

Network Working Group  
Request for Comments: 4707  
Category: Experimental

P. Grau  
V. Heinau  
H. Schlichting  
R. Schuettler  
Freie Universitaet Berlin  
October 2006

## Netnews Administration System (NAS)

### Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2006).

### IESG Note

This RFC is not a candidate for any level of Internet Standard. The IETF disclaims any knowledge of the fitness of this RFC for any purpose, and in particular notes that the decision to publish is not based on IETF review for such things as security, congestion control or inappropriate interaction with deployed protocols. The RFC Editor has chosen to publish this document at its discretion. Readers of this document should exercise caution in evaluating its value for implementation and deployment.

### Abstract

The Netnews Administration System (NAS) is a framework to simplify the administration and usage of network news (also known as Netnews) on the Internet. Data for the administration of newsgroups and hierarchies are kept in a distributed hierarchical database and are available through a client-server protocol.

The database is accessible by news servers, news administrators, and news readers. News servers can update their configuration automatically; administrators are able to get the data manually. News reader programs are able to get certain information from an NAS server, automatically or at a user's discretion, which provides detailed information about groups and hierarchies to the user.

NAS is usable in coexistence with the current, established process of control messages; an unwanted interference is impossible. Furthermore, NAS is able to reflect the somewhat chaotic structure of Usenet in a hierarchical database. NAS can be used without modification of existing news relay, news server, or news reader software; however, some tasks will be better accomplished with NAS-compliant software.

## Table of Contents

1. Introduction .....	3
2. Overview .....	4
3. Protocol Level .....	5
4. Description of Functions .....	6
5. Definitions .....	7
6. Specification of the NAS Protocol (TCP) .....	8
6.1. Responses .....	8
6.1.1. Overview .....	8
6.1.2. Response Code Values, Structure, and Meaning .....	8
6.2. Connection Setup .....	9
6.3. Commands .....	10
6.3.1. Structure .....	10
6.3.2. Overview .....	10
6.3.3. Detailed Description .....	10
6.3.3.1. HELP .....	11
6.3.3.2. INFO .....	12
6.3.3.3. DATE .....	13
6.3.3.4. VERS .....	14
6.3.3.5. QUIT .....	15
6.3.3.6. LIST .....	16
6.3.3.7. LSTR .....	18
6.3.3.8. HIER .....	19
6.3.3.9. DATA .....	21
6.3.3.10. GETP .....	22
6.3.3.11. GETA .....	25
6.3.3.12. Unknown Commands and Syntax Errors .....	27
6.3.4. Data Headers .....	27
6.4. Status Indicators .....	41
6.5. Newsgroup Types .....	41
6.6. Hierarchy Types .....	42
6.7. PGP Keys .....	42
7. Specification of the NAS Protocol (UDP) .....	44
8. IANA Considerations .....	44
9. Security Considerations .....	44
10. Response Codes (Overview) .....	45
11. Data Headers for DATA and HIER Commands (Overview) .....	45

12. References .....	46
12.1. Normative References .....	46
12.2. Informative References .....	47

## 1. Introduction

An increasing number of newsgroups, hierarchies, and articles has made the administration of news servers a complex and time-consuming task. The tools for the administration have remained unchanged for ten years and are no longer appropriate. Many hierarchies are inconsistent; many new newsgroups are not created or only with a large delay; removed groups keep lurking in the configuration files for a long period of time. There is no administration tool that utilizes the power of the Internet, and it is not possible to check the consistency of the news server at a given point of time.

Users find it difficult to get an overview of the newsgroups, the charter of a particular one, which language is preferred, or whether a group is moderated. Renaming, the status change from moderated to unmoderated or vice versa, and the splitting of a group into several others are dynamic processes. These processes are in common use, but it takes a long time until every news server is aware of these changes.

An increasing number of faked control messages has appeared in the last few years. Purposely or accidentally, control messages were sent to foreign news servers to create or remove a certain group, although this was not approved according to the rules of the hierarchy in question. Due to this fact, automatic creation and removal are disabled on many news servers, and several dead groups have not been deleted. It is very difficult for users to determine the current status of a group, and in some cases they simply cannot tell that the group they are posting to is not an active group but a dead or invalid one.

It is the design goal of Netnews Administration System (NAS) to provide an out-of-band system that helps to maintain, propagate, and deliver the required information. There will not be any interference with current protocols and standards. It is not intended to make use of control messages or some special Network News Transfer Protocol (NNTP) commands. The advantage of NAS is that it provides more information in a more structured format than that of control messages. Not only news server administrators but also Usenet users can get more detailed information about newsgroups and hierarchies.

Due to the fact that a client connects to a server and the server asks for authentication, this is a more reasonable procedure for transmitting information than that for control messages.

Furthermore, it is possible to check for changes on a regular basis at customized intervals to keep local data up-to-date.

## 2. Overview

NAS is based on a database that contains information about certain groups and hierarchies. This database is structured in a hierarchical manner and distributed to various servers, and it is able to receive queries at any time. The service is comparable to directory services like DNS, Lightweight Directory Access Protocol (LDAP), or Network Information Service (NIS). The NAS protocol is inspired by protocols like NNTP and SMTP. The port 991 is reserved for NAS and registered by the Internet Assigned Numbers Authority (IANA) [IANA-PN].

The organizational structure of NAS is hierarchical; this means that a NAS root server collects data from the sub-servers that are authoritative for certain hierarchies. The root server signs the data and distributes it authoritatively. Replication of database entries is possible. The hierarchical structure can consist of multiple levels. Usage of the database is possible for news servers, news readers, and special client programs. The communication is based on TCP and UDP.

Taking the real world into account, there might be some policy problems with a single root server. But it is possible to establish a structure like that of the current Usenet system, where some hierarchies have a good administration with a well-defined system of rules, and where some are not well maintained. The goal is to get as much information as possible under one hat, but there can be no "official" force to achieve this.

During the startup phase, it is quite likely that there will be a root server, handling just hierarchies with strict rules and accepted authorities (e.g., BIG8, de.\*, us.\*, bln.\*, fr.\*, it.\*).

However, it is also imaginable to have some NAS servers providing data on, for example, alt.!binaries, some providing data on alt.\*, and even some providing alt.\* following special policies or sets of rules.

An administrator using NAS will have the choice to use just one root server (and all its data) or to use another NAS server for special hierarchies.

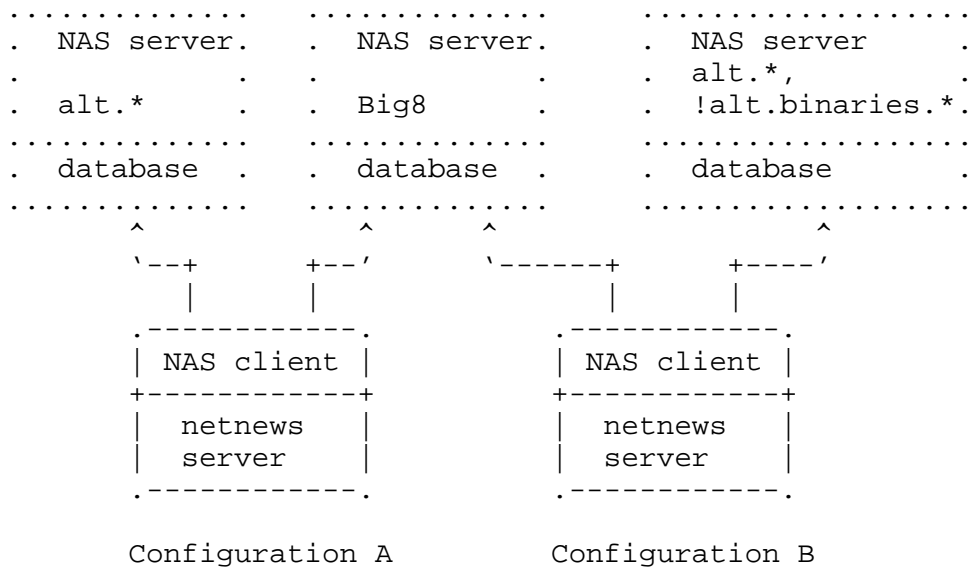


Figure 1

NAS contains information about newsgroups and complete hierarchies. Furthermore, it contains information about the hierarchies' inheritable entries and default values for a single newsgroup.

### 3. Protocol Level

It is expected that the real-life use of NAS will change the requirements for the Netnews Administration System. On the one hand, the protocol has to be extensible and flexible in order to implement improvements; on the other hand, it must ensure compatibility between different versions. A simultaneous migration of all sites using NAS to a new protocol version is not likely to happen. To solve this problem, NAS has a protocol level. This protocol level describes the current functionality. The protocol level, being a number between 1 and 32767, is negotiated at connection setup. Enhancements and modifications must use a different protocol level than that of their predecessors. (Usually the protocol level is incremented by 1 with every new version of the protocol specification.) Every current or future implementation **MUST** be compatible with protocol level 1 in order to fall back to this level if communication on a higher level fails.

An implementation of higher protocol levels should be able to emulate the behavior of lower levels, even if this implies a loss of features. The negotiation of the protocol level between client and server is described in the specification of the command `VERS`. If there is no agreement on the protocol level, only commands of the

protocol level 1 MUST be used. Documents enhancing or modifying the NAS standard MUST specify on which level these changes take place and how the behavior should be in other protocol levels.

This document describes protocol level 1.

#### 4. Description of Functions

In order to use an NAS server, a connection must be opened by the client. The NAS server can be located in the same domain or somewhere else on the Internet.

The NAS system is hierarchical. The idea is to have an NAS root server like the DNS root servers. The root server distributes the data collected from client NAS servers that are authoritative servers for their hierarchy. The maintenance of the authoritative data is possible on any system. The root server collects the data and makes them available to other servers, which can in turn distribute these data to other servers. The administrator has the opportunity to make use of either all data or only parts of the database. NAS servers can ask multiple NAS servers for data. An attached time stamp makes it possible to distinguish between new and old data and to avoid loops in the propagation.

To describe the NAS in greater detail, it is necessary to emphasize the hierarchical design of the NAS system. The following figure shows the propagation of data along the server hierarchy.

Authoritative data for a newsgroup or a hierarchy are collected and written into a database. These data are available through a local NAS server and are collected from this authoritative server by upstream NAS servers.

There may also be NAS servers that are not authoritative servers; these servers merely provide the information they collect from other NAS servers to clients such as news servers, administration programs, and news readers.

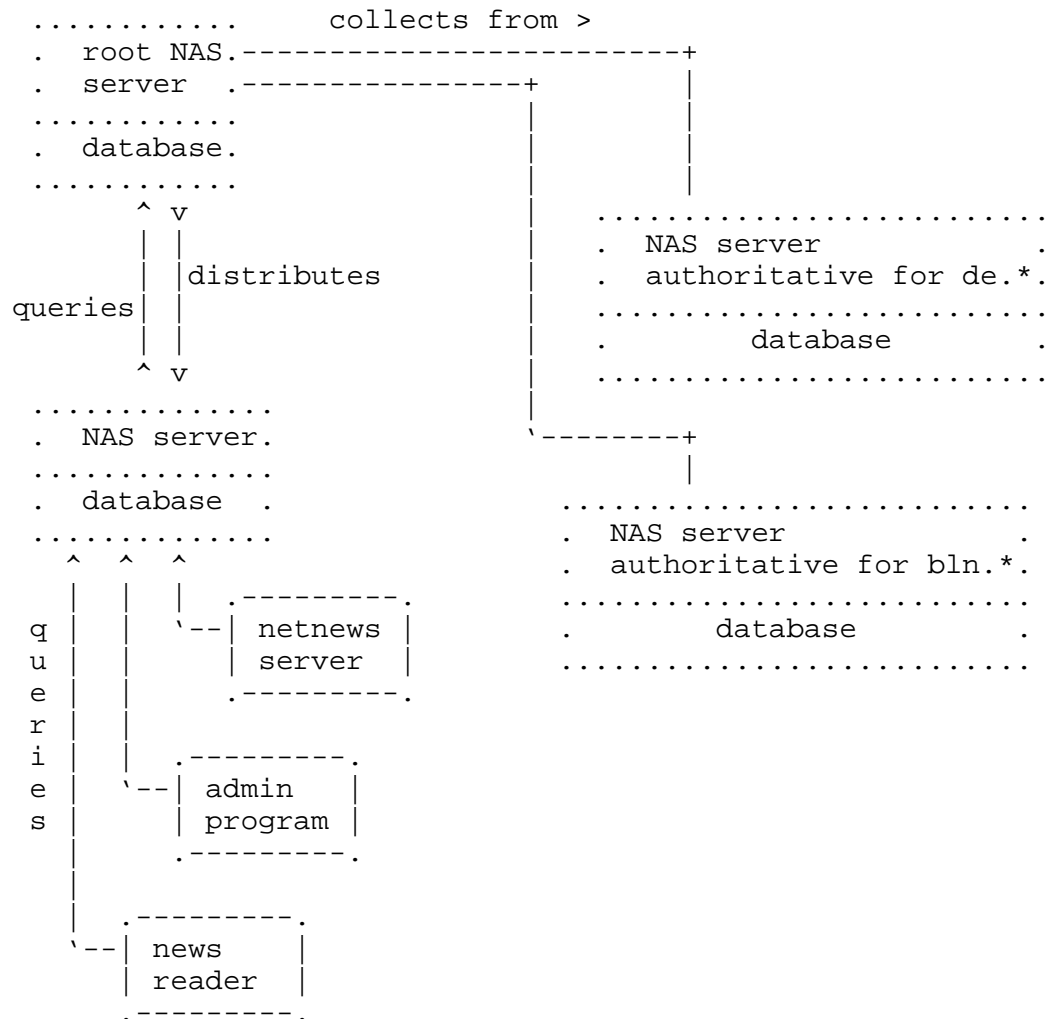


Figure 2

Requests to an NAS server originating at a client (as well as at another server) are accomplished in several steps: establishing a connection, authentication (optional), negotiating a protocol level (optional), queries on the database, and termination.

## 5. Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 6. Specification of the NAS Protocol (TCP)

### 6.1. Responses

#### 6.1.1. Overview

An answer starts with a response code (a three-digit number), optionally followed by white space and a textual message. Then the actual text/data follows. Text is sent as a series of successive lines of textual matter, each terminated with CRLF. A single line containing only a single period ('.') is sent to indicate the end of the text (i.e., the server will send a CRLF at the end of the last line of text, a period, and another CRLF).

```
Answer = response-code [answertext] CRLF
        text CRLF
        "." CRLF
```

If the original text contains a period as the first character of the text line, that first period is doubled. Therefore, the client must examine the first character of each line received and, for those beginning with a period, determine either that this is the end of the text or that it should collapse the doubled period to a single one.

#### Example

```
<-- INFO
--> 101 Information follows
    Server: nas.example.org (192.0.2.100)
    Uptime: 2 weeks, 3 days, 5 hours, 9 minutes
    Software: NAS 1.0
    Client: client.example.org (192.0.2.123)
    Connection: 9 minutes
    Highest protocol level supported: 1
    Requested protocol level: 1
    Protocol level used: 1
    .
```

#### 6.1.2. Response Code Values, Structure, and Meaning

The first digit of the response code indicates the message type (i.e., information, success, warning, error, or data):

```
1xx Information
2xx Request successful
3xx Request successful, data follow
4xx Request accepted, but no operation possible
```



5xx Request is wrong (syntax error), is not implemented, or leads to an internal error

6xx Request successful, data follow until end mark

The second digit specifies the message category:

- x0x Connection-related stuff
- x1x Queries, answers, or data
- x2x Server-server communication
- x3x Authentication, authorization
- x8x Non-standard extensions
- x9x Debugging output

The actual response code for a specific command is listed in the description of the commands. Answers of the type 1xx, 2xx, 4xx, and 5xx can have a text after the numerical code. 3xx answers contain one or more parameters with data; the exact format is explained in the description of the commands.

An answer to an incorrect request may be longer than one line.

## 6.2. Connection Setup

NAS typically uses port 991, which is reserved by IANA [IANA-PN]. If a connection is set up by the client, the server answers immediately (without a request) with the greeting message, which will start with code 200:

```
--> 200 Welcome!
    nas.example.org ready
    .
```

If a connection is refused because the client has no permission to access the server, the answer code is 434. That decision can be made on connection startup based on the client's IP address. When the server is currently out of service, the answer code is 404.

Examples:

```
--> 434 You have no permission to retrieve data. Good bye.
    .

--> 404 Maintenance time
    .
```

After sending a 404 or 434 message, the connection will be closed.

### 6.3. Commands

#### 6.3.1. Structure

A command consists of a command word, sometimes followed by a parameter. Parameters are separated from the command word by white space.

Commands used in the NAS protocol are not case sensitive. A command word or parameter may be uppercase, lowercase, or any mixture of upper- and lowercase.

The length of a command line is not limited. If the need to limit the length of command lines in real-life implementations arises, answer code 513 (line too long) should be returned.

The protocol level described in this document uses command words with a length of exactly four characters each.

In examples, octets sent to the NAS server are preceded by "<-- " and those sent by the NAS server by "--> ". The indicator is omitted if the direction of the dialog does not change.

#### 6.3.2. Overview

The commands described below are defined using the Augmented Backus-Naur Form (ABNF) defined in [RFC4234]. The definitions for 'ALPHA', 'CRLF', 'DIGIT', 'WSP' and 'VCHAR' are taken from appendix B of [RFC4234] and not repeated here.

The following ABNF definitions constitute the set of NAS commands that can be sent from the client to an NAS server.

#### 6.3.3. Detailed Description

Some overall definitions follow:

```

text           = %d1-9 /                ; all octets except
                %d11-12 /               ; US-ASCII NUL, CR and LF
                %d14-255

answertext     = WSP *( ALPHA / DIGIT / "+" / "-" / "/" / "_" /
                "." / "," / ":" / "=" / "?" / "!" / SP )

utc-time       = 14DIGIT ; the date and time of the server in UTC
                ; YYYYMMDDhhmmss

response-code  = 3DIGIT ; three digit number

```

Newsgroup names and hierarchy names are defined according to the following ABNF definitions. Since a hierarchy name can be the same as a newsgroup name (e.g., hierarchy bln.announce.fub.\* and newsgroup name bln.announce.fub), there is no difference between the two.

```

name           = plain-component *("." component)
component       = plain-component / encoded-word
encoded-word    = 1*( lowercase / DIGIT /
                    "+" / "-" / "/" / "_" / "=" / "?" )
plain-component = component-start *component-rest
component-start = lowercase / DIGIT
lowercase       = %x61-7A ; letter a-z lowercase
component-rest  = component-start / "+" / "-" / "_"

```

NOTE: This definition of newsgroup name is in reference to "News Article Format and Transmission" [SON1036]. When the document "News Article Format" [USEFOR] is established as an RFC, its definitions should be integrated into a higher protocol level of NAS.

#### 6.3.3.1. HELP

##### Description

This command prints a short help text on a given command. If called without parameters, it will display a complete list of commands.

```
help-cmd = "HELP" [WSP commandname] CRLF
```

```

commandname = "DATA" / "DATE" / "GETP" / "GETA" /
               "HELP" / "HIER" / "INFO" / "LIST" /
               "LSTR" / "QUIT" / "VERS"

```

##### Possible answers

```

100: Command overview, command description
410: Indicates that the server is not giving any information

```

```

help-answer = "410" [answertext] CRLF
              text CRLF
              "." CRLF
help-answer =/ "100" [answertext] CRLF
              text CRLF
              "." CRLF

```

##### Examples

```
<-- HELP
```

--> 100 NAS server nas.example.org - Version 1.0

Supported commands:

DATA - data for a newsgroup  
DATE - show time of server in UTC  
GETP - get package  
GETA - get data from an authoritative server  
HELP - show this help  
HIER - data for a hierarchy  
INFO - show info on current connection  
LIST - list newsgroups or hierarchies  
LSTR - recursive list newsgroups or hierarchies  
QUIT - close the connection  
VERS - show or set current protocol level

Contact address nas@example.org

.

<-- HELP LIST

--> 100 LIST

LIST - list newsgroups or hierarchies  
Syntax: LIST hierarchy ...  
Get a list of newsgroups and sub-hierarchies  
directly under the parameter hierarchy

.

<-- HELP NOOP

--> 410

unknown command "NOOP"

.

#### 6.3.3.2. INFO

##### Description

Prints information about the current connection, the server, and the client.

info-cmd = "INFO" CRLF

##### Possible answers

101: Normal answer; prints some information about client  
and server

400: Indicates that the server is not giving any information

```
info-answer = "400" [answertext] CRLF
              text CRLF
              "." CRLF
info-answer =/ "101" [answertext] CRLF
              text CRLF
              "." CRLF
```

#### Examples

```
<-- INFO
--> 101 Information follows
    Server: nas.example.org (192.0.2.100)
    Uptime: 2 weeks, 3 days, 5 hours, 9 minutes
    Software: NAS 1.0
    Client: client.example.org (192.0.2.123)
    Connection: 9 minutes
    Highest protocol level supported: 1
    Requested protocol level: 1
    Protocol level used: 1

    End
    .

<-- INFO
--> 400
    No information available.
    .
```

#### 6.3.3.3. DATE

##### Description

Prints the current time of the server in UTC (Universal Coordinated Time) in the format YYYYMMDDhhmmss, followed by an optional comment. The DATE command is only for informational use and to check the server time. For regular transmission of time over the network, the Network Time Protocol (NTP) [RFC1305] should be used.

```
date-cmd = "DATE" CRLF
```

##### Possible answers

300: Print the UTC time in specified format; see below  
511: Error; print an error message

```
date-answer = "511" [answertext] CRLF
              text CRLF
              "." CRLF
```

```
date-answer =/ "300" [answertext] CRLF
              utc-time [answertext] CRLF
              "." CRLF
```

#### Examples

```
<-- DATE
--> 300
    19990427135230 UTC
    .
```

```
<-- DATE
--> 511
    Time is unknown
    .
```

#### 6.3.3.4. VERS

##### Description

The VERS command is used to determine the protocol level to use between client and server. The parameter is a protocol level that the client supports and wants to use. The server will respond with the highest level accepted. This version number MUST not be higher than that requested by the client. Client and server MUST only use commands from the level that the server has confirmed. It is possible, but seldom necessary, to change the protocol level during a session by client request (VERS [protocol level]). When no option is given, the current protocol level will be printed. When no protocol level is negotiated, the protocol level 1 will be used. Commands of a higher level are not allowed without successful negotiation. The protocol level can be followed by an optional comment.

```
vers-cmd = "VERS" [WSP level] CRLF
```

level = 1\*5DIGIT ; the valid range is 1 - 32767

##### Possible answers

202: Returns current protocol level  
302: Requested level accepted  
402: Requested level too high; falling back to lower level  
510: Syntax error

```
vers-answer = "202" [answertext] CRLF
              level [answertext] CRLF
              "." CRLF
```

```
vers-answer =/ "302" [answertext] CRLF
               level [answertext] WSP level CRLF
               "." CRLF
vers-answer =/ "402" [answertext] CRLF
               level [answertext] WSP level CRLF
               "." CRLF
vers-answer =/ "510" [answertext] CRLF
               level [answertext] CRLF
               "." CRLF
```

#### Examples

```
<-- VERS
--> 202
    2 Current protocol level is 2
    .

<-- VERS 2
--> 302
    2 My max protocol level is 10
    .

<-- VERS 11
--> 402
    10 Falling back to level 10
    .

<-- VERS BAL
--> 510
    1 Syntax error
    .
```

#### 6.3.3.5. QUIT

##### Description

Terminates the connection.

```
quit-cmd = "QUIT" CRLF
```

##### Possible answers

201: Termination of the connection

```
quit-answer = "201" [answertext] CRLF
```

### Example

```
<-- QUIT
--> 201 Closing connection. Bye.
```

### 6.3.3.6. LIST

#### Description

To obtain a list of newsgroups and sub-hierarchies in the requested hierarchies, the command LIST is used. The status of the hierarchies is also given. The highest level consists of all top-level hierarchies and is labeled "\*". It can be obtained this way, too.

The data consist of a newsgroup- or hierarchy-name/status indicator pair per line. Name and status indicator must be separated by at least one white space. The status indicator is a single word (see Section 6.4). The interpretation is not case sensitive.

```
list-cmd = "LIST" ( WSP "*" / 1*(WSP name)) CRLF
```

#### Possible answers

```
401: Permission denied
510: Syntax error
610: Normal response with all requested data
```

```
list-answer = "610" [answertext] CRLF
              *(listdata CRLF)
              "." CRLF
list-answer =/ "401" [answertext] CRLF
              text CRLF
              "." CRLF
list-answer =/ "510" [answertext] CRLF
              text CRLF
              "." CRLF
```

```
listdata     = name WSP list-status
```

The list-status is the status of a newsgroup or hierarchy according to Section 6.4.

```
list-status = "Complete"      /
              "Incomplete"    /
              "Obsolete"      /
              "Unknown"       /
              "Unmoderated"   /
              "Readonly"      /
```



```
"Moderated"    /  
"Removed"      ; list-status is case-insensitive
```

#### Examples

```
<-- LIST *  
--> 610 data follow  
    alt Incomplete  
    comp Complete  
    de Incomplete  
    rec Complete  
    sub Obsolete  
    .  
  
<-- LIST de  
--> 610 data follow  
    de.admin Complete  
    de.alt Incomplete  
    de.comm Complete  
    de.comp Complete  
    de.etc Complete  
    de.markt Complete  
    de.newusers Complete  
    de.org Complete  
    de.rec Complete  
    de.sci Complete  
    de.soc Complete  
    de.answers Moderated  
    de.test Unmoderated  
    .  
  
<-- LIST foo  
--> 610 data follow  
    foo Unknown  
    .  
  
<-- LIST  
--> 510 Syntax error  
    missing parameter hierarchy  
    .  
  
<-- LIST de  
--> 401 Something is wrong  
    Permission denied  
    .
```

## 6.3.3.7. LSTR

## Description

To obtain a recursive list of newsgroups and sub-hierarchies in the named hierarchy, the command LSTR is used. The status of the hierarchies is also given. The highest level consists of all top-level hierarchies and is labeled "\*". It can be obtained this way, too.

The use of "\*" as a wildcard pattern following the beginning of a hierarchy name is also possible; so a "LSTR de.a\*" would return a list of all newsgroups and hierarchies starting with "de.a".

```
lstr-cmd = "LSTR" ( WSP "*" / 1*(WSP name ["*" / ".*"]) ) CRLF
```

## Possible answers

401: Permission denied

510: Syntax error

610: Normal answer with all requested data

```
lstr-answer = "610" [answertext] CRLF
              *(listdata CRLF)
              "." CRLF
```

```
lstr-answer =/ "401" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
lstr-answer =/ "510" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
listdata    = name WSP list-status
```

The list-status is the status of a newsgroup or hierarchy according to Section 6.4.

```
list-status = "Complete"      /
              "Incomplete"    /
              "Obsolete"      /
              "Unknown"       /
              "Unmoderated"   /
              "Readonly"      /
              "Moderated"     /
              "Removed"       ; list-status is case-insensitive
```

## Example

```

<-- LSTR de.admin
--> 610 recursive mode
    de.admin Complete
    de.admin.infos Moderated
    de.admin.lists Moderated
    de.admin.misc Unmoderated
    de.admin.net-abuse Complete
    de.admin.net-abuse.announce Moderated
    de.admin.net-abuse.mail Unmoderated
    de.admin.net-abuse.misc Unmoderated
    de.admin.net-abuse.news Unmoderated
    de.admin.news Complete
    de.admin.news.announce Moderated
    de.admin.news.groups Unmoderated
    de.admin.news.misc Unmoderated
    de.admin.news.nocem Unmoderated
    de.admin.news.regeln Unmoderated
.

```

## 6.3.3.8. HIER

## Description

The command HIER lists all information available about the hierarchy. With the data header "Name", a new data block for each hierarchy is started. The header "Name" gives the name of the hierarchy. The data headers are described in Section 6.3.4. The default is to transmit all available information. It can be limited to a list of desired headers ("Name" and "Status" are always given). A set of comma-separated headers, as an option to the HIER command, will return the requested header fields.

```
hier-cmd = "HIER" 1*(WSP name) [WSP selection] CRLF
```

```
selection = *( "," header )           ; Describes the data fields
                                         ; that are requested
```

```
header     = ALPHA *( ALPHA / "-" ) ; According to section 6.3.4
```

## Example for selection

```
,Followup,Description : For all entries list Name, Status, Followup
                        and Description
```

## Possible answers

```
401: Permission denied
```

510: Syntax error  
 611: Regular answer with all requested data

```
hier-answer = "611" [answertext] CRLF
              *(hierdata CRLF)
              "." CRLF
hier-answer =/ "510" [answertext] CRLF
              *(text CRLF)
              "." CRLF
hier-answer =/ "401" [answertext] CRLF
              *(text CRLF)
              "." CRLF

hierdata     = "Name:" WSP text CRLF
              "Status:" WSP text CRLF
              *(header ":" WSP text CRLF)
              [ ("Ctl-PGP-Key:" CRLF PGP-answer /
                "Mod-PGP-Key:" CRLF PGP-answer)]
```

PGP-answer: The exact format is described in Section 6.7.

#### Examples

```
<-- HIER de
--> 611 Data coming
    Name: de
    Status: Complete
    Serial: 20020823120306
    Description: Internationale deutschsprachige Newsgruppen
    Netiquette: http://www.kirchwitz.de.example/~amk/dni/netiquette
    FAQ: http://www.kirchwitz.de.example/~amk/dai/einrichtung
    Ctl-Send-Adr: moderator@dana.de.example
    Ctl-Newsgroup: de.admin.news.announce
    Mod-Wildcard: %s@moderators.dana.de.example
    Language: DE
    Charset: ISO-8859-1
    Encoding: text/plain
    Newsgroup-Type: Discussion
    Hier-Type: Global
    Comp-Length: 14
    Date-Creation: 19920106000000
    .

<-- HIER bln
--> 401
    Permission denied
    .
```

```
<-- HIER
--> 510 Syntax error
    missing parameter hierarchy
    .
```

#### 6.3.3.9. DATA

##### Description

The DATA command corresponds to the HIER command, as explained in 6.3.3.8, but it is used for information about a newsgroup. A summary of codes can be found in Section 6.3.4.

```
data-cmd  = "DATA" 1*(WSP name) [WSP selection] CRLF
```

##### Possible answers

```
401: Permission denied
510: Syntax error
612: Regular answer with all requested data
```

```
data-answer = "612" [answertext] CRLF
              *(datadata CRLF)
              "." CRLF
```

```
data-answer =/ "510" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
data-answer =/ "401" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
datadata    = "Name:" WSP text CRLF
              "Status:" WSP text CRLF
              *(header ":" WSP text CRLF)
              [ ("Ctl-PGP-Key:" CRLF PGP-answer /
                "Mod-PGP-Key:" CRLF PGP-answer)]
```

##### Examples

```
<-- DATA de.comp.os.unix.linux.moderated
--> 612 data follow
    Name: de.comp.os.unix.linux.moderated
    Status: Moderated
    Serial: 20020823120312
    Description: Linux und -Distributionen.
                  <dcoulm-moderators@linux-config.de.example>
    Charter: http://www.dana.de.example/mod/chartas/de.html
    Netiquette: http://www.kirchwitz.de.example/~amk/dni/netiquette
```

```
Netiquette: ftp://ftp.fu-berlin.de.example/doc/usenet/german
                                         /Netiquette
Mod-Sub-Adr: dcoulm-moderators@linux-config.de.example
Mod-Group-Info: http://wpxx02.toxi.uni-wuerzburg.de.example
                                         /~dcoulmod/
Newsgroup-Type: Discussion

.

<-- DATA de.foo
--> 612 data follow
    Name: de.foo
    Status: Unknown

.

<-- DATA de
--> 401
    Permission denied

.

<-- DATA
--> 510 Syntax error
    missing parameter newsgroup

.
```

#### 6.3.3.10. GETP

##### Description

GETP is used for server-server communication. It requests the data for the hierarchy specified by the parameter "name". The format of the data is the same as for the commands "HIER" and "LIST". If "\*" is given as hierarchy name, all data the server is offering will be transmitted.

The "timestamp" attached to a package consists of the date and time that the package was created. The timestamp for a package is transmitted together with the package data by the server and marks a specific revision for the package data.

When a client requests a package with GETP, it transmits the timestamp attached to the package in its database so that the server can check whether the data on the client side is still valid or if it is too old. If the data on the client side is still valid, a 213 answer is sent, so the client knows that its data is OK. If the timestamp is "0", the server is forced to transmit the data.

Timestamps set by the server must be increasing and may not be more than 12 hours in the future.

The data for a successful request are signed and sent in ASCII armor according to [RFC2440], so a client can check the signature or ignore it. The actual data will be surrounded by the armor start and end sections, according to Section 6.2 of [RFC2440].

```
getp-cmd = "GETP" WSP username WSP password WSP timestamp
           WSP ( name / "*" ) CRLF
```

```
username = *1( VCHAR ) / "0" ; Length of VCHAR >= 1
```

```
password = *1( VCHAR ) / "0" ; Length of VCHAR >= 1
```

```
timestamp = utc-time / ; date and time of the last retrieval
            "0"         ; force the transmission of data
```

Possible answers

213: Current data at the client side

411: No hierarchy with that name

430: Permission denied

510: Syntax error

613: Hierarchy data

```
getp-answer = "613" [answertext] CRLF
              pgp-ascii-armor-start ; this is according to [RFC2440]
              *(getpdata CRLF)
              pgp-ascii-armor-end   ; this is according to [RFC2440]
              "." CRLF
```

```
getp-answer =/ "213" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
getp-answer =/ "430" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
getp-answer =/ "411" [answertext] CRLF
               text CRLF
               "." CRLF
```

```
getp-answer =/ "510" [answertext] CRLF
               text CRLF
               "." CRLF
```

pgp-ascii-armor-start and the pgp-ascii-armor-end are built according to [RFC2440], Section 6.2., "Forming ASCII Armor".

```
getpdata    =  "Name:" WSP text CRLF
                "Status:" WSP text CRLF
                "Serial:" WSP timestamp CRLF
                *(header ":" WSP text CRLF)
                [ ("Ctl-PGP-Key:" CRLF PGP-answer /
                  "Mod-PGP-Key:" CRLF PGP-answer) ]
```

#### Examples

```
<-- GETP 0 0 0 humanities
--> 615 data follow
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Name: humanities
Status: Complete
Serial: 20020821094529
Description: Branches of learning that investigate human
             constructs and concerns as opposed to natural processes.
Netiquette: ftp://rtfm.mit.edu.example/pub/usenet
             /news.announce.newusers
             /A_Primer_on_How_to_Work_With_the_Usenet_Community
Rules: http://www.uvv.org.example/docs/howto.txt
Ctl-Send-Adr: group-admin@isc.org.example
Ctl-Newsgroup: news.announce.newgroup
Language: EN
Charset: US-ASCII
Encoding: text/plain
Newsgroup-Type: Discussion
Hier-Type: Global
Comp-Length: 14
Date-Creation: 19950417143009

Name: humanities.answers
Status: Moderated
Serial: 20020821094533
Description: Repository for periodic USENET articles. (Moderated)
Mod-Sub-Adr: news-answers@mit.edu.example
Mod-Adm-Adr: news-answers-request@mit.edu.example
Newsgroup-Type: Announce
Date-Creation: 19950725182040
Name: humanities.classics
[...]
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.0.7 (IRIX64)

iD8DBQE9Zj/Wn13IYldLZg8RAhWiAJ4y7o+3FzBpRjYJj2HWwXyG2g8FoQCfeEsH
rRynPhhjveiY/XBkkrrZFho=
```



```
=muK4
-----END PGP SIGNATURE-----
.

<-- GETP 0 0 19990909101000 de
--> 213
    You are up-to-date
.

<-- GETP foo
--> 510 Syntax error
    Missing parameters
.

<-- GETP guest test 0 de
--> 430
    You have no permission to retrieve the data
.
```

#### 6.3.3.11. GETA

##### Description

The GETA command is used for server-server communication; it is used to collect authoritative data and will request packages that the server is authoritative for. A package is the authoritative data either for a newsgroup or a hierarchy. Each package has a "timestamp" attached to mark the revision of the package. This timestamp is set by the server to the date of the last modification of the package data in UTC format. A timestamp of "0" indicates that the package MUST be retrieved. If the retrieving client has a recent package (i.e., no modification on the authoritative server), the server sends only a 215 response. The format of the data is the same as that for the commands "HIER" and "LIST".

```
geta-cmd = "GETA" WSP username WSP password WSP
           timestamp WSP name CRLF
```

##### Possible answers

```
215: The client already has the current data
430: Permission denied
411: No hierarchy with that name
510: Syntax error
615: Regular answer with all requested data
```

```

geta-answer = "615" [answertext] CRLF
               pgp-ascii-armor-start ; this is according to [RFC2440]
               *(getadata CRLF)
               pgp-ascii-armor-end   ; this is according to [RFC2440]
               "." CRLF
geta-answer =/ "215" [answertext] CRLF
               text CRLF
               "." CRLF
geta-answer =/ "430" [answertext] CRLF
               text CRLF
               "." CRLF
geta-answer =/ "411" [answertext] CRLF
               text CRLF
               "." CRLF
geta-answer =/ "510" [answertext] CRLF
               text CRLF
               "." CRLF

getadata      = "Name:" WSP text CRLF
                "Status:" WSP text CRLF
                "Serial:" WSP timestamp CRLF
                *(header ":" WSP text CRLF)
                [ ("Ctl-PGP-Key:" CRLF PGP-answer/
                  "Mod-PGP-Key:" CRLF PGP-answer)]

```

#### Example

```

<-- GETA 0 0 0 humanities
--> 613 data follow
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Name: humanities
Status: Complete
Serial: 20020821094529
Description: Branches of learning that investigate human
             constructs and concerns as opposed to natural processes.
Netiquette: ftp://rtfm.mit.edu.example/pub/usenet
             /news.announce.newusers
             /A_Primer_on_How_to_Work_With_the_Usenet_Community
Rules: http://www.uvv.org.example/docs/howto.txt
Ctl-Send-Adr: group-admin@isc.org.example
Ctl-Newsgroup: news.announce.newgroup
Language: EN
Charset: US-ASCII
Encoding: text/plain
Newsgroup-Type: Discussion
Hier-Type: Global

```

```

Comp-Length: 14
Date-Create: 19950417143009

Name: humanities.answers
Status: Moderated
Serial: 20020821094533
Description: Repository for periodic USENET articles. (Moderated)
Mod-Sub-Adr: news-answers@mit.edu.example
Mod-Adm-Adr: news-answers-request@mit.edu.example
Newsgroup-Type: Announce
Date-Create: 19950725182040

Name: humanities.classics
[...]
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.0.7 (IRIX64)

iD8DBQE9Zj/Wn13IYldLZg8RAhWiAJ4y7o+3FzBpRjYJj2HWwXyG2g8FoQCfeEsh
rRynPhhjveiY/XBkkrrZFho=
=muK4
-----END PGP SIGNATURE-----
.

```

#### 6.3.3.12. Unknown Commands and Syntax Errors

If a command is recognized as unknown, a 519 return code (unknown command) is given. If an error occurs after the command string (e.g., a missing parameter), a 510 return code (Syntax error: Missing parameter) is given.

#### 6.3.4. Data Headers

The following paragraphs describe key words and key terms that support retrieval and storing of information. Every header has a unique English name.

The content of a header is inheritable within a hierarchy, as long as the header is marked as inheritable. The content is the default value for all downstream newsgroups and sub-hierarchies. For example, in the hierarchy "de", the language header has the value "DE" (German); therefore, this value is "DE" for all newsgroups in this hierarchy, except for those that explicitly define a language code of their own.

Hierarchies and newsgroups must have at least values for the headers "Name" and "Status". Unknown hierarchies or groups get the status "Unknown".

The header used in the NAS protocol are not case sensitive. A header may be uppercase, lowercase, or any mixture of upper- and lowercase. It is recommended that the first letter of the header and the first letter after a dash be uppercase and that all other characters be lowercase.

#### Name

Header: Name

Used for: hierarchy

Mandatory: yes

Inheritable: no

Repeatable: no

Description: Name of a hierarchy.

Comment: Start of a new data block.

Example: Name: comp

Used for: newsgroup

Mandatory: yes

Repeatable: no

Description: Name of a newsgroup

Comment: Start of a new data block.

Example: Name: de.admin.news.announce

#### Status

Header: Status

Used for: hierarchy

Mandatory: yes

Inheritable: no

Repeatable: no

Description: Status of a hierarchy.

Comment: For a detailed description, see Section 6.4.

Example: Status: Hierarchy-Complete

Used for: newsgroup

Mandatory: yes

Repeatable: no

Description: Status of a newsgroup.

Comment: For a detailed description, see Section 6.4.

Example: Status: Group-Moderated

## Serial

Header: Serial

Used for: hierarchy

Mandatory: no

Inheritable: no

Repeatable: no

Description: Timestamp for hierarchy data.

Comment: For a detailed description, see Section 6.4.

Example: Serial: 20020821102413

Used for: newsgroup

Mandatory: no

Inheritable: no

Repeatable: no

Description: Timestamp for newsgroup data.

Comment: For a detailed description, see Section 6.4.

Example: Serial: 20020821102413

## Group for followup

Header: Followup

Used for: newsgroup

Mandatory: no

Repeatable: no

Description: Name of the newsgroup that will take the followup postings of a moderated group.

Comment: The value can be used as default value for the "Followup-To:" header on postings to a moderated group. This value is only useful on groups that are moderated (Status Group-Moderated) and have a dedicated discussion group.

Example: Followup: bln.announce.fub.zedat.d  
(for the moderated group bln.announce.fub.zedat)

## Short description

Header: Description

Used for: hierarchy

Mandatory: no

Inheritable: no

Repeatable: no

Description: Short description of a hierarchy.

Example: Description: Angelegenheiten, die den Grossraum Berlin  
betreffen  
(for the hierarchy bln)

Used for: newsgroup

Mandatory: no

Repeatable: no

Description: Short description of a newsgroup.

Comment: This information is often presented to the news reader  
upon selection of the newsgroup, and it should be a  
brief but meaningful description of the topic.

Example: Description: Technisches zur Newssoftware  
(for de.admin.news.software)

## Charter-URL

Header: Charter

Used for: hierarchy

Mandatory: no

Inheritable: no

Repeatable: yes

Description: URL that points to the charter of a hierarchy.

Example: Charter: ftp://ftp.fu-berlin.de.example/doc/news/bln/bln  
(for the hierarchy bln)

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: URL that points to the charter of a newsgroup.

Comment: This information should be presented to the  
news reader upon selection of the newsgroup.

Example: Charter: ftp://ftp.fu-berlin.de.example/doc/news/bln  
/bln.markt.arbeit

## Netiquette-URL

Header: Netiquette

Used for: hierarchy  
Mandatory: no  
Inheritable: yes  
Repeatable: yes  
Description: URL that points to the netiquette of a hierarchy.  
Comment: Since the netiquettes are often valid for a complete hierarchy, this is inheritable.  
Example: Netiquette:  
<http://www.kirchwitz.de.example/~amk/dni/netiquette>

Used for: newsgroup  
Mandatory: no  
Repeatable: yes  
Description: URL for Netiquette.  
Comment: If a group has some special rules, this is the pointer to these rules.  
Example: Netiquette: <http://go.to.example/bln.markt>  
(for bln.markt)

## Frequently Asked Questions (FAQ)

Header: FAQ

Used for: Newsgroup  
Mandatory: no  
Repeatable: yes  
Description: URL for the FAQ of a newsgroup.  
Example: FAQ: <http://www.dard.de.example/>

## Administration rules

Header: Rules

Used for: hierarchy  
Mandatory: no  
Inheritable: yes  
Repeatable: yes  
Description: URL pointing to a document that describes the rules for creating, deleting, or renaming newsgroups in this hierarchy.  
Comment: Normally inherited from the toplevel hierarchy.

Example: Rules: `http://www.kirchwitz.de.example/~amk/dai`  
`/einrichtung`

## Control Email

Header: Ctl-Send-Adr

```
Used for:      hierarchy
Mandatory:     no
Inheritable:   yes
Repeatable:    yes
Description:    Email address of the sender of control messages.
Comment:       Multiple addresses are valid.
Example:       Ctl-Send-Adr: group-admin@isc.org.example
```

## Control newsgroup

Header: Ctl-NewsGroup

```
Used for:      hierarchy
Mandatory:    no
Inheritable:  yes
Repeatable:   yes
Description:   Name of the newsgroup that will get the postings for
               checkgroups, rmgroup, and newsgroup control messages.
Example:       Ctl-Newsgroup: de.admin.news.groups
```

## Moderators

Header: Mod-Wildcard

```
Used for:      hierarchy
Mandatory:     no
Inheritable:   yes
Repeatable:    no
Description:    Moderator wildcard for this hierarchy.
Comment:       This information can be used for the configuration of
                the news software, for example, to configure the
                moderators file in INN.
Example:       Mod-Wildcard: %s@moderators.dana.de.example
                (for the hierarchy de)
```



## Submission address

Header: Mod-Sub-Adr

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: Email address for submissions to the newsgroup.

Comment: If there is no "Mod-Sub-Adr" for a moderated newsgroup, "Mod-Wildcard" of the hierarchy is used. This is useful only for moderated groups (Status Group-Moderated).

Example: Mod-Sub-Adr: news-answers@mit.edu.example  
(for the newsgroup news.answers)

## Moderator's address (email)

Header: Mod-Adm-Adr

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: Email address of the moderator of the newsgroup.

Comment: If there is no code "Mod-Adm-Adr" for a moderated newsgroup, "Mod-Wildcard" of the hierarchy is used. This is useful only for moderated groups (Status Group-Moderated).

Example: Mod-Adm-Adr: news-answers-request@mit.edu.example  
(for the newsgroup news.answers)

## Info-URL

Header: Mod-Group-Info

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: URL that points to a document where the moderator presents information about the newsgroup and the submission of articles.

Example: Mod-Group-Info: <http://www.example.org/cola-submit.html>  
(for comp.os.linux.announce)

## Language

Header: Language

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: yes

Description: The language that will normally be used in postings.

Comment: The notation is according to the "Content-Language" field of [RFC2616]. The languages not preferred are enclosed in parentheses.

Example: Language: DE  
(for the hierarchy de)

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: The language that will normally be used in postings.

Comment: The notation is according to the "Content-Language" field of [RFC2616]. The languages not preferred are enclosed in parentheses.

Example: Language: TR  
Language: DE  
Language: (EN)  
(for the newsgroup bln.kultur.tuerkisch)

## Charset

Header: Charset

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: yes

Description: Charset that will normally be used in postings in this hierarchy.

Comment: The complete set of charset names is defined by [RFC2277] and the IANA Character Set registry [IANA-CS]. The charsets that are not the preferred charsets are enclosed in parentheses.

Example: Charset: ISO-8859-1  
(for the hierarchy de)

Used for: newsgroup  
Mandatory: no  
Repeatable: yes  
Description: Charset that will normally be used in  
postings in this group.  
Comment: The complete set of charset names is defined by  
[RFC2277] and the IANA Character Set registry  
[IANA-CS]. The charsets that are not the preferred  
charsets are enclosed in parentheses.  
Example: Charset: ISO-8859-9  
Charset: ISO-8859-1  
(for the newsgroup bln.kultur.tuerkisch)

#### Encoding

Header: Encoding

Used for: hierarchy  
Mandatory: no  
Inheritable: yes  
Repeatable: yes  
Description: Encoding for this hierarchy according to MIME [RFC2045].  
Comment: This is the media type used in this hierarchy; a list of  
registered media types can be found at [IANA-MT]. The  
encodings not preferred are enclosed in parentheses.  
Example: Encoding text/plain

Used for: newsgroup  
Mandatory: no  
Repeatable: yes  
Description: Encoding for this newsgroup according to MIME [RFC2045].  
Comment: This is the media type used in this newsgroup; a list of  
registered media types can be found at [IANA-MT]. The  
encodings not preferred are enclosed in parentheses.  
Example: Encoding: text/plain

## Type of newsgroup

Header: Newsgroup-Type

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: yes

Description: Default newsgroup type in this hierarchy.

Comment: This header has no concrete meaning for a hierarchy but is used for the inheritance to newsgroups in the hierarchy.

Specification of the types can be found in Section 6.5.

Example: Newsgroup-Type: Discussion  
(for the hierarchy de)

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: Type of newsgroup.

Comment: Specification of the types can be found in Section 6.5.

Example: Newsgroup-Type: Announce  
(for de.admin.news.announce)

## Type of hierarchy

Header: Hier-Type

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: yes

Description: Type of hierarchy.

Comment: Specification of the types can be found in Section 6.6.

Example: Hier-Type: Regional  
(for hierarchy bln)

## Regional or Organizational Area

Header: Area

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: yes

Description: Description of the geographical region or organization of this hierarchy.

Comment: This code is useful when the hierarchy type (Hier-Type) is "Regional" or "Organization".

Example: Area: Grossraum Berlin  
(for the hierarchy bln)

## Name length of group names

Header: Name-Length

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: no

Description: Maximum length of a newsgroup name.

Example: Name-Length: 72  
(for the hierarchy bln)

## Component length of group names

Header: Comp-Length

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: no

Description: Maximum length of a single component in the newsgroup name.

Example: Comp-Length: 14  
(for the hierarchy de)

## Article length

Header: Article-Length

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: no

Description: Maximum length of an article in bytes.

Comment: This header has no concrete meaning for a hierarchy but is used for the inheritance to newsgroups in the hierarchy.

Example: Article-Length: 50000

Used for: newsgroup

Mandatory: no

Repeatable: no

Description: Maximum length of an article in bytes.

Example: Article-Length: 50000

## Date of creation

Header: Date-Create

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: no

Description: Creation date of a hierarchy; can even be in the future.

Comment: The format is the same as in the DATE command.

Example: Date-Create: 19970330101514

Used for: newsgroup

Mandatory: no

Repeatable: no

Description: Creation date of a newsgroup; can even be in the future.

Comment: The format is the same as in the DATE command.

Example: Date-Create: 19970330101514

## Date of removal

Header: Date-Delete

Used for: hierarchy

Mandatory: no

Inheritable: yes

Repeatable: no

Description: Date of removal of a hierarchy; can even be in the future.

Comment: The format is the same as in the DATE command.

Example: Date-Delete: 19970330101514

Used for: newsgroup

Mandatory: no

Repeatable: no

Description: Date of removal of a newsgroup; can even be in the future.

Comment: The format is the same as in the DATE command.

Example: Date-Delete: 19970330101514

## Successor

Header: Replacement

Used for: hierarchy

Mandatory: no

Inheritable: no

Repeatable: yes

Description: Name of the hierarchy that replaced a removed hierarchy if status is "Hierarchy-Obsolete" or will replace a hierarchy if the date of removal is in the future.

Example: Replacement: de  
(for the hierarchy sub)

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: Name of the newsgroup or newsgroups that will replace a removed newsgroup if status is "Group-Removed" or will replace the newsgroup if the date of removal is in the future.

Example: Replacement: bln.markt.arbeit  
(for bln.jobs)

## Source

Header: Source

Used for: hierarchy  
 Mandatory: no  
 Inheritable: yes  
 Repeatable: no  
 Description: Pointer to an organization or person responsible for this hierarchy. SHOULD be a URL or an email address.  
 Example: Source: <http://www.dana.de.example/mod/>  
 (for the hierarchy de)

E: This is for tracking the maintainer of a hierarchy.

## Control PGP key

Header: Ctl-PGP-Key

Used for: hierarchy  
 Mandatory: no  
 Inheritable: yes  
 Repeatable: yes  
 Description: PGP key (with additional information: key owner, key-id, etc.) of the sender of control messages in this hierarchy.  
 Comment: The exact format is described in Section 6.7.  
 Example: Ctl-PGP-Key:  
 U de.admin.news.announce  
 B 1024  
 I D3033C99  
 L <http://www.dana.de.example/mod/pgp/dana.asc>  
 L <ftp://ftp.isc.org.example/pub/pgpcontrol/PGPKEYS.gz>  
 F 5B B0 52 88 BF 55 19 4F 66 7D C2 AE 16 26 28 25  
 V 2.6.3ia  
 K-----BEGIN PGP PUBLIC KEY BLOCK-----  
 K-Version: 2.6.3ia  
 K-  
 K-mQCNEALZ+Xfm/WDCEMXM48gK1PlKG6TkV3SLbXt4CnzpGM0tOMa  
 K-HjlHqMlwEGUHD5hw/BL/heR5Tq+C5IEyXQQmYwkrgeVFMOz/rAQ  
 [...]  
 K-SDw+iQgAAAtN6zrYOhHFBp+  
 K-VpvRovMz+lSOy9Zcsbs+5t8Pj9ZVAQyfxBkqD5A=  
 K-=Xwgc  
 K -----END PGP PUBLIC KEY BLOCK-----



## Moderator's PGP key

Header: Mod-PGP-Key

Used for: newsgroup

Mandatory: no

Repeatable: yes

Description: Public PGP key (with additional information: key owner, key-id, etc.) of this newsgroup's moderator.

Comment: The exact format is described in Section 6.7

Example: See Section 6.7.

## 6.4. Status Indicators

The status indicator uniquely determines the status of a hierarchy or newsgroup. The indicator is case insensitive.

Indicator	Type	Description
Complete	hierarchy	Authorized, complete known hierarchy
Incomplete	hierarchy	Not completely known hierarchy (like free.*)
Obsolete	hierarchy	Obsolete hierarchy; should contain only newsgroups with status "Removed"
Unknown	hierarchy	No information available; unknown hierarchy
Unmoderated	newsgroup	Posting allowed; unmoderated
Readonly	newsgroup	Posting not allowed
Moderated	newsgroup	Moderated group; articles must be sent to the moderator
Removed	newsgroup	Deleted or renamed newsgroup; no posting or transport
Unknown	newsgroup	Unknown group; no information available

## 6.5. Newsgroup Types

A Newsgroup Type is a comprehensive overview about some characteristics of a newsgroup, being a test group, a binary group, or some other kind. The Newsgroup Type is case insensitive.

Type	Meaning
Discussion	Discussion (text postings)
Binary	(Encoded) binary postings
Sources	Source postings (e.g., comp.unix.sources)
Announce	Announcements, press releases, RfD/CfV
Test	Test postings, sometimes reflectors (e.g., de.test)

Robots	Automatic postings (like the former comp.mail.maps)
Experiment	Experimental, other

## 6.6. Hierarchy Types

To describe a hierarchy, the following Hierarchy Types are used. These Types are used to mark some properties of a news hierarchy. They are case insensitive.

Type	Meaning
Global	International, global hierarchy (e.g., the hierarchies comp, de, rec)
Regional	Regional hierarchy (e.g., the hierarchies ba, bln, tor)
Alt	Alternative hierarchy, simpler rules for creating a group, no formal structure (e.g., the hierarchy alt)
Non-commercial	Only for personal use; commercial use is prohibited (e.g., the hierarchy de)
Commercial	Commercial use permitted (e.g., the hierarchy biz)
Organization	Hierarchy bound to an organization (e.g., the hierarchy gnu)

## 6.7. PGP Keys

PGP keys for Ctrl-PGP-Key and Mod-PGP-Key are transmitted in the following structure:

```
PGP-answer = "V" SP Version CRLF
            "U" SP User-ID CRLF
            "B" SP Bits CRLF
            "I" SP Key-ID CRLF
            "F" SP Finger CRLF
            *("L" SP Location CRLF)
            *("K-" Keyblock CRLF)
            "K" SP Keyblock CRLF
```

```
Version    = text
User-ID    = text
Bits       = text
Key-ID     = text
Finger     = text
Location   = text
Keyblock   = text
```

Key	Name	Mandatory	Description
---	-----	-----	-----
K	Keyblock	yes	Public key block in ASCII armor format [RFC2440]
V	Version	yes	PGP-Version
U	User-ID	no	Key user id
B	Bits	no	Number of bits
I	Key-ID	no	Key id, without leading "0x"
F	Finger	no	Fingerprint
L	Location	no	URL that points to the public key
---	-----	-----	-----

A hyphen following the code indicates that the block is continued on the next line. In the last message row, there MUST be white space after the code; this is also true for a single line code.

#### Example

```
<-- HIER de
--> 611 Data coming
    Name: de
    Status: Hierarchy
    [...]
    Ctl-PGP-Key:
    U de.admin.news.announce
    B 1024
    I D3033C99
    L http://www.dana.de.example/mod/pgp/dana.asc
    L ftp://ftp.fu-berlin.de.example/unix/news/pgpcontrol/PGPKEYS.gz
    F 5B B0 52 88 BF 55 19 4F 66 7D C2 AE 16 26 28 25
    V 2.6.3ia
    K-----BEGIN PGP PUBLIC KEY BLOCK-----
    K-Version: 2.6.3ia
    K-
    K-mQCNAzGeB/YAAAEALZ+Xfm/WDCEMXM48gK1PlKG6TkV3SLbXt4CnzpGMtOM
    K-HjlHaU6Xco5ijAuqMlwEGUHD5hw/BL/heR5Tq+C5IEyXQQmYwkrgeVFMO/rA
    [...]
    K-SDw+Id0JPF09AWOiQgAAtn6zrYOhHFBp+68h9k674Yg9IHqj3BWdRjJF6PKo
    K-VpvRovMz+lSOy9Zcsbs+5t8Pj9ZVAQyfxBkqD5A=
    K-=Xwgc
    K -----END PGP PUBLIC KEY BLOCK-----
    [...]
    .
```

## 7. Specification of the NAS Protocol (UDP)

UDP is intended for reading programs (news readers); it is not in the scope of this document. The use of UDP for NAS will be described in a separate paper.

## 8. IANA Considerations

The IANA has registered the application/nasdata media type as defined by the following information:

Media type name: application  
Media subtype name: nasdata  
Required parameters: none  
Optional parameters: level

The NAS protocol level number for the enclosed NAS data package. If not present, the protocol level defaults to 1.

Encoding scheme: NAS data is plain text; no special encodings are needed.

Security considerations: see below

## 9. Security Considerations

Security issues are only addressed in respect to server-server communication in this protocol level. Username and password combinations in the GETA and GETP commands can be used to make sure that connections are only accepted from authorized clients. PGP keys according to [RFC2440] are used to sign NAS data in server-server communication in order to validate that the data is authentic and has not been tampered with.

Every server does have the possibility (in both server-server and server-client communication) to deny some commands or the whole connection according to the client's IP number.

No mechanisms are defined in the current protocol level to allow a client to validate that it is talking to a legitimate server or that the data it receives is authentic.

A stronger authentication scheme will be provided in a higher protocol level.

## 10. Response Codes (Overview)

Code	Description
-----	-----
100	Command overview, Information, command description (HELP)
101	Information about connection, client and server (INFO)
200	Greeting message (Connection Setup)
201	Termination of the connection (QUIT)
202	Returns current protocol level (VERS)
213	Valid data at the client side (GETP)
215	The client already has the current data (GETA)
300	Time in UTC (DATE)
302	Answer to a successful request (VERS)
400	Indicates that the server is not giving any information (INFO)
401	Permission denied (LIST, LSTR, HIER, DATA)
402	Requested level too high; falling back to lower level (VERS)
404	Server currently out of service (Connection Setup)
410	Indicates that the server is not giving any information (HELP)
411	No hierarchy with that name (GETP, GETA)
430	Permission denied (GETP, GETA)
434	Client has no permission to talk to server (Connection Setup)
510	Syntax error
511	Internal error (TIME)
513	Line too long
519	Unknown command
610	Regular answer with all requested data (LIST, LSTR)
611	Regular answer with all requested data (HIER)
612	Regular answer with all requested data (DATA)
613	hierarchy data (GETP)
615	Regular answer with all requested data (GETA)
----	-----

## 11. Data Headers for DATA and HIER Commands (Overview)

Header	Mandatory	Use	Multiple	Description
-----	-----	---	-----	-----
Name	yes	H/N	no	Name of a hierarchy or newsgroup (Start of a new data block)
Status	yes	H/N	no	Status of hierarchy or newsgroup
Serial	no	H/N	no	Revision of hierarchy /newsgroup data
Followup	no	N	no	Group for followup
Description	no	H/N	no	Short description of a hierarchy/newsgroup
Charter	no	H/N	yes	Charter-URL
Netiquette	no	H/N	yes	Netiquette-URL

FAQ	no	N	yes	FAQ-URL
Rules	no	H	yes	Administration rules URL
Ctl-Send-Adr	no	H	yes	Control email
Ctl-Newsgroup	no	H	yes	Control newsgroup
Mod-Wildcard	no	H	no	Moderator wildcard
Mod-Sub-Adr	no	N	no	Submission address
Mod-Adm-Adr	no	N	yes	Moderator's address (email)
Mod-Group-Info	no	N	yes	Info-URL
Language	no	H/N	yes	Language
Charset	no	H/N	yes	Charset
Encoding	no	H/N	yes	Encoding
Newsgroup-Type	no	H/N	yes	Type of newsgroup
Hier-Type	no	H	yes	Type of hierarchy
Area	no	H	yes	Regional or organizational area
Name-Length	no	H	no	Total length of group names
Comp-Length	no	H	no	Component length of group names
Article-Length	no	H	no	Article length
Date-Create	no	H/N	no	Date of creation
Date-Delete	no	H/N	no	Date of removal
Replacement	no	H/N	yes	Successor
Source	no	H	yes	Source of data
Ctl-PGP-Key	no	H	yes	Control PGP key
Mod-PGP-Key	no	N	yes	Moderator's PGP key
-----	-----	---	-----	-----

N: Newsgroup, H: Hierarchy

## 12. References

### 12.1. Normative References

- [IANA-CS] IANA: Character Sets,  
<<http://www.iana.org/assignments/character-sets>>.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

- [RFC2440] Callas, J., Donnerhackle, L., Finney, H., and R. Thayer, "OpenPGP Message Format", RFC 2440, November 1998.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.
- [RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

## 12.2. Informative References

- [IANA-MT] IANA: Media Types, <<http://www.iana.org/assignments/>>.
- [IANA-PN] IANA: Assigned Port Numbers, <<http://www.iana.org/assignments/port-numbers>>.
- [RFC1305] Mills, D., "Network Time Protocol", RFC 1305, University of Delaware, March 1992.
- [SON1036] H. Spencer, "News Article Format and Transmission", A Draft for an RFC 1036 Successor, <<ftp://zoo.toronto.edu/pub/news.txt.Z>>.
- [USEFOR] USEFOR Working Group, "News Article Format", Work in Progress.

## Acknowledgement

This work has been supported by the German Academic Network Organization (DFN-Verein) with funds from the German Federal Ministry of Education and Research (Bundesministerium fuer Bildung und Forschung).

## Authors' Addresses

Philipp Grau  
Vera Heinau  
Heiko Schlichting  
Robert Schuettler

Freie Universitaet Berlin  
ZEDAT  
Fabeckstr. 32  
14195 Berlin  
Germany

Phone: +49 30 838-74707  
Fax: +49 30 838-56721

EMail: [nas@fu-berlin.de](mailto:nas@fu-berlin.de)  
URL: <http://nas.fu-berlin.de/>



## Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78 and at [www.rfc-editor.org/copyright.html](http://www.rfc-editor.org/copyright.html), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

