Samba4 Status - April 2004

Andrew Tridgell
Samba Team

Major Features

- The basic goals of Samba4 are quite ambitious, but achievable:
    - protocol completeness
    - extreme testability
    - non-POSIX backends
    - fully asynchronous internals
    - flexible process models
    - auto-generated RPC infrastructure
    - flexible database architecture

Samba 3.5 – half way to Samba4

- We plan on creating a Samba3.5 release in the next few months
    - extract finished code from Samba4 now
    - get team members used to the Samba4 structures and methods
    - provide essential infrastructure for some new features
    - only merge easily 'separable' parts of Samba4

Samba 3.5 – pieces to merge

- We plan on merging the following:
  - the libcli/ SMB client library
  - the IDL based RPC client library
  - the ldb database
  - the smbclient code
  - the smbtorture test suite
  - the schannel code
  - some of the source directory structue

Protocol Completeness

- CIFS/SMB is a huge protocol, but is not infinite.
- In previous versions of Samba we implemented new protocol elements “on demand”, only adding an element when we saw an application using it.
- In Samba4 the new attitude is “implement everything”

Old testing method

- The Samba project has previously developed testsuites of 3 main kinds:
    - ad-hoc tests for a range of specific conditions
    - full-coverage tests for a very small range of operations
    - randomised testing for a very small range of operations
- This approach did work to some extent, but suffered from some major drawbacks:
    - many parts of the protocol remained completely untested
    - many fields untested within the tested parts of the protocol
    - difficult to expand to be comprehensive

New approach: extreme testability

- The new testing system in Samba4 is based on a few basic components:
    - a comprehensive raw client library
    - individual tests covering every field of every call
    - a randomised dual-server tester with broad coverage
    - a "CIFS on CIFS" storage backend for the Samba4 server
- These components work together to provide a testing capability far beyond what could be achieved with our earlier testsuites

CIFS Plugfest

Raw Client Library

- The heart of the new testing system is a 'raw' comprehensive client library. Unlike our previous client library this allows easy generation of all SMBs, with control over all fields in each request
- New features include:
    - async interfaces
    - oplock support
    - no 'smarts' - send exactly what is asked for
- Note that it takes a lot code to use the new interface compared to the old one. The old interface is still available as a wrapper

C interface to raw library

Old interface:

```
int fnum = cli_open(cli, "\\test.dat", O_RDWR, DENY_READ);
```

New Interface:

```
NTSTATUS status;
union smb_open io;

io.generic.level = RAW_OPEN_OPENX;
io.openx.in.flags = OPENX_FLAGS_ADDITIONAL_INFO;
io.openx.in.open_mode = OPEN_MODE_ACCESS_RDWR;
io.openx.in.search_attrs = FILE_ATTRIBUTE_SYSTEM|FILE_ATTRIBUTE_HIDDEN;
io.openx.in.file_attrs = 0;
io.openx.in.write_time = 0;
io.openx.in.open_func = OPENX_OPEN_FUNC_OPEN;
io.openx.in.size = 0;
io.openx.in.timeout = 0;
io.openx.in.fname = "\\test.dat";

req = smb_raw_open_send(tree, &io);
status = smb_raw_open_recv(req, mem_ctx, &io);
```

3.5

CIFS Backend

- A new feature in Samba4 is the ability to define arbitrary storage backends at the 'raw' CIFS level
- A backend that has proved incredibly useful for testing is the 'CIFS' backend, that uses a remote CIFS server for all operations:
    - uses the raw client library for remote server access
    - ideal for testing core server infrastructure
    - combined with the individual tests and gentest it allows the server side CIFS parsing to be tested in isolation

gentest

- gentest is the 'big gun' CIFS test program that I have wanted to build for many years. Basic features include:
    - dual server, dual instance testing
    - randomised, broad coverage request generation
    - automatic backtracking for finding minimal request subset
    - can cover all fields of all requests
    - full async oplock testing

Dual Server Testing

- The basis of gentest is 'dual server testing', the same basic technique used in the 'locktest' program from earlier versions of Samba:
  - The test program establishes two connections to each of two servers
  - Random requests are then generated, with identical requests sent to the two servers
  - At each step gentest compares every field of every response between the two servers
  - When a response differs gentest uses backtracking to find the minimal subset of the requests sent so far that generates a difference in response

Backtracking

- When a difference is discovered between the two servers gentest goes into 'analyze' mode, using a backtracking technique to find the minimal subset of requests that produce a difference:
    - successively smaller chunks of the request streams are blocked out
    - If a difference is still reported when a chunk is blocked out then that chunk is not needed and can be discarded
    - reconnects to the servers and wipes all files at each pass
    - The final pattern of requests can be replayed for analysis with a network sniffer

Process Models

- Samba3 only supported a “one client, one fork” process model
- In Samba4 the process model is pluggable, allowing the model to match the environment and backend
- Three process model modules are currently available:
    - 'single' - one process for all clients
    - 'standard' - the old Samba3 model
    - 'thread' - a pthread per client

pidl - autogenerated RPC

- In Samba4 we are finally moving to auto-generated RPC code, using a new IDL compiler called “pidl”
    - extended IDL syntax to support Microsoft “handwritten” RPC, including relative and subcontext RPC
    - auto-generation of test suite support code makes test suite generation easy
    - auto-generates both client and server code
    - work in progress to auto-generated server backends using ldb API
- Over 100k lines of Samba3 code have been replaced with less than 10k lines in Samba4

ldb - a new database API

- A little known fact is that internally Samba is database driven, using the tdb “trivial” database
- In Samba4 we will use ldb
    - a mid-point between LDAP and TDB
    - allows for “no-schema” operation
    - LDAP-like API
    - can either use a TDB or LDAP backend
    - very fast indexing
    - supports LDAP search expressions
- ldb will be used for all persistant databases. tdb will be used for temporary databases

Active Directory and PDC

- Our aim is to make Samba4 be a full ADS domain controller, plus a full NT4 domain controller
- We will use auto-generated mappings from IDL to ldb to store directory information
- The work to make Samba4 a domain controller is only just beginning, but the basic infrastructure looks good

The move to UTF-16

- In Samba3 we finally moved to full UCS-2 unicode support, greatly improving support for multi-byte languages
- For Samba4 we will move to UTF-16, to allow for support of those characters not in UCS-2.
- A new technique should mean that languages like Chinese and Japanese will actually be much faster than English in Samba4

Easier Install

- For Samba4 I want Samba to be much easier to install and configure
    - builtin web configuration in smbd - no extra setup
    - no base config file needed, just start daemon and use browser
    - new GUI for SWAT, including functionality from current command line tools
    - ldb+tdb means no messing about with LDAP setup

Current Status

- The effort to build Samba4 has so far taken 3 people about 14 months
    - RAW client library done
    - test suite done
    - NTVFS layer done
    - CIFS backend done
    - RPC/pidl infrastructure done
    - ldb done
- To get this far we have dropped a great deal of fundamental functionality.

More Info

- So, you want to help? Good!
    - Get the code from the svn.samba.org
    - Join the samba-technical IRC channel and mailing list
    - Not for the faint of heart! This is not production code yet
    - See http://samba.org/ftp/samba/slides/samba4_sambaxp04.pdf for a copy of these slides

Questions?