and_Clustered_Samba ／https://ctdb.samba.org/manpages/ctdb.7.html ↩

5. https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Domain_Member（smbd -b）／https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（include・VARIABLE SUBSTITUTIONS・server string）／https://www.samba.org/samba/docs/current/man-html/smbd.8.html（smb.conf の既定位置・再読込）↩

6. https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（server role・security）↩

7. https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（server string・server services・vfs objects）↩

8. https://wiki.samba.org/index.php/Back_up_and_Restoring_a_Samba_AD_DC（private/tls）／https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（tls enabled・tls keyfile・tls certfile・tls cafile・tls dh params file・tls priority）／https://www.samba.org/samba/docs/current/man-html/smbcontrol.1.html（reload-certs）↩

9. https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（config backend・registry shares・include）／https://www.samba.org/samba/docs/current/man-html/samba-

SAMPLE

regedit.8.html ／ https://learn.microsoft.com/en-us/windows/win32/sysinfo/registry-value-types ↩

10. https://www.samba.org/samba/docs/current/man-html/net.8.html（REGISTRY・CONF） ↩

11. https://wiki.samba.org/index.php/Setting_up_Samba_as_a_Standalone_Server ／ https://www.samba.org/samba/docs/current/man-html/testparm.1.html ／ https://www.samba.org/samba/docs/current/man-html/samba.7.html ↩

12. https://wiki.samba.org/index.php/Installing_RSAT ／ https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc771022(v=ws.10)（LDP） ／ https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（レジストリ設定へのアクセス） ↩

13. https://wiki.samba.org/index.php/Managing_the_Samba_AD_DC_Service_Using_Systemd ／ https://www.samba.org/samba/docs/current/man-html/smbd.8.html（シグナル） ↩

14. https://www.samba.org/samba/docs/current/man-html/smbcontrol.1.html ／ https://www.samba.org/samba/docs/current/man-html/smbstatus.1.html ↩

SAMPLE

15. https://gitlab.com/samba-team/samba/-/raw/master/lib/tdb/man/tdbbackup.8.xml ／ https://gitlab.com/samba-team/samba/-/raw/master/lib/tdb/man/tdbrestore.8.xml ／ https://wiki.samba.org/index.php/Using_the_samba_backup_script ↩

16. https://wiki.samba.org/index.php/Back_up_and_Restoring_a_Samba_AD_DC ／ https://wiki.samba.org/index.php/Domain_rename_tool ／ https://wiki.samba.org/index.php/Domain_backup_and_rename ／ https://www.samba.org/samba/docs/current/man-html/samba-tool.8.html ↩

17. https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/introduction-to-active-directory-domain-services-ad-ds-virtualization-level-100 ／ https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/virtualized-domain-controller-architecture ／ https://wiki.samba.org/index.php/Back_up_and_Restoring_a_Samba_AD_DC ↩

18. https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（log level・debuglevel・max log size）／ https://wiki.samba.org/index.php/Configuring_Logging_on_a_Samba_Server ／ https://www.samba.org/samba/docs/current/man-html/testparm.1.html（-d） ↩

SAMPLE

19. https://www.samba.org/samba/docs/current/man-html/smbpasswd.8.html ／
https://www.samba.org/samba/docs/current/man-html/pdbedit.8.html ／
https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（passdb backend） ↩

20. https://wiki.samba.org/index.php/TDB_Locations ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/tdb/man/tdbdump.8.xml ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/tdb/man/tdbtool.8.xml ／
https://www.samba.org/samba/docs/current/man-html/net.8.html（registry check） ↩

21. https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/tools/cmdline.c ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/man/ldbsearch.1.xml ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/man/ldbmodify.1.xml ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/man/ldbedit.1.xml ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/man/ldbadd.1.xml ／
https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/man/ldbdel.1.xml ／
https://wiki.samba.org/index.php/LDB ↩

SAMPLE

22. https://wiki.samba.org/index.php/Dbcheck ／ https://wiki.samba.org/index.php/Samba_4.5_Features_added/changed ／ https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/adac/Advanced-AD-DS-Management-Using-Active-Directory-Administrative-Center--Level-200- ／ https://gitlab.com/samba-team/samba/-/raw/master/lib/ldb/tools/cmdline.c ／ https://www.samba.org/samba/docs/current/man-html/samba-tool.8.html（domain tombstones expunge） ↩

23. https://wiki.samba.org/index.php/Domain_rename_tool ／ https://wiki.samba.org/index.php/Domain_backup_and_rename ／ https://wiki.samba.org/index.php/Event_Logging ／ https://www.samba.org/samba/docs/current/man-html/eventlogadm.8.html ／ https://www.samba.org/samba/docs/current/man-html/smb.conf.5.html（eventlog list） ↩

24. https://www.samba.org/samba/docs/current/man-html/rpcclient.1.html ↩

SAMPLE

LPIC-3 300試験 学習用テキスト(無料サンプル)

版
2026年11月版(LPI公式 LPIC-3 Exam 300 Objectives Version 3.0(300-300・2026-11-24 取得)準拠)

発行
IT資格道場(https://www.shikaku-doujou.com)

お問い合わせ
info@shikaku-doujou.com

LPI および LPIC は Linux Professional Institute Inc. の商標または登録商標です。Linux は Linus Torvalds 氏の米国およびその他の国における商標または登録商標です。記載されているその他の会社名・製品名・サービス名は、各社の商標または登録商標です。

本教材は公式シラバスの内容を基に作成したもので、Linux Professional Institute との提携・承認関係はありません。

本文・図表・設問はすべてオリジナルの著作物です。無断での転載・複製・再配布・二次販売はできません。お気づきの点は info@shikaku-doujou.com までご連絡ください。

SAMPLE