

NAME

leafnode – NNTP server for small (dialup) sites

SYNOPSIS

leafnode

DESCRIPTION

Leafnode is a USENET package intended for small sites, where there are few users and little disk space, but where a large number of groups is desired.

The design of **leafnode** is intended to self-repair after problems, and to require no manual maintenance.

The **leafnode** program itself is the NNTP server. It is run from **inetd**(8), **xinetd**(8) or **tcpserver** when someone wants to read news. The other parts of the package, **fetchnews** and **texpire**, are responsible for fetching new news from another server, and for deleting old news.

ACCESS CONTROL

No authentication or access control is supported. This is a deliberate omission: Implementing this is a job which should not be redone for each and every service.

It is mandatory that you use external access control mechanisms like tcpd, inetd/xinetd compiled with libwrap support, tcpserver with -x option and the like and that these are in effect. tcpd and libwrap are components of Wietse Venema's fine **tcp_wrappers** package.

As a very rough last line of defense against abuse, leafnode will drop connections from outside your LANs by default. You can configure leafnode to let go of this restriction (look for the allowstrangers option), but don't do that unless tight access control is in place. Someone will abuse your computer sooner or later. Promised.

I recommend that either firewalling or tcpd be used for access control.

FILES

All these files and directories *must* be readable by the user "news". It is recommended that, unless otherwise stated, that the user "news" be the only user in the group "news" and these files belong to "root:news" (user:group) so leafnode cannot modify your configuration or filter files.

/opt/leafnode-1/etc should not be writable by the user "news", but it must be executable for at least any of the group that the user "news" is in. */opt/leafnode-1/etc/config* contains the configuration parameters for **leafnode**. It **must not** be writable by the user "news". Set this to owner root:news and mode 640. For details, see CONFIGURATION below.

/opt/leafnode-1/var/spool/news must also be readable and writable by the user "news". It contains the news articles; e.g. */opt/leafnode-1/var/spool/news/alt/fan/agulbra* contains the articles in the *alt.fan.agulbra* group. Each directory contains articles in numbered files (decimal numbers, monotonically increasing), and a special file called *.overview* which contains the "Subject", "From", "Date", "Message-ID", "References", "Bytes" and "Lines" headers for each article in the group.

Several subdirectories are special:

/opt/leafnode-1/var/spool/news/leafnode contains the files that leafnode creates during operation, for example the **groupinfo** file which contains information about each USENET newsgroup. This file is built by **fetchnews** (8). You can force a complete rebuild of the groupinfo file by calling **fetchnews** with the parameter -f (see **fetchnews** (8)).

/opt/leafnode-1/var/spool/news/out.going contains local postings that **fetchnews**(8) is to pass to the upstream NNTP server. After a posting has been successfully written to disk, its u+r permission flag is set. This flag is interpreted by **fetchnews**(8) as "you may post this article". This prevents fetchnews from posting articles that are still being received from newsreaders. (Note: versions 1.9.23 to 1.9.32 inclusively used u+x instead, which caused some "stuck post" problems with articles in the spool when a prior leafnode

version was updated to one of these 10 versions. Updating to leafnode 1.9.33 or later fixes the problem.)

/opt/leafnode-1/var/spool/news/failed.postings contains local postings that the upstream server rejected. **fetchnews(8)** will create files in this directory, but none of the **leafnode** programs will delete anything in it.

/opt/leafnode-1/var/spool/news/message.id contains hard links to each message; this is used in place of the **dbz** database typically used by bigger servers. (A directory such as this is probably more efficient for the small servers **leafnode** is designed for but scales very badly.)

/opt/leafnode-1/var/spool/news/interesting.groups contains one file for each group an NNTP client has asked to read. **leafnode** will update the **ctime** (ls -l usually shows the **mtime**, try ls -lc) of the relevant file when a LISTGROUP, XOVER, XHDR, STAT, HEAD, BODY or ARTICLE command is issued, when a GROUP or LIST ACTIVE command (the latter only with a single group, not with patterns) is issued for an interesting group (to avoid unsubscribing low-traffic groups that are still read) and **fetchnews(8)** will retrieve all new articles in all groups whose files have been either

- touched during the past two days, or
- touched more than once, and at least once within the past week.

The timeout is configurable through the *config* file variables **timeout_short** and **timeout_long**. See also **fetchnews(8)** for the **-n** option.

/etc/inetd.conf or */etc/xinetd.conf* contains the configuration which starts **leafnode**. It is strongly recommended to start **leafnode** as user news.

ENVIRONMENT

LN_REJECT_POST_PRE

If this variable exists, all POST commands are rejected with a 400 code. Use only for debugging clients.

LN_REJECT_POST_POST

If this variable exists, the POST command is rejected with a 400 code after the article and CRLF.CRLF has been received. Use only for debugging clients.

CONFIGURATION

All configuration is done using the file */opt/leafnode-1/etc/config*, which may include a filter description file, *filterfile* for short, as described below.

For the purposes of this section, whitespace shall be defined as an arbitrary sequence consisting of one or more SPACE or HTAB characters, ASCII positions 32 and 9, respectively.

The configuration file is strictly line-oriented with LF or CRLF as line terminator.

Empty lines and lines consisting of only whitespace, possibly followed by a comment (introduced by a hash mark (#) and extending through the end of the line), are skipped.

All other lines have exactly three mandatory fields, a plain text parameter, an assignment character (=) optionally surrounded by whitespace and a value. The value is either plain text or - new since leafnode v1.11 - a string in double quotes with trivial backslash escape (see below).

Plain text starts at the first non-whitespace character and extends through the last non-whitespace character on the line that is not a comment. A trailing comment on a line is skipped.

Quoted strings are enclosed in double quote characters ("). The backslash character (\) is skipped, but it copies the immediately following character verbatim, so that you can specify the backslash itself by doubling it (\\) or a double quote character as part of the string by preceding it with a backslash (\"); the hash mark has no special meaning as command introducer inside quoted strings. Text after the end of the string is silently ignored (this may change in future versions). Comments after quoted strings are ignored.

MANDATORY PARAMETERS

These parameters must be specified for leafnode to work.

`server = news02.example.com`

"server" is used by **fetchnews** (8) to select what NNTP server(s) to retrieve news from and to post your articles to. You can specify more than one news server; in that case, the servers will be queried from the top down. If you want to post articles, at least one of your servers should allow you to post. In the example above, news02.example.com is the news server.

This parameter can be given more than once. Each **server** starts with a fresh set of default configuration options, no inheritance takes place from the previous server definition. Only options explicitly marked "server-specific" can be set on a per server basis, "general" options are set for all servers at the same time.

`expire = 5`

"expire" is the number of days an article should be kept around. In the example, five days after the article has last been read, it is deleted by **texpire** (8). This value **MUST** be at least 1. This parameter is global, see the introductory paragraph of the following **GENERAL OPTION PARAMETERS** section to find out what this means.

GENERAL OPTIONAL PARAMETERS

These options can only be configured once in the configuration file, and take effect for leafnode as a whole. It does not matter where these are specified relative to `server=` options, but for clarity, you are encouraged to place these before the first `server=` line. Specifying each of the global options more than once lets the last copy take effect, but may cause errors in the future.

`hostname = host.domain.country`

By default, leafnode tries hard to figure the host name of your computer, skipping inadequate (non-unique) names if possible. It will look up your computer's host name with `gethostname(3)` and then try to qualify the name with `gethostbyname(3)` if necessary. Common sources for the full name therefore are `/etc/hosts`, NIS and DNS, but consult your system documentation for details.

If leafnode fails to determine the host name, this is usually a hint that your system is not configured properly, or it has a hostname that is unsuitable for the domain part of a Message-ID, for example, "localhost.localdomain", and you should fix the name service configuration. Adding a unique fully-qualified host name to `/etc/hosts` is usually sufficient. Please see *README-FQDN* for more details.

You can configure the unique fully-qualified host name here as well, but this is not recommended and discouraged.

`create_all_links = 1`

Normally, fetchnews will store articles only in the newsgroups which it considers interesting. With this option set, fetchnews will create hardlinks for all newsgroups in the Newsgroups: header that it knows about. This may be of interest if you want to apply a score- or killfile to the local Xref: line.

`maxfetch = 1000`

"maxfetch" specifies the maximum number of articles **fetchnews** (8) should fetch from the upstream server in each group. Its use is not advised, because if you use it you will not see all the traffic in a group. By default there is no limit.

`initialfetch = 1`

"initialfetch" defines how many articles from a newly subscribed group should be fetched. The default is to fetch all old articles, which can get quite time-consuming when subscribing to a very busy group. This is equivalent to setting initialfetch to zero. If you want to get no old articles when subscribing to a new group, you should set initialfetch to one, as in the example above.

`groupexpire very.crowded.group = 1`

groupexpire very.crowded.hierarchy.* = 1

"groupexpire" makes it possible to adjust expiry times for individual groups. Expiry times are given in days. 0 means "use the default", negative values prevent the expire process for this group altogether (you can consider this an archive mode). This value is used by **texpire** (8). You can specify as many groupexpire lines as you like. It is possible to specify **glob** (7)-like wildcard expressions.

maxage = 10

If an article turns up on your upstream news server which is older than "maxage" days it will not be fetched even if you don't have it yet. This is useful if your upstream server gets occasional "hiccups". The default is set to 10. If you want to switch this feature off, set maxage to some very large value, such as 20000 (this is equivalent to roughly 54 years).

maxold = 10

Is synonymous to maxage, see above.

maxlines = 2000

If you want to avoid receiving very large articles, you may set the "maxlines" parameter to the maximal number of lines an article should have. By default, this feature is switched off.

minlines = 2

Sometimes newsgroups are spammed with empty postings. To reject these postings, you can set the "minlines" parameter. Setting minlines to a value larger 4 is probably not a good idea since you will also start to kill "real" postings then. By default, this feature is switched off.

maxbytes = 100000

If you want to avoid receiving very large articles, instead of using the "maxlines" parameter you can also use the "maxbytes" parameter. By default, this feature is switched off.

maxcrosspost = 5

If you want to combat spam, you can filter out all postings that are posted to more than a certain number of newsgroups. The number is defined by setting "maxcrosspost". Setting this parameter to very low values is probably a bad idea. This feature is switched off by default.

maxgroups = 5

Synonymous for maxcrosspost. See above.

filterfile = /opt/leafnode-1/etc/filters

Leafnode can filter the input headers for arbitrary regular expressions. These are stored in a file designated "filterfile". The format of "filterfile" is very simple: one perl-compatible regular expression per line. If one of the regular expressions fits to a header to be downloaded, the body of that article will be rejected. This feature is switched off by default. The format of the regular expressions is described in **pcre**(3).

timeout_short = 2

By default, a group that has been accidentally touched is being fetched for two days. You can change this time by changing timeout_short.

timeout_long = 7

By default, a group that has not been read at all is being fetched for seven days before being unsubscribed. This interval can be changed by setting timeout_long to a different value.

timeout_active = 90

By default, active files from the upstream servers are re-read every 90 days. This interval can be changed by setting timeout_active to a different value. Be aware that reading an active file transfers about one MB of information if the server that you are using carries a reasonable number of groups (i. e. around 20,000).

timeout_client = 900 (since v1.9.23)

By default, leafnode will drop the connection 900 seconds (15 minutes) after seeing the last command from the client. You can change the timeout here. Setting it too low (like below 5 minutes) will annoy your users and consume more system resources for re-reading all the files.

`timeout_fetchnews = 300` (since v1.9.52)

Fetchnews will, since v1.9.52, assume the upstream server has become wedged after waiting for a reply for 300 seconds. You can change the timeout here.

`timeout_lock = 5` (since v1.9.54)

Configure how many seconds the leafnode programs (`applyfilter`, `checkgroups`, `fetchnews`, `texpire`) will wait for the lock file before aborting. Setting this to 0 means to wait indefinitely. **NOTE:** you can override this by setting the environment variable `LN_LOCK_TIMEOUT` (note it is not `LN_TIMEOUT_LOCK`). The default is 5 seconds.

`delaybody = 1`

With this option set, **fetchnews** (8) fetches only the headers of an article for visual inspection. Only when the headers have been read, the bodies of the articles will be retrieved the next time **fetchnews** (8) is called. This can save a huge amount of download time and disk space.

`delaybody_in_situ = 1` (since v1.9.41)

This is only applicable with `delaybody=1`.

By default, leafnode will give the full downloaded article a new article number so they appear as new in your newsreader. This does not work for all newsreaders. With this option set, leafnode will retain the original article number. You'll have to figure out how to tell your newsreader to show old articles. This option defaults to 0. It is **highly** recommended to leave it unset.

`debugmode = 1`

With this option set, **fetchnews** (8), **texpire** (8) and **leafnode** (8) will start to log lots of debugging output via **syslog** (8) at facility news and priority debug. Use it for tracking down problems with your feed. `debugmode` should be left at 0 for regular use because it can log enormous amounts of data. The higher the number, the more will be logged. Choosing a figure greater than 3 will not make a difference at the moment.

`allow_8bit_headers = 1` (since v1.9.25)

By default, leafnode rejects local posts that have 8-bit characters in their headers, because they violate relevant standards, particularly RFC-2822 (which RFC-1036 is based on) that demands that Usenet news headers (as mail headers) must be pure 7-bit US-ASCII, with only whitespace allowed from the control characters.

However, as UTF-8 is to come, and some national hierarchies, particularly the Norwegian and Danish (no.*, dk.*) seem to have agreed on preferring just-send-eight over RFC-2047, you can set this option to allow 8-bit data in headers. Leafnode will however add a warning header if 8-bit data is present, stating that the site administrator allowed this.

There is no way to make leafnode accept non-whitespace control characters in headers.

`allowSTRANGERS = MAGIC` (since v1.9.23)

By default, leafnode refuses connections from outside your LANs. Check `config.example` for how to use this parameter to let strangers connect to your leafnode. Instead of `MAGIC`, you have to write a number as mentioned in `config.example`. Note that capitalization matters.

`linebuffer = 1`

By default, `stdout` and sometimes `stderr` of applications are set to "fully buffered" unless connected to terminals. Use this option to explicitly request line buffered mode for `stdout` and `stderr`.

`clamp_maxage = 0`

By default, leafnode will derive a "maxage" argument from the expire time that is applicable to the group (`groupexpire` if set, `expire` otherwise), to prevent fetching articles again that were once there and then cleared by **texpire**(8). Set `clamp_maxage=0` to get rid of this behaviour.

`article_despite_filter = 1` (since v1.9.33)

By default, fetchnews will request HEAD and BODY separately if a filter file is defined and delay-body is off. For high latency, high throughput links (such as interleaved DSL or satellite links), it may be faster to request head and body together with an ARTICLE command and ignore the body if the filters apply (though it may not be cheaper if you pay per MByte), enabling this option will force leafnode to use the ARTICLE command in spite of filters being defined. (Note that in delay-body mode, HEAD and BODY will ALWAYS be requested separately.)

`newsadmin = news@leafnode.example.org` (since v1.9.47)

This option sets the From: address for the placeholder article, it should be the news administrator's mail address. It defaults to news@HOSTNAME, where HOSTNAME is leafnode's hostname.

SERVER-SPECIFIC OPTIONAL PARAMETERS

These options can only be placed after the `server=` line of the server to which you would like these to apply, and they always pertain to the preceding `server=` line. Specifying them before the first `server=` line is an error.

`username = myname`

If any of your news servers requires authentication, you can enter your username on that server here. This field may occur multiple times, once after each server definition. See the introduction of this CONFIGURATION section for information on how to quote *myname*.

`password = mypassword`

If any of your news servers requires authentication, you can enter your password on that server here. This field may occur multiple times, once after each server definition. Since the password is available in clear text, it is recommended that you set the rights on the config file as restrictive as possible, otherwise other users of your computer will be able to get your password(s) from that file. See the introduction of this CONFIGURATION section for information on how to quote *mypassword*.

`port = 8000`

By default, fetchnews tries to connect to its upstream news servers on the NNTP port (119). If your servers run on a different port, you can specify those here. This field may occur multiple times, once after each server definition.

Note: to modify the port your own leafnode servers listens on, change the `inetd.conf`, `xinetd.conf` configuration file or the `tcpsvd/tcpserver` command line. leafnode does not set up its listen port itself.

`timeout = 30`

By default, leafnode tries to connect for 10 seconds to a server and then gives up. If you have a slow server, you can try for a longer time by setting the timeout higher (in this example, 30 seconds). The timeout can be tuned individually for each server.

`noactive = ANYTHING` (v1.9.25 ... v1.11.4)

`noactive = 1` (since v1.11.5)

If this parameter is set, the active file is never downloaded from this server. Use this for very slow servers unless they carry groups that other servers don't offer. Leafnode versions 1.9.25 to 1.11.4 would always assume that "ANYTHING" had been 1. "noactive = 0" is supported since v1.11.5.

`nodesc = ANYTHING` (until v1.11.4)

`nodesc = 1` (since v1.11.5)

Some servers do not deliver news groups descriptions correctly because they cannot parse the XGTITLE and LIST NEWSGROUPS commands. In that case, put this line after the "server" line. Leafnode versions up to v1.11.4 would always assume that "ANYTHING" had been 1. "nodesc = 0" is supported since v1.11.5.

nopost = 1 (since v1.9.23)

Prevent posting to this server. You can use this if the upstream won't let you post but still greet leafnode with 200 or if the upstream doesn't forward your postings reliably.

noread = 1 (since v1.9.33)

Prevent fetching news articles or active files from this server. You can use this if the upstream is good to post, but too slow to fetch news from.

noxovert = 1 (since v1.9.47)

Prevent the use of XOVER on the current server. Fetchnews will use XHDR instead.

only_groups_match_all = 1 (since v1.9.52)

Usually, when cross-posting an article, fetchnews will post the article if ANY group listed in the Newsgroups: header is matched by the PCRE. With this option on, ALL groups listed in the Newsgroups: header must match. This can be used to avoid "poison" groups when you have multiple upstream servers.

only_groups_pcre = PCRE (since v1.9.28)

This parameter lists the Perl-compatible regular expression of groups that are fetched or posted to this server. The PCRE is automatically anchored at the left hand side, so you can omit the leading `^`. Remember to escape dots, as in:

```
de\.comp\.|de\.comm\.
```

If this parameter is omitted, all groups are fetched from and posted to this server.

Note: you must run fetchnews with the `-f` option after changing, adding or removing any `only_groups_pcre` option.

Hint: you can use something like this to check your `only_groups_pcre` settings:

```
cut -f1 -d" " @spooldir@/leaf.node/groupinfo \
| pcregrep 'PATTERN'
```

post_anygroup = 1 (since v1.9.37)

This parameter makes leafnode post on this server without checking if it carries the group an article is posted to. The default is `post_anygroup = 0`, which means that leafnode will check with a "GROUP" command if the server carries the group the articles is posted into. Use this on post-only servers that don't allow the "GROUP" command. Note: inconsiderate use of this parameter may cause articles to end up in the failed.postings directory.

OBsolete PARAMETERS

supplement

is synonymous to server. Don't use it on new installations.

fqdn

is synonymous to hostname. Don't use it on new installations.

PROTOCOL

Here are the NNTP commands supported by this server:

ARTICLE, BODY, DATE, GROUP, HDR, HEAD, HELP, LAST, LIST, LIST ACTIVE, LIST ACTIVE.TIMES, LIST EXTENSIONS, LIST NEWSGROUPS, LIST OVERVIEW.FMT, LISTGROUP, MODE, NEWGROUPS, NEXT, POST, OVER, SLAVE, STAT, XHDR, XOVER. These commands follow RFC-977 and RFC-2980, except HDR and OVER which are recognized in anticipation of current NNTP drafts.

Note that the syntax of HDR and OVER may change.

BUGS

Leafnode is totally unaware of UTF-8 and will reject a client that posts UTF-8 characters in the header. Current Usefor drafts claim all article headers UTF-8 encoded Unicode. Leafnode still expects RFC-2047 instead which may eventually be phased out in favour of UTF-8.

Leafnode stops reading a line at the first NUL character.

Leafnode may not cope well with crosspostings that cross hierarchies if you have multiple upstream feeds and use the `only_groups_pcre` configuration option.

Leafnode will only bother to determine the time zone offset for generated Date: headers for posts that lack them on systems that offer the `tm_gmtoff` member in struct `tm` (commonly BSD and GNU systems).

AUTHOR

Written by Arnt Gulbrandsen <agulbra@troll.no> and copyright 1995 Troll Tech AS, Postboks 6133 Etterstad, 0602 Oslo, Norway, fax +47 22646949.

Modified by Cornelius Krasel <krasel@wpxx02.toxi.uni-wuerzburg.de>, Randolph Skerka <Randolf.Skerka@gmx.de> and Markus Enzenberger <enz@cip.physik.uni-muenchen.de>. Copyright of the modifications 1997-1999.

Modified by Matthias Andree <matthias.andree@gmx.de>, Copyright 1999 - 2002. Modified by Ralf Wildenhues <ralf.wildenhues@gmx.de>, Copyright 2002.

Jonathan Larmour <jifl@jifvik.org> contributed the `timeout_client` feature.

Andreas Meininger <a.meininger@gmx.net> contributed the code to let `texpire` ignore `groupexpire = -1` groups.

Mark Brown <broonie@debian.org> added the `noactive` option.

Numerous contributions by other people.

The initial development of **leafnode** has been paid for by Uninett AS (<http://www.uninett.no/>).

SEE ALSO

applyfilter(8), **checkgroups(8)**, **fetchnews(8)**, **newsq(1)**, **texpire(8)**.

tcpd(8), **hosts_access(5)**, **glob(7)**, **pcre(3)**, **RFC 977**, **RFC 2980**.