# Internationalization in Samba : Past, Present and Future


## By Jeremy Allison

**Development Team**

**email: jra@samba.org**

# Internationalization in the SMB/CIFS protocol

- Like NFS, early versions of the SMB/CIFS protocol ignored internationalization (I18N).
- DOS/Windows clients deal with I18N by having a "code page" representing the current character set being used.
- Client code page is important to the server in SMB/CIFS as most operations are done in a case insensitive way.
  - This is code page specific.
- Prior to Windows NT, SMB/CIFS assumed all clients had the same code page.

# I18N Extensions to SMB/CIFS

- Windows NT is a completely UNICODE aware operating system.
  - All system calls are UNICODE entry points.
  - Entire OS kernel is UNICODE native.
  - UCS2 (16 bit, little endian) is used internally.
- NT extensions to SMB/CIFS allowed server to inform clients it is UNICODE aware.
- Clients then may send some or all of their requests in UNICODE.
- Bugs mean some SMBs still use 8-bit requests.

# Early I18N work in Samba

- Like most software developed by English speaking authors, Samba initially ignored I18N issues.
- First I18N patches came from Japanese Samba users (Fujitsu).
  - Patches were Japan-specific.
  - Multibyte conversions had to be selected at compile time.
- Conversion to/from DOS codepage and UNIX character set added for filesystem code.
  - Single byte character sets - ISO8859-x only.

# Commercial I18N pressures

- Whilst working at Whistle, the need arose to be able to ship a single set of Samba binaries worldwide.
- Multi-byte character set changes had to be dynamically selectable.
- "client code page" parameter used to load a case conversion mapping file dynamically.
- "character set" parameter selects specific UNIX conversion.
- Multi-byte conversion names recognized internally.

# Current state of I18N in Samba (2.0.x)

- Internally to Samba, all strings are kept in DOS codepage format (specified by "client code page" setting).
  - This means no conversion when reading/writing filenames off the wire from/to the client.
  - Case insensitivity checking is done by converting to upper case using the current dynamically loaded codepage map.
- Only one client code page may be active at a time.
  - Clients from multiple non-ASCII codepages will see incorrect filenames.

# Current state of I18N in Samba (2.0.x) continued.

- Conversions from/to DOS codepage to UNIX character set are done just before writing/reading filenames from native system call interfaces.
- UNIX user/group databases (and plaintext passwords) are converted just before use.
- Data read from the smb.conf (Samba config file) is assumed to be in UNIX character set format and converted to DOS codepage format on read.
  - This means the "character set" parameter must be first in the config file.

# Current state of I18N in Samba (2.0.x) - Good enough ?

- Few customers use multiple code pages to a server.
  - SMB/CIFS is NOT an Internet protocol. Don't believe the marketing :-).
  - Commercial companies find Samba I18N support "good enough" to sell product.
- From the point of view of the Samba Team - No !
  - Code page to UNIX character set conversions are still statically compiled in.
  - Bugs in Windows clients when dealing with code page requests in NT specific SMB calls make exact NT semantics impossible without UNICODE.

# The move to UNICODE

- Samba needs to support UNICODE on the wire to Windows clients.
- Two possible decisions :
  - Use UTF8 internally, convert to 16 bit UNICODE when going to/from the wire.
  - Use UCS2 internally, just convert to native endian format when going to/from the wire.
- Under both systems, unless UNIX locale is set to UTF8, information can be lost when mapping from UNICODE to UNIX locale.
  - UNIX doesn't do multiple simultaneous character sets well - no UNICODE on disk or in system databases.

# Option 1: Samba with UTF8 internals

- Initially very attractive solution.
- Allows most of existing Samba code to stay unchanged.
  - Samba already deals with multi-byte character sets.
- UTF8 can be used directly for file names and system databases.
- Thin layer would convert between UTF8 and little endian UCS2 on read/writes to the wire.
- Downsides :
  - Native UNIX locale still a problem.
  - Character case conversion means converting to UCS2 (could optimize for single byte characters).

# Option 2: Samba with UCS2 internals

- Allows Samba to dispense with all multi-byte character code walk functions. All character manipulation assumes fixed length characters.
- Puts non-english speaking users on the same footing as English speaking users (a UNICODE coding bug will affect everyone :-).
- Allows reading/writing from the wire to be a simple matter of little-endian to native conversion.
- Downsides :
  - Buffer size doubles.
  - Native UNIX locale still a problem.

# Intractable problems

- Unless all applications are using the UTF8 locale, UNIX has a problem with storing multi-lingual characters.
  - Filenames from different locales seen incorrectly.
  - User and group databases cannot contain characters from different locales.
- This is a legacy POSIX interface problem that Windows NT does not have.
  - NT is native UNICODE throughout.
  - All system interfaces, databases and the native filesystem (NTFS) is UNICODE aware.
  - Samba cannot (single-handedly :-) fix POSIX.

# Our current planned solution

- Currently, we are planning to use UCS2 internally, and then attempt to convert to the specified UNIX locale when interfacing with the system.
  - Use Samba "character set" parameter, to keep consistency.
  - If UNIX locale is UTF8 then a correct mapping can be done.
  - If UNIX locale is not UTF8 then data will be mangled when attempting to convert client names to local UNIX format. No possible solution to this with current UNIX interfaces.

# Localization issues

- Most of Samba text is error messages designed to help the Samba team find and fix bugs.
  - It does not make sense to localize these strings.
  - Daemon arguments will not be localized.
- Localization is planned for user tools and the one component with a Web user interface (SWAT, the Samba Web Administration Tool).
- As with most Open Source projects, localization will depend upon user submissions.
  - Most core Team members primarily speak English.

# References.

- Samba Web site : http://samba.org
  - This contains current documentation about Samba I18N, and also a copy of the O'Reilly Samba book.
- CIFS Specification: http://www.cifs.com
  - Contains many useful CIFS references.
- Unicode consortium Web site: http://www.unicode.org
  - Contains code page to UNICODE mapping tables.
  - Contains case mapping tables for UNICODE characters.
- Slide availability:
  - ftp://samba.org/pub/samba/slides