Netnews Architecture and Protocols  
draft-allbery-usefor-usepro-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on June 3, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).
Abstract

This document defines the architecture of Netnews systems and specifies the correct manipulation and interpretation of Netnews articles by software which originates, distributes, stores, and displays them. It also specifies the requirements that must be met by any protocol used to transport and serve Netnews articles.

Table of Contents

1. Introduction ................................................. 4
   1.1. Basic Concepts ........................................... 4
   1.2. Scope .................................................. 4
   1.3. Requirements Notation ..................................... 4
   1.4. Definitions ............................................. 4
2. Transport ................................................... 6
3. Duties of Agents ............................................. 7
   3.1. General Principles ........................................ 7
   3.2. Path Identities ......................................... 8
   3.3. Duties of a Posting Agent ................................ 9
      3.3.1. Proto-articles ...................................... 9
      3.3.2. Reinjection of Articles ............................. 10
      3.3.3. Followups .......................................... 10
      3.3.4. Construction of the References Header Field ....... 11
   3.4. Duties of an Injecting Agent ............................ 12
      3.4.1. Forwarding Messages to a Moderator ................. 14
   3.5. Duties of a Relaying Agent .............................. 15
      3.5.1. Path Header Field Example ........................... 17
   3.6. Duties of a Serving Agent ............................... 18
   3.7. Duties of a Reading Agent ............................... 19
   3.8. Duties of a Moderator ................................... 19
   3.9. Duties of a Gateway ..................................... 21
      3.9.1. Duties of an Outgoing Gateway ...................... 22
      3.9.2. Duties of an Incoming Gateway ...................... 23
      3.9.3. Gateway Example ................................... 25
4. Media Types .................................................. 26
   4.1. application/news-transmission ........................... 26
   4.2. application/news-groupinfo .............................. 27
   4.3. application/news-checkgroups ............................ 28
5. Control Messages ............................................ 30
   5.1. Authentication and Authorization ......................... 30
   5.2. Group Control Messages ................................ 31
5.2.1. The newgroup Control Message ................. 31
5.2.2. The rmgroup Control Message ................. 33
5.2.3. The checkgroups Control Message ............ 33
5.3. The cancel Control Message .................... 34
5.4. The Supersedes Header Field ................... 35
5.5. The ihave and sendme Control Messages ........... 35
5.6. Obsolete Control Messages ..................... 36
6. Security Considerations ............................ 37
   6.1. Compromise of System Integrity ................ 37
   6.2. Denial of Service .............................. 38
   6.3. Leakage ...................................... 39
7. IANA Considerations ................................. 40
8. References .......................................... 41
   8.1. Normative References .......................... 41
   8.2. Informative References ........................ 41
Appendix A. Changes to the Existing Protocols ........ 42
Appendix B. Acknowledgements ........................ 43
Authors' Addresses ................................... 44
Intellectual Property and Copyright Statements ........ 45
1. Introduction

1.1. Basic Concepts

"Netnews" is a set of protocols for generating, storing and retrieving news "articles" whose format is defined in [USEFOR], and for exchanging them amongst a readership that is potentially widely distributed. It is organized around "newsgroups", with the expectation that each reader will be able to see all articles posted to each newsgroup in which he participates. These protocols most commonly use a flooding algorithm which propagates copies throughout a network of participating servers. Typically, only one copy is stored per server, and each server makes it available on demand to readers able to access that server.

"Usenet" is a particular worldwide publicly accessible network based on the Netnews protocols. It is only one such possible network; there are deployments of the Netnews protocols other than Usenet (such as ones internal to particular organizations). This document discusses the more general Netnews architecture and protocols.

1.2. Scope

This document defines the architecture of Netnews systems and specifies the correct manipulation and interpretation of Netnews articles by software which originates, distributes, stores, and displays them. It addresses protocol issues that are independent of transport protocols such as NNTP [RFC3977] and specifies the requirements Netnews places on those underlying transport protocols.
It also specifies the handling of control messages.

The format and syntax of Netnews articles are specified in [USEFOR], which should be read in conjunction with this document.

Comments are solicited and should be addressed to the Usenet Format Working Group at ietf-usefor@imc.org or to the editor.

1.3. Requirements Notation

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].

1.4. Definitions

Any term used in this document that is defined in Section 1.5 of [USEFOR] is used with the definition given there. In addition, the following terms will be used:

A "hierarchy" is the set of all newsgroups whose names share a first <component> (as defined in Section 3.1.4 of [USEFOR]). A "sub-hierarchy" is the set of all newsgroups whose names share several initial components.

A "news server" is further distinguished into the roles of "injecting agent", "relaying agent", and "serving agent". An "injecting agent" accepts a proto-article with the goal of distributing it to relaying and serving agents and hence to readers. A "relaying agent" accepts articles from other relaying agents or injecting agents and distributes them to other relaying agents or serving agents. A "serving agent" receives an article from a relaying agent or injecting agent and makes it available to readers.

A "user agent" is further distinguished into the roles of "posting agent" and "reading agent". A "posting agent" is software which assists in the preparation of a proto-article and then passes it to an injecting agent. A "reading agent" is software which retrieves articles from a serving agent for presentation to a reader.

"Injecting" an article is the processing of a proto-article by an injecting agent. Normally this action is done once and only once for
"Reinjecting" an article is passing an already-injected article to an injection agent.

A "gateway" is software which receives news articles and converts them to messages of some other kind (such as [RFC2822] mail messages), receives messages of some other kind and converts them to news articles, or conveys articles between two separate Netnews networks.

2. Transport

The exact means used to transmit articles from one agent to another is not specified. NNTP [RFC3977] is the most common transport mechanism for Netnews networks. Other methods in use include the UUCP protocol [RFC0976] (extensively used in the early days of Usenet) and physically delivered magnetic and optical media. Any mechanism may be used in conjunction with this protocol provided that it can meet the requirements specified here.

Transports for Netnews articles MUST treat news articles as uninterpreted sequences of octets, excluding the values 0 (which may not occur in Netnews articles) and 13 and 10 (which MUST only appear in Netnews articles as a pair in that order and which together denote a line separator). These octets are the US-ASCII [ASCII] characters
NOTE: This corresponds to the range of octets permitted in MIME 8bit data [RFC2045]. Transports for Netnews are not required to support transmission of MIME binary data.

In particular, transports MUST convey all header fields (including header fields within message/rfc822 objects in article bodies) unmodified even if they contain octets in the range 128 to 255. Furthermore, transports for relaying and serving agents MUST, and transports for other agents SHOULD, convey lines even if they exceed 998 characters in length, especially in article bodies. (This requirement is stricter than MIME 8bit data.) These requirements include the transport paths between posting agents, injecting agents, serving agents, and reading agents.

3. Duties of Agents

The following section specifies the duties of the agents involved in the creation, relaying, and serving of Netnews articles. This protocol is described by following the life of a typical Usenet article: it is prepared by a posting agent, given to an injecting agent, transferred through one or more relaying agents, accepted by a serving agent, and finally retrieved by a reading agent. Articles
submitted to moderated groups go through an additional process, which is described separately. Finally, the additional duties and requirements of a gateway are discussed.

At each step, each agent has a set of checks and transformations of the article that it is required to perform. These are described as sequences of steps to be followed, but it should be understood that it is the effect of these sequences that is important, and implementations may use any method that gives rise to the same effect.

Many news servers combine the functions of injecting agent, relaying agent, and serving agent in a single software package. For the purposes of this specification, such combined agents should conceptually be treated as an injecting agent which sends articles to a serving agent and optionally a relaying agent. The requirements of all three agents MUST be still met when the news server is performing the functions of those agents.

Control messages may have additional effects than those described below on news servers which accept them. Those effects are described in Section 5.

3.1. General Principles

There are two important principles that news implementors and administrators need to keep in mind. The first is the well-known Internet Robustness Principle:

Be liberal in what you accept, and conservative in what you send.

As applied to Netnews, this primarily means that unwanted or non-compliant articles SHOULD be rejected as early as possible, but once they are in general circulation, relaying and serving agents may wish to accept them where possible rather than lose information. Posting agents and injecting agents SHOULD therefore be maximally strict in their application of both this protocol and [USEFOR], and reading agents SHOULD be robust in the presence of violations of the Netnews article format where possible.
derived from a much older code of practice, the Hippocratic Oath (we may thus call this the Hippocratic Principle):

First, do no harm.

It is vital to realize that decisions which might be merely suboptimal in a smaller context can become devastating mistakes when amplified by the actions of thousands of hosts within a few minutes.

No Netnews agent is ever required to accept any article. It is common for injecting, relaying, and serving agents to reject well-formed articles for reasons of local policy (such as not wishing to carry a particular newsgroup or attempting to filter out unwanted articles). This document specifies how articles are to be treated if they are accepted and specifies some cases where they must be rejected, but an agent MAY always reject any article for other reasons than those stated here.

A primary goal of the Netnews protocol is to ensure that all readers receiving a particular article (as uniquely identified by the content of its Message-ID header field) see the identical article, apart from allowable divergence in trace headers and local metadata. Accordingly, agents (other than moderators) MUST NOT modify articles in ways other than described here. Unacceptable articles MUST be rejected rather than corrected.

3.2. Path Identities

All news server components (injecting agents, relaying agents, and serving agents) MUST identify themselves, when processing an article, by prepending their <path-identity> (defined in Section 3.1.5 of [USEFOR]) to the Path header field. Injecting agents MUST also use the same identity in Injection-Info header fields they add, and serving and relaying agents SHOULD use the same identity in any Xref header fields they add.

The <path-identity> used by an agent may be chosen via one of the following methods (in decreasing order of preference):

1. The fully-qualified domain name (FQDN) of the system on which the agent is running.

2. A fully-qualified domain name (FQDN) within a domain affiliated with the administrators of the agent and guaranteed to be unique by the administrators of that domain. For example, the uniqueness of server.example.org could be guaranteed by the administrator of example.org even if there is no DNS record for
server.example.org itself.

3. Some other (arbitrary) name in the form <path-nodot> believed to be unique and registered at least with all the other news servers to which that relaying agent or injecting agent sends articles. This option SHOULD NOT be used unless the earlier options are unavailable or unless the name is of longstanding usage.

Path identities are case sensitive. To avoid unintentional variation in a news server's identity, the <path-identity> SHOULD be all lowercase.

3.3. Duties of a Posting Agent

A posting agent is the component of a user agent that assists a poster in creating a valid proto-article and forwarding it to an injecting agent.

Posting agents SHOULD ensure that proto-articles they create are valid according to [USEFOR] and any other applicable policies. They MUST NOT create any Injection-Date or Injection-Info header fields; these headers will be added by the injecting agent.

Contrary to [RFC2822], which implies that the mailbox or mailboxes in the From header field should be that of the poster or posters, a poster who does not, for whatever reason, wish to use his own mailbox MAY use any mailbox ending in the top level domain ".invalid" [RFC2606].

Posting agents meant for use by ordinary posters SHOULD reject any attempt to post an article which cancels or Supersedes another article of which the poster is not the author or sender.

3.3.1. Proto-articles

A proto-article is an article in the format used by a posting agent for offering to an injecting agent. It may omit certain header fields which can be better-supplied by the injecting agent and will not contain header fields that are added by the injecting agent. A proto-article is only for transmission to an injecting agent and SHOULD NOT be transmitted to any other agent.

A proto-article has the same format as a normal article except that the Injection-Date, Injection-Info, and Xref header fields MUST NOT be present; the Path header field MUST NOT contain a "POSTED" <diag-keyword>; and any of the following mandatory header fields MAY be
omitted: Message-ID, Date, and Path. In all other respects, a proto-article MUST be a valid Netnews article. In particular, the header fields which may be omitted MUST NOT be present with invalid content.

If a posting agent intends to offer the same proto-article to multiple injecting agents, the header fields Message-ID and Date MUST be present and identical in all copies of the proto-article.

3.3.2. Reinjection of Articles

A given article SHOULD be processed by an injecting agent once and only once. The Injection-Date or Injection-Info header fields are added by an injecting agent and are not permitted in a proto-article. Their presence (or the presence of other unstandardized or obsolete trace headers such as NNTP-Posting-Host, NNTP-Posting-Date, or X-Trace) indicates that the proto-article is instead an article and has already been processed by an injecting agent. A posting agent SHOULD normally reject such articles.

In the exceptional case that an article needs to be reinjected for some reason (such as transferring an article from one Netnews to another where those networks have no relaying agreement), the posting agent doing the reinjection MUST convert the article back into a proto-article before passing it to an injecting agent (such as by renaming the Injection-Info and Injection-Date header fields and removing any Xref header field) and MUST perform the date checks on the existing Injection-Date or Date header fields that would otherwise be done by the injecting agent.

Reinjecting articles may cause loops, loss of trace information, and other problems and should only be done with care and when there is no available alternative. A posting agent that does reinjection is a limited type of gateway and as such is subject to all of the requirements of an incoming gateway in addition to the requirements of a posting agent.

3.3.3. Followups

A followup is an article that contains a response to the contents of an earlier article, its precursor. In addition to its normal duties, a posting agent preparing a followup is also subject to the following
requirements. Wherever in the following it is stated that by default a header field is said to be inherited from one of those header fields in the precursor, it means that its initial content is to be a copy of the content of that precursor header field (with changes in folding permitted). However, posters MAY then override that default before posting.

Despite the historic practice of some posting agents, the Keywords header field SHOULD NOT be inherited by default from the precursor article.

1. If the Followup-To header field of the precursor article consists of "poster", the followup MUST NOT be posted by default but by default is to be emailed to the address given in the precursor's Reply-To or From header field following the rules for an email reply [RFC2822]. This action MAY be overridden by the poster, in which case the posting agent should continue as if the Followup-To header field in the precursor did not exist.

2. The Newsgroups header field SHOULD by default be inherited from the precursor's Followup-To header field if present, and otherwise from the precursor's Newsgroups header field.

3. The Subject header field SHOULD by default be inherited from that of the precursor. The case-sensitive string "Re: " (including the space after the colon) MAY be prepended to the content of its Subject header field unless it already begins with that string.

   NOTE: Prepending "Re: " serves no protocol function and hence is not required, but it is widely expected and not doing so would be surprising.

4. The Distribution header field SHOULD by default be inherited from the precursor's Distribution header field, if present.

5. The followup MUST have a References header field referring to its precursor constructed in accordance with Section 3.3.4.

3.3.4. Construction of the References Header Field

The following procedure is to be used whenever some previous article
(the "parent") is to be referred to in the References header field of a new article, whether because the new article is a followup and the parent is its precursor or for some other reason.

The content of the new article's References header field MUST be formed from the content of the parent's References header field if present and the content of the Message-ID header field of the parent. If the parent had a References header, FWS as defined in [USEFOR] MUST be added between its content and the Message-ID header field content.

If the resulting References header field would, after unfolding, exceed 998 characters in length (including its field name but not the final CRLF), it SHOULD be trimmed (and otherwise MAY be trimmed). Trimming means removing any number of message identifiers from its content, except that the first message identifier and the last two

3.4. Duties of an Injecting Agent

An injecting agent takes a proto-article from a posting agent and either forwards it to a moderator or passes it to a relaying or serving agent or agents. An injecting agent bears the primary responsibility for ensuring that any article it injects conforms with the rules of the Netnews standards. The administrator of an injecting agent is also expected to bear some responsibility towards the rest of the Netnews network to which it is connected for the articles the injecting agent accepts.

Injecting agents, when rejecting articles, are encouraged to communicate the reason for rejection to the posting agent using whatever facility is provided by the underlying transport. The injecting agent is in a unique position to communicate the reason for rejection; relaying agents and serving agents normally have to reject messages silently. The injecting agent therefore bears much of the burden of diagnosing broken posting agents or communicating policy
violations to posters.

An injecting agent MUST have available a list (possibly empty) of moderated groups for which it accepts articles and the corresponding submission addresses. It SHOULD have available a list of valid newsgroups to catch articles not posted to a valid newsgroup and therefore likely to be silently discarded by relaying and serving agents. Usually, an injecting agent is deployed in conjunction with a serving agent and maintains these lists based on control messages received by the serving agent.

An injecting agent processes proto-articles as follows:

1. It SHOULD verify that the article is from a trusted source (by, for example, relying on the authorization capability of the underlying transport used to talk to the posting agent).

2. It MUST reject any proto-article that does not have the proper mandatory header fields for a proto-article; that has Injection-Date, Injection-Info, or Xref header fields; that has a Path header field containing the "POSTED" <diag-keyword>; or that is not syntactically valid as defined by [USEFOR]. It SHOULD reject any proto-article which contains a header field deprecated for Netnews. It MAY reject any proto-article that contains trace header fields indicating that it was already injected by an injecting agent that did not add Injection-Info or Injection-Date.

3. It SHOULD reject any article whose Date header field is more than 24 hours into the future (and MAY use a margin less than 24 hours). It SHOULD reject any article whose Date header appears to be stale (more than 72 hours into the past, for example, or too old to still be recorded in the database of a relaying agent the injecting agent will be using) since not all news servers support Injection-Date.

4. It SHOULD reject any proto-article whose Newsgroups header field does not contain at least one <newsgroup-name> for a valid group, or containing a <newsgroup-name> reserved for specific purposes by Section 3.1.4 of [USEFOR] unless that specific purpose or local agreement applies to the proto-article being
processed. Crossposting to unknown newsgroups is not precluded provided that at least one of the newsgroups in the Newsgroups header is valid.

5. The Message-ID and Date header fields with appropriate contents MUST be added when not present in the proto-article.

6. The injecting agent MUST NOT alter the body of the article in any way (including any change of Content-Transfer-Encoding). It MAY add other header fields not already provided by the poster, but injecting agents are encouraged to use the Injection-Info header for such information and to minimize the addition of other headers. It SHOULD NOT alter, delete, or reorder any existing header field except the Path header.

7. If the Newsgroups header contains one or more moderated groups and the proto-article does not contain an Approved header field, the injecting agent SHOULD forward it to a moderator as specified in Section 3.4.1. If the article cannot be forwarded to a moderator for some reason, it MUST be rejected. This forwarding MUST be done after adding the Message-ID and Date headers if required, and before adding the Injection-Info and Injection-Date headers.

8. Otherwise, a Path header field with a <tail-entry> MUST be added if not already present.

9. The injecting agent SHOULD then prepend the <path-identity> of the injecting agent followed by "!.POSTED", optionally "." and the FQDN or IP address of the source, and a further "!" to the content of the Path header field. If the injecting agent does not support the use of <diag-keyword>, it MUST instead prepend its <path-identity> followed by "!"; one or the other of these mechanisms MUST be used.

10. The relaying agent MAY fold the Path header field by inserting FWS immediately after the <path-identity> it added.

11. An Injection-Info header field SHOULD be added identifying the source of the article and possibly other trace information as described in Section 3.2.8 of [USEFOR].
12. The injecting agent MUST then add an Injection-Date header field containing the current date and time.

13. Finally, the injecting agent forwards the article to one or more relaying agents, and the injection process is complete.

3.4.1. Forwarding Messages to a Moderator

An injecting agent MUST forward the proto-article to the moderator of the leftmost moderated group listed in the Newsgroups header field, customarily via email. There are two standard ways in which it may do this:

1. The complete proto-article is encapsulated, header fields and all, within the email. This SHOULD be done by creating an email message with a Content-Type of application/news-transmission with the usage parameter set to "moderate". The body SHOULD NOT contain any content other than the message. This method has the advantage of removing any possible conflict between Netnews and email header fields and any changes to those fields during transport through email.

2. The proto-article is sent as an email with the addition of any header fields (such as a To header field) required for an email. The existing Message-ID header field SHOULD be retained.

Although both of these methods have been used in the past and the first has clear technical advantages, the second is in more common use and many moderators are not prepared to deal with messages in the first format. Accordingly, the first method SHOULD NOT be used unless the moderator to which it is being forwarded is known to be able to handle this method.

NOTE: Deriving the email address of the moderator of a group is outside the scope of this document. It is worth mentioning, however, that a common method is to use a forwarding service that handles submissions for many moderated groups. For maximum compatibility with existing news servers, such forwarding services generally form the submission address for a moderated group by replacing each "." in the <newsgroup-name> with "-" and then using
that value as the <local-part> of a <mailbox> formed by appending a set domain.

Forwarding proto-articles to moderators via email is the most general method and most common in large Netnews networks such as Usenet, but any means of forwarding the article that preserves it without injecting it MAY be used. For example, if the injecting agent has access to a database used by the moderator to store proto-articles awaiting processing, it may place the proto-article directly into that database. Such methods may be more appropriate for smaller Netnews networks.

3.5. Duties of a Relaying Agent

A relaying agent accepts injected articles from injecting and other relaying agents and passes them on to relaying or serving agents. To avoid bypass of injecting agent policies and forgery of Path and Injector-Info headers, relaying agents SHOULD accept articles only from trusted agents.

An article SHOULD NOT be relayed unless the sending agent has been configured to supply and the receiving agent to receive at least one of the <newsgroup-name>s in its Newsgroups header field and at least one of the <dist-name>s in its Distribution header field (if present). Exceptionally, control messages creating new newsgroups (newgroup control messages) SHOULD be relayed if the sending agent has been configured to supply and the receiving agent to receive the newsgroup affected by the control message, even if that newsgroup does not currently exist and even if the control message does not contain that group in its Newsgroups header field.

In order to avoid unnecessary relaying attempts, an article SHOULD NOT be relayed if the <path-identity> of the receiving agent (or some known alias thereof) appears as a <path-identity> (excluding within the <tail-entry> or following a "POSTED" <diag-keyword>) in its Path header field.

A relaying agent processes an article as follows:

1. It MUST reject any article without a Newsgroups header field or Message-ID header field, or without either an Injection-Date or Date header field.
2. It MUST reject any article that has already been successfully sent to it, based on the Message-ID header field of the article. To satisfy this requirement, a relaying agent normally keeps a database of message identifiers it has already accepted.

3. It MUST examine the Injection-Date header field or, if absent, the Date header field, and reject the article if that date predates the earliest articles of which it keeps record or if that date is more than 24 hours into the future. It MAY reject articles with dates in the future with a smaller margin than 24 hours.

4. It SHOULD reject any article that does not include all the mandatory header fields. It MAY reject any article that contains header fields that do not have valid contents.

5. It SHOULD reject any article that matches an already-received cancel control message or the contents of the Supersedes header field of an accepted article, provided that the relaying agent chose (on the basis of local site policy) to honor that cancel control message or Supersedes header field.

6. It MAY reject any article without an Approved header field posted to a newsgroup known to be moderated. This practice is strongly encouraged but the information necessary to do so is not required to be maintained by a relaying agent.

7. If the relaying agent is processing an article from an injecting agent that is part of the same news server, it MAY leave the Path header field unmodified. Otherwise, it SHOULD compare the expected <path-identity> of the source of the article with the leftmost <path-identity> of the Path header field's content. If those identities match, it SHOULD then prepend "!!" to that content. If those identities do not match, it SHOULD instead prepend "!.MISMATCH.", the true established <path-identity> of the source or its IP address, and a further "!". If the relaying agent is not able or willing to verify the path identity of the source of the article, it MUST prepend "!" to the Path header field's content, optionally preceded by "!.SEEN." and the FQDN, IP address, or expected <path-identity> of the source. Regardless of which method it used, the relaying agent MUST then prepend its own <path-identity> to the Path header field content.

8. If it added a <path-identity>, the relaying agent MAY fold the Path header field by inserting FWS immediately after the final
9. It MAY delete any Xref header field present and MAY add a new Xref header field with any valid content. The Xref header field is not used by relaying agents, but relaying agents that are also serving agents may generate Xref header fields for their own internal purposes.

10. Finally, it passes the article on to other relaying and serving agents to which it is configured to send articles.

Relaying agents SHOULD, where possible in the underlying transport, inform the agent that passed the article to the relaying agent if the article was rejected. Relaying agents MUST NOT inform any other external entity of the rejection of an article unless that external entity has explicitly requested that it be informed of such errors.

Relaying agents MUST NOT alter, delete, or rearrange any part of an article except for the Path and Xref header fields. They MUST NOT modify the body of articles in any way. If an article is not acceptable as-is, the article MUST be rejected rather than modified.

3.5.1. Path Header Field Example

Here is an example of a Path header field created following the rules for injecting and relaying agents.

```
Path: foo.isp.example!.SEEN.isp.example!!foo-news
   !.MISMATCH.2001:DB:0:0:8:800:200C:417A!bar.isp.example
   !!old.site.example!barbaz!!baz.isp.example
   !.POSTED.dialup123.baz.isp.example!not-for-mail
```

This article was injected by baz.isp.example as indicated by the <diag-keyword> "POSTED". The injector has recorded that it received the article from dialup123.baz.isp.example. "not-for-mail is a common <tail-entry>.

The article was relayed to the relaying agent known, at least to old.site.example, as "barbaz".

barbaz relayed it to old.site.example, which does not support <diag-keyword> and therefore used the old "!" delimiter. This indicates
that the identity of "barbaz" was not verified and may have been forged.

old.site.example relayed it to a news server using the <path-identity> of bar.isp.example and claiming (by using the "!!" <path-delimiter>) to have verified that it came from old.site.example.

bar.isp.example relayed it to foo-news which, not being convinced that it truly came from bar.isp.example, inserted the <diag-keyword> "MISMATCH" and then stated that it received the article from the IPv6 address [2001:DB8:0:0:8:800:200C:417A]. (This is not to say that bar.isp.example was not a correct <path-identity> for that source but simply that that identity did not match the expectations of foo-news.

foo-news then passed the article to foo.isp.example, which declined to validate its <path-identity> and instead appended the <diag-keyword> "SEEN" to indicate it knows the source of the article as isp.example. This may be either an expected <path-identity> or the FQDN of the system from which it received the article. Presumably foo.isp.example is a serving agent that then delivered the article to a reading agent.

baz.isp.example, bar.isp.example, and foo-news folded the Path header field.

### 3.6. Duties of a Serving Agent

A serving agent accepts articles from a relaying agent or injecting agent, stores it, and makes it available to reading agents. Articles are normally indexed by newsgroup and <article-locator> ([Section 3.2.14 of USEFOR](https://example.com)), usually in the form of a decimal number.

If the serving agent stores articles by newsgroup, control messages SHOULD NOT be stored in the newsgroups listed in the control message's Newsgroups header field. Instead, they SHOULD be stored in a newsgroup in the hierarchy "control", which is reserved for this purpose. Conventionally, control messages are stored in newsgroups named for the type of control message (such as "control.cancel" for cancel control messages).

A serving agent MUST have available a list (possibly empty) of
moderated groups for which it accepts articles so that it can reject unapproved articles posted to moderated groups. Frequently a serving agent is deployed in combination with an injecting agent and can use the same list as the injecting agent.

A serving agent processes articles as follows:

1. It MUST reject any article that does not include all the mandatory header fields or any article which contains header fields that do not have valid contents.

2. It MUST examine the Injection-Date header field or, if absent, the Date header field, and reject the article if that date predates the earliest articles of which it keeps record or if that date is more than 24 hours into the future. It MAY reject articles with dates in the future with a smaller margin than 24 hours.

3. It SHOULD reject any article that has already been successfully sent to it or that matches an already-received and honored cancel message or Supersedes header field following the same rules as a relaying agent (Section 3.5).

4. It MUST reject any article without an Approved header field posted to any newsgroup listed as moderated.

5. It MUST modify the Path header field following the same rules as for a relaying agent (Section 3.5).

6. It MUST (except when specially configured to preserve the <article-locator>s set by the sending site) remove any Xref header field from each article. It then MAY (and usually will) generate a fresh Xref header field.

7. Finally, it stores the article and makes it available for reading agents.

Serving agents MUST NOT create new newsgroups simply because an unrecognized <newsgroup-name> occurs in a Newsgroups header field. Newsgroups are normally created via control messages (Section 5.2.1).
Serving agents MUST NOT alter, delete, or rearrange any part of an article except for the Path and Xref header fields. They MUST NOT modify the body of the articles in any way. If an article is not acceptable as-is, the article MUST be rejected rather than modified.

3.7. Duties of a Reading Agent

Since a reading agent is only a passive participant in a Netnews network, there are no specific protocol requirements placed on it. See [USEAGE] for best-practice recommendations.

3.8. Duties of a Moderator

A moderator receives news articles, customarily by email, decides whether to approve them and, if so, either passes them to an injecting agent or forwards them to further moderators.

Articles are normally received by the moderator in email either encapsulated as an object of Content-Type application/news-transmission (or possibly encapsulated but without an explicit Content-Type header field), or else directly as an email already containing all the header fields appropriate for a Netnews article (see Section 3.4.1). Moderators who may receive articles via email SHOULD be prepared to accept articles in either format.

Moderators are entirely free within the Netnews protocol to accept or reject messages based on any criteria and to make arbitrary modifications to articles (both header fields and body). See [USEAGE] for best-practice recommendations. Moderators need to be aware that modifications made to articles may invalidate signatures created by the poster or previous moderators.

Moderators process articles as follows:

1. They decide whether to approve or reject an article, and if approved, make whatever modifications to the article (if any) they choose. If the article is rejected, it is normally rejected for all newsgroups to which it was posted and nothing further is done. If it is approved, the moderator proceeds with the following steps.
2. If the Newsgroups header field contains further moderated
newsgroups for which approval has not already been given, they
may either reach some agreement with the other moderators on the
disposition of the article or, more generally, add an indication
(identifying both the moderator and the name of the newsgroup)
that they approve the article and then forward it to the
moderator of the leftmost unapproved newsgroup. This forwarding
SHOULD be done following the procedure in Section 3.4.1 and MAY
be done by rotating the <newsgroup-name>s in the Newsgroups
header field so that the leftmost unapproved newsgroup is the
leftmost moderated newsgroup in that field and then posting it,
letting the injecting agent do the forwarding. However, if using
this mechanism, they MUST first ensure that the article contains
no Approved header field.

3. If the Newsgroups header field contains no further unapproved
moderated groups, they add an Approved header field (see Section
3.2.1 of [USEFOR]) identifying the moderator and, insofar as is
possible, all the other moderators who have approved the article.
The moderator who takes this step assumes responsibility for
ensuring that the article was approved by the moderators of all
moderated newsgroups to which it was posted.

4. Moderators are encouraged to retain the Message-ID header field
if it is valid, and also retain the Date header field unless it
appears to be stale (72 hours or more in the past) for reasons
understood by the moderator (such as delays in the moderation
process) in which case they MAY substitute the current date. Any
Injection-Date, Injection-Info, or Xref header fields already
present (though there should be none) MUST be removed.

5. Any Path header field MUST either be removed or truncated to only
those entries following its "POSTED" <diag-keyword>, if any.

6. The moderator then passes the article to an injecting agent,
having first observed all the duties of a posting agent.

3.9. Duties of a Gateway

A gateway transforms an article into the native message format of
another medium, or translates the messages of another medium into
news articles, or transforms articles into proto-articles for injection into a separate Netnews network. Encapsulation of a news article into a message of MIME type application/news-transmission, or the subsequent undoing of that encapsulation, is not gatewaying, since it involves no transformation of the article.

There are two basic types of gateway, the outgoing gateway that transforms a news article into a different type of message, and the incoming gateway that transforms a message from another network into a news proto-article and injects it into a Netnews network. These are handled separately below.

Transformation of an article into another medium stands a very high chance of discarding or interfering with the protection inherent in the news system against duplicate articles. The most common problem caused by gateways is loops that repeatedly reinject previously posted articles. To prevent this, a gateway MUST take precautions against loops, as detailed below.

The transformations applied to the message SHOULD be as minimal as possible while still accomplishing the gatewaying. Every change made by a gateway potentially breaks a property of one of the media or loses information, and therefore only those transformations made necessary by the differences between the media should be applied.

If bidirectional gatewaying (both an incoming and an outgoing gateway) is being set up between Netnews and some other medium, the incoming and outgoing gateways SHOULD be coordinated to avoid unintended reinjection of gated articles. Circular gatewaying (gatewaying a message into another medium and then back into Netnews) SHOULD NOT be done; encapsulation of the article SHOULD be used instead where this is necessary.

Safe bidirectional gatewaying between a mailing list and a newsgroup is far easier if the newsgroup is moderated. Posts to the moderated group and submissions to the mailing list can then go through a single point that does the necessary gatewaying and then sends the message out to both the newsgroup and the mailing list at the same time, eliminating most of the possibility of loops. Bidirectional gatewaying between a mailing list and an unmoderated newsgroup, in contrast, is difficult to do correctly and is far more fragile.
Newsgroups intended to be bidirectionally gated to a mailing list SHOULD therefore be moderated where possible, even if the moderator is a simple gateway and injecting agent that correctly handles crossposting to other moderated groups and otherwise passes all traffic.

### 3.9.1. Duties of an Outgoing Gateway

From the perspective of Netnews, an outgoing gateway is just a special type of reading agent. The exact nature of what the outgoing gateway will need to do to articles depends on the medium to which the articles are being gated. The operation of the outgoing gateway is subject to additional constraints due to the possibility of one or more corresponding incoming gateways back from that medium to Netnews, since this raises the danger of loops.

The following practices are encouraged for all outgoing gateways, regardless of whether there is known to be a related incoming gateway, both as precautionary measures and as a guideline to quality of implementation:

1. The message identifier of the news article should be preserved if at all possible, preferably as or within the corresponding unique identifier of the other medium, but if not at least as a comment in the message. This helps greatly with preventing loops.

2. The Date and Injection-Date of the news article should also be preserved if possible, for similar reasons.

3. The message should be tagged in some way so as to prevent its reinjection into Netnews. This may be impossible to do without knowledge of potential incoming gateways, but it is better to try to provide some indication even if not successful; at the least, a human-readable indication that the article should not be gated back to Netnews can help locate a human problem.

4. Netnews control messages should not be gated to another medium unless they would somehow be meaningful in that medium.
3.9.2. Duties of an Incoming Gateway

The incoming gateway has the responsibility of ensuring that all of the requirements of this protocol are met by the articles that it forms. In addition to its special duties as a gateway, it bears all of the duties and responsibilities of a posting agent as well, and additionally has the same responsibility of a relaying agent to reject articles that it has already gatewayed.

An incoming gateway MUST NOT gate the same message twice. It may not be possible to ensure this in the face of mangling or modification of the message, but at the very least a gateway, when given a copy of a message it has already gated identical except for trace header fields (like Received in Email or Path in Netnews) MUST NOT gate the message again. An incoming gateway SHOULD take precautions against having this rule bypassed by modifications of the message that can be anticipated.

News articles prepared by gateways MUST be valid news proto-articles (see Section 3.3.1). This often requires the gateway to synthesize a conforming article from non-conforming input. The gateway MUST then pass the article to an injecting agent, not directly to a relaying agent.

Incoming gateways MUST NOT pass control messages (articles containing a Control or Supersedes header field) without removing or renaming that header field. Gateways MAY, however, generate cancel control messages for messages they have gatewayed. If a gateway receives a message that it can determine is a valid equivalent of a cancel control message in the medium it is gatewaying, it SHOULD discard that message without gatewaying it, generate a corresponding cancel control message of its own, and inject that cancel control message.

NOTE: It is not unheard of for mail-to-news gateways to be used to post control messages, but encapsulation should be used for these cases instead. Gateways by their very nature are particularly prone to loops. Spews of normal articles are bad enough; spews of control messages with special significance to the news system, possibly resulting in high processing load or even email sent for every message received, are catastrophic. It is far preferable to construct a system specifically for posting control messages that can do appropriate consistency checks and authentication of the originator of the message.

If there is a message identifier that fills a role similar to that of the Message-ID header field in news, it SHOULD be used in the formation of the message identifier of the news article, perhaps with transformations required to meet the uniqueness requirement of
Netnews and with the removal of any comments so as to comply with the syntax in Section 3.1.3 of [USEFOR]. Such transformations SHOULD be designed so that two messages with the same identifier generate the same Message-ID header field.

NOTE: Message identifiers play a central role in the prevention of duplicates, and their correct use by gateways will do much to prevent loops. Netnews does, however, require that message identifiers be unique, and therefore message identifiers from other media may not be suitable for use without modification. A balance must be struck by the gateway between preserving information used to prevent loops and generating unique message identifiers.

Exceptionally, if there are multiple incoming gateways for a particular set of messages, each to a different newsgroup(s), each one SHOULD generate a message identifier unique to that gateway. Each incoming gateway nonetheless MUST ensure that it does not gate the same message twice.

NOTE: Consider the example of two gateways of a given mailing list into two separate Usenet newsgroups, both of which preserve the email message identifier. Each newsgroup may then receive a portion of the messages (different sites seeing different portions). In these cases, where there is no one "official" gateway, some other method of generating message identifiers has to be used to avoid collisions. It would obviously be preferable for there to be only one gateway which crossposts, but this may not be possible to coordinate.

If no date information is available, the gateway MAY supply a Date header field with the gateway's current date. If only partial information is available (such as date but not time), this SHOULD be fleshed out to a full Date by adding default values rather than discarding this information. Only in very exceptional circumstances should Date information be discarded, as it plays an important role in preventing reinjection of old messages.

An incoming gateway MUST add a Sender header field to the news article it forms containing the <mailbox> of the administrator of the gateway. Problems with the gateway may be reported to this <mailbox>. The <display-name> portion of this <mailbox> SHOULD
indicate that the entity responsible for injection of the message is a gateway. If the original message already had a Sender header field, it SHOULD be renamed so that its contents can be preserved.

3.9.3. Gateway Example

To illustrate the type of precautions that should be taken against loops, here is an example of the measures taken by one particular combination of mail-to-news and news-to-mail gateways designed to handle bidirectional gatewaying between mailing lists and unmoderated groups:

1. The news-to-mail gateway preserves the message identifier of the news article in the generated email message. The mail-to-news gateway likewise preserves the email message identifier provided that it is syntactically valid for Netnews. This allows the news system's built-in suppression of duplicates to serve as the first line of defense against loops.

2. The news-to-mail gateway adds an X-Gateway header field to all messages it generates. The mail-to-news gateway discards any incoming messages containing this header field. This is robust against mailing list managers that replace the message identifier, and against any number of email hops, provided that the other message header fields are preserved.

3. The mail-to-news gateway prepends the host name from which it received the email message to the content of the Path header field. The news-to-mail gateway refuses to gateway any message that contains the list server name in its Path header field (including in the tail section). This is robust against any amount of munging of the message header fields by the mailing list, provided that the email only goes through one hop.

4. The mail-to-news gateway is designed never to generate bounces to the envelope sender. Instead, articles that are rejected by the news server (for reasons not warranting silent discarding of the message) result in a bounce message sent to an errors address known not to forward to any mailing lists, so that they can be
handled by the news administrators.

These precautions have proven effective in practice at preventing loops for this particular application (bidirectional gateways between mailing lists and locally distributed newsgroups where both gateways can be designed together). General gateways to world-wide newsgroups poses additional difficulties; one must be very wary of strange configurations, such as a newsgroup gated to a mailing list which is in turn gated to a different newsgroup.

4. Media Types

This document defines several media types, which shall be registered with IANA as provided for in [RFC4288].

The media type message/news, as previously registered with IANA, is hereby declared obsolete. It was never widely implemented, and its default treatment as application/octet-stream by agents that did not recognize it was counter-productive. The media type message/rfc822 SHOULD be used in its place.

4.1. application/news-transmission

The media type application/news-transmission is intended for the encapsulation of complete news articles where the intention is that the recipient should then inject them into Netnews. This application type provides one of the methods for mailing articles to moderators (see Section 3.4.1) and may be used to convey messages to an injecting agent. This encapsulation removes the need to transform an email message into a Netnews proto-article and provides a way to send a Netnews article using MIME through a transport medium that does not support 8bit data.

The MIME media type definition of application/news-transmission is:

|MIME type name:       | application         |
|MIME subtype name:    | news-transmission   |
|Required parameters:  | none                |
Optional parameters: One and only one of "usage=moderate", "usage=infect”, or "usage=relay”.

Encoding considerations: A transfer-encoding different from that of the article transmitted MAY be supplied to ensure correct transmission over some 7bit transport medium.

Security considerations: A news article may be a control message, which if processed could have effects on the recipient host's system beyond just storage of the article.

Published specification: This specification.

Body part: A complete proto-article ready for injection into Netnews or an article being relayed to another agent.

usage=moderate indicates the article is intended for a moderator, usage=inject for an injecting agent, and usage=relay for a relaying agent. The entity receiving the article may only implement one type of agent, in which case the parameter MAY be omitted.

4.2. application/news-groupinfo

The application/news-groupinfo media type is used in conjunction with the newgroup control message (see Section 5.2.1). Its body part contains brief information about a newsgroup: the newsgroup name, its description, and its moderation status.

The MIME media type definition of application/news-transmission is:

MIME type name: application
MIME subtype name: news-groupinfo
Required parameters: charset
Optional parameters: charset, which MUST be a charset registered for use with MIME text types and has the same syntax as the parameter defined for text/plain [RFC2046]. Specifies the charset of the body part. If not given, the charset defaults to US-ASCII [ASCII].

Disposition: by default, inline
Encoding considerations: 7bit or 8bit MUST be used to maintain compatibility.
The content of the application/news-groupinfo body part is defined as:

```
  groupinfo-body = [ newsgroups-tag CRLF ]
                   newsgroups-line CRLF
  newsgroups-tag = %x46.6F.72 SP %x79.6F.75.72 SP
                   %x6E.65.77.73.67.72.6F.75.70.73 SP
                   %x66.69.6C.65.3A
                   ; case sensitive
                   ; "For your newsgroups file:"
  newsgroups-line = newsgroup-name
                   [ 1*HTAB newsgroup-description ]
                   [ 1*WSP moderation-flag ]
  newsgroup-description = utext *( *WSP utext )
  moderation-flag = %x28.4D.6F.64.65.72.61.74.65.64.29
                   ; case sensitive "(Moderated)"
```

This unusual format is backward-compatible with the scanning of the body of newgroup control messages for descriptions done by Netnews implementations that predate this specification. Although optional, the <newsgroups-tag> SHOULD be included for backward compatibility.

The <newsgroup-description> MUST NOT contain any occurrence of the string "(Moderated)" within it. Moderated newsgroups MUST be marked by appending the case sensitive text " (Moderated)" at the end.

While a charset parameter is defined for this media type, most existing software does not correctly handle descriptions in a variety of charsets. Using a charset of US-ASCII where possible is therefore RECOMMENDED.

### 4.3. application/news-checkgroups

The application/news-checkgroups media type contains a list of newsgroups within a hierarchy or hierarchies, including their descriptions and moderation status. It is primarily for use with the checkgroups control message (see Section 5.2.3).
The MIME media type definition of application/news-checkgroups is:

MIME type name:          application
MIME subtype name:       news-checkgroups
Required parameters:     none
Optional parameters:     charset, which MUST be a charset registered for use with MIME text types and has the same syntax as the parameter defined for text/plain [RFC2046]. Specifies the charset of the body part. If not given, the charset defaults to US-ASCII [ASCII].
Disposition:             by default, inline
Encoding considerations: 7bit or 8bit MUST be used to maintain compatibility.
Security considerations: This media type provides only a means for conveying a list of newsgroups and does not provide any information indicating whether the sender is authorized to state which newsgroups should exist within a hierarchy. Such authorization must be accomplished by other means.
Published specification: This specification.

The content of the application/news-groupinfo body part is defined as:

```
checkgroups-body  = *( valid-group CRLF )
valid-group       = newsgroups-line ; see 4.2
```

The same restrictions on <newsgroup-description> apply for this media type as for application/news-groupinfo.

One application/news-checkgroups message may contain information for one or more hierarchies and is considered complete for any hierarchy for which it contains a <valid-group>. In other words, an application/news-checkgroups body part consisting of:

```
example.moderated       A moderated newsgroup (Moderated)
example.test            An unmoderated test group
```
is a statement that the example.* hierarchy contains two newsgroups, example.moderated and example.test, and no others. This media type therefore MUST NOT be used for conveying partial information about a hierarchy; if a group from a given hierarchy is present, all groups that exist in that hierarchy MUST be listed.
5. Control Messages

A control message is an article which contains a Control header field and thereby indicates that some action should be taken by an agent other than distribution and display. Any article containing a Control header field (defined in Section 3.2.3 of [USEFOR]) is a control message. Additionally, the action of an article containing a Supersedes header field is described here; while such an article is not a control message, it specifies an action similar to the cancel control message.

The <control-command> of a Control header field comprises a <verb>, which indicates the action to be taken, and one or more <argument> values, which supply the details. For some control messages, the body of the article is also significant. Each recognized <verb> (the control message type) is described in a separate section below. Agents MAY accept other control message types than those specified below, and MUST either ignore or reject control messages with unrecognized types. Syntactic definitions of valid <argument> values and restrictions on control message bodies are given in the section for each control message type.

Contrary to [RFC1036], the presence of a Subject header field starting with the string "cmsg " MUST NOT cause an article to be interpreted as a control message. Agents MAY reject an article with no Control header field and such a Subject header field as ambiguous. Likewise, the presence of a <newsgroup-name> ending in ".ctl" in the Newsgroups header field or the presence of an Also-Control header field MUST NOT cause the article to be interpreted as a control message.

5.1. Authentication and Authorization

Control messages specify actions above and beyond the normal processing of an article and are therefore potential vectors of abuse and unauthorized action. There is, at present, no standardized means of authenticating the sender of a control message or verifying that the contents of a control message were sent by the claimed sender. There are, however, some unstandardized authentication mechanisms in common use.

Agents acting on control messages SHOULD take steps to authenticate control messages before acting on them, as determined by local authorization policy. Whether this is done via the use of an unstandardized authentication protocol, by comparison with information obtained through another protocol, by human review, or by some other means is left unspecified by this document. Further extensions or revisions of this protocol are expected to standardize
a digital signature mechanism.

Agents are expected to have their own local authorization policies for which control messages will be honored. No Netnews agent is ever required to act on any control message. The following descriptions specify the actions that a control message requests, but an agent MAY always decline to act on any given control message.

5.2. Group Control Messages

A group control message is any control message type that requests some update to the list of newsgroups known to a news server. The standard group control message types are "newgroup", "rmgroup", and "checkgroups".

Before honoring any group control message, an agent MUST check the newsgroup or newsgroups affected by that control message and decline to create any newsgroups not in conformance with the restrictions in Section 3.1.4 of [USEFOR].

All of the group control messages MUST have an Approved header field (Section 3.2.1 of [USEFOR]). Group control messages without an Approved header field SHOULD NOT be honored.

5.2.1. The newgroup Control Message

The newgroup control message requests the specified group be created or, if already existing, have its moderation status or description changed. The syntax of its Control header field is:

```
control-command =/ Newgroup-command
Newgroup-command = "newgroup" Newgroup-arguments
Newgroup-arguments = 1*WSP newsgroup-name [ 1*WSP newgroup-flag ]
newgroup-flag = "moderated"
```

If the request is honored, the moderation status of the group SHOULD be set in accordance with the presence or absence of the <newgroup-flag> "moderated". "moderated" is the only flag defined by this protocol. Other flags MAY be defined by extensions to this protocol and accepted by agents. If an agent does not recognize the <newgroup-flag> of a newgroup control message, it SHOULD ignore that control message.
The body of a newgroup message SHOULD contain an entity of type application/news-groupinfo specifying the description of the newsgroup, either as the entire body or as an entity within a multipart/related object [RFC2046]. If such an entity is present, the moderation status specified therein MUST match the moderation status specified by the <newgroup-flag>. The body of a newgroup message MAY contain other entities (encapsulated in multipart/related) that provide additional information about the newsgroup or the circumstances of the control message.

In the absence of an application/news-groupinfo entity, a news server MAY search the body of the message for the line "For your newsgroups file:" and take the following line as a <newsgroups-line>. Prior to the standardization of application/news-groupinfo, this was the convention for providing a newsgroup description.

If the request is honored and contains a newsgroup description, and if the news server honoring it stores newsgroup descriptions, the stored newsgroup description SHOULD be updated to the description specified in the control message, even if no other property of the group has changed.

5.2.1.1. newgroup Control Message Example

A newgroup control message requesting creation of the moderated newsgroup example.admin.info.

From: "example.* Administrator" <admin@noc.example>
Newsgroups: example.admin.info
Date: 27 Feb 2002 12:50:22 +0200
Subject: cmgs newgroup example.admin.info moderated
Approved: admin@noc.example
Control: newgroup example.admin.info moderated
Message-ID: <ng-example.admin.info-20020227@noc.example>
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="nxtprt"
Content-Transfer-Encoding: 8bit

This is a MIME control message.
--nxtprt
Content-Type: application/news-groupinfo
For your newsgroups file:
example.admin.info      About the example.* groups (Moderated)

--nxtprt
Content-Type: text/plain

A moderated newsgroup for announcements about new newsgroups in
the example.* hierarchy.

--nxtprt--

Internet-Draft     Netnews Architecture and Protocols      November 2006

5.2.2. The rmgroup Control Message

The rmgroup control message requests the specified group be removed
from a news server's list of valid groups. The syntax of its Control
header field is:

```
control-command     =/ Rmgroup-command
Rmgroup-command     = "rmgroup" Rmgroup-arguments
Rmgroup-arguments   = 1*WSP newsgroup-name
```

The body of the control message MAY contain anything, usually an
explanatory text.

5.2.3. The checkgroups Control Message

The checkgroups control message contains a list of all the valid
groups in a hierarchy with descriptions and moderation status. It
requests a news server update its valid newsgroup list for that
hierarchy to include the groups specified, remove any groups not
specified, and update group descriptions to match those given in the
checkgroups control message. The syntax of its Control header field is:

```
control-command     =/ Checkgroup-command
Checkgroup-command  = "checkgroups" Checkgroup-arguments
Checkgroup-arguments= [ chkscope ] [ chksernr ]
chkscope            = 1*( FWS ["!"] newsgroup-name )
chksernr            = FWS "#" 1*DIGIT
```
A checkgroups message is interpreted as an exhaustive list of the valid groups in all hierarchies or sub-hierarchies with a prefix listed in the <chkscope> argument, excluding any sub-hierarchy where the <chkscope> argument is prefixed by "!". If no <chkscope> argument is given, it applies to all hierarchies for which group statements appear in the body of the message. Since much existing software does not honor the <chkscope> argument, the body of the checkgroups control message MUST NOT contain group statements for newsgroups outside the intended scope and SHOULD contain a correct newsgroup list even for sub-hierarchies excluded with "!" <chkscope> terms. News servers, however, MUST honor <chkscope> as specified here.

The <chksernr> argument may be any positive integer. If present, it MUST increase with every change to the newsgroup list and MUST NOT ever decrease. If provided, news servers SHOULD remember the <chksernr> value of the previous checkgroups control message honored for a particular hierarchy or sub-hierarchy and decline to honor any subsequent checkgroups control message for the same hierarchy or sub-

For example, the following Control header field

Control: checkgroups de !de.alt #248

indicates the body of the message will list every newsgroup in the de.* hierarchy, excepting the de.alt.* sub-hierarchy, and should not be honored if a checkgroups control message with a serial number greater than 248 was previously honored.

The body of the message is an entity of type application/news-checkgroups. It SHOULD be declared as such with appropriate MIME headers, but news servers SHOULD interpret checkgroups messages that lack the appropriate MIME headers as if the body were of type application/news-checkgroups for backward compatibility.

5.3. The cancel Control Message

The cancel control message requests that a target article be withdrawn from circulation and access. The syntax of its Control header field is:
control-command =/ Cancel-command
Cancel-command = "cancel" Cancel-arguments
Cancel-arguments = FWS msg-id [FWS]

The argument identifies the article to be cancelled by its message identifier. The body of the control message MAY contain anything, usually an explanatory text.

A serving agent that elects to honor a cancel message SHOULD make the article unavailable to reading agents (perhaps by deleting it completely). If the cancel control message arrives before the article it targets, news servers choosing to honor it SHOULD remember the message identifier that was cancelled and reject the cancelled article when it arrives.

Cancel control messages listing moderated newsgroups in their Newsgroups header field MUST contain an Approved header field like any other article in a moderated newsgroup. This means that cancels posted to a moderated newsgroup will normally be sent to the moderator first for approval. Outside of moderated newsgroups, cancel messages are not required to contain an Approved header field.

Contrary to [RFC1036], cancel control messages are not required to contain From and Sender header fields matching the target message. This requirement only encouraged cancel issuers to conceal their identity and provided no security.

5.4. The Supersedes Header Field

The presence of a Supersedes header field in an article requests that the message identifier given in that header field be withdrawn in exactly the same manner as if it were the target of a cancel control message. Accordingly, news servers SHOULD use the same authentication and authorization checks for deciding whether to honor a Supersedes header field as they use for cancel control messages. If the Supersedes header field is honored, the news server SHOULD take the same actions as it would take when honoring a cancel control message for the given target article.

5.5. The ihave and sendme Control Messages
The ihave and sendme control messages implement a predecessor of the NNTP [RFC3977] protocol. They are largely obsolete on the Internet but still see use in conjunction with some transport protocols such as UUCP. News servers are not required to support them.

ihave and sendme control messages share similar syntax for their Control header fields and bodies:

```
control-command   =/ Ihave-command
Ihave-command     = "ihave" [Ihave-argument]
Ihave-argument    = 1*WSP *( msg-id 1*WSP ) [relayer-name]
control-command   =/ Sendme-command
Sendme-command     = "sendme" Sendme-argument
Sendme-argument    = Ihave-argument
relayer-name       = path-identity ; see 3.1.5 of [USEFOR]
ihave-body         = *( msg-id CRLF )
sendme-body        = ihave-body
```

The body of the article consists of a list of <msg-id>s, one per line. The message identifiers SHOULD be put in the body of the article, not in the Control header field, but news servers MAY recognize and process message identifiers in the Control header field for backward compatibility.

The ihave message states that the named relaying agent has received articles with the specified message identifiers, which may be of interest to the relaying agents receiving the ihave message. The sendme message requests that the agent receiving it send the articles having the specified message identifiers to the named relaying agent. If <relayer-name> is not given, it is determined from the origin of the control message.

Upon receipt of the sendme message (and a decision to honor it), the receiving agent sends the article or articles requested. The mechanism for this transmission is unspecified by this document and is arranged between the sites involved.

These control messages are normally sent as point-to-point articles between two sites and not then sent on to other sites. Newsgroups beginning with "to." are reserved for such point-to-point
communications.

5.6. Obsolete Control Messages

The following control message types are declared obsolete by this document and SHOULD NOT be sent or honored:

  sendsys
  version
  whogets
  senduuname

6. Security Considerations
Netnews is designed for broad dissemination of public messages and offers little in the way of security. What protection Netnews has against abuse and impersonation is provided primarily by the underlying transport layer. In large Netnews networks where news servers cannot be relied upon to enforce authentication and authorization requirements at the transport layer, articles may be trivially forged and widely read, and the identities of article senders and privacy of articles cannot be assured.

See Section 5 of [USEFOR] for further security considerations related to the format of articles.

6.1. Compromise of System Integrity

Control messages pose a particular security concern since acting on unauthorized control messages may cause newsgroups to be removed, articles to be deleted, and unwanted newsgroups to be created. Administrators of news servers SHOULD therefore take steps to verify the authenticity of control messages as discussed in Section 5.1. Articles containing Supersedes header fields are effectively cancel control messages and SHOULD be subject to the same checks as discussed in Section 5.4. Currently, many sites are ignoring all cancel control messages and Supersedes header fields due to the difficulty of authenticating them and their widespread abuse.

All agents should be aware that all article content, most notably including the content of the Control header field, is potentially untrusted and malicious. Articles may be constructed in syntactically invalid ways to attempt to overflow internal buffers, violate hidden assumptions, or exploit implementation weaknesses. For example, some news server implementations have been successfully attacked via inclusion of Unix shell code in the Control header field. All article contents, and particularly control message contents, SHOULD be handled with care and rigorously verified before any action is taken on the basis of the contents of the article.

A malicious poster may add an Approved header field to bypass the moderation process of a moderated newsgroup. Injecting agents SHOULD verify that messages approved for a moderated newsgroup are being injected by the moderator using authentication information from the underlying transport or some other authentication mechanism arranged with the moderator.

A malicious news server participating in a Netnews network may bypass checks performed by injecting agents, forge Path header fields and other trace information (such as Injection-Info header fields), and
otherwise compromise the authorization requirements of a Netnews network. News servers SHOULD use the facilities of the underlying transport to authenticate their peers and reject articles from injecting and relaying agents that do not follow the requirements of this protocol or the Netnews network.

6.2. Denial of Service

The proper functioning of individual newsgroups can be disrupted by the excessive posting of unwanted articles; by the repeated posting of identical or near identical articles; by posting followups unrelated to their precursors or which quote their precursors in full with the addition of minimal extra material (especially if this process is iterated); by crossposting to, or requesting followups to, totally unrelated newsgroups; and by abusing control messages and the Supersedes header field to delete articles or newsgroups.

Such articles intended to deny service, or other articles of an inflammatory nature, may also have their From or Reply-To addresses set to valid but incorrect email addresses, thus causing large volumes of email to descend on the true owners of those addresses. Users and agents should always be aware that identifying information in articles may be forged.

A malicious poster may prevent an article from being seen at a particular site by including in the Path header field of the proto-article the <path-identity> of that site. Use of the <diag-keyword> "POSTED" by injecting agents to mark the point of injection can prevent this attack.

Primary responsibility for preventing such attacks lies with injecting agents, which can apply authentication and authorization checks via the underlying transport and prevent those attempting denial of service attacks from posting messages. If specific injecting agents fail to live up to their responsibilities, they may be excluded from the Netnews network by configuring relaying agents to reject articles originating from them.

A malicious complainer may submit a modified copy of an article (with an altered Injection-Info header field, for instance) to the administrator of an injecting agent in an attempt to discredit the author of that article and even to have his posting privileges removed. Administrators SHOULD therefore obtain a genuine copy of the article from their own serving agent before taking action in response to such a complaint.
6.3. Leakage

Articles which are intended to have restricted distribution are dependent on the goodwill of every site receiving them. Restrictions on dissemination and retention of articles may be requested via the Archive and Distribution header fields, but such requests cannot be enforced by the protocol.

The flooding algorithm used by Netnews transports such as NNTP [RFC3977] is extremely good at finding any path by which articles can leave a subnet with supposedly restrictive boundaries, and substantial administrative effort is required to avoid this. Organizations wishing to control such leakage are advised to designate a small number of gateways to handle all news exchange with the outside world.

The sendme control message Section 5.5, insofar as it is still used, can be used to request articles the requester should not have access to.
7. IANA Considerations

IANA is requested to register the following media types, described elsewhere in this document, for use with the Content-Type header field, in the IETF tree in accordance with the procedures set out in [RFC4288].

    application/news-transmission  (4.1)
    application/news-groupinfo      (4.2)
    application/news-checkgroups    (4.3)

IANA is also requested to change the status of the following media type to "OBSOLETE".

message/news

message/rfc822 should be used instead.
8. References

8.1. Normative References

[ASCII] "American National Standard for Information Systems -
Coded Character Sets - 7-Bit American National Standard
Code for Information Interchange (7-Bit ASCII)",
ANSI X3.4, 1986.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail
Extensions (MIME) Part Two: Media Types", RFC 2046,
November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate

[RFC2822] Resnick, P., "Internet Message Format", RFC 2822,
April 2001.

[RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and

Article Format".
Informative References


[USEAGE] Lindsey, C., "Usenet Best Practice".

Appendix A. Changes to the Existing Protocols

This document prescribes many changes, clarifications, and new features since the protocol described in [RFC1036]. Most notably:

- A new, backward-compatible Path header field format that permits standardized embedding of additional trace and authentication information is now RECOMMENDED. See Section 3.4 and Section 3.5. Folding of the Path header is permitted.

- Trimming of the References header field is permitted and a mechanism for doing so is defined.

- Addition of the new Injection-Date header field is required for injecting agents (Section 3.4) and must be used by news servers for date checks (Section 3.5). Injecting agents are strongly encouraged to put all local trace information in the new Injection-Info header field.
A new media type is defined for transmitting Netnews articles through other media (Section 4.1), and moderators should prepare to receive submissions in that format (Section 3.4.1).

Certain control messages (Section 5.6) are declared obsolete, and the special significance of "cmsg" at the start of a Subject header field is removed.

Additional media types are defined for improved structuring, specification, and automated processing of control messages (Section 4.2 and Section 4.3).

Two new optional parameters are added to the checkgroups control message.

The message/news media type is declared obsolete.

Cancel control messages are no longer required to have From and Sender header fields matching those of the target message, as this requirement added no real security.

In addition, many protocol steps and article verification requirements unmentioned or left ambiguous by [RFC1036] but widely implemented by Netnews servers have been standardized and specified in detail.

Appendix B. Acknowledgements

This document is the result of a ten year effort and the number of people that have contributed to its content are too numerous to mention individually. Many thanks go out to all past and present members of the USEFOR Working Group of the Internet Engineering Task Force (IETF) and the accompanying mailing list.

Special thanks are due to Henry Spencer, whose Son-of-1036 draft served as the initial basis for this document.
P.O. Box 20066
Stanford, CA 94309
US

Email: rra@stanford.edu
URI: http://www.eyrie.org/~eagle/

Charles H. Lindsey
University of Manchester
5 Clerewood Avenue
Heald Green
Cheadle
Cheshire SK8 3JU
United Kingdom

Phone: +44 161 436 6131
Email: chl@clerew.man.ac.uk
Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in BCP 78, 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.

Acknowledgment

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