WendzelNNTPd Version 2.2 Documentation

Steffen Wendzel and contributors

October 8, 2025

Contents

1 Introduction

WendzelNNTPd is a tiny but easy to use Usenet server (NNTP server) for Linux, *nix and BSD. The server is written in C.

1.1 Features

1.1.1 License

WendzelNNTPd uses the GPLv3 license.

1.1.2 Database Abstraction Layer

The server contains a database abstraction layer. Currently supported database systems are SQLite3 and MySQL (and experimental PostgreSQL support). New databases can be added in an easy way.

1.1.3 Security

WendzelNNTPd contains different security features, the most important features are probably Access Control Lists (ACLs) and the Role Based Access Control (RBAC) system. ACL and RBAC are described in a own chapter. WendzelNNTPd was probably the first Usenet server with support for RBAC.

Encrypted connections are feasible as WendzelNNTPd supports TLS v1.0 to v1.3 including STARTTLS!

Another feature that was introduced by WendzelNNTPd (and later adopted by other servers) are so-called "invisible newsgroups": If access control is activated, a user without permission to access the newsgroup will not be able to see the existence of the newsgroup. In case the user knows about the existence of the newsgroup nevertheless, he will not be able to post to or read from the newsgroup.

However, **please note** that the salting for password hashing is using SHA-256, but with a global user-definable salt that is concatenated with the username and password, rendering it less secure than using unique random hashes per password.

1.1.4 Auto-prevention of double-postings

In case a user sends a posting that lists the same newsgroup multiple times within one post command's "Newsgroups:" header tag, the server will add it only once to that newsgroup to save memory on the server and the time of the readers.

1.1.5 IPv6

WendzelNNTPd supports IPv6. The server can listen on multiple IP addresses as well as multiple ports.

1.1.6 Why this is not a perfect Usenet server

WendzelNNTPd does not implement all NNTP commands, but the (most) important ones. Another problem is that the regular expression library used is not 100% compatible with the NNTP matching in commands like "XGTITLE". Another limitation is that WendzelNNTPd cannot share messages with other NNTP servers.

1.1.7 Supported RFCs, NNTP commands and capabilities

The initial RFC 977 (Network News Transfer Protocol) for NNTP is partially supported:

- Supported commands: ARTICLE, BODY, HEAD, STAT, GROUP, HELP, LIST, POST, QUIT
- Unsupported commands: IHAVE, LAST, NEWGROUPS, NEWNEWS, NEXT, SLAVE

WendzelNNTPd also supports some commands from RFC 2980 (Common NNTP Extensions): LIST NEWSGROUPS, LIST OVERVIEW.FMT, LISTGROUP, MODE READER, XGTITLE, XHDR, XOVER, DATE

The newer NNTP standard RFC 3977 (Network News Transfer Protocol) supersedes RFC 977 and RFC 2980. WendzelNNTPd supports this standard partially. The new command CAPABILITIES is supported. The RFC contains some changes to the commands from RFC 977 and 2980. WendzelNNTPd does not support all changes. It is generally safer to assume that WendzelNNTPd behaves as described in RFC 977 and 2980.

RFC 4642 (Using Transport Layer Security (TLS) with NNTP) is supported (including the new command STARTTLS).

RFC 4643 (Extension for authentication) is partially supported. WendzelNNTPd supports authentication with username and password (the commands AUTHINFO USER and AUTHINFO PASS). Authentication with the Simple Authentication and Security Layer (SASL) is not supported (command AUTHINFO SASL).

The RFCs 4644 (Extension for Streaming Feeds), 6048 (Additions to LIST Command) and 8054 (Extension for Compression) are unsupported.

Supported capabilities: AUTHINFO, LIST, MODE-READER, POST, STARTTLS, VERSION

1.2 Contribute

See the *CONTRIBUTING* file.

1.3 History

The project started in 2004 and was written by Steffen Wendzel. Version 1.0 was released in 2007, version 2.0 in 2011, finally version 2.1 in 2021.

Since around 2018 the development was strongly supported by Wendzel's students at the University of Hagen. See the *AUTHORS* file for a list of major contributors.

A detailed history can be found in the *HISTORY* file.

2 Installation

This chapter provides a guide on how to install WendzelNNTPd 2.x.

2.1 Linux/*nix/BSD from source

To install WendzelNNTPd from source you can either download the provided archive file of a stable version (e.g., *v-2.x.y.tar.gz*) and extract it[1] or you can clone the current WendzelNNTPd development repository. Afterwards, run `./configure`. Please note that configure indicates missing libraries and packages that you may first need to install using the package system of your operating system.

```
# 1. Unpack the content of the tarball (or: use git clone).
$ tar -xzf v2.2.0.tar.gz

# 2. Switch into the unpacked directory (the name of the
#    directory could be different on your system, e.g.,
#    cdpxe-WendzelNNTPd-17d557d.
$ cd wendzelnntpd

# 3. Run the configure script:
$ ./configure
...
```

Please Note: *If you wish to compile WITHOUT MySQL or WITHOUT SQLite support*, then run `./configure --disable-mysql` or `./configure --disable-sqlite`, respectively.

Please Note: Run `configure` as well as `make` in the `bash` shell (under some BSDs you first need to install `bash`).*

Please Note: *If you wish to compile WITHOUT TLS support*, then run `./configure --disable-tls`.

Dependencies WendzelNNTPd depends on some programs and libraries. `./configure` will inform you about missing dependencies. Here is a list of packages for some distributions which provide the dependencies. The list may omit some packages which are already installed by default. Dependencies for the experimental PostgreSQL support are in brackets:

[1] On some *nix-like operating systems you need to first run `gzip -d wendzelnntpd-VERSION.tgz` and then `tar -xf wendzelnntpd-VERSION.tar` instead of letting `tar` do the whole job as in the listing.

- Debian/Ubuntu: gcc flex bison sqlite3 libsqlite3-dev libmariadb-dev-compat ca-certificates libmariadb-dev libmhash-dev make openssl libssl-dev (libpq-dev)
- Fedora: gcc flex bison sqlite sqlite-devel mariadb-connector-c-devel ca-certificates mhash-devel make openssl openssl-devel (libpq-devel)
- openSUSE Leap: gcc flex bison sqlite3 sqlite3-devel libmariadb-devel ca-certificates mhash-devel make openssl libopenssl-devel (postgresql-devel)
- Arch Linux: gcc flex bison sqlite mariadb-libs ca-certificates mhash make openssl (postgresql-libs)
- FreeBSD 14: bash sqlite3 bison mhash mariadb114-client (postgresql17-client)
- OpenBSD: bash bison mhash mariadb-client (postgresql-client)
- NetBSD 10: bash bison mhash mariadb-client (postgresql17-client)

Compiling WendzelNNTPd After `configure` has finished, run `make`:

```
$ make
...
```

Installing WendzelNNTPd To install WendzelNNTPd on your system, you need superuser access. Run `make install` to install it to the default location */usr/local/**.

```
$ sudo make install
```

Please Note (Upgrades): Run `sudo make upgrade` instead of `sudo make install` for an upgrade. Please cf. Section Upgrading.

Please Note (MySQL): *If you plan to run MySQL*, then no database was set up during `make install`. Please refer to Section Basic Configuration to learn how to generate the MySQL database.

Generating SSL certificates TLS is enabled by default in *wendzelnntpd.conf* as long as WendzelNNTPd has not been compiled without TLS support. `make install` generates a self-signed certificate for usage on localhost so that TLS can be used out-of-the-box.

If you want to generate an SSL certificate, which is signed by Let's Encrypt, or a new self-signed certificate, you can use the helper script `create_certificate`:

```
$ sudo create_certificate \
    --environment letsencrypt \
    --email <YOUR-EMAIL> \
    --domain <YOUR-DOMAIN>
```

For the parameter `--environment`, "*local*" is also a valid value. In that case, the certificate is generated only for usage on localhost and is self-signed. The location of the generated certificates can be adjusted with the parameter `--targetdir`. You also need to adjust the paths in *wendzelnntpd.conf* if you use a non-default location (check Section Encrypted connections over TLS).

2.1.1 Automatic startup

There is an init script and a systemd service unit in the directory *scripts/startup* for automatic startup of `wendzelnntpd`. More information can be found in Automating Start, Stop, and Restart

2.2 Packages for Linux and BSD

WendzelNNTPd is available as a package for some Linux and BSD distributions. You can use them to install WendzelNNTPd instead of installing it from source. Please consult the documentation of your distribution for further information about package installation. Here is a list of known packages:

- *Slackware Linux*:
  - Slackware: Slackbuilds.org Build Script Slackware 14.2, Slackware 15.0
  - Slackware64-current: Slackware package (Installation via `installpkg (filename)`)
- *NetBSD*: WendzelNNTPd port at pkgsrc
- *Debian* and *Ubuntu*:
  - WendzelNNTPd at openSUSE Build Service (provides the packages wendzelnntpd and wendzelnntpd-doc, the doc package contains the documentation in HTML and PDF format)

2.3 Docker image for Linux

WendzelNNTPd is available as a Docker image on Docker Hub.

Run WendzelNNTPd in a Docker container

```
$ docker run --name wendzelnntpd -d -p 119:119 -p 563:563 cdpxe/wendzelnntpd
```

Specify volumes for the database and configuration files

```
$ docker run --name my-wendzelnntpd -d -p 119:119 -p 563:563 \
    -v wendzelnntpd_config:/usr/local/etc/wendzelnntpd \
    -v wendzelnntpd_data:/var/spool/news/wendzelnntpd \
    cdpxe/wendzelnntpd
```

Administration with wendzelnntpadm

```
$ docker run --rm -v wendzelnntpd_config:/usr/local/etc/wendzelnntpd \
    -v wendzelnntpd_data:/var/spool/news/wendzelnntpd \
     cdpxe/wendzelnntpd wendzelnntpadm
```

Find further information regarding the administration of WendzelNNTPd in Running.

Create new certificates

```
$ docker run --rm -v wendzelnntpd_config:/usr/local/etc/wendzelnntpd \
    cdpxe/wendzelnntpd create_certificate
```

Further information in Generating SSL certificates.

Get the configuration file Copy the default configuration file from the image to the host:

```
$ docker run --rm --entrypoint=cat cdpxe/wendzelnntpd \
    /usr/local/etc/wendzelnntpd/wendzelnntpd.conf \
    > /home/youruser/wendzelnntpd.conf
```

Or copy the current configuration file from your container:

```
$ docker cp my-wendzelnntpd:/usr/local/etc/wendzelnntpd/wendzelnntpd.conf \
    /home/youruser/wendzelnntpd.conf
```

Edit the configuration file Find further information regarding the configuration of WendzelNNTPd in Configuration.

Provide the configuration file to the container Copy the configuration file back to the container:

```
$ docker cp /home/youruser/wendzelnntpd.conf \
    my-wendzelnntpd:/usr/local/etc/wendzelnntpd/wendzelnntpd.conf
```

Or bind mount the configuration file (the file needs to be owned by root in this case):

```
$ sudo chown 0:0 /home/youruser/wendzelnntpd.conf
$ docker run --name wendzelnntpd -d -p 119:119 -p 563:563 \
    -v wendzelnntpd_config:/usr/local/etc/wendzelnntpd \
    -v wendzelnntpd_data:/var/spool/news/wendzelnntpd \
    -v /home/youruser/wendzelnntpd.conf:\
/usr/local/etc/wendzelnntpd/wendzelnntpd.conf:ro \
     -d cdpxe/wendzelnntpd
```

Or build a new image with your configuration file:

```
FROM cdpxe/wendzelnntpd
COPY wendzelnntpd.conf /usr/local/etc/wendzelnntpd/wendzelnntpd.conf
```

Create the Docker image from source You can also build the Docker image from source instead of using the pre-built image from Docker Hub. Therefore, you need to get the source code of WendzelNNTPd like explained in Linux/*nix/BSD from source. After that you can build the image with `make`:

```
$ make docker-build
```

2.4 Unofficial Note: Mac OS X

A user reported WendzelNNTPd-2.0.0 is installable under Mac OS X 10.5.8. The only necessary change was to add the flag `-arch x86_64` to compile the code on a 64 bit system. However, I never tried to compile WendzelNNTPd on a Mac.

2.5 Windows

Not supported.

2.6 Installed files

This documentation assumes that you have installed WendzelNNTPd with default paths to `/usr/local`. The installation prefix as well as the path for the subdirectories like `sbin` or `share/man/man8` can be changed when calling `./configure` or `make install`. Please consult `./configure --help` for a list of available options. Here is an overview of the installed files as well as files which are created during runtime:

Path	Description
/usr/local/etc/wendzelnntpd/wendzelnntpd.conf	Cofiguration file
/usr/local/etc/wendzelnntpd/ssl/*	SSL certificates for encrypted connections over TLS
/usr/local/sbin/create_certificate	Script for generating SSL certificates
/usr/local/sbin/wendzelnntpadm	Administration tool
/usr/local/sbin/wendzelnntpd	The Usenet server
/usr/local/share/doc/wendzelnntpd/*	Various documentation files
/usr/local/share/man/man5/*	Man pages for configuration files
/usr/local/share/man/man8/*	Man pages for commands
/usr/local/share/wendzelnntpd/mysql_db_struct.sql	SQL file to create the database schema of the MySQL database
/usr/local/share/wendzelnntpd/openssl.cnf	openssl config for create_certificate
/usr/local/share/wendzelnntpd/usenet.db_struct	SQL file to create the database schema of the SQLite database
/var/log/wendzelnntpd	Logfile
/var/spool/news/wendzelnntpd/cdp*	Message bodies of the postings
/var/spool/news/wendzelnntpd/nextmsgid	Next unique message id
/var/spool/news/wendzelnntpd/usenet.db	SQLite database

3 Basic Configuration

This chapter will explain how to configure WendzelNNTPd after installation.

Note: The configuration file for WendzelNNTPd is named */usr/local/etc/wendzelnntpd/wendzelnntpd.conf*. The format of the configuration file should be self-explanatory and the default configuration file includes many comments which will help you to understand its content.

Note: On *nix-like operating systems the default installation path is */usr/local/**, i.e., the configuration file of WendzelNNTPd will be */usr/local/etc/wendzelnntpd/wendzelnntpd.conf*, and the binaries will be placed in */usr/local/sbin*.

3.1 Choosing a database engine

The first and most important step is to choose a database engine. You can use either SQLite3 (this is the default case and easy to use, but not suitable for larger systems with many thousand postings or users) or MySQL (which is the more advanced solution, but also a little bit more complicated to realize). By default, WendzelNNTPd is configured for SQLite3 and is ready to run. If you want to keep this setting, you do not have to read this section.

3.1.1 Modifying wendzelnntpd.conf

In the configuration file you will find a parameter called **database-engine**. You can choose to use either MySQL or SQLite as the backend storage system by appending either **sqlite** or **mysql**. Experimental support for PostgreSQL can be activated with **postgres**.

```
database-engine mysql
```

If you choose to use MySQL then you will also need to specify the user and password which WendzelNNTPd must use to connect to the MySQL server. If your server does not run on localhost or uses a non-default MySQL port then you will have to modify these values too.

```
; Your database hostname (not needed for sqlite3)
database-server 127.0.0.1

; the database connection port (not needed for sqlite3)
; Comment out to use the default port of your database engine
database-port 3306
```

```
; Server authentication (not needed for sqlite3)
database-username mysqluser
database-password supercoolpass
```

3.1.2 Generating your database tables

Once you have chosen your database backend you will need to create the database and the required tables.

SQLite

If you chose SQLite as your database backend then you can skip this step as running `make install` does this for you.

Note: The SQLite database file as well as the posting management files will be stored in */var/spool/news/wendzelnntpd/*.

MySQL

For MySQL, an SQL script file called *mysql_db_struct.sql* is included. It creates the WendzelNNTPd database and all the needed tables. Use the MySQL console tool to execute the script.

```
$ cd /path/to/your/extracted/wendzelnntpd-archive/
$ mysql -u YOUR-USER -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 5.1.37-1ubuntu5.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> source mysql\_db\_struct.sql
...
mysql> quit
Bye
```

PostgreSQL

Similarly to MySQL, there is a SQL script file (*postgres_db_struct.sql*) to create the WendzelNNTPd database. Create and setup a new database (and a corresponding user) and use the `psql(1)` command line client to load table and function definitions:

```
$ psql --username USER -W wendzelnntpd
wendzelnntpd=> begin;
wendzelnntpd=> \i database/postgres_db_struct.sql
wendzelnntpd=> commit; quit;
```

3.1.3 Store message body and count in the database

WendzelNNTPd stores the message body and the unique message index over the filesystem normally. They can also be stored in the database instead (currently **only supported for PostgreSQL**). `message-body-in-db` enables the storage of the message body in the database and `message-count-in-db` enables the storage of the unique message index in the database. Both settings should not be changed after initial utilization of the server as old message bodies would not be locatable anymore and the message counter would be corrupted.

```
message-body-in-db
message-count-in-db
```

3.2 Network Settings

For each type of IP address (IPv4 and/or IPv6) you have to define an own connector. You can find a minimal example for NNTP over port 119 on localhost over IPv4 and IPv6 below.

```
<connector>
    port        119
    listen      127.0.0.1
</connector>

<connector>
    port        119
    listen      ::1
</connector>
```

3.2.1 Encrypted connections over TLS

WendzelNNTPd supports encrypted connections over TLS. There are two ways to use TLS: STARTTLS and dedicated TLS (SNNTP). Both ways require an SSL certificate, which is created during installation by default, but you can also create a new certificate or provide your own (see Generating SSL certificates for more information). The configuration options `tls-server-certificate`, `tls-server-key` and `tls-ca-certificate` contain the paths to the SSL certificate, the private key of the certificate and the certificate of the certificate authority (CA) which signed the SSL certificate. These options are required for using TLS or STARTTLS with NNTP. All other TLS-related options are optional.

STARTTLS

STARTTLS can be added to an existing port and makes it possible to create encrypted and unencrypted connections over the same port. Encrypted connections are created by starting an unencrypted connection and switching to TLS using the STARTTLS NNTP command. The example below is for NNTP over port 119.

```
<connector>
    ;; enables STARTTLS for this port
    enable-starttls
    port        119
    listen      127.0.0.1
    ;; configure SSL server certificate (required)
    tls-server-certificate "/usr/local/etc/wendzelnntpd/ssl/server.crt"
    ;; configure SSL private key (required)
    tls-server-key "/usr/local/etc/wendzelnntpd/ssl/server.key"
    ;; configure SSL CA certificate (required)
    tls-ca-certificate "/usr/local/etc/wendzelnntpd/ssl/ca.crt"
    ;; configure TLS ciphers for TLSv1.3
    tls-cipher-suites "TLS_AES_128_GCM_SHA256"
    ;; configure TLS ciphers for TLSv1.1 and TLSv1.2
    tls-ciphers "ALL:!COMPLEMENTOFDEFAULT:!eNULL"
    ;; configure allowed TLS version (1.0-1.3)
    tls-version "1.2-1.3"
</connector>
```

Dedicated TLS (SNNTP)

To use dedicated TLS with NNTP (SNNTP) you can define another connector. This connector only accepts encrypted connections. The example below is for SNNTP over port 563.

```
<connector>
    ;; enables TLS for this port
    enable-tls
    port        563
    listen      127.0.0.1
    ;; configure SSL server certificate (required)
    tls-server-certificate "/usr/local/etc/wendzelnntpd/ssl/server.crt"
    ;; configure SSL private key (required)
    tls-server-key "/usr/local/etc/wendzelnntpd/ssl/server.key"
    ;; configure SSL CA certificate (required)
    tls-ca-certificate "/usr/local/etc/wendzelnntpd/ssl/ca.crt"
    ;; configure TLS ciphers for TLSv1.3
    tls-cipher-suites "TLS_AES_128_GCM_SHA256"
    ;; configure TLS ciphers for TLSv1.1 and TLSv1.2
    tls-ciphers "ALL:!COMPLEMENTOFDEFAULT:!eNULL"
    ;; configure allowed TLS version (1.0-1.3)
    tls-version "1.2-1.3"
</connector>
```

Mutual authentication (mTLS) and certificate revocation lists (CRLs)

WendzelNNTPd supports mutual authentication over TLS (mTLS). The option `tls-verify-client` enables mTLS, which accepts the following options:

- require: the client certificate is requested and validated by the server
- optional: the client certificate is not requested by the server, but validated if sent by the client
- none: the client certificate is neither requested nor validated by the server

The option `tls-verify-client-depth` defines the depth of the validation of the certificate chain. It needs to be set to 0 if selfi-signed certificates are used.

It is possible to check the client certificates against a certificate revocation list (CRL). The option `tls-crl` enables CRL checking, which accepts the following options:

- chain: the entire certificate chain of the client certificate is checked against the CRL
- leaf: the client certificate is checked against the CRL
- none: CRL checking is disabled

```
; mutual auth needed
<connector>
    enable-tls
    port        563
    listen      127.0.0.1

    tls-server-certificate "/usr/local/etc/wendzelnntpd/ssl/server.crt"
    tls-server-key "/usr/local/etc/wendzelnntpd/ssl/server.key"
    tls-ca-certificate "/usr/local/etc/wendzelnntpd/ssl/ca.crt"
    tls-cipher-suites "TLS_AES_128_GCM_SHA256"
    tls-ciphers "ALL:!COMPLEMENTOFDEFAULT:!eNULL"
    tls-version "1.3-1.3"

    ; possibility to force the client to authenticate with
    ; client certificate (none | optional | require)
    tls-verify-client "required"
    tls-verify-client-depth 0

    ; possibility to use certificate revocation list
    ; (none | leaf | chain)
    tls-crl "leaf"
    tls-crl-file "/usr/local/etc/wendzelnntpd/ssl/ssl.crl"
</connector>
```

Mandatory TLS

It is possible to make TLS mandatory with the parameter `tls-is-mandatory`. This is a global setting that will be applied to all connectors and needs to be set outside of the connectors.

```
tls-is-mandatory
```

3.3 Setting the Allowed Size of Postings

To change the maximum size of a post to be accepted by the server, change the variable **max-size-of-postings**. The value must be set in Bytes and the default value is 20971520 (20 MBytes).

```
max-size-of-postings 20971520
```

3.4 Verbose Mode

If you have any problems running WendzelNNTPd or if you simply want more information about what is happening, you can uncomment the **verbose-mode** line.

```
; Uncomment 'verbose-mode' if you want to find errors or if you
; have problems with the logging subsystem. All log strings are
; written to stderr too, if verbose-mode is set. Additionally, all
; commands sent by clients are written to stderr too (but not to
; logfile)
verbose-mode
```

3.5 Security Settings

3.5.1 Authentication and Access Control Lists (ACL)

WendzelNNTPd contains an extensive access control subsystem. If you want to only allow authenticated users to access the server, you should uncomment **use-authentication**. This gives every authenticated user access to each newsgroup.

```
; Activate authentication
use-authentication
```

If you need a slightly more advanced authentication system, you can activate Access Control Lists (ACL) by uncommenting **use-acl**. This activates the support for Role-based ACL too.

```
; If you activated authentication, you can also activate access
; control lists (ACL)
use-acl
```

3.5.2 Anonymized Message-ID

By default, WendzelNNTPd makes a user's hostname or IP address part of new message IDs when a user sends a post using the NNTP POST command. If you do not want that, you can force WendzelNNTPd not to do so by uncommenting **enable-anonym-mids**, which enables anonymized message IDs.

```
; This prevents that IPs or Hostnames will become part of the
; message ID generated by WendzelNNTPd what is the default case.
; Uncomment it to enable this feature.
enable-anonym-mids
```

3.5.3 Changing the Default Salt for Password Hashing

When uncommenting the keyword **hash-salt**, the default salt value that is used to enrich the password hashes can be changed. Please note that you have to define the salt *before* you set-up the first password since it will otherwise be stored hashed using an old salt, rendering it unusable. For this reason, it is necessary to define your salt right after running **make install** (or at least before the first creation of NNTP user accounts).

```
; This keyword defines a salt to be used in conjunction with the
; passwords to calculate the cryptographic hashes. The salt must
; be in the form [a-zA-Z0-9.:\/-_]+.
; ATTENTION: If you change the salt after passwords have been
; stored, they will be rendered invalid! If you comment out
; hash-salt, then the default hash salt defined in the source
; code will be used.
hash-salt 0.hG4//3baA-::_\
```

WendzelNNTPd applies the SHA-2 hash algorithm using a 256 bit hash value. Please also note that the final hash is calculated using a string that combines salt, username and password as an input to prevent password-identification attacks when an equal password is used by multiple users. However, utilizing the username is less secure than having a completely separate salt for every password.[1]

3.5.4 Encrypted communication (TLS) and mutual authentication (mTLS)

Please look at section Encrypted connections over TLS when you want to use encryption and mutual authentication over TLS.

3.6 Configuration options reference

[1] Patches are appreciated!

Configuration option	Description
database-engine	Specify a database engine (see Choosing a database engine)
database-password	Password for authentication with the database (see Choosing a database engine)
database-port	Your database port (see Choosing a database engine)
database-server	Your database hostname (see Choosing a database engine)
database-username	Username for authentication with the database (see Choosing a database engine)
port	The port number that the server will listen on (see Network Settings)
enable-anonym-mids	Prevents that IPs or hostnames will become part of the message ID (see Anonymized Message-ID)
enable-starttls	Enables STARTTLS for this port (see STARTTLS)
enable-tls	Enables TLS for this port (see Dedicated TLS (SNNTP))
hash-salt	The salt to be used in conjunction with the passwords to calculate the cryptographic hashes (see Changing the Default Salt for Password Hashing)
listen	The IP address that the server will listen on (see Network Settings)
max-size-of-postings	The max. allowed size of a single posting (in bytes) (see Setting the Allowed Size of Postings)
message-body-in-db	Store the Message Body in the database (only possible with postgres) (see Store message body and count in the database)
message-count-in-db	Store/load the unique message index within the database system (only possible with postgres) (see Store message body and count in the database)
tls-ca-certificate	Configure SSL CA certificate (see Encrypted connections over TLS)
tls-cipher-suites	Configure TLS ciphers for TLSv1.3 (see Encrypted connections over TLS)
tls-ciphers	Configure TLS ciphers for TLSv1.1 and TLSv1.2 (see Encrypted connections over TLS)
tls-crl	Enables verification of client certificates against CRLs for mTLS (see mTLS and CRLs)
tls-crl-file	Path to the certificate revocation list (see mTLS and CRLs)
tls-is-mandatory	Make TLS mandatory for all connectors (see Mandatory TLS)

Configuration option	Description
tls-server-certificate	Configure SSL server certificate (see Encrypted connections over TLS)
tls-server-key	Configure SSL private key (see Encrypted connections over TLS)
tls-verify-client	Enables mutual authentication over TLS (see mTLS and CRLs)
tls-verify-client-depth	Depth of the validation of the certificate chain (see mTLS and CRLs)
tls-versions	Configure allowed TLS versions (see Encrypted connections over TLS
use-authentication	Activate authentication (see Authentication and Access Control Lists)
use-acl	Activate access control lists (ACL) (see Authentication and Access Control Lists)
verbose-mode	Write additional log messages and write logs to STDERR (see Verbose Mode)

4 Starting and Running WendzelNNTPd

4.1 Starting the Service

Once your WendzelNNTPd installation has been configured, you can run the server (in the default case you need superuser access to do that since this is required to bind WendzelNNTPd to the default NNTP port 119) by starting **/usr/local/sbin/wendzelnntpd**.

```
$ /usr/local/sbin/wendzelnntpd
WendzelNNTPd: version 2.0.7 'Berlin'  - (Oct 26 2015 14:10:20 #2544) is ready.
```

Note (Daemon Mode): If you want to run WendzelNNTPd as a background daemon process on *nix-like operating systems, you should use the parameter **-d**.

4.2 Stopping and Restarting the Service

The server can be stopped by terminating its process:

```
$ pkill wendzelnntpd
```

The server has a handler for the termination signal that allows to safely shutdown using **pkill** or **kill**.

To restart the service, terminate and start the service.

4.2.1 Automating Start, Stop, and Restart

init.d script

The script **init.d_script** in the directory *scripts/startup* of the tarball can be used to start, restart, and stop WendzelNNTPd. It is a standard *init.d* script for Linux operating systems that can usually be copied to */etc/init.d* (it must be executable).

```
$ cp scripts/startup/init.d_script /etc/init.d/wendzelnntpd
$ chmod +x /etc/init.d/wendzelnntpd
```

Note: Please note that some operating systems use different directories than */etc/init.d* or other startup script formats. In such cases, the script works nevertheless but should simply be installed to */usr/local/sbin* instead.

To start, stop, and restart WendzelNNTPd, the following commands can be used afterwards:

```
$ /etc/init.d/wendzelnntpd start
Starting WendzelNNTPd ... done.
WendzelNNTPd: version 2.0.7 'Berlin'  - (Oct 26 2015 14:10:20 #2544) is ready.

$ /etc/init.d/wendzelnntpd restart
Stopping WendzelNNTPd ... done.
Starting WendzelNNTPd ... done.
WendzelNNTPd: version 2.0.7 'Berlin'  - (Oct 26 2015 14:10:20 #2544) is ready.

$ /etc/init.d/wendzelnntpd stop
Stopping WendzelNNTPd ... done.
```

systemd service unit

The project also includes a systemd service unit file that can be used to start, stop, and restart wendzelnntpd on systems which use systemd as their init system. The file *scripts/startup/wendzelnntpd.service* is present after building the project with `make`. You can install it by copying it to */etc/systemd/system* and then reloading the unit files:

```
$ sudo cp scripts/startup/wendzelnntpd.service /etc/systemd/system/
$ sudo systemctl daemon-reload
```

To start, stop, and restart WendzelNNTPd, to enable and disable the service and to show the status, the following commands can be used afterwards:

```
$ sudo systemctl start wendzelnntpd.service
$ sudo systemctl status wendzelnntpd
wendzelnntpd.service - WendzelNNTPd Usenet server
     Loaded: loaded (/etc/systemd/system/wendzelnntpd.service; enabled; preset: enabled)
     Active: active (running) since Sat 2025-09-27 21:25:30 CEST; 1min 11s ago
       Docs: man:wendzelnntpd(8)
   Main PID: 105845 (wendzelnntpd)
      Tasks: 1 (limit: 38345)
     Memory: 1.5M
        CPU: 1min 11.173s
     CGroup: /system.slice/wendzelnntpd.service
...
$ sudo systemctl restart wendzelnntp
$ sudo systemctl stop wendzelnntpd
$ sudo systemctl enable wendzelnntpd
Created symlink /etc/systemd/system/multi-user.target.wants/wendzelnntpd.service
    → /etc/systemd/system/wendzelnntpd.service.
$ sudo systemctl disable wendzelnntpd
Removed "/etc/systemd/system/multi-user.target.wants/wendzelnntpd.service".
```

4.3 Administration Tool 'wendzelnntpadm'

Use the command line tool **wendzelnntpadm** to configure users, roles and newsgroups of your WendzelNNTPd installation. To get an overview of supported commands, run `wendzelnntpadm help`:

```
$ wendzelnntpadm help
usage: wendzelnntpd <command> [parameters]
*** Newsgroup Administration:
 <listgroups>
 <addgroup | modgroup> <newsgroup> <posting-allowed-flag (y/n)>
 <delgroup> <newsgroup>
*** User Administration:
 <listusers>
 <adduser> <username> [<password>]
 <deluser> <username>
*** ACL (Access Control List) Administration:
 <listacl>
 <addacluser | delacluser> <username> <newsgroup>
 <addaclrole | delaclrole> <role>
 <rolegroupconnect | rolegroupdisconnect> <role> <newsgroup>
 <roleuserconnect | roleuserdisconnect> <role> <username>
```

4.4 Creating/Listing/Deleting Newsgroups

You can either list, create or delete newsgroups using **wendzelnntpadm**.

4.4.1 Listing existing newsgroups

```
$ wendzelnntpadm listgroups
Newsgroup, Low-, High-Value, Posting-Flag
-----------------------------------------
alt.test 10 1 y
mgmt.talk 1 1 y
secret.project-x 20 1 y
done.
```

4.4.2 Creating a new newsgroup

To create a new newsgroup run the following command:

```
$ wendzelnntpadm addgroup my.cool.group y
Newsgroup my.cool.group does not exist. Creating new group.
done.
```

You can also change the "posting allowed" flag of a newsgroup:

```
$ wendzelnntpadm modgroup my.cool.group y
Newsgroup my.cool.group exists: okay.
done.
$ wendzelnntpadm modgroup my.cool.group n
Newsgroup my.cool.group exists: okay.
done.
```

4.4.3 Deleting a newsgroup

```
$ wendzelnntpadm delgroup my.cool.group
Newsgroup my.cool.group exists: okay.
Clearing association class ... done
Clearing ACL associations of newsgroup my.cool.group... done
Clearing ACL role associations of newsgroup my.cool.group... done
Deleting newsgroup my.cool.group itself ... done
Cleanup: Deleting postings that do not belong to an existing newsgroup ... done
done.
```

4.5 User Accounts Administration

The easiest way to give only some people access to your server is to create user accounts (please make sure you activated authentication in your configuration file). You can add, delete and list all users.

4.5.1 Listing Users (and Passwords)

This command always prints the (hashed) password of the users:

```
$ wendzelnntpadm listusers
Username, Password
------------------
developer1, 8d67111f9e4fd067cd5e420267c87ea07cbedffa0a21bdcf296de17c7745ae22
developer2, 162d3865cbc7cd028a782c9cf48e287bbc7cdd6206260e0110c76560d3c24da0
manager1, 6b33743ed8443f561dce6bab8740a36d2855b4f14a626bdc232693f10b69d635
manager2, 9115cefd91e130dfe0e7fb66dc4d5f42c19cfc5ff8c19b680871116cec0476d7
swendzel, 66a86c99f02cc2f7b3a3c40a6a1549eeaeaa92b2e5e38b2554446e18069268d0
swendzel2, 97aefc76b7d35254d51c13a10cd9f059d9ff343042448d9449f9d225c1ccc5f4
swendzel3, 64b762f32fba01816dec50f834a2a97dd897ef20fbe33a1840dfdf4484e344e9
swendzel4, ea9c61de864bda5bf75ec3c2912a310918537a40cdb9aef075a536e3d149cd16
done.
```

4.5.2 Creating a new user

You can either enter the password as additional parameter (useful for scripts that create users automatically) ...

```
$ wendzelnntpadm adduser UserName HisPassWord
User UserName does currently not exist: okay.
done.
```

... or you can type it using the prompt (in this case the input is shadowed):

```
$ wendzelnntpadm adduser UserName2
Enter new password for this user (max. 100 chars):
User UserName2 does currently not exist: okay.
done.
```

Please Note: A password must include at least 8 characters and may not include more than 100 characters.

4.5.3 Deleting an existing user

```
$ wendzelnntpadm deluser UserName2
User UserName2 exists: okay.
Clearing ACL associations of user UserName2... done
Clearing ACL role associations of user UserName2... done
Deleting user UserName2 from database ... done
done.
```

4.6 Access Control List Administration (in case the standard NNTP authentication is not enough)

Welcome to the advanced part of WendzelNNTPd. WendzelNNTPd includes a powerful role-based access control system. You can either only use normal access control lists where you can configure which user will have access to which newsgroup. Or you can use the advanced role system: You can add users to roles (e.g., the user "boss99" to the role "management") and give a role access to a group (e.g., role "management" shall have access to "discuss.management").

Note: Please note that you must activate the ACL feature in your configuration file to use it.

Note: To see *all* data related to the ACL subsystem of your WendzelNNTPd installation, simply use "wendzelnntpadm listacl".

4.6.1 Invisible Newsgroups

WendzelNNTPd includes a feature called "Invisible Newsgroups" which means that a user without access to a newsgroup will neither see the newsgroup in the list of newsgroups, nor will he be able to post to such a newsgroup or will be able to read it.

4.6.2 Simple Access Control

We start with the simple access control component where you can define which user will have access to which newsgroup.

Giving a user access to a newsgroup

```
$ wendzelnntpadm addacluser swendzel alt.test
User swendzel exists: okay.
Newsgroup alt.test exists: okay.
done.
$ wendzelnntpadm listacl
List of roles in database:
Roles
-----

Connections between users and roles:
Role, User
----------

Username, Has access to group
-----------------------------
swendzel, alt.test

Role, Has access to group
-------------------------
done.
```

Removing a user's access to a newsgroup

```
$ wendzelnntpadm delacluser swendzel alt.test
User swendzel exists: okay.
Newsgroup alt.test exists: okay.
done.
```

4.6.3 Adding and Removing ACL Roles

If you have many users, some of them should have access to the same newsgroup (e.g., the developers of a new system should have access to the development newsgroup of the system). With roles you do not have to give every user explicit access to such a group. Instead, you add the users to a role and give the role access to the group. (One advantage is that you can easily give the complete role access to another group with only one command instead of adding each of its users to the list of people who have access to the new group).

In the following examples, we give the users "developer1", "developer2", and "developer3" access to the development role of "project-x" and connect their role to the newsgroups

"project-x.discussion" and "project-x.support". To do so, we create the three users and the two newsgroups first:

```
$ wendzelnntpadm adduser developer1
Enter new password for this user (max. 100 chars):
User developer1 does currently not exist: okay.
done.
$ wendzelnntpadm adduser developer2
Enter new password for this user (max. 100 chars):
User developer2 does currently not exist: okay.
done.
$ wendzelnntpadm adduser developer3
Enter new password for this user (max. 100 chars):
User developer3 does currently not exist: okay.
done.

$ wendzelnntpadm addgroup project-x.discussion y
Newsgroup project-x.discussion does not exist. Creating new group.
done.
$ wendzelnntpadm addgroup project-x.support y
Newsgroup project-x.support does not exist. Creating new group.
done.
```

Creating an ACL Role

Before you can add users to a role and before you can connect a role to a newsgroup, you have to create an ACL *role* (you have to choose an ASCII name for it). In this example, the group is called "project-x".

```
$ wendzelnntpadm addaclrole project-x
Role project-x does not exists: okay.
done.
```

Deleting an ACL Role

You can delete an ACL role by using "delaclrole" instead of "addaclrole" like in the example above.

4.6.4 Connecting and Disconnecting Users with/from Roles

To add (connect) or remove (disconnect) a user to/from a role, you need to use the admin tool too.

Connecting a User with a Role

The second parameter ("project-x") is the role name and the third parameter ("developer1") is the username. Here we add our three developer users from the example above to the group project-x:

```
$ wendzelnntpadm roleuserconnect project-x developer1
Role project-x exists: okay.
User developer1 exists: okay.
Connecting role project-x with user developer1 ... done
done.
$ wendzelnntpadm roleuserconnect project-x developer2
Role project-x exists: okay.
User developer2 exists: okay.
Connecting role project-x with user developer2 ... done
done.
$ wendzelnntpadm roleuserconnect project-x developer3
Role project-x exists: okay.
User developer3 exists: okay.
Connecting role project-x with user developer3 ... done
done.
```

Disconnecting a User from a Role

```
$ wendzelnntpadm roleuserdisconnect project-x developer1
Role project-x exists: okay.
User developer1 exists: okay.
Dis-Connecting role project-x from user developer1 ... done
done.
```

4.6.5 Connecting and Disconnecting Roles with/from Newsgroups

Even if a role is connected with a set of users, it is still useless until you connect the role with a newsgroup.

Connecting a Role with a Newsgroup

To connect a role with a newsgroup, we have to use the command line tool for a last time (the 2nd parameter is the role, and the 3rd parameter is the name of the newsgroup). Here we connect our "project-x" role to the two newsgroups "project-x.discussion" and "project-x.support":

```
$ wendzelnntpadm rolegroupconnect project-x project-x.discussion
Role project-x exists: okay.
Newsgroup project-x.discussion exists: okay.
Connecting role project-x with newsgroup project-x.discussion ... done
done.
$ wendzelnntpadm rolegroupconnect project-x project-x.support
Role project-x exists: okay.
Newsgroup project-x.support exists: okay.
Connecting role project-x with newsgroup project-x.support ... done
done.
```

Disconnecting a Role From a Newsgroup

Disconnecting is done like in the example above but you have to use the command "rolegroup**dis**connect" instead of "rolegroupconnect".

4.6.6 Listing Your Whole ACL Configuration Again

Like mentioned before, you can use the command "listacl" to list your whole ACL configuration (except the list of users that will be listed by the command "listusers").

```
$ wendzelnntpadm listacl
List of roles in database:
Roles
-----
project-x

Connections between users and roles:
Role, User
----------
project-x, developer1
project-x, developer2
project-x, developer3

Username, Has access to group
-----------------------------
swendzel, alt.test

Role, Has access to group
-------------------------
project-x, project-x.discussion
project-x, project-x.support
done.
```

Saving time

As mentioned above, you can save time by using roles. For instance, if you add a new developer to the system, and the developer should have access to the two groups "project-x.discussion" and "project-x.support", you do not have to assign the user to both (or even more) groups by hand. Instead, you just add the user to the role "project-x" that is already connected to both groups.

If you want to give all developers access to the group "project-x.news", you also do not have to connect every developer with the project. Instead, you just connect the role with the newsgroup, what is one command instead of n commands. Of course, this time-saving concept also works if you want to delete a user.

4.7 Hardening

Besides the already mentioned authentication, ACL and RBAC features, the security of the server can be improved by putting WendzelNNTPd in a *chroot* environment or letting it run under an unprivileged user account (the user then needs write access to */var/spool/news/wendzelnntpd* and read access to (*/usr/local)/etc/wendzelnntpd/wendzelnntpd.conf*). An unprivileged user under Unix-like systems is also not able to create a listen socket on the default NNTP port (119) since all ports up to 1023 are usually[1] reserved. This means that the server should use a port greater than or equal to 1024 if it is started by a non-root user.

In case you use MySQL or Postgres databases with authentication, your *wendzelnntpd.conf* contains a username and password to access the database. Make sure that only the server's user has read (and write) access to the configuration file.

Please also note that WendzelNNTPd can be easily identified due to its welcoming 'banner' (desired code '200' message of NNTP). Tools such as `nmap` provide rules to identify WendzelNNTPd and its version this way. Theoretically, this could be changed by a slight code modification (welcome message, HELP output and other components that make the server identifiable). However, I do not recommend this as it is just a form of 'security by obscurity'.

[1] Some *nix-like systems may have a different range of privileged ports.

5 Development

For development purposes you can start WendzelNNTPd on your host system. If you are using an unsupported operating system or just not run WenzdelNNTPd on your host system, you can use the provided Dockerfiles to run WendzelNNTPd in a Docker container.

5.0.1 Initial setup

When you are on a UNIX-based system (like macOS) you can use the provided make commands:

```
$ make docker-dev-build
$ make docker-dev-run
```

To stop the Docker container, you can use the following command:

```
$ make docker-dev-stop
```

If you are not on a UNIX-based system (like Windows) use the following native docker commands:

```
$ docker build -f ./docker/Dockerfile -t wendzelnntpd-dev:latest .
$ docker run --name wendzelnntpd-dev --rm -it \
    -p 118:118 -p 119:119 -p 563:563 -p 564:564 \
    -v ${PWD}:/wendzelnntpd \
    -v wendzelnntpd_data:/var/spool/news/wendzelnntpd \
    wendzelnntpd-dev:latest
```

To stop the Docker container you can use the following command:

```
$ docker stop wendzelnntpd-dev
```

5.0.2 Test new code

The container is built without code included. The code is automatically mounted as volume into the container. After each change of source code, the application is compiled again.

5.0.3 Edit configure or Makefile

This project uses `autoconf` to generate the *configure* script from *configure.ac*. The files *configure*, *aclocal.m4*, *config.guess*, *config.sub* and *install-sh* are autogenerated. Do not edit them directly, but regenerate them with `autoreconf -fi` after editing *configure.ac* or a macro in the *m4* directory.

The *Makefile* is generated by the *configure* script from the *Makefile.in*.

6 Upgrading

6.1 Upgrade from version 2.1.y to 2.2.x

Please stop WendzelNNTPd and check the *wendzelnntpd.conf*. There is a new configuration style that breaks parts of the previous configuration style (especially due to the introduction of "connectors"). Additionally, the configuration file has been moved into the subdirectory *wendzelnntpd*. So, for example, if the file was previously located at */usr/local/etc/wendzelnntpd.conf*, it must now be moved to */usr/local/etc/wendzelnntpd/wendzelnntpd.conf*.

Existing user passwords will no longer work. You need to recreate all users and their passwords using **wendzelnntpadm**. Additionally, you must reapply any previously configured ACL roles and group memberships for the recreated users.

The behavior of `./configure` has changed. The environment variables for enabling and disabling features are no longer supported and replaced by CLI flags. The support for the databases can now be enabled/disabled. Examples: *--disable-mysql*, *--enable-sqlite*, *--enable-postgres*. The support for TLS can be disabled by *--disable-tls*. You can use `./configure --help` for an overview of the available CLI flags.

6.2 Upgrade from version 2.1.x to 2.1.y

Same as upgrading from v.2.0.x to v.2.0.y, see Section Upgrade from version 2.0.x to 2.0.y.

6.3 Upgrade from version 2.0.x to 2.1.y

Please follow the upgrade instructions for upgrading from 2.0.x to 2.0.y below. However, once you use cryptographic hashes in your *wendzelnntpd.conf*, your previous passwords will not work anymore, i.e., you need to reset all passwords or deactivate the hashing feature.

6.4 Upgrade from version 2.0.x to 2.0.y

Stop WendzelNNTPd if it is currently running. Install WendzelNNTPd as described but run `make upgrade` instead of `make install`. Afterwards, start WendzelNNTPd again.

6.5 Upgrade from version 1.4.x to 2.0.x

Acknowledgement: I would like to thank Ann from Href.com for helping a lot with finding out how to upgrade from 1.4.x to 2.0.x!

An upgrade from version 1.4.x was not foreseen due to the limited available time I have for the development. However, here is a dirty hack:

Preparation Step: You **need to create a backup of your existing installation first**, at least everything from **/var/spool/news/wendzelnntpd** as you will need all these files later. **Perform the following steps on your own risk – it is possible that they do not work on your system as only two WendzelNNTPd installations were tested!**

First Step: Install Wendzelnntpd-2.x on a Linux system (Windows is not supported anymore). This requires some libraries and tools.

Under *Ubuntu* they all come as packages:

```
$ sudo apt-get install libmysqlclient-dev libsqlite3-dev flex bison sqlite3
```

Under *CentOS* they come as packages as well:

```
$ sudo yum install make gcc bison flex sqlite-devel
```

Other operating systems should provide the same or similar packages/ports.

Run `MYSQL=NO ./configure`, followed by `make`, and `sudo make install`. This will compile, build and install WendzelNNTPd without MySQL support as you only rely on SQLite3 from v.1.4.x (and it would be significantly more difficult to bring the SQLite database content to a MySQL database).

Second Step: Please make sure WendzelNNTPd-2 is configured in a way that we can *hopefully* make it work with your own database file. Therefore, in the configuration file **/usr/local/etc/wendzelnntpd.conf** make sure that WendzelNNTPd uses sqlite3 instead of mysql:

```
database-engine sqlite3
```

Third Step: Now comes the tricky part. The install command should have created **/var/spool/news/wendzelnntpd/usenet.db**. However, it is an empty Usenet database file in the new format. Now REPLACE that file with the file you use on your existing WendzelNNTPd installation, which uses the old 1.4.x format. Also copy all of your old **cdp*** files and the old **nextmsgid** file from your Windows system/from your backup directory to **/var/spool/news/wendzelnntpd/**.

The following step is a very dirty hack but I hope it works for you. It is not 100% perfect as important table columns are then still of the type 'STRING' instead of the type 'TEXT'!

Load the sqlite3 tool with your replaced database file:

```
$ sudo sqlite3 /var/spool/news/wendzelnntpd/usenet.db
```

This will open the new file in editing mode. We now add the tables which are not part of v.1.4.x to your existing database file. Therefore enter the following commands:

```
CREATE TABLE roles (role TEXT PRIMARY KEY);
CREATE TABLE users2roles (username TEXT, role TEXT, PRIMARY KEY(username, role));
CREATE TABLE acl_users (username TEXT, ng TEXT, PRIMARY KEY(username, ng));
CREATE TABLE acl_roles (role TEXT, ng TEXT, PRIMARY KEY(role, ng));
.quit
```

Fix Postings You will probably see no post bodies right now if posts are requested by your client. Therefore, switch into **/var/spool/news/wendzelnntpd** and run (as superuser) this command, it will replace the broken trailings with corrected ones:

```
$ cd /var/spool/news/wendzelnntpd
$ for filn in `/bin/ls cdp*`; do echo $filn; cat $filn | \
sed 's/\.\r/.\r\n/' > new;  num=`wc -l new| \
awk '{$minone=$1-1; print $minone}'` ; \
head -n $num new > $filn; done
```

Last Step (Checking whether it works!): First check, whether the database file is accepted at all:

```
$ sudo wendzelnntpadm listgroups
```

It should list all your newsgroups

```
$ sudo wendzelnntpadm listusers
```

It should list all existing users. Accordingly

```
$ sudo wendzelnntpadm listacl
```

should list all access control entries (which will be empty right now but if no error message appears, the related tables are now part of your database file!).

Now start WendzelNNTPd via `sudo wendzelnntpd` and try to connect with a NNTP client to your WendzelNNTPd and then try reading posts, sending new posts and replying to these posts.

If this works, you can now run v2.x on 32bit and 64bit Linux :) Congrats, you made it and chances are you are the second user who did that. Let me know via e-mail if it worked for you.

7 Uninstallation

WendzelNNTPd can be uninstalled with `make uninstall` if you installed it from source. `./configure` needs to be called with the same parameters as during installation. You need superuser access to uninstall WendzelNNTPd. It is recommended to use the same version of the sources of WendzelNNTPd as the one installed. The configuration files, SQLite database, postings and log files are left in place to prevent data loss. You need to remove them manually if desired.

```
$ ./configure
$ sudo make uninstall
```