                        Common NNTP Extensions

Status of this Memo

Copyright Notice

Abstract

   In this document, a number of popular extensions to the Network News
   Transfer Protocol (NNTP) protocol defined in RFC 977 are documented
   and discussed. While this document is not intended to serve as a
   standard of any kind, it will hopefully serve as a reference document
   for future implementers of the NNTP protocol. In the role, this
   document would hopefully create the possibility for some level of
   interoperability among implementations that make use of extensions.

Introduction

   RFC 977 [1] defines the NNTP protocol and was released over a decade
   ago. Since then, NNTP has become one of the most popular protocols
   in use on the Internet. Many implementations of the protocol have
   been created on many different platforms and operating systems. With
   the growth in use of the protocol, work began on a revision to NNTP
   in 1991, but that work did not result in a new standard protocol
   specification. However, many ideas from that working group did find
   their way into many implementations of NNTP. Additionally, many
   other extensions, often created by newsreader authors, are also in
   use. This document will capture and define all known extensions to
   NNTP available in official NNTP server releases of some type as of
   this writing. Where possible, the server software first implementing
   a particular extension will be noted. It is the hope of the author
   that using this document in tandem with RFC 977 will limit the
   addition of new extensions that essentially do the same thing.
   Software developers may wish to use this document and others [2] as a
   resource for the development of new software.

This document does not specify an Internet Standard of any kind. It
only attempts to document current practices. While this document may
clarify some ambiguity in RFC 977, RFC 977 should be regarded as
authoritative in all cases. There are some implementations that are
not strictly RFC 977 compliant and where necessary, these deviations
from the standard will be noted. This document does reflect the work
of the IETF NNTP-EXT working group chaired by Ned Freed and Stan
Barber.

This document is provided to help implementers have a uniform source
of information about extensions, however, it is important for any
prospective implementer to understand that the extensions listed here
are NOT part of any current standard for NNTP. In fact, some of the
ones listed in this document should not be included in new NNTP
implementations as they should no longer be used modern NNTP
environments. Such commands should be considered historic and are
documented as such in this document.

Extensions fall into three categories: transport, newsreader and
other. Transport extensions are additions to the NNTP specification
that were made specifically to move news articles from one server to
another server. Newsreader extensions are additions to the NNTP
specification that were made to assist NNTP clients in selecting and
retrieving news articles from servers. Other extensions to the NNTP
specification are those which did not specifically fall into either
of the other two categories. Examples of other extensions include
authentication and time-of-day extensions. For each command, the
format of section 3 of RFC 977 will be used.

1. Transport Extensions

A transport extension is one which is primarily used in inter-server
communications. Following are the descriptions of each transport
extension commands and the responses which will be returned by those
commands.

Each command is shown in upper case for clarity, although case is
ignored in the interpretation of commands by the NNTP server. Any
parameters are shown in lower case. A parameter shown in [square
brackets] is optional. For example, [GMT] indicates that the
triglyph GMT may present or omitted. A parameter that may be
repeated is followed by an ellipsis.

1.1.1 The CHECK command

   CHECK <message-id>

   CHECK is used by a peer to discover if the article with the specified
   message-id should be sent to the server using the TAKETHIS command.
   The peer does not have to wait for a response from the server before
   sending the next command.

   From using the responses to the sequence of CHECK commands, a list of
   articles to be sent can be constructed for subsequent use by the
   TAKETHIS command.

   The use of the CHECK command for streaming is optional. Some
   implementations will directly use the TAKETHIS command and send all
   articles in the send queue on that peer for the server.

   On some implementations, the use of the CHECK command is not
   permitted when the server is in slave mode (via the SLAVE command).

   Responses that are of the form X3X must specify the message-id in the
   response.

1.1.2. Responses

       238 no such article found, please send it to me
       400 not accepting articles
       431 try sending it again later
       438 already have it, please don't send it to me
       480 Transfer permission denied
       500 Command not understood

1.2.1 The MODE STREAM command

   MODE STREAM

   MODE STREAM is used by a peer to indicate to the server that it would
   like to suspend the lock step conversational nature of NNTP and send
   commands in streams. This command should be used before TAKETHIS and
   CHECK. See the section on the commands TAKETHIS and CHECK for more
   details.

1.2.2. Responses

       203 Streaming is OK
       500 Command not understood

1.3.1 The TAKETHIS command

    TAKETHIS <message-id>

    TAKETHIS is used to send articles to a server when in streaming mode.
    The entire article (header and body, in that sequence) is sent
    immediately after the peer sends the TAKETHIS command. The peer does
    not have to wait for a response from the server before sending the
    next command and the associated article.

    During transmission of the article, the peer should send the entire
    article, including header and body, in the manner specified for text
    transmission from the server. See RFC 977, Section 2.4.1 for
    details.

    Responses that are of the form X3X must specify the message-id in the
    response.

1.3.2. Responses

        239 article transferred ok
        400 not accepting articles
        439 article transfer failed
        480 Transfer permission denied
        500 Command not understood

1.4.1 The XREPLIC command

    XREPLIC ggg:nnn[,ggg:nnn...]

    The XREPLIC command makes is possible to exactly duplicate the news
    spool structure of one server in another server. It first appeared
    in INN.

    This command works similarly to the IHAVE command as specified in RFC
    977. The same response codes are used. The command line arguments
    consist of entries separated by a single comma. Each entry consists
    of a news group name, a colon, and an article number. If the server
    responds with a 335 response, the article should be filed in the news
    group(s) and article number(s) specified in the XREPLIC command line.
    If the server cannot do successfully install the article once it has
    accepted it, a 436 or 437 response code can be used to indicate the
    failure.

    This command should only be used when the receiving server is being
    fed by only one other server. It is likely that when used with
    servers that have multiple feeds that this command will frequently
    fail.

XREPLIC slaving has been deprecated in INN version 1.7.2 and later.
INN now has the ability to slave servers via transparent means,
simply by having the article's Xref header transferred. (In previous
versions, this header was generated locally and stripped off on
outgoing feeds.)

It is likely that future versions of INN will no longer support
XREPLIC.

1.4.2. Responses

```
235 article transferred ok
335 send article to be transferred.  End with <CR-LF>.<CR-LF>
435 article not wanted - do not send it
436 transfer failed - try again later
437 article rejected - do not try again
```

2. Newsreader Extensions

Newsreader extensions are those which are primarily used by
newsreading clients. Following are the descriptions of each
newsreader extension commands and the responses which will be
returned by those commands.

Each command is shown in upper case for clarity, although case is
ignored in the interpretation of commands by the NNTP server. Any
parameters are shown in lower case. A parameter shown in [square
brackets] is optional. For example, [GMT] indicates that the
triglyph GMT may present or omitted. A parameter that may be
repeated is followed by an ellipsis. Mutually exclusive parameters
are separated by a vertical bar (|) character. For example,
ggg|<message-id> indicates that a group name or a <message-id> may
be specified, but not both. Some parameters, notably <message-id>,
is case specific. See RFC 1036 for these details.

Also, certain commands make use of a pattern for selection of
multiple news groups. The pattern in all cases is based on the
wildmat [4] format introduced by Rich Salz in 1986. Arguments
expected to be in wildmat format will be represented by the string
wildmat. This format is discussed in detail in section 3.3 of this
document.

2.1.1 Extensions to the LIST command

The original LIST command took no arguments in RFC 977 and returned
the contents of the active file in a specific format. Since the
original newsreaders made use of other information available in the
news transport software in addition to the active file, extensions to

the LIST command were created to make that information available to
NNTP newsreaders. There may be other extensions to the LIST command
that simply return the contents of a file. This approach is
suggested over the addition of over verbs. For example, LIST MOTD
could be used instead of adding XMOTD.

2.1.2 LIST ACTIVE

   LIST ACTIVE [wildmat]

   LIST ACTIVE is exactly the same as the LIST command specified in RFC
   977. The responses and the format should exactly match the LIST
   command without arguments. If the optional matching parameter is
   specified, the list is limited to only the groups that match the
   pattern. Specifying a single group is usually very efficient for the
   server, and multiple groups may be specified by using wildmat
   patterns (described later in this document), not regular expressions.
   If nothing is matched an empty list is returned, not an error. This
   command first appeared in the UNIX reference version.

2.1.3 LIST ACTIVE.TIMES

   LIST ACTIVE.TIMES

   The active.times file is maintained by some news transports systems
   to contain information about the when and who created a particular
   news group. The format of this file generally include three fields.
   The first field is the name of the news group. The second is the
   time when this group was created on this news server measured in
   seconds since January 1, 1970. The third is the email address of the
   entity that created the news group. When executed, the information
   is displayed following the 215 response. When display is completed,
   the server will send a period on a line by itself. If the
   information is not available, the server will return the 503 error
   response. This command first appeared in the UNIX reference version.

2.1.3.1 Responses

      215 information follows
      503 program error, function not performed

2.1.4 LIST DISTRIBUTIONS

   LIST DISTRIBUTIONS

   The distributions file is maintained by some news transport systems
   to contain information about valid values for the Distribution: line
   in a news article header and about what the values mean. Each line

contains two fields, the value and a short explanation on the meaning
of the value. When executed, the information is displayed following
the 215 response. When display is completed, the server will send a
period on a line by itself. If the information is not available, the
server will return the 503 error response. This command first
appeared in the UNIX reference version.

2.1.4.1 Responses

     215 information follows
     503 program error, function not performed

2.1.5 LIST DISTRIB.PATS

   LIST DISTRIB.PATS

   The distrib.pats file is maintained by some news transport systems to
   contain default values for the Distribution: line in a news article
   header when posting to particular news groups. This information
   could be used to provide a default value for the Distribution: line
   in the header when posting an article. The information returned
   involves three fields separated by colons. The first column is a
   weight. The second is a group name or a pattern that can be used to
   match a group name in the wildmat format. The third is the value of
   the Distribution: line that should be used when the group name
   matches and the weight value is the highest. All this processing is
   done by the news posting client and not by the server itself. The
   server just provides this information to the client for it to use or
   ignore as it chooses. When executed, the information is displayed
   following the 215 response. When display is completed, the server
   will send a period on a line by itself. If the information is not
   available, the server will return the 503 error response. This
   command first appeared in INN.

2.1.5.1 Responses

     215 information follows
     503 program error, function not performed

2.1.6 LIST NEWSGROUPS

   LIST NEWSGROUPS [wildmat]

   The newsgroups file is maintained by some news transport systems to
   contain the name of each news group which is active on the server and
   a short description about the purpose of each news group. Each line
   in the file contains two fields, the news group name and a short
   explanation of the purpose of that news group. When executed, the

information is displayed following the 215 response. When display is
completed, the server will send a period on a line by itself. If the
information is not available, the server will return the 503
response. If the optional matching parameter is specified, the list
is limited to only the groups that match the pattern (no matching is
done on the group descriptions). Specifying a single group is
usually very efficient for the server, and multiple groups may be
specified by using wildmat patterns (similar to file globbing), not
regular expressions. If nothing is matched an empty list is
returned, not an error.

When the optional parameter is specified, this command is equivalent
to the XGTITLE command, though the response code are different.

        215 information follows
        503 program error, function not performed

2.1.7 LIST OVERVIEW.FMT

        LIST OVERVIEW.FMT

The overview.fmt file is maintained by some news transport systems to
contain the order in which header information is stored in the
overview databases for each news group. When executed, news article
header fields are displayed one line at a time in the order in which
they are stored in the overview database [5] following the 215
response. When display is completed, the server will send a period
on a line by itself. If the information is not available, the server
will return the 503 response.

Please note that if the header has the word "full" (without quotes)
after the colon, the header's name is prepended to its field in the
output returned by the server.

Many newsreaders work better if Xref: is one of the optional fields.

It is STRONGLY recommended that this command be implemented in any
server that implements the XOVER command. See section 2.8 for more
details about the XOVER command.

2.1.7.1 Responses

        215 information follows
        503 program error, function not performed

2.1.8 LIST SUBSCRIPTIONS

   LIST SUBSCRIPTIONS

   This command is used to get a default subscription list for new users
   of this server. The order of groups is significant.

   When this list is available, it is preceded by the 215 response and
   followed by a period on a line by itself. When this list is not
   available, the server returns a 503 response code.

2.1.8.1 Responses

      215 information follows
      503 program error, function not performed

2.2 LISTGROUP

   LISTGROUP [ggg]

   The LISTGROUP command is used to get a listing of all the article
   numbers in a particular news group.

   The optional parameter ggg is the name of the news group to be
   selected (e.g. "news.software.b"). A list of valid news groups may
   be obtained from the LIST command. If no group is specified, the
   current group is used as the default argument.

   The successful selection response will be a list of the article
   numbers in the group followed by a period on a line by itself.

   When a valid group is selected by means of this command, the
   internally maintained "current article pointer" is set to the first
   article in the group. If an invalid group is specified, the
   previously selected group and article remain selected. If an empty
   news group is selected, the "current article pointer" is in an
   indeterminate state and should not be used.

   Note that the name of the news group is not case-dependent. It must
   otherwise match a news group obtained from the LIST command or an
   error will result.

2.2.1 Responses

      211 list of article numbers follow
      412 Not currently in newsgroup
      502 no permission

2.3 MODE READER

   MODE READER is used by the client to indicate to the server that it
   is a news reading client. Some implementations make use of this
   information to reconfigure themselves for better performance in
   responding to news reader commands. This command can be contrasted
   with the SLAVE command in RFC 977, which was not widely implemented.
   MODE READER was first available in INN.

2.3.1 Responses

      200 Hello, you can post
      201 Hello, you can't post

2.4 XGTITLE

   XGTITLE [wildmat]

   The XGTITLE command is used to retrieve news group descriptions for
   specific news groups.

   This extension first appeared in ANU-NEWS, an NNTP implementation for
   DEC's VMS. The optional parameter is a pattern in wildmat format.
   When executed, a 282 response is given followed by lines that have
   two fields, the news group name (which matches the pattern in the
   argument) and a short explanation of the purpose of the news group.
   When no argument is specified, the default argument is the current
   group name. When display is completed, the server sends a period on
   a line by itself.

   Please note that this command and the LIST NEWSGROUP command provide
   the same functionality with different response codes.

   Since this command provides the same functionality as LIST NEWSGROUP
   it is suggested that this extension be deprecated and no longer be
   used in newsreading clients.

   Note that there is a conflict in one of the response codes from
   XGTITLE and some of the authentication extensions.

2.5.1 Responses

      481 Groups and descriptions unavailable
      282 list of groups and descriptions follows

2.6 XHDR

   XHDR header [range|<message-id>]

   The XHDR command is used to retrieve specific headers from specific
   articles.

   The required parameter is the name of a header line (e.g. "subject")
   in a news group article. See RFC 1036 for a list of valid header
   lines. The optional range argument may be any of the following:

                an article number
                an article number followed by a dash to indicate
                   all following
                an article number followed by a dash followed by
                   another article number

   The optional message-id argument indicates a specific article. The
   range and message-id arguments are mutually exclusive. If no
   argument is specified, then information from the current article is
   displayed. Successful responses start with a 221 response followed
   by a the matched headers from all matched messages. Each line
   containing matched headers returned by the server has an article
   number (or message ID, if a message ID was specified in the command),
   then one or more spaces, then the value of the requested header in
   that article. Once the output is complete, a period is sent on a
   line by itself. If the optional argument is a message-id and no such
   article exists, the 430 error response is returned. If a range is
   specified, a news group must have been selected earlier, else a 412
   error response is returned. If no articles are in the range
   specified, a 420 error response is returned by the server. A 502
   response will be returned if the client only has permission to
   transfer articles.

   Some implementations will return "(none)" followed by a period on a
   line by itself if no headers match in any of the articles searched.
   Others return the 221 response code followed by a period on a line by
   itself.

   The XHDR command has been available in the UNIX reference
   implementation from its first release. However, until now, it has
   been documented only in the source for the server.

2.6.1 Responses

     221 Header follows
     412 No news group current selected
     420 No current article selected
     430 no such article
     502 no permission

2.7 XINDEX

   XINDEX ggg

   The XINDEX command is used to retrieve an index file in the format of
   originally created for use by the TIN [6] news reader.

   The required parameter ggg is the name of the news group to be
   selected (e.g. "news.software.b"). A list of valid news groups may
   be obtained from the LIST command.

   The successful selection response will return index file in the
   format used by the TIN news reader followed by a period on a line by
   itself.

   When a valid group is selected by means of this command, the
   internally maintained "current article pointer" is set to the first
   article in the group. If an invalid group is specified, the
   previously selected group and article remain selected. If an empty
   news group is selected, the "current article pointer" is in an
   indeterminate state and should not be used.

   Note that the name of the news group is not case-dependent. It must
   otherwise match a news group obtained from the LIST command or an
   error will result.

   The format of the tin-style index file is discussed in the
   documentation for the TIN newsreader. Since more recent versions of
   TIN support the news overview (NOV) format, it is recommended that
   this extension become historic and no longer be used in current
   servers or future implementations.

2.7.1 Responses

     218 tin-style index follows
     418 no tin-style index is available for this news group

2.8 XOVER

   XOVER [range]

   The XOVER command returns information from the overview database for
   the article(s) specified. This command was originally suggested as
   part of the OVERVIEW work described in "The Design of a Common
   Newsgroup Overview Database for Newsreaders" by Geoff Collyer. This
   document is distributed in the Cnews distribution. The optional
   range argument may be any of the following:

                  an article number
                  an article number followed by a dash to indicate
                     all following
                  an article number followed by a dash followed by
                     another article number

   If no argument is specified, then information from the current
   article is displayed. Successful responses start with a 224 response
   followed by the overview information for all matched messages. Once
   the output is complete, a period is sent on a line by itself. If no
   argument is specified, the information for the current article is
   returned. A news group must have been selected earlier, else a 412
   error response is returned. If no articles are in the range
   specified, a 420 error response is returned by the server. A 502
   response will be returned if the client only has permission to
   transfer articles.

   Each line of output will be formatted with the article number,
   followed by each of the headers in the overview database or the
   article itself (when the data is not available in the overview
   database) for that article separated by a tab character. The
   sequence of fields must be in this order: subject, author, date,
   message-id, references, byte count, and line count. Other optional
   fields may follow line count. Other optional fields may follow line
   count. These fields are specified by examining the response to the
   LIST OVERVIEW.FMT command. Where no data exists, a null field must
   be provided (i.e. the output will have two tab characters adjacent to
   each other). Servers should not output fields for articles that have
   been removed since the XOVER database was created.

   The LIST OVERVIEW.FMT command should be implemented if XOVER is
   implemented. A client can use LIST OVERVIEW.FMT to determine what
   optional fields and in which order all fields will be supplied by
   the XOVER command. See Section 2.1.7 for more details about the LIST
   OVERVIEW.FMT command.

Note that any tab and end-of-line characters in any header data that
is returned will be converted to a space character.

2.8.1 Responses

    224 Overview information follows
    412 No news group current selected
    420 No article(s) selected
    502 no permission

2.9 XPAT

    XPAT header range|<message-id> pat [pat...]

    The XPAT command is used to retrieve specific headers from specific
    articles, based on pattern matching on the contents of the header.
    This command was first available in INN.

    The required header parameter is the name of a header line (e.g.
    "subject") in a news group article. See RFC 1036 for a list of valid
    header lines. The required range argument may be any of the
    following:

                    an article number
                    an article number followed by a dash to indicate
                        all following
                    an article number followed by a dash followed by
                        another article number

    The required message-id argument indicates a specific article. The
    range and message-id arguments are mutually exclusive. At least one
    pattern in wildmat must be specified as well. If there are
    additional arguments the are joined together separated by a single
    space to form one complete pattern. Successful responses start with
    a 221 response followed by a the headers from all messages in which
    the pattern matched the contents of the specified header line. This
    includes an empty list. Once the output is complete, a period is
    sent on a line by itself. If the optional argument is a message-id
    and no such article exists, the 430 error response is returned. A
    502 response will be returned if the client only has permission to
    transfer articles.

2.9.1 Responses

    221 Header follows
    430 no such article
    502 no permission

2.10 The XPATH command

   XPATH <message-id>

   The XPATH command is used to determine the filenames in which an
   article is filed. It first appeared in INN.

   The required parameter message-id is the message id of an article as
   shown in that article's message-id header. According to RFC 1036
   [3], all message ids for all articles within the netnews environment
   are unique, but articles may be crossposted to multiple groups. The
   response to an XPATH command will include a listing of all filenames
   in which an article is stored separated by spaces or a response
   indicating that no article with the specified message-id exists. The
   returned data is only useful if the news client knows the
   implementation details of the server. Because of this, it is
   recommended that client avoid using this command.

2.10.1 Responses

      223 path1[path2 ...]
      430 no such article on server

2.11 The XROVER command

   XROVER [range]

   The XROVER command returns reference information from the overview
   database for the article(s) specified. This command first appeared
   in the Unix reference implementation. The optional range argument
   may be any of the following:

                an article number
                an article number followed by a dash to indicate
                    all following
                an article number followed by a dash followed by
                    another article number

   Successful responses start with a 224 response followed by the
   contents of reference information for all matched messages. Once the
   output is complete, a period is sent on a line by itself. If no
   argument is specified, the information for the current article is
   returned. A news group must have been selected earlier, else a 412
   error response is returned. If no articles are in the range
   specified, a 420 error response is returned by the server. A 502
   response will be returned if the client only has permission to
   transfer articles.

The output will be formatted with the article number, followed by the
contents of the References: line for that article, but does not
contain the field name itself.

This command provides the same basic functionality as using the XHDR
command and "references" as the header argument.

2.11.1 Responses

      224 Overview information follows
      412 No news group current selected
      420 No article(s) selected
      502 no permission

2.12 XTHREAD

   XTHREAD [DBINIT|THREAD]

   The XTHREAD command is used to retrieve threading information
   in format of originally created for use by the TRN [6] news
   reader.

   The command XTHREAD DBINIT may be issued prior to entering
   any groups to see if a thread database exists. If it does,
   the database's byte order and version number are returned
   as binary data.

   If no parameter is given, XTHREAD THREAD is assumed.

   To use XTHREAD THREAD, a news group must have been selected
   earlier, else a 412 error response is returned.

   A 502 response will be returned if the client only has
   permission to transfer articles. A 503 response is returned
   if the threading files are not available.

   The format of the trn-style thread format is discussed in
   the documentation for the TRN newsreader. Since more recent
   versions of TRN support the news overview (NOV) format, it
   is recommended that this extension become historic and no
   longer be used in current servers or future implementations.

2.12.1 Responses

      288 Binary data to follow
      412 No newsgroup current selected
      502 No permission
      503 program error, function not performed

3. Other Extensions

3.1 AUTHINFO

   AUTHINFO is used to inform a server about the identity of a user of
   the server. In all cases, clients must provide this information when
   requested by the server. Servers are not required to accept
   authentication information that is volunteered by the client.
   Clients must accommodate servers that reject any authentication
   information volunteered by the client.

   There are three forms of AUTHINFO in use. The original version, an
   NNTP v2 revision called AUTHINFO SIMPLE and a more recent version
   which is called AUTHINFO GENERIC.

3.1.1 Original AUTHINFO

   AUTHINFO USER username
   AUTHINFO PASS password

   The original AUTHINFO is used to identify a specific entity to the
   server using a simple username/password combination. It first
   appeared in the UNIX reference implementation.

   When authorization is required, the server will send a 480 response
   requesting authorization from the client. The client must enter
   AUTHINFO USER followed by the username. Once sent, the server will
   cache the username and may send a 381 response requesting the
   password associated with that username. Should the server request a
   password using the 381 response, the client must enter AUTHINFO PASS
   followed by a password and the server will then check the
   authentication database to see if the username/password combination
   is valid. If the combination is valid or if no password is required,
   the server will return a 281 response. The client should then retry
   the original command to which the server responded with the 480
   response. The command should then be processed by the server
   normally. If the combination is not valid, the server will return a
   502 response.

   Clients must provide authentication when requested by the server. It
   is possible that some implementations will accept authentication
   information at the beginning of a session, but this was not the
   original intent of the specification. If a client attempts to
   reauthenticate, the server may return 482 response indicating that
   the new authentication data is rejected by the server. The 482 code
   will also be returned when the AUTHINFO commands are not entered in
   the correct sequence (like two AUTHINFO USERs in a row, or AUTHINFO
   PASS preceding AUTHINFO USER).

All information is passed in cleartext.

When authentication succeeds, the server will create an email address
for the client from the user name supplied in the AUTHINFO USER
command and the hostname generated by a reverse lookup on the IP
address of the client. If the reverse lookup fails, the IP address,
represented in dotted-quad format, will be used. Once authenticated,
the server shall generate a Sender: line using the email address
provided by authentication if it does not match the client-supplied
From: line. Additionally, the server should log the event, including
the email address. This will provide a means by which subsequent
statistics generation can associate newsgroup references with unique
entities - not necessarily by name.

3.1.1.1 Responses

        281 Authentication accepted
        381 More authentication information required
        480 Authentication required
        482 Authentication rejected
        502 No permission

3.1.2 AUTHINFO SIMPLE

    AUTHINFO SIMPLE
    user password

    This version of AUTHINFO was part of a proposed NNTP V2
    specification, which was started in 1991 but never completed, and is
    implemented in some servers and clients. It is a refinement of the
    original AUTHINFO and provides the same basic functionality, but the
    sequence of commands is much simpler.

    When authorization is required, the server sends a 450 response
    requesting authorization from the client. The client must enter
    AUTHINFO SIMPLE. If the server will accept this form of
    authentication, the server responds with a 350 response. The client
    must then send the username followed by one or more space characters
    followed by the password. If accepted, the server returns a 250
    response and the client should then retry the original command to
    which the server responded with the 450 response. The command should
    then be processed by the server normally. If the combination is not
    valid, the server will return a 452 response.

    Note that the response codes used here were part of the proposed NNTP
    V2 specification and are violations of RFC 977. It is recommended
    that this command not be implemented, but use either or both of the
    other forms of AUTHINFO if such functionality if required.

3.1.2.1 Responses

        250 Authorization accepted
        350 Continue with authorization sequence
        450 Authorization required for this command
        452 Authorization rejected

3.1.3 AUTHINFO GENERIC

    AUTHINFO GENERIC authenticator arguments...

    AUTHINFO GENERIC is used to identify a specific entity to the server
    using arbitrary authentication or identification protocols. The
    desired protocol is indicated by the authenticator parameter, and any
    number of parameters can be passed to the authenticator.

    When authorization is required, the server will send a 480 response
    requesting authorization from the client. The client should enter
    AUTHINFO GENERIC followed by the authenticator name, and the
    arguments if any. The authenticator and arguments must not contain
    the sequence "..".

    The server will attempt to engage the server end authenticator,
    similarly, the client should engage the client end authenticator.
    The server end authenticator will then initiate authentication using
    the NNTP sockets (if appropriate for that authentication protocol),
    using the protocol specified by the authenticator name. These
    authentication protocols are not included in this document, but are
    similar in structure to those referenced in RFC 1731 [8] for the
    IMAP-4 protocol.

    If the server returns 501, this means that the authenticator
    invocation was syntactically incorrect, or that AUTHINFO GENERIC is
    not supported. The client should retry using the AUTHINFO USER
    command.

    If the requested authenticator capability is not found, the server
    returns the 503 response code.

    If there is some other unspecified server program error, the server
    returns the 500 response code.

    The authenticators converse using their protocol until complete. If
    the authentication succeeds, the server authenticator will terminate
    with a 281, and the client can continue by reissuing the command that
    prompted the 380. If the authentication fails, the server will
    respond with a 502.

The client must provide authentication when requested by the server.
The server may request authentication at any time. Servers may
request authentication more than once during a single session.

When the server authenticator completes, it provides to the server
(by a mechanism herein undefined) the email address of the user, and
potentially what the user is allowed to access. Once authenticated,
the server shall generate a Sender: line using the email address
provided by the authenticator if it does not match the user-supplied
From: line. Additionally, the server should log the event, including
the user's authenticated email address (if available). This will
provide a means by which subsequent statistics generation can
associate newsgroup references with unique entities - not necessarily
by name.

Some implementations make it possible to obtain a list of
authentication procedures available by sending the server AUTHINFO
GENERIC with no arguments. The server then returns a list of
supported mechanisms followed by a period on a line by itself.

3.1.3.1 Responses

    281 Authentication succeeded
    480 Authentication required
    500 Command not understood
    501 Command not supported
    502 No permission
    503 Program error, function not performed
    nnn authenticator-specific protocol.

3.2 DATE

    DATE

    The first NNTP working group discussed and proposed a syntax for this
    command to help clients find out the current time from the server's
    perspective. At the time this command was discussed (1991-1992), the
    Network Time Protocol [9] (NTP) was not yet in wide use and there was
    also some concern that small systems may not be able to make
    effective use of NTP.

    This command returns a one-line response code of 111 followed by the
    GMT date and time on the server in the form YYYYMMDDhhmmss.

3.2.1 Responses

    111 YYYYMMDDhhmmss

3.3 The WILDMAT format

   The WILDMAT format was first developed by Rich Salz based on the
   format used in the UNIX "find" command to articulate file names. It
   was developed to provide a uniform mechanism for matching patterns in
   the same manner that the UNIX shell matches filenames. Patterns are
   implicitly anchored at the beginning and end of each string when
   testing for a match. There are five pattern matching operations
   other than a strict one-to-one match between the pattern and the
   source to be checked for a match. The first is an asterisk (*) to
   match any sequence of zero or more characters. The second is a
   question mark (?) to match any single character. The third specifies
   a specific set of characters. The set is specified as a list of
   characters, or as a range of characters where the beginning and end
   of the range are separated by a minus (or dash) character, or as any
   combination of lists and ranges. The dash can also be included in
   the set as a character it if is the beginning or end of the set.
   This set is enclosed in square brackets. The close square bracket
   (]) may be used in a set if it is the first character in the set.
   The fourth operation is the same as the logical not of the third
   operation and is specified the same way as the third with the
   addition of a caret character (^) at the beginning of the test string
   just inside the open square bracket. The final operation uses the
   backslash character to invalidate the special meaning of the a open
   square bracket ([), the asterisk, backslash or the question mark.
   Two backslashes in sequence will result in the evaluation of the
   backslash as a character with no special meaning.

3.3.1 Examples

   a. [^]-] -- matches any single character other than a close square
               bracket or a minus sign/dash.

   b. *bdc -- matches any string that ends with the string "bdc"
               including the string "bdc" (without quotes).

   c. [0-9a-zA-Z] -- matches any single printable alphanumeric ASCII
               character.

   d. a??d -- matches any four character string which begins
               with a and ends with d.

3.4 Additional Headers

   Many NNTP implementations add headers to Usenet articles when then
   are POSTed via NNTP. These headers are discussed in this section.
   None of these headers conflict with those specified in RFC 1036 and
   should be passed unchanged by Usenet transports conforming to RFC
   1036.

3.4.1 NNTP-Posting-Host

   This line is added to the header of a posted article by the server.
   The contents of the header is either the IP address or the fully
   qualified domain name of the client host posting the article. The
   fully qualified domain name should be determined by doing a reverse
   lookup in the DNS on the IP address of the client. If the client
   article contains this line, it is removed by the server before
   acceptance of the article by the Usenet transport system.

   This header provides some idea of the actual host posting the article
   as opposed to information in the Sender or From lines that may be
   present in the article. This is not a fool-proof methodology since
   reverse lookups in the DNS are vulnerable to certain types of
   spoofing, but such discussions are outside the scope of this
   document.

3.4.2 X-Newsreader and others

   There are other lines that are added by clients as well. Most of
   these indicate the type of newsreader software that is posting the
   article.

4.0 Common Implementation Issues

   Many NNTP implementations do not follow the specifications in RFC
   977. In this section, some common implementation issues are
   summarized.

4.1 The Response to the LIST command

   RFC 977 says that the fourth field of the "list of valid newsgroups
   associated information" returned must be "either 'y' or 'n'
   indicating whether posting to this newsgroup is allowed ('y') or
   prohibited ('n'). Most implementations simply output the exact
   contents of the transport system's active newsgroup list. For more
   implementations, the fourth field usually has more values that 'y' or
   'n'.

4.2 The Required Headers in an Article and the POST command

   RFC 977 notes in section 3.10.1 that articles presented "should
   include all required header lines." In fact, modern implementations
   only require From, Subject, and Newsgroups header lines and will
   supply the rest; further, many implementers believe that it is best
   for clients to generate as few headers as possible, since clients
   often do not format other headers correctly.

   This implementation behavior is consistent with both Bnews and Cnews
   which would supply missing headers for articles directly submitted to
   them.

4.3 Article Numbering

   RFC 977 does not directly address the rules concerning articles
   number. However, the current practice is simple: article numbers are
   monotonically increasing, articles may disappear, and therefore the
   high and low water marks returned in a GROUP command should be
   treated as maximum minima, and minimum maxima, respectively.

4.4 Availability of commands defined in RFC 977

   Some implementations permit administrators to disable commands
   defined RFC 977. Some implementations have some set of commands
   disabled by default. This means that client implementations cannot
   depend on the availability of the disabled set of commands. This
   increases the complexity of the client and does not encourage
   implementors to optimize the implementation of commands that don't
   perform well.

   NEWNEWS is one of the commands frequently disabled.

4.5 The Distribution header and NEWNEWS

   In section 12.4 of RFC 977, the optional distributions argument is
   described. This argument, according to RFC 977, would limit the
   responses to articles that were in newsgroups with prefixes that
   matched the optional distributions argument.

   Some implementations implement this by matching the Distributions
   header in articles to the distribution argument. Others do the match
   against segments of the newsgroup's name.

   This variation is probably best explained by the evolution of the
   USENET article format. At the time RFC 977 was specified, the
   newsgroup name defined how the group was distributed throughout
   USENET. RFC 1036 changed this convention. So, those that are

strictly implementing RFC 977 would match the newsgroup name prefix
against the distribution argument and only display matches. Those
that implement against the intent of the command (as modified by the
redefinition of the article format)would match the Distributions
header against the distribution argument and only display those
matches.

5.0 Further Work

   With the continued use of NNTP on the Internet, there remains an
   interest in creating an optimized transport protocol for server-to-
   server transfers and an optimized client protocol for client-to-
   server interactions. There is also considerable interest is building
   better mechanisms to provide audit information on which news groups
   are being read by which users.

   An IETF working group has been formed and it is the hope of this
   author that these issues will be addressed in that forum.

6.0 Security Considerations

   The use of the AUTHINFO is optional. This command as documented has
   a number of security implications. In the original and simple forms,
   all passwords are passed in plaintext and could be discovered by
   various forms of network or system surveillance. The AUTHINFO
   GENERIC command has the potential for the same problems if a
   mechanism is used that also passes cleartext passwords. RFC 1731 [8]
   discusses these issues in greater detail.

7.0 References

   [1] Kantor, B and P. Lapsley, "Network News Transfer Protocol", RFC
        977, February 1986.

   [2] Limoncelli, Tom, "Read This Before You Write a Newsreader",
        http://mars.superlink.net/tal/news-software-authors.html, June,
        1996.

   [3] Horton, M. and R. Adams, "Standard for interchange of USENET
        messages", RFC 1036, December 1987.

   [4] Salz, Rich, Manual Page for wildmat(3) from the INN 1.4
        distribution, UUNET Technologies, Revision 1.10, April, 1992.

   [5] Robertson, Rob, "FAQ: Overview database / NOV General
        Information", ftp://ftp.uu.net/networking/news/nntp/inn/faq-
        nov.Z, January, 1995.

   [6] Lea, Iain, "FAQ about the TIN newsreader",
        http://www.cs.unca.edu/~davidson/handouts/tinfaq.html

   [7] Kappesser, Peter, "[news.software.readers] trn newsreader FAQ",
        2 parts, ftp://rtfm.mit.edu/pub/usenet-by-hierarchy/news/
        software/readers/%5Bnews.software.readers%5D_trn_newsreader
        _FAQ%2C_part_1%3A_Basics and ftp://rtfm.mit.edu/pub/usenet-by
        -hierarchy/news/software/readers/%5Bnews.software.readers
        %5D_trn_news-reader_FAQ%2C_part_2%3A_Advanced, February, 1995.

   [8] Meyers, J., "IMAP4 Authentication Mechanisms", RFC 1731,
        December 1994.

   [9] Mills, D., "Network Time Protocol (Version 3), Specification,
        Implementation and Analysis", RFC 1305, March 1992.

8.0 Notes

   DEC is a registered trademark of Compaq Computer Corporation. UNIX
   is a registered trademark of The Open Group. VMS is a registered
   trademark of Compaq Computer Corporation.

9.0 Acknowledgments

   The author gratefully acknowledges the comments and additional
   information provided by the following individuals:

   Wayne Davison <davison@armory.com>
   Chris Lewis <clewis@bnr.ca>
   Tom Limoncelli <tal@lucent.com>
   Eric Schnoebelen <eric@egsner.cirr.com>
   Rich Salz <rsalz@osf.org>

   This work was precipitated by the work of various newsreader authors
   and newsserver authors which includes those listed below:

   Rick Adams   -- Original author of the NNTP extensions to the RN
                     newsreader and last maintainer of Bnews
   Stan Barber   -- Original author of the NNTP extensions to the
                     newsreaders that are part of Bnews.
   Geoff Collyer -- Original author of the OVERVIEW database proposal and
                     one of the original authors of CNEWS
   Dan Curry   -- Original author of the xvnews newsreader
   Wayne Davison -- Author of the first threading extensions to the
                     RN newsreader (commonly called TRN).
   Geoff Huston   -- Original author of ANU NEWS

        Phil Lapsey   -- Original author of the UNIX reference
                         implementation
        Iain Lea   -- Original maintainer of the TIN newsreader
        Chris Lewis   -- First known implementor of the AUTHINFO GENERIC
                         extension
        Rich Salz   -- Original author of INN
        Henry Spencer -- One of the original authors of CNEWS
        Kim Storm   -- Original author of the NN newsreader

10.0 Author's Address

        Stan Barber
        P.O. Box 300481
        Houston, Texas, 77230

        EMail: sob@academ.com

11.0 Full Copyright Statement