                 "Son of 1036": News Article Format and Transmission

Abstract

   By the early 1990s, it had become clear that RFC 1036, then the
   specification for the Interchange of USENET Messages, was badly in
   need of repair. This "Internet-Draft-to-be", though never formally
   published at that time, was widely circulated and became the de facto
   standard for implementors of News Servers and User Agents, rapidly
   acquiring the nickname "Son of 1036". Indeed, under that name, it
   could fairly be described as the best-known Internet Draft (n)ever
   published, and it formed the starting point for the recently adopted
   Proposed Standards for Netnews.

   It is being published now in order to provide the historical
   background out of which those standards have grown. Present-day
   implementors should be aware that it is NOT NOW APPROPRIATE for use
   in current implementations.

Copyright Notice

Table of Contents

Preface

   Although [RFC1036] was published in 1987, for many years it remained
   the only formally published specification for Netnews format and
   processing. It was widely considered obsolete within a few years,
   and it has now been superseded by the work of the USEFOR Working
   Group, leading to the publication of [RFC5536] and [RFC5537].
   However, there was an intermediate step that is of some historical
   interest.

   In 1993-4, Henry Spencer wrote and informally circulated a document
   that became known as "Son of 1036", meant as a first draft of a
   replacement for [RFC1036]. It went no further at the time (although,
   more recently, the USEFOR Working Group started from it), but has
   nevertheless seen considerable use as a technical reference and even
   a de facto standard, despite its informal status.

   The USEFOR work has eliminated any further relevance of Son of 1036
   as a technical reference, but it remains of historical interest. The
   USEFOR Working Group has asked that it be published as an Historic
   RFC, to ensure its preservation in an accessible form and facilitate
   referencing it.

   This document is identical to the last distributed version of Son of
   1036, dated 2 June 1994, except for reformatting, correction of a few
   minor factual or formatting errors, completion of the then-empty
   Appendix D and of the References section, minor editing to match
   preferred RFC style, and changes to leading and trailing material.
   Remarks enclosed within "{...}" indicate explanatory material not
   present in the original version. References to the current MIME
   standards (and a few others) have been added (that was an unresolved
   issue in 1994).

   The technical content remains unchanged, including the references to
   the document itself as a Draft rather than an RFC and the presence of
   unresolved issues. The original section numbering has been
   preserved, although the original pagination has not (among other
   reasons, it did not fully follow IETF formatting standards).

   READERS ARE CAUTIONED THAT THIS DOCUMENT IS OBSOLETE AND SHOULD NOT
   BE USED AS A TECHNICAL REFERENCE. Although Son of 1036 largely
   documented existing practice, it also proposed some changes, some of
   which did not catch on or are no longer considered good ideas. (Of
   particular note, the MIME type "message/news" should not be used.)
   Consult [RFC5536] and [RFC5537] for modern technical information.

Although a number of people contributed useful comments or criticism
during the preparation of this document, its contents are entirely
the opinions of the author circa 1994. Not even the author himself
agrees with them all now.

The author thanks Charles Lindsey for his assistance in getting this
document cleaned up and formally published at last (not least, for
supplying some prodding to actually get it done!).

The author thanks Luc Rooijakkers for supplying the MIME summary that
Appendix B is based on.

Original Abstract

This Draft defines the format and procedures for interchange of
network news articles. It is hoped that a later version of this
Draft will obsolete RFC 1036, reflecting more recent experience and
accommodating future directions.

Network news articles resemble mail messages but are broadcast to
potentially large audiences, using a flooding algorithm that
propagates one copy to each interested host (or group thereof),
typically stores only one copy per host, and does not require any
central administration or systematic registration of interested
users. Network news originated as the medium of communication for
Usenet, circa 1980. Since then, Usenet has grown explosively, and
many Internet sites participate in it. In addition, the news
technology is now in widespread use for other purposes, on the
Internet and elsewhere.

This Draft primarily codifies and organizes existing practice. A few
small extensions have been added in an attempt to solve problems that
are considered serious. Major extensions (e.g., cryptographic
authentication) that need significant development effort are left to
be undertaken as independent efforts.

1. Introduction

Network news articles resemble mail messages but are broadcast to
potentially large audiences, using a flooding algorithm that
propagates one copy to each interested host (or groups thereof),
typically stores only one copy per host, and does not require any
central administration or systematic registration of interested
users. Network news originated as the medium of communication for
Usenet, circa 1980. Since then, Usenet has grown explosively, and
many Internet sites participate in it. In addition, the news
technology is now in widespread use for other purposes, on the
Internet and elsewhere.

The earliest news interchange used the so-called "A News" article
format. Shortly thereafter, an article format vaguely resembling
Internet mail was devised and used briefly. Both of those formats
are completely obsolete; they are documented in Appendix A for
historical reasons only. With the publication of [RFC850] in 1983,
news articles came to closely resemble Internet mail messages, with
some restrictions and some additional headers. In 1987, [RFC1036]
updated [RFC850] without making major changes.

In the intervening five years, the [RFC1036] article format has
proven quite satisfactory, although minor extensions appear desirable
to match recent developments in areas such as multi-media mail.
[RFC1036] itself has not proven quite so satisfactory. It is often
rather vague and does not address some issues at all; this has caused
significant interoperability problems at times, and implementations
have diverged somewhat. Worse, although it was intended primarily to
document existing practice, it did not precisely match existing
practice even at the time it was published, and the deviations have
grown since.

This Draft attempts to specify the format of articles, and the
procedures used to exchange them and process them, in sufficient
detail to allow full interoperability. In addition, some tentative
suggestions are made about directions for future development, in an
attempt to avert unnecessary divergence and consequent loss of
interoperability. Major extensions (e.g., cryptographic
authentication) that need significant development effort are left to
be undertaken as independent efforts.

   NOTE: One question all of this may raise is: why is there no News-
   Version header, analogous to MIME-Version, specifying a version
   number corresponding to this specification? The answer is: it
   doesn't appear to be useful, given news's backward-compatibility
   constraints. The major use of a version number is indicating
   which of several INCOMPATIBLE interpretations is relevant. The
   impossibility of orchestrating any sort of simultaneous change
   over news's installed base makes it necessary to avoid such
   incompatible changes (as opposed to extensions) entirely. MIME
   has a version number mostly because it introduced incompatible
   changes to the interpretation of several "Content-" headers. This
   Draft attempts no changes in interpretation, and it appears
   doubtful that future Drafts will find it feasible to introduce
   any.

   UNRESOLVED ISSUE: Should this be reconsidered? Only if the header
   has SPECIFIC IDENTIFIABLE uses today. Otherwise, it's just
   useless added bulk.

As in this Draft's predecessors, the exact means used to transmit
articles from one host to another is not specified. Network News
Transfer Protocol (NNTP) [RFC977] {since replaced by [RFC3977]} is
probably the most common transmission method on the Internet, but a
number of others are known to be in use, including the Unix-To-Unix
Copy Protocol [UUCP], which was extensively used in the early days of
Usenet and is still much used on its fringes today.

Several of the mechanisms described in this Draft may seem somewhat
strange or even bizarre at first reading. As with Internet mail,
there is no reasonable possibility of updating the entire installed
base of news software promptly, so interoperability with old software
is crucial and will remain so. Compatibility with existing practice
and robustness in an imperfect world necessarily take priority over
elegance.

2. Definitions, Notations, and Conventions

2.1. Textual Notations

Throughout this Draft, "MAIL" is short for "[RFC822] as amended by
[RFC1123]". ([RFC1123]'s amendments are mostly relatively small, but
they are not insignificant.) See also the discussion in Section 3
about this Draft's relationship to MAIL. "MIME" is short for
"[RFC1341] and [RFC1342]" (or their {since} updated replacements
{[RFC2045], [RFC2046], and [RFC2047]}).

   UNRESOLVED ISSUE: Update these numbers {now resolved!}.

   {NOTE: Since the original publication of this Draft [RFC822] has
   been updated, firstly to [RFC2822] and more recently to [RFC5322];
   however, this Draft is firmly rooted in the original [RFC822].
   Similarly, [RFC821] has also received two upgrades in the
   meantime.}

"ASCII" is short for "the ANSI X3.4 character set" [X3.4]. While
"ASCII" is often misused to refer to various character sets somewhat
similar to X3.4, in this Draft, "ASCII" means [X3.4] and only [X3.4].

   NOTE: The name is traditional (to the point where the ANSI
   standard sanctions it), even though it is no longer an acronym for
   the name of the standard.

   NOTE: ASCII, X3.4, contains 128 characters, not all of them
   printable. Character sets with more characters are not ASCII,
   although they may include it as a subset.

Certain words used to define the significance of individual
requirements are capitalized. "MUST" means that the item is an
absolute requirement of the specification. "SHOULD" means that the
item is a strong recommendation: there may be valid reasons to ignore
it in unusual circumstances, but this should be done only after
careful study of the full implications and a firm conclusion that it
is necessary, because there are serious disadvantages to doing so.
"MAY" means that the item is truly optional, and implementors and
users are warned that conformance is possible but not to be relied
on.

The term "compliant", applied to implementations, etc., indicates
satisfaction of all relevant "MUST" and "SHOULD" requirements. The
term "conditionally compliant" indicates satisfaction of all relevant
"MUST" requirements but violation of at least one relevant "SHOULD"
requirement.

This Draft contains explanatory notes using the following format.
These may be skipped by persons interested solely in the content of
the specification. The purpose of the notes is to explain why
choices were made, to place them in context, or to suggest possible
implementation techniques.

   NOTE: While such explanatory notes may seem superfluous in
   principle, they often help the less-than-omniscient reader grasp
   the purpose of the specification and the constraints involved.
   Given the limitations of natural language for descriptive
   purposes, this improves the probability that implementors and
   users will understand the true intent of the specification in
   cases where the wording is not entirely clear.

All numeric values are given in decimal unless otherwise indicated.
Octets are assumed to be unsigned values for this purpose. Large
numbers are written using the North American convention, in which ","
separates groups of three digits but otherwise has no significance.

2.2. Syntax Notation

Although the mechanisms specified in this Draft are all described in
prose, most are also described formally in the modified BNF notation
of [RFC822]. Implementors will need to be familiar with this
notation to fully understand this specification and are referred to
[RFC822] for a complete explanation of the modified BNF notation.
Here is a brief illustrative example:

```
sentence  = clause *( punct clause ) "."
punct     = ":" / ";"
clause    = 1*word [ "(" clause ")" / "," 1*word ]
word      = <any English word>
```

This defines a sentence as some clauses separated by puncts and ended
by a period, a punct as a colon or semicolon, a clause as at least
one <word> optionally followed by either a parenthesized clause or a
comma and at least one more <word>, and a <word> as (informally) any
English word. The characters "<>" are used to enclose names when
(and only when) distinguishing them from surrounding text is useful.
The full form of the repetition notation is "<m>*<n><thing>",
denoting <m> through <n> repetitions of <thing>; <m> defaults to
zero, <n> to infinity, and the "*" and <n> can be omitted if <m> and
<n> are equal, so 1*word is one or more words, 1*5word is one through
five words, and 2word is exactly two words.

The character "\" is not special in any way in this notation.

This Draft is intended to be self-contained; all syntax rules used in
it are defined within it, and a rule with the same name as one found
in MAIL does not necessarily have the same definition. The lexical
layer of MAIL is NOT, repeat NOT, used in this Draft, and its
presence must not be assumed; notably, this Draft spells out all
places where white space is permitted/required and all places where
constructs resembling MAIL comments can occur.

   NOTE: News parsers historically have been much less permissive
   than MAIL parsers.

2.3. Definitions

The term "character set", wherever it is used in this Draft, refers
to a coded character set, in the sense of ISO character set
standardization work, and must not be misinterpreted as meaning
merely "a set of characters".

In this Draft, ASCII character 32 is referred to as "blank"; the word
"space" has a more generic meaning.

An "article" is the unit of news, analogous to a MAIL "message".

A "poster" is a human being (or software equivalent) submitting a
possibly compliant article to be "posted", i.e., made available for
reading on all relevant hosts. A "posting agent" is software that
assists posters to prepare articles, including determining whether
the final article is compliant, passing it on to a relayer for
posting if so, and returning it to the poster with an explanation if

not. A "relayer" is software that receives allegedly compliant
articles from posting agents and/or other relayers, files copies in a
"news database", and possibly passes copies on to other relayers.

   NOTE: While the same software may well function both as a relayer
   and as part of a posting agent, the two functions are distinct and
   should not be confused. The posting agent's purpose is (in part)
   to validate an article, supply header information that can or
   should be supplied automatically, and generally take reasonable
   actions in an attempt to transform the poster's submission into a
   compliant article. The relayer's purpose is to move already-
   compliant articles around efficiently without damaging them.

A "reader" is a human being reading news articles. A "reading agent"
is software that presents articles to a reader.

   NOTE: Informal usage often uses "reader" for both these meanings,
   but this introduces considerable potential for confusion and
   misunderstanding, so this Draft takes care to make the
   distinction.

A "newsgroup" is a single news forum, a logical bulletin board,
having a name and nominally intended for articles on a specific
topic. An article is "posted to" a single newsgroup or several
newsgroups. When an article is posted to more than one newsgroup, it
is said to be "cross-posted"; note that this differs from posting the
same text as part of each of several articles, one per newsgroup. A
"hierarchy" is the set of all newsgroups whose names share a first
component (see the name syntax in Section 5.5).

A newsgroup may be "moderated", in which case submissions are not
posted directly, but mailed to a "moderator" for consideration and
possible posting. Moderators are typically human but may be
implemented partially or entirely in software.

A "followup" is an article containing a response to the contents of
an earlier article (the followup's "precursor"). A "followup agent"
is a combination of reading agent and posting agent that aids in the
preparation and posting of a followup.

Text comparisons are "case-sensitive" if they consider uppercase
letters (e.g., "A") different from lowercase letters (e.g., "a"), and
"case-insensitive" if letters differing only in case (e.g., "A" and
"a") are considered identical. Categories of text are said to be
case-(in)sensitive if comparisons of such texts to others are case-
(in)sensitive.

A "cooperating subnet" is a set of news-exchanging hosts that is
sufficiently well-coordinated (typically via a central administration
of some sort) that stronger assumptions can be made about hosts in
the set than about news hosts in general. This is typically used to
relax restrictions that are otherwise required for worst-case
interoperability; members of a cooperating subnet MAY interchange
articles that do not conform to this Draft's specifications, provided
all members have agreed to this and provided the articles are not
permitted to leak out of the subnet. The word "subnet" is used to
emphasize that a cooperating subnet is typically not an isolated
universe; care must be taken that traffic leaving the subnet complies
with the restrictions of the larger net, not just those of the
cooperating subnet.

A "message ID" is a unique identifier for an article, usually
supplied by the posting agent that posted it. It distinguishes the
article from every other article ever posted anywhere (in theory).
Articles with the same message ID are treated as identical copies of
the same article even if they are not in fact identical.

A "gateway" is software that receives news articles and converts them
to messages of some other kind (e.g., mail to a mailing list), or
vice versa; in essence, it is a translating relayer that straddles
boundaries between different methods of message exchange. The most
common type of gateway connects newsgroup(s) to mailing list(s),
either unidirectionally or bidirectionally, but there are also
gateways between news networks using this Draft's news format and
those using other formats.

A "control message" is an article that is marked as containing
control information; a relayer receiving such an article will
(subject to permissions, etc.) take actions beyond just filing and
passing on the article.

   NOTE: "Control article" would be more consistent terminology, but
   "control message" is already well established.

An article's "reply address" is the address to which mailed replies
should be sent. This is the address specified in the article's From
header (see Section 5.2), unless it also has a Reply-To header (see
Section 6.3).

The notation (for example) "(ASCII 17)" following a name means "this
name refers to the ASCII character having value 17". An "ASCII
printable character" is an ASCII character in the range 33-126. An
"ASCII control character" is an ASCII character in the range 0-31, or
the character DEL (ASCII 127). A "non-ASCII character" is a
character having a value exceeding 127.

   NOTE: Blank is neither an "ASCII printable character" nor an
   "ASCII control character".

2.4. End-of-Line

   How the end of a text line is represented depends on the context and
   the implementation. For Internet transmission via protocols such as
   SMTP [RFC821], an end-of-line is a CR (ASCII 13) followed by an LF
   (ASCII 10). ISO C [ISO/IEC9899] and many modern operating systems
   indicate end-of-line with a single character, typically ASCII LF (aka
   "newline"), and this is the normal convention when news is
   transmitted via UUCP. A variety of other methods are in use,
   including out-of-band methods in which there is no specific character
   that means end-of-line.

   This Draft does not constrain how end-of-line is represented in news,
   except that characters other than CR and LF MUST NOT be usurped for
   use in end-of-line representations. Also, obviously, all software
   dealing with a particular copy of an article must agree on the
   convention to be used. "EOL" is used to mean "whatever end-of-line
   representation is appropriate"; it is not necessarily a character or
   sequence of characters.

      NOTE: If faced with picking an EOL representation in the absence
      of other constraints, use of a single character simplifies
      processing, and the ASCII standard [X3.4] specifies that if one
      character is to be used for this purpose, it should be LF (ASCII
      10).

      NOTE: Inside MIME encodings, use of the Internet canonical EOL
      representation (CR followed by LF) is mandatory. See [RFC2049].

2.5. Case-Sensitivity

   Text in newsgroup names, header parameters, etc. is case-sensitive
   unless stated otherwise.

      NOTE: This is at variance with MAIL, which is case-insensitive
      unless stated otherwise, but is consistent with news historical
      practice and existing news software. See the comments on backward
      compatibility in Section 1.

2.6. Language

   Various constant strings in this Draft, such as header names and
   month names, are derived from English words. Despite their
   derivation, these words do NOT change when the poster or reader
   employing them is interacting in a language other than English.

Posting and reading agents SHOULD translate as appropriate in their
interaction with the poster or reader, but the forms that actually
appear in articles are always the English-derived ones defined in
this Draft.

3. Relation to MAIL (RFC822, etc.)

The primary intent of this Draft is to completely describe the news
article format as a subset of MAIL's message format (augmented by
some new headers). Unless explicitly noted otherwise, the intent
throughout is that an article MUST also be a valid MAIL message.

   NOTE: Despite obvious similarities between news and mail, opinions
   vary on whether it is possible or desirable to unify them into a
   single service. However, it is unquestionably both possible and
   useful to employ some of the same tools for manipulating both mail
   messages and news articles, so there is specific advantage to be
   had in defining them compatibly. Furthermore, there is no
   apparent need to re-invent the wheel when slight extensions to an
   existing definition will suffice.

Given that this Draft attempts to be self-contained, it inevitably
contains considerable repetition of information found in MAIL. This
raises the possibility of unintentional conflicts. Unless
specifically noted otherwise, any wording in this Draft that permits
behavior that is not MAIL-compliant is erroneous and should be
followed only to the extent that the result remains compliant with
MAIL.

   NOTE: [RFC1036] said "where this standard conflicts with the
   Internet Standard, RFC 822 should be considered correct and this
   standard in error". Taken literally, this was obviously
   incorrect, since [RFC1036] imposed a number of restrictions not
   found in [RFC822]. The intent, however, was reasonable: to
   indicate that UNINTENTIONAL differences were errors in [RFC1036].

Implementors and users should note that MAIL is deliberately an
extensible standard, and most extensions devised for mail are also
relevant to (and compatible with) news. Note particularly MIME,
summarized briefly in Appendix B, which extends MAIL in a number of
useful ways that are definitely relevant to news. Also of note is
the work in progress on reconciling Privacy Enhanced Mail (PEM),
which defines extensions for authentication and security) with MIME,
after which this may also be relevant to news.

   UNRESOLVED ISSUE: Update the MIME/PEM information.

Similarly, descriptions here of MIME facilities should be considered
correct only to the extent that they do not require or legitimize
practices that would violate those RFCs. (Note that this Draft does
extend the application of some MIME facilities, but this is an
extension rather than an alteration.)

4. Basic Format

4.1. Overall Syntax

The overall syntax of a news article is:

```
article        = 1*header separator body
header         = start-line *continuation
start-line     = header-name ":" space [ nonblank-text ] eol
continuation   = space nonblank-text eol
header-name    = 1*name-character *( "-" 1*name-character )
name-character = letter / digit
letter         = <ASCII letter A-Z or a-z>
digit          = <ASCII digit 0-9>
separator      = eol
body           = *( [ nonblank-text / space ] eol )
eol            = <EOL>
nonblank-text  = [ space ] text-character *( space-or-text )
text-character = <any ASCII character except NUL (ASCII 0),
                    HT (ASCII 9), LF (ASCII 10), CR (ASCII 13),
                    or blank (ASCII 32)>
space          = 1*( <HT (ASCII 9)> / <blank (ASCII 32)> )
space-or-text  = space / text-character
```

An article consists of some headers followed by a body. An empty
line separates the two. The headers contain structured information
about the article and its transmission. A header begins with a
header name identifying it, and can be continued onto subsequent
lines by beginning the continuation line(s) with white space. (Note
that Section 4.2.3 adds some restrictions to the header syntax
indicated here.) The body is largely unstructured text significant
only to the poster and the readers.

    NOTE: Terminology here follows the current custom in the news
    community, rather than the MAIL convention of (sometimes)
    referring to what is here called a "header" as a "header field" or
    "field".

Note that the separator line must be truly empty, and not just a line
containing white space. Further empty lines following it are part of
the body, as are empty lines at the end of the article.

   NOTE: Some systems make no distinction between empty lines and
   lines consisting entirely of white space; indeed, some systems
   cannot represent entirely empty lines. The grammar's requirement
   that header continuation lines contain some printable text is
   meant to ensure that the empty/space distinction cannot confuse
   identification of the separator line.

   NOTE: It is tempting to authorize posting agents to strip empty
   lines at the beginning and end of the body, but such empty lines
   could possibly be part of a preformatted document.

Implementors are warned that trailing white space, whether alone on
the line or not, MAY be significant in the body, notably in early
versions of the "uuencode" encoding for binary data. Trailing white
space MUST be preserved unless the article is known to have
originated within a cooperating subnet that avoids using significant
trailing white space, and SHOULD be preserved regardless. Posters
SHOULD avoid using conventions or encodings that make trailing white
space significant; for encoding of binary data, MIME's "base64"
encoding is recommended. Implementors are warned that ISO C
implementations are not required to preserve trailing white space,
and special precautions may be necessary in implementations that do
not.

   NOTE: Unfortunately, the signature-delimiter convention (described
   in Section 4.3.2) does use significant trailing white space. It's
   too late to fix this; there is work underway on defining an
   organized signature convention as part of MIME, which is a
   preferable solution in the long run.

Posters are warned that some very old relayer software misbehaves
when the first non-empty line of an article body begins with white
space.

4.2. Headers

4.2.1. Names and Contents

Despite the restrictions on header-name syntax imposed by the
grammar, relayers and reading agents SHOULD tolerate header names
containing any ASCII printable character other than colon (":",
ASCII 58).

   NOTE: MAIL header names can contain any ASCII printable character
   (other than colon) in theory, but in practice, arbitrary header
   names are known to cause trouble for some news software. Section
   4.1's restriction to alphanumeric sequences separated by hyphens
   is believed to permit all widely used header names without causing

      problems for any widely used software. Software is nevertheless
      encouraged to cope correctly with the full range of possibilities,
      since aberrations are known to occur.

   Relayers MUST disregard headers not described in this Draft (that is,
   with header names not mentioned in this Draft) and pass them on
   unaltered.

   Posters wishing to convey non-standard information in headers SHOULD
   use header names beginning with "X-". No standard header name will
   ever be of this form. Reading agents SHOULD ignore "X-" headers, or
   at least treat them with great care.

   The order of headers in an article is not significant. However,
   posting agents are encouraged to put mandatory headers (see
   Section 5) first, followed by optional headers (see Section 6),
   followed by headers not defined in this Draft.

      NOTE: While relayers and reading agents must be prepared to handle
      any order, having the significant headers (the precise definition
      of "significant" depends on context) first can noticeably improve
      efficiency, especially in memory-limited environments where it is
      difficult to buffer up an arbitrary quantity of headers while
      searching for the few that matter.

   Header names are case-insensitive. There is a preferred case
   convention, which posters and posting agents SHOULD use: each hyphen-
   separated "word" has its initial letter (if any) in uppercase and the
   rest in lowercase, except that some abbreviations have all letters
   uppercase (e.g., "Message-ID" and "MIME-Version"). The forms used in
   this Draft are the preferred forms for the headers described herein.
   Relayers and reading agents are warned that articles might not obey
   this convention.

      NOTE: Although software must be prepared for the possibility of
      random use of case in header names (and other case-independent
      text), establishing a preferred convention reduces pointless
      diversity and may permit optimized software that looks for the
      preferred forms before resorting to less-efficient case-
      insensitive searches.

   In general, a header can consist of several lines, with each
   continuation line beginning with white space. The EOLs preceding
   continuation lines are ignored when processing such a header,
   effectively combining the start-line and the continuations into a
   single logical line. The logical line, less the header name, colon,
   and any white space following the colon, is the "header content".

4.2.2. Undesirable Headers

   A header whose content is empty is said to be an empty header.
   Relayers and reading agents SHOULD NOT consider presence or absence
   of an empty header to alter the semantics of an article (although
   syntactic rules, such as requirements that certain header names
   appear at most once in an article, MUST still be satisfied). Posting
   agents SHOULD delete empty headers from articles before posting them.

   Headers that merely state defaults explicitly (e.g., a Followup-To
   header with the same content as the Newsgroups header, or a MIME
   Content-Type header with contents "text/plain; charset=us-ascii") or
   state information that reading agents can typically determine easily
   themselves (e.g., the length of the body in octets) are redundant,
   conveying no information whatsoever. Headers that state information
   that cannot possibly be of use to a significant number of relayers,
   reading agents, or readers (e.g., the name of the software package
   used as the posting agent) are useless and pointless. Posters and
   posting agents SHOULD avoid including redundant or useless headers in
   articles.

      NOTE: Information that someone, somewhere, might someday find
      useful is best omitted from headers. (There's quite enough of it
      in article bodies.) Headers should contain information of known
      utility only. This is not meant to preclude inclusion of
      information primarily meant for news-software debugging, but such
      information should be included only if there is real reason,
      preferably based on experience, to suspect that it may be
      genuinely useful. Articles passing through gateways are the only
      obvious case where inclusion of debugging information appears
      clearly legitimate. (See Section 10.1.)

      NOTE: A useful rule of thumb for software implementors is: "if I
      had to pay a dollar a day for the transmission of this header,
      would I still think it worthwhile?".

4.2.3. White Space and Continuations

   The colon following the header name on the start-line MUST be
   followed by white space, even if the header is empty. If the header
   is not empty, at least some of the content MUST appear on the start-
   line. Posting agents MUST enforce these restrictions, but relayers
   (etc.) SHOULD accept even articles that violate them.

      NOTE: MAIL does not require white space after the colon, but it is
      usual. [RFC1036] required the white space, even in empty headers,
      and some existing software demands it. In MAIL, and arguably in
      [RFC1036] (although the wording is vague), it is technically

legitimate for the white space to be part of a continuation line
rather than the start-line, but not all existing software will
accept this. Deleting empty headers and placing some content on
the start-line avoids this issue; this is desirable because
trailing blanks, easily deleted by accident, are best not made
significant in headers.

In general, posters and posting agents SHOULD use blank (ASCII 32),
not tab (ASCII 9), where white space is desired in headers. Existing
software does not consistently accept tab as synonymous with blank in
all contexts. In particular, [RFC1036] appeared to specify that the
character immediately following the colon after a header name was
required to be a blank, and some news software insists on that, so
this character MUST be a blank. Again, posting agents MUST enforce
these restrictions but relayers SHOULD be more tolerant.

Since the white space beginning a continuation line remains a part of
the logical line, headers can be "broken" into multiple lines only at
white space. Posting agents SHOULD NOT break headers unnecessarily.
Relayers SHOULD preserve existing header breaks, and SHOULD NOT
introduce new breaks. Breaking headers SHOULD be a last resort;
relayers and reading agents SHOULD handle long header lines
gracefully. (See the discussion of size limits in Section 4.6.)

4.3. Body

Although the article body is unstructured for most of the purposes of
this Draft, structure MAY be imposed on it by other means, notably
MIME headers (see Appendix B).

4.3.1. Body Format Issues

The body of an article MAY be empty, although posting agents SHOULD
consider this an error condition (meriting returning the article to
the poster for revision). A posting agent that does not reject such
an article SHOULD issue a warning message to the poster and supply a
non-empty body. Note that the separator line MUST be present even if
the body is empty.

   NOTE: An empty body is probably a poster error except, arguably,
   for some control messages, and even they really ought to have a
   body explaining the reason for the control message. Some old
   reading agents are known to generate empty bodies for "cancel"
   control messages, so posting agents might opt not to reject
   bodyless articles in such cases (although it would be better to
   fix the reading agents to request a body). However, some existing
   news software is known to react badly to bodyless articles, hence
   the request for posting agents to insert a body in such cases.

NOTE: A possible posting-agent-supplied body text (already used by
one widespread posting agent) is "This article was probably
generated by a buggy news reader". (The use of "reader" to refer
to the reading agent is traditional, although this Draft uses more
precise terminology.)

NOTE: The requirement for the separator line even in a bodyless
article is inherited from MAIL and also distinguishes legitimately
bodyless articles from articles accidentally truncated in the
middle of the headers.

Note that an article body is a sequence of lines terminated by EOLs,
not arbitrary binary data, and in particular it MUST end with an EOL.
However, relayers SHOULD treat the body of an article as an
uninterpreted sequence of octets (except as mandated by changes of
EOL representation and by control-message processing) and SHOULD
avoid imposing constraints on it. See also Section 4.6.

4.3.2. Body Conventions

Although body lines can in principle be very long (see Section 4.6
for some discussion of length limits), posters SHOULD restrict body
line lengths to circa 70-75 characters. On systems where text is
conventionally stored with EOLs only at paragraph breaks and other
"hard return" points, with software breaking lines as appropriate for
display or manipulation, posting agents SHOULD insert EOLs as
necessary so that posted articles comply with this restriction.

NOTE: News originated in environments where line breaks in plain
text files were supplied by the user, not the software. Be this
good or bad, much reading-agent and posting-agent software assumes
that news articles follow this convention, so it is often
inconvenient to read or respond to articles that violate it. The
"70-75" number comes from the widespread use of display devices
that are 80 columns wide (with the number reduced to provide a bit
of margin for quoting, see below).

Reading agents confronted with body lines much longer than the
available output-device width SHOULD break lines as appropriate.
Posters are warned that such breaks may not occur exactly where the
poster intends.

NOTE: "As appropriate" would typically include breaking lines when
supplying the text of an article to be quoted in a reply or
followup, something that line-breaking reading agents often
neglect to do now.

Although styles vary widely, for plain text it is usual to use no
left margin, leave the right edge ragged, use a single empty line to
separate paragraphs, and employ normal natural-language usage on
matters such as upper/lowercase. (In particular, articles SHOULD NOT
be written entirely in uppercase. In environments where posters have
access only to uppercase, posting agents SHOULD translate it to
lowercase.)

    NOTE: Most people find substantial bodies of text entirely in
    uppercase relatively hard to read, while all-lowercase text merely
    looks slightly odd. The common association of uppercase with
    strong emphasis adds to this.

Tone of voice does not carry well in written text, and
misunderstandings are common when sarcasm, parody, or exaggeration
for humorous effect is attempted without explicit warning. It has
become conventional to use the sequence ":-)", which (on most output
devices) resembles a rotated "smiley face" symbol, as a marker for
text not meant to be taken literally, especially when humor is
intended. This practice aids communication and averts unintended
ill-will; posters are urged to use it. A variety of analogous
sequences are used with less-standardized meanings [Sanderson].

The order of arrival of news articles at a particular host depends
somewhat on transmission paths, and occasionally articles are lost
for various reasons. When responding to a previous article, posters
SHOULD NOT assume that all readers understand the exact context. It
is common to quote some of the previous article to establish context.
This SHOULD be done by prefacing each quoted line (even if it is
empty) with the character ">". This will result in multiple levels
of ">" when quoted context itself contains quoted context.

    NOTE: It may seem superfluous to put a prefix on empty lines, but
    it simplifies implementation of functions such as "skip all quoted
    text" in reading agents.

Readability is enhanced if quoted text and new text are separated by
an empty line.

Posters SHOULD edit quoted context to trim it down to the minimum
necessary. However, posting agents SHOULD NOT attempt to enforce
this by imposing overly simplistic rules like "no more than 50% of
the lines should be quotes".

    NOTE: While encouraging trimming is desirable, the 50% rule
    imposed by some old posting agents is both inadequate and
    counterproductive. Posters do not respond to it by being more
    selective about quoting; they respond by padding short responses,

or by using different quoting styles to defeat automatic analysis.
The former adds unnecessary noise and volume, while the latter
also defeats more useful forms of automatic analysis that reading
agents might wish to do.

NOTE: At the very least, if a minimum-unquoted quota is being set,
article bodies shorter than (say) 20 lines, or perhaps articles
that exceed the quota by only a few lines, should be exempt. This
avoids the ridiculous situation of complaining about a 5-line
response to a 6-line quote.

NOTE: A more subtle posting-agent rule, suggested for experimental
use, is to reject articles that appear to contain quoted
signatures (see below). This is almost certainly the result of a
careless poster not bothering to trim down quoted context. Also,
if a posting agent or followup agent presents an article template
to the poster for editing, it really should take note of whether
the poster actually made any changes, and refrain from posting an
unmodified template.

Some followup agents supply "attribution" lines for quoted context,
indicating where it first appeared and under whose name. When
multiple levels of quoting are present and quoted context is edited
for brevity, "inner" attribution lines are not always retained. The
editing process is also somewhat error-prone. Reading agents (and
readers) are warned not to assume that attributions are accurate.

UNRESOLVED ISSUE: Should a standard format for attribution lines
be defined? There is already considerable diversity, but
automatic news analysis would be substantially aided by a standard
convention.

Early difficulties in inferring return addresses from article headers
led to "signatures": short closing texts, automatically added to the
end of articles by posting agents, identifying the poster and giving
his network addresses, etc. If a poster or posting agent does append
a signature to an article, the signature SHOULD be preceded with a
delimiter line containing (only) two hyphens (ASCII 45) followed by
one blank (ASCII 32). Posting agents SHOULD limit the length of
signatures, since verbose excess bordering on abuse is common if no
restraint is imposed; 4 lines is a common limit.

NOTE: While signatures are arguably a blemish, they are a well-
understood convention, and conveying the same information in
headers exposes it to mangling and makes it rather less
conspicuous. A standard delimiter line makes it possible for
reading agents to handle signatures specially if desired.

(This is unfortunately hampered by extensive misunderstanding of,
and misuse of, the delimiter.)

NOTE: The choice of delimiter is somewhat unfortunate, since it
relies on preservation of trailing white space, but it is too
well-established to change. There is work underway to define a
more sophisticated signature scheme as part of MIME, and this will
presumably supersede the current convention in due time.

NOTE: Four 75-column lines of signature text is 300 characters,
which is ample to convey name and mail-address information in all
but the most bizarre situations.

4.4. Characters and Character Sets

Header and body lines MAY contain any ASCII characters other than CR
(ASCII 13), LF (ASCII 10), and NUL (ASCII 0).

NOTE: CR and LF are excluded because they clash with common EOL
conventions. NUL is excluded because it clashes with the C
end-of-string convention, which is significant to most existing
news software. These three characters are unlikely to be
transmitted successfully.

However, posters SHOULD avoid using ASCII control characters except
for tab (ASCII 9), formfeed (ASCII 12), and backspace (ASCII 8). Tab
signifies sufficient horizontal white space to reach the next of a
set of fixed positions; posters are warned that there is no standard
set of positions, so tabs should be avoided if precise spacing is
essential. Formfeed signifies a point at which a reading agent
SHOULD pause and await reader interaction before displaying further
text. Backspace SHOULD be used only for underlining, done by a
sequence of underscores (ASCII 95) followed by an equal number of
backspaces, signifying that the same number of text characters
following are to be underlined. Posters are warned that underlining
is not available on all output devices and is best not relied on for
essential meaning. Reading agents SHOULD recognize underlining and
translate it to the appropriate commands for devices that support it.

NOTE: Interpretation of almost all control characters is device-
specific to some degree, and devices differ. Tabs and underlining
are supported, to some extent, by most modern devices and reading
agents, hence the cautious exemptions for them. The underlining
method is specified because the inverse method, text and then
underscores, is tempting to the naive; however, if sent unaltered
to a device that shows only the most recent of several overstruck
characters rather than a composite, the result can be utterly
unreadable.

NOTE: A common interpretation of tab is that it is a request to
space forward to the next position whose number is one more than a
multiple of 8, with positions numbered sequentially starting at 1.
(So tab positions are 9, 17, 25, ...) Reading agents not
constrained by existing system conventions might wish to use this
interpretation.

NOTE: It will typically be necessary for a reading agent to catch
and interpret formfeed, not just send it to the output device.
The actions performed by typical output devices on receiving a
formfeed are neither adequate for, nor appropriate to, the pause-
for-interaction meaning.

Cooperating subnets that wish to employ non-ASCII character sets by
using escape sequences (employing, e.g., ESC (ASCII 27), SO
(ASCII 14), and SI (ASCII 15)) to alter the meaning of superficially
ASCII characters MAY do so, but MUST use MIME headers to alert
reading agents to the particular character set(s) and escape
sequences in use. A reading agent SHOULD NOT pass such an escape
sequence through, unaltered, to the output device unless the agent
confirms that the sequence is one used to affect character sets and
has reason to believe that the device is capable of interpreting that
particular sequence properly.

NOTE: Cooperating-subnet organizers are warned that some very old
relayers strip certain control characters out of articles they
pass along. ESC is known to be among the affected characters.

NOTE: There are now standard Internet encodings for Japanese
[RFC1345] and Vietnamese [RFC1456] in particular.

Articles MUST NOT contain any octet with value exceeding 127, i.e.,
any octet that is not an ASCII character.

NOTE: This rule, like others, may be relaxed by unanimous consent
of the members of a cooperating subnet, provided suitable
precautions are taken to ensure that rule-violating articles do
not leak out of the subnet. (This has already been done in many
areas where ASCII is not adequate for the local language(s).)
Beware that articles containing non-ASCII octets in headers are a
violation of the MAIL specifications and are not valid MAIL
messages. MIME offers a way to encode non-ASCII characters in
ASCII for use in headers; see Section 4.5.

NOTE: While there is great interest in using 8-bit character sets,
not all software can yet handle them correctly, hence the
restriction to cooperating subnets. MIME encodings can be used to
transmit such characters while remaining within the octet
restriction.

In anticipation of the day when it is possible to use non-ASCII
characters safely anywhere, and to provide for the (substantial)
cooperating subnets that are already using them, transmission paths
SHOULD treat news articles as uninterpreted sequences of octets
(except perhaps for transformations between EOL representations) and
relayers SHOULD treat non-ASCII characters in articles as ordinary
characters.

NOTE: 8-bit enthusiasts are warned that not all software conforms
to these recommendations yet. In particular, standard NNTP
[RFC977] is a 7-bit protocol {but in [RFC3977] it has been upped
to 8-bit}, and there may be implementations that enforce this
rule. Be warned, also, that it will never be safe to send raw
binary data in the body of news articles, because changes of EOL
representation may (will!) corrupt it.

Except where cooperating subnets permit more direct approaches, MIME
headers and encodings SHOULD be used to transmit non-ASCII content
using ASCII characters; see Section 4.5, Appendix B, and the MIME
RFCs for details. If article content can be expressed in ASCII, it
SHOULD be. Failing that, the order of preference for character sets
is that described in MIME.

NOTE: Using the MIME facilities, it is possible to transmit ANY
character set, and ANY form of binary data, using only ASCII
characters. Equally important, such articles are self-describing
and the reading agent can tell which octet-to-symbol mapping is
intended! Designation of some preferred character sets is
intended to minimize the number of character sets that a reading
agent must understand in order to display most articles properly.

Articles containing non-ASCII characters, articles using ASCII
characters (values 0 through 127) to refer to non-ASCII symbols, and
articles using escape sequences to shift character sets SHOULD
include MIME headers indicating which character set(s) and
conventions are being used. They MUST do so unless such articles are
strictly confined to a cooperating subnet that has its own pre-agreed
conventions. MIME encodings are preferred over all of these
techniques. If it comes to a relayer's attention that it is being
asked to pass an article using such techniques outward across what it
knows to be the boundary of such a cooperating subnet, it MUST report

   this error to its administrator and MAY refuse to pass the article
   beyond the subnet boundary. If it does pass the article, it MUST
   re-encode it with MIME encodings to make it conform to this Draft.

      NOTE: Such re-encoding is a non-trivial task, due to MIME rules
      such as the prohibition of nested encodings. It's not just a
      matter of pouring the body through a simple filter.

   Reading agents SHOULD note MIME headers and attempt to show the
   reader the closest possible approximation to the intended content.
   They SHOULD NOT just send the octets of the article to the output
   device unaltered, unless there is reason to believe that the output
   device will indeed interpret them correctly. Reading agents MUST NOT
   pass ASCII control characters or escape sequences, other than as
   discussed above, unaltered to the output device; only by chance would
   the result be the desired one, and there is serious potential for
   harmful side effects, either accidental or malicious.

      NOTE: Exactly what to do with unwanted control
      characters/sequences depends on the philosophy of the reading
      agent, but passing them straight to the output device is almost
      always wrong. If the reading agent wants to mark the presence of
      such a character/sequence in circumstances where only ASCII
      printable characters are available, translating it to "#" might be
      a suitable method; "#" is a conspicuous character seldom used in
      normal text.

      NOTE: Reading agents should be aware that many old output devices
      (or the transmission paths to them) zero out the top bit of octets
      sent to them. This can transform non-ASCII characters into ASCII
      control characters.

   Followup agents MUST be careful to apply appropriate transformations
   of representation to the outbound followup as well as the inbound
   precursor. A followup to an article containing non-ASCII material is
   very likely to contain non-ASCII material itself.

4.5. Non-ASCII Characters in Headers

   All octets found in headers MUST be ASCII characters. However, it is
   desirable to have a way of encoding non-ASCII characters, especially
   in "human-readable" headers such as Subject. MIME provides a way to
   do this. Full details may be found in the MIME specifications;
   herewith a quick summary to alert software authors to the issues.

```
        encoded-word  = "=?" charset "?" encoding "?" codes "?="
        charset       = 1*tag-char
        encoding      = 1*tag-char
        tag-char      = < ASCII printable character except
                               !()<>@,;:\"[]/?= >
        codes         = 1*code-char
        code-char     = <ASCII printable character except ?>
```

An encoded word is a sequence of ASCII printable characters that
specifies the character set, encoding method, and bits of
(potentially) non-ASCII characters. Encoded words are allowed only
in certain positions in certain headers. Specific headers impose
restrictions on the content of encoded words beyond that specified in
this section. Posting agents MUST ensure that any material
resembling an encoded word (complete with all delimiters), in a
context where encoded words may appear, really is an encoded word.

    NOTE: The syntax is a bit ugly, but it was designed to minimize
    chances of confusion with legitimate header contents, and to
    satisfy difficult constraints on use within existing headers.

An encoded word MUST NOT be more than 75 octets long. Each line of a
header containing encoded word(s) MUST be at most 76 octets long, not
counting the EOL.

    NOTE: These limits are meant to bound the lookahead needed to
    determine whether text that begins with "=?" is really an encoded
    word.

The details of charsets and encodings are defined by MIME; the
sequence of preferred character sets is the same as MIME's. Encoded
words SHOULD NOT be used for content expressible in ASCII.

When an encoded word is used, other than in a newsgroup name (see
Section 5.5), it MUST be separated from any adjacent non-space
characters (including other encoded words) by white space. Reading
agents displaying the contents of encoded words (as opposed to their
encoded form) should ignore white space adjacent to encoded words.

    UNRESOLVED ISSUE: Should this section be deleted entirely, or made
    much more terse? The material is relevant, but too complex to
    discuss fully.

    NOTE: The deletion of intervening white space permits using
    multiple encoded words, implicitly concatenated by the deletion,
    to encode text that will not fit within a single 75-character
    encoded word.

Reading-agent implementors are warned that although this Draft
completely specifies where encoded words may appear in the headers it
defines, there are other headers (e.g., the MIME Content-Description
header) that MAY contain them.

4.6. Size Limits

Implementations SHOULD avoid fixed constraints on the sizes of lines
within an article and on the size of the entire article.

Relayers SHOULD treat the body of an article as an uninterpreted
sequence of octets (except as mandated by changes of EOL
representation and processing of control messages), not to be altered
or constrained in any way.

If it is absolutely necessary for an implementation to impose a limit
on the length of header lines, body lines, or header logical lines,
that limit shall be at least 1000 octets, including EOL
representations. Relayers and transmission paths confronted with
lines beyond their internal limits (if any) MUST NOT simply inject
EOLs at random places; they MAY break headers (as described in
Section 4.2.3) as a last resort, and otherwise they MUST either pass
the long lines through unaltered, or refuse to pass the article at
all (see Section 9.1 for further discussion).

   NOTE: The limit here is essentially the same minimum as that
   specified for SMTP mail [RFC821]. Implementors are warned that
   Path (see Section 5.6) and References (see Section 6.5) headers,
   in particular, often become several hundred characters long, so
   1000 is not an overly generous limit.

All implementations MUST be able to handle an article totalling at
least 65,000 octets, including headers and EOL representations,
gracefully and efficiently. All implementations SHOULD be able to
handle an article totalling at least 1,000,000 (one million) octets,
including headers and EOL representations, gracefully and
efficiently. "Gracefully and efficiently" is intended to preclude
not only failures, but also major loss of performance, serious
problems in error recovery, or resource consumption beyond what is
reasonably necessary.

   NOTE: The intent here is to prohibit lowering the existing de
   facto limit any further, while strongly encouraging movement
   towards a higher one. Actually, although improvements are
   desirable in some cases, much news software copes reasonably well
   with very large articles. The same cannot be said of the
   communications software and protocols used to transmit news from
   one host to another, especially when slow communications links are

involved. Occasional huge articles that appear now (by accident
or through ignorance) typically leave trails of failing software,
system problems, and irate administrators in their wake.

   NOTE: It is intended that the successor to this Draft will raise
   the "MUST" limit to 1,000,000 and the "SHOULD" limit still
   further.

Posters SHOULD limit posted articles to at most 60,000 octets,
including headers and EOL representations, unless the articles are
being posted only within a cooperating subnet that is known to be
capable of handling larger articles gracefully. Posting agents
presented with a large article SHOULD warn the poster and request
confirmation.

   NOTE: The difference between this and the earlier "MUST" limit is
   due to margin for header growth, differing EOL representations,
   and transmission overheads.

   NOTE: Disagreeable though these limits are, it is a fact that in
   current networks, an article larger than 64K (after header growth,
   etc.) simply is not transmitted reliably. Note also the comments
   above on the trauma caused by single extremely large articles now;
   the problems are real and current. These problems arguably should
   be fixed, but this will not happen network-wide in the immediate
   future, hence the restriction of larger articles to cooperating
   subnets, for now.

Posters using non-ASCII characters in their text MUST take into
account the overhead involved in MIME encoding, unless the article's
propagation will be entirely limited to a cooperating subnet that
does not use MIME encodings for non-ASCII characters. For example,
MIME base64 encoding involves growth by a factor of approximately
4/3, so an article that would likely have to use this encoding should
be at most about 45,000 octets before encoding.

Posters SHOULD use MIME "message/partial" conventions to facilitate
automatic reassembly of a large document split into smaller pieces
for posting. It is recommended that the content identifier used
should be a message ID, generated by the same means as article
message IDs (see Section 5.3), and that all parts should have a
See-Also header (see Section 6.16) giving the message IDs of at least
the previous parts and preferably all of the parts.

   NOTE: See-Also is more correct for this purpose than References,
   although References is in common use today (with less-formal
   reassembly arrangements). MIME reassemblers should probably

    examine articles suggested by References headers if See-Also
    headers are not present to indicate the whereabouts of the other
    parts of "message/partial" articles.

   To repeat: implementations SHOULD avoid fixed constraints on the
   sizes of lines within an article and on the size of the entire
   article.

4.7. Example

   Here is a sample article:

       From: jerry@eagle.ATT.COM (Jerry Schwarz)
       Path: cbosgd!mhuxj!mhuxt!eagle!jerry
       Newsgroups: news.announce
       Subject: Usenet Etiquette -- Please Read
       Message-ID: <642@eagle.ATT.COM>
       Date: Mon, 17 Jan 1994 11:14:55 -0500 (EST)
       Followup-To: news.misc
       Expires: Wed, 19 Jan 1994 00:00:00 -0500
       Organization: AT&T Bell Laboratories, Murray Hill

       body
       body
       body

5. Mandatory Headers

   An article MUST have one, and only one, of each of the following
   headers: Date, From, Message-ID, Subject, Newsgroups, Path.

      NOTE: MAIL specifies (if read most carefully) that there must be
      exactly one Date header and exactly one From header, but otherwise
      does not restrict multiple appearances of headers. (Notably, it
      permits multiple Message-ID headers!) This appears singularly
      useless, or even harmful, in the context of news, and much current
      news software will not tolerate multiple appearances of mandatory
      headers.

   Note also that there are situations, discussed in the relevant parts
   of Section 6, where References, Sender, or Approved headers are
   mandatory.

   In the discussions of the individual headers, the content of each is
   specified using the syntax notation. The convention used is that the
   content of, for example, the Subject header is defined as
   <Subject-content>.

5.1. Date

   The Date header contains the date and time when the article was
   submitted for transmission:

       Date-content = [weekday "," space] date space time
       weekday   = "Mon" / "Tue" / "Wed" / "Thu"
                     / "Fri" / "Sat" / "Sun"
       date   = day space month space year
       day   = 1*2digit
       month   = "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun"
                     / "Jul" / "Aug" / "Sep" / "Oct" / "Nov" / "Dec"
       year   = 4digit / 2digit
       time   = hh ":" mm [":" ss] space timezone
       timezone   = "UT" / "GMT"
                     / ("+" / "-") hh mm [space "(" zone-name ")"]
       hh   = 2digit
       mm   = 2digit
       ss   = 2digit
       zone-name   = 1*(<ASCII printable character except ()\>
                     / space)

   This is a restricted subset of the MAIL date format.

   If a weekday is given, it MUST be consistent with the date. The
   modern Gregorian calendar is used, and dates MUST be consistent with
   its usual conventions; for example, if the month is May, the day must
   be between 1 and 31 inclusive. The year SHOULD be given as four
   digits, and posting agents SHOULD enforce this; however, relayers
   MUST accept the two-digit form, and MUST interpret it as having the
   implicit prefix "19".

      NOTE: Two-digit year numbers can, should, and must be phased out
      by 1999.

   The time is given on the 24-hour clock, e.g., two hours before
   midnight is "22:00" or "22:00:00". The hh must be between 00 and 23
   inclusive, the mm between 0 and 59 inclusive, and the ss between 0
   and 60 inclusive.

      NOTE: Leap seconds very occasionally result in minutes that are 61
      seconds long.

   The date and time SHOULD be given in the poster's local time zone,
   including a specification of that time zone as a numeric offset
   (which SHOULD include the time zone name, e.g., "EST", supplied in
   parentheses like a MAIL comment). If not, they MUST be given in
   Universal Time (abbreviated "UT"; "GMT" is a historical synonym for

"UT"). The time zone name in parentheses, if present, is a comment;
software MUST ignore it, except that reading agents might wish to
display it to the reader. Time zone names other than "UT" and "GMT"
MUST appear only in the comment.

   NOTE: Attempts to deal with a full set of time zone names have all
   foundered on the vast number of such names in use and the
   duplications (for example, there are at least FIVE different time
   zones called "EST" by somebody). Even the limited set of North
   American zone names authorized by MAIL is subject to confusion and
   misinterpretation, hence the flat ban on non-UT time zone names,
   except as comments.

   NOTE: [RFC1036] specified that use of GMT (aka UT, UTC) was
   preferred. However, the local time (in the poster's time zone) is
   arguably information of possible interest to the reader, and this
   requires some indication of the poster's time zone. Numeric
   offsets are an unambiguous way of doing this, and their use was
   indeed sanctioned by [RFC1036] (that is, this is a change of
   preference only).

   NOTE: There is frequent confusion, including errors in some news
   software, regarding the sign of numeric time zones. Zones west of
   Greenwich have negative offsets. For example, North American
   Eastern Standard Time is zone -0500 and North American Eastern
   Daylight Time is zone -0400.

   NOTE: Implementors are warned that the hh in a time zone can go up
   to about 14; it is not limited to 12. This is because the
   International Date Line does not run exactly along the boundary
   between zone -1200 and zone +1200.

   NOTE: The comments in Section 2.6 regarding translation to other
   languages are relevant here. The Date-content format, and the
   spellings of its components, as found in articles themselves, are
   always as defined in this Draft, regardless of the language used
   to interact with readers and posters. Reading and posting agents
   should translate as appropriate. Actually, even English-language
   reading and posting agents will probably want to do some degree of
   translation on dates, if only to abbreviate the lengthy format and
   (perhaps) translate to and from the reader's time zone.

5.2. From

   The From header contains the electronic address, and possibly the
   full name, of the article's author:

      From-content = address [space "(" paren-phrase ")"]
                    / [plain-phrase space] "<" address ">"
      paren-phrase = 1*(paren-char / space / encoded-word)
      paren-char   = <ASCII printable character except ()<>\>
      plain-phrase = plain-word *(space plain-word)
      plain-word   = unquoted-word / quoted-word / encoded-word
      unquoted-word = 1*unquoted-char
      unquoted-char = <ASCII printable character except !()<>@,;:\".[]>
      quoted-word   = quote 1*(quoted-char / space) quote
      quote   = <" (ASCII 34)>
      quoted-char   = <ASCII printable character except "()<>\>
      address   = local-part "@" domain
      local-part   = unquoted-word *("." unquoted-word)
      domain   = unquoted-word *("." unquoted-word)

   (Encoded words are described in Section 4.5.) The full name is
   distinguished from the electronic address either by enclosing the
   former in parentheses (making it resemble a MAIL comment, after the
   address) or by enclosing the latter in angle brackets. The second
   form is preferred. In the first form, encoded words inside the full
   name MUST be composed entirely of <paren-char>s. In the second form,
   encoded words inside the full name may not contain characters other
   than letters (of either case), digits, and the characters "!", "*",
   "+", "-", "/", "=", and "_". The local part is case-sensitive
   (except that all case counterparts of "postmaster" are deemed
   equivalent), the domain is case-insensitive, and all other parts of
   the From content are comments that MUST be ignored by news software
   (except insofar as reading agents may wish to display them to the
   reader). Posters and posting agents MUST restrict themselves to this
   subset of the MAIL From syntax; relayers MAY accept a broader subset,
   but see the discussion in Section 9.1.

      NOTE: The syntax here is a restricted subset of the MAIL From
      syntax, with quoting particularly restricted, for simple parsing.
      In particular, the presence of "<" in the From content indicates
      that the second form is being used; otherwise, the first form is
      being used. The major restrictions here are those already de
      facto imposed by existing software.

      NOTE: Overly lenient posting agents sometimes permit the second
      form with a full name containing "(" or ")", but it is extremely
      rare for a full name to contain "<" or ">", even in mail.

Accordingly, reading agents wishing to robustly determine which
form is in use in a particular article should key on the presence
or absence of "<", not the presence or absence of "(".

The address SHOULD be a valid and complete Internet domain address,
capable of being successfully mailed to by an Internet host (possibly
via an MX (Mail Exchange) record and a forwarder). The pseudo-domain
".uucp" MAY be used for hosts registered in the UUCP maps (e.g., name
"xyz.uucp" for registered site "xyz"), but such hosts SHOULD
discontinue this usage (either by arranging a proper Internet address
and forwarder, or by using the "% hack" (see below)), as soon as
possible. Bitnet hosts SHOULD use Internet addresses, avoiding the
obsolescent ".bitnet" pseudo-domain. Other forms of address MUST NOT
be used.

    NOTE: "Other forms" specifically include UK-style "backward"
    domains ("uk.oxbridge.cs" is in the Czech Republic, not the UK),
    pure-UUCP addressing ("knee!shin!foot" instead of
    "foot%shin@knee.uucp"), and abbreviated domains ("zebra.zoo"
    instead of "zebra.zoo.toronto.edu").

If it is necessary to use the local part to specify a routing
relative to the nearest Internet host, this MUST be done using the "%
hack", using "%" as a secondary "@". For example, to specify that
mail to the address should go to Internet host "foo.bar.edu", then to
non-Internet host "ein", then to non-Internet host "deux", for
delivery there to mailbox "fred", a suitable address would be:

    fred%deux%ein@foo.bar.edu

Analogous forms using "!" in the local part MUST NOT be used, as they
are ambiguous; they should be expressed in the "%" form.

    NOTE: "a!b@c" can be interpreted as either "b%c@a" or "b%a@c", and
    there is no consistency in which choice is made. Such addresses
    consequently are unreliable. The "%" form does not suffer from
    this problem, and although its use is officially discouraged, it
    is a de facto standard, to the point that MAIL recognizes it.

Relayers MUST NOT, repeat MUST NOT, repeat MUST NOT, rewrite From
lines, in any way, however minor or seemingly innocent. Trying to
"fix" a non-conforming address has a very high probability of making
things worse. Either pass it along unchanged or reject the article.

    NOTE: An additional reason for banning the use of "!" addressing
    is that it has a much higher probability of being rewritten into
    mangled unrecognizability by old relayers.

Posters and posting agents SHOULD avoid use of the characters "!" and
"@" in full names, as they may trigger unwanted header rewriting by
old, simple-minded news software.

   NOTE: Also, the characters "." and ",", not infrequently found in
   names (e.g., "John W. Campbell, Jr."), are NOT, repeat NOT,
   allowed in an unquoted word. A From header like the following
   MUST NOT be written without the quotation marks:

   From: "John W. Campbell, Jr." <editor@analog.com>

5.3. Message-ID

   The Message-ID header contains the article's message ID, a unique
   identifier distinguishing the article from every other article:

      Message-ID-content = message-id
      message-id   = "<" local-part "@" domain ">"

   As with From addresses, a message ID's local part is case-sensitive,
   and its domain is case-insensitive. The "<" and ">" are parts of the
   message ID, not peculiarities of the Message-ID header.

      NOTE: News message IDs are a restricted subset of MAIL message
      IDs. In particular, no existing news software copes properly with
      MAIL quoting conventions within the local part, so they are
      forbidden. This is unfortunate, particularly for X.400 gateways
      that often wish to include characters that are not legal in
      unquoted message IDs, but it is impossible to fix net-wide. See
      the notes on gatewaying in Section 10.

   The domain in the message ID SHOULD be the full Internet domain name
   of the posting agent's host. Use of the ".uucp" pseudo-domain (for
   hosts registered in the UUCP maps) or the ".bitnet" pseudo-domain
   (for Bitnet hosts) is permissible but SHOULD be avoided.

   Posters and posting agents MUST generate the local part of a message
   ID using an algorithm that obeys the specified syntax (words
   separated by ".", with certain characters not permitted) (see Section
   5.2 for details) and will not repeat itself (ever). The algorithm
   SHOULD NOT generate message IDs that differ only in case of letters.
   Note the specification in Section 6.5 of a recommended convention for
   indicating subject changes. Otherwise, the algorithm is up to the
   implementor.

      NOTE: The crucial use of message IDs is to distinguish circulating
      articles from each other and from articles circulated recently.
      They are also potentially useful as permanent indexing keys, hence

the requirement for permanent uniqueness, but indexers cannot
absolutely rely on this because the earlier RFCs urged it but did
not demand it. All major implementations have always generated
permanently unique message IDs by design, but in some cases this
is sensitive to proper administration, and duplicates may have
occurred by accident.

   NOTE: The most popular method of generating local parts is to use
   the date and time, plus some way of distinguishing between
   simultaneous postings on the same host (e.g., a process number),
   and encode them in a suitably restricted alphabet. An older but
   now less-popular alternative is to use a sequence number,
   incremented each time the host generates a new message ID; this is
   workable but requires careful design to cope properly with
   simultaneous posting attempts, and it is not as robust in the
   presence of crashes and other malfunctions.

   NOTE: Some buggy news software considers message IDs completely
   case-insensitive, hence the advice to avoid relying on case
   distinctions. The restrictions placed on the "alphabet" of local
   parts and domains in Section 5.2 have the useful side effect of
   making it unnecessary to parse message IDs in complex ways to
   break them into case-sensitive and case-insensitive portions.

The local part of a message ID MUST NOT be "postmaster" or any other
string that would compare equal to "postmaster" in a case-insensitive
comparison. Message IDs MUST be no longer than 250 octets, including
the "<" and ">".

   NOTE: "Postmaster" is an irksome exception to case-sensitivity in
   local parts, inherited from MAIL, and simply avoiding it is the
   best way to deal with it (not that it's likely, but the issue
   needs to be dealt with). The length limit is undesirable but is
   present in widely used existing software. The limit is actually
   255, but a small safety margin is wise.

5.4. Subject

The Subject header's content (the "subject" of the article) is a
short phrase describing the topic of the article:

   Subject-content = ["Re: "] nonblank-text

Encoded words MAY appear in this header.

If the article is a followup, the subject SHOULD begin with "Re: " (a
"back reference"). If the article is not a followup, the subject
MUST NOT begin with a back reference. Back references are case-
insensitive, although "Re: " is the preferred form. A followup agent
assisting a poster in preparing a followup SHOULD prepend a back
reference, UNLESS the subject already begins with one. If the poster
determines that the topic of the followup differs significantly from
what is described in the subject, a new, more descriptive subject
SHOULD be substituted (with no back reference). An article whose
subject begins with a back reference MUST have a References header
referencing the precursor.

    NOTE: A back reference is FOUR characters, the fourth being a
    blank. [RFC1036] was confused about this. Observe also that only
    ONE back reference should be present.

    NOTE: There is a semi-standard convention, often used, in which a
    subject change is flagged by making the new Subject-content of the
    form:

    new topic (was: old topic)

    possibly with "old topic" somewhat truncated. Posters wishing to
    do something like this are urged to use this exact form, to
    simplify automated analysis.

For historical reasons, the subject MUST NOT begin with "cmsg " (note
that this sequence ends with a blank).

    NOTE: Some old news software takes a subject beginning with
    "cmsg " as an indication that the article is a control message
    (see Sections 6.6 and 7). This mechanism is obsolete and
    undesirable, but accidental triggering of it is still possible.

The subject SHOULD be terse. Posters SHOULD avoid trying to cram
their entire article into the headers; even the simplest query
usually benefits from a sentence or two of elaboration and context,
and the details of header display vary widely among reading agents.

    NOTE: All-in-the-subject articles are sometimes the result of
    misunderstandings over the interaction protocol of a posting
    agent. Posting agents might wish to give special attention to the
    possibility that a poster specifying a very long subject might
    have thought he was typing the body of the article.

5.5. Newsgroups

   The Newsgroups header's content specifies to which newsgroup(s) the
   article is posted:

      Newsgroups-content = newsgroup-name *(ng-delim newsgroup-name)
      newsgroup-name   = plain-component *("." component)
      component   = plain-component / encoded-word
      plain-component   = component-start *13component-rest
      component-start   = lowercase / digit
      lowercase   = <letter a-z>
      component-rest   = component-start / "+" / "-" / "_"
      ng-delim   = ","

   Encoded words used in newsgroup names MUST NOT contain characters
   other than letters, digits, "+", "-", "/", "_", "=", and "?"
   (although they may encode them).

   A newsgroup name consists of one or more components, which may be
   plain components or (except for the first) encoded words. A plain
   component MUST contain at least one letter, MUST begin with a letter
   or digit, and MUST NOT be longer than 14 characters. The first
   component MUST begin with a letter; subsequent components SHOULD
   begin with a letter. Newsgroup names MUST NOT contain uppercase
   letters, except where required by encodings in encoded words. The
   sequences "all" and "ctl" MUST NOT be used as components.

      NOTE: The alphabet and syntax specified encompasses all existing
      names of widespread newsgroups, while avoiding various forms that
      are known to cause problems. Important existing software uses
      various non-alphanumeric characters as punctuation adjacent to
      newsgroup names. (It would, in fact, be preferable to ban "+"
      from newsgroup names, were it not that several widespread
      newsgroups related to the C++ programming language already use
      it.)

      NOTE: Much existing software converts the newsgroup name into a
      directory path and stores the articles themselves using numeric
      filenames, so all-digit name components can be troublesome; the
      "Great Renaming" early in the history of Usenet included revisions
      of several newsgroup names to eliminate such components.

      NOTE: The same storage technique is the reason for the
      14-character limit. The limit is now largely historical, since
      most modern systems have much larger limits on the length of a
      directory entry's name, but many old systems are still in use.
      Systems with shorter limits also exist, but news software on such
      systems has had to deal with the problem already, since there are

several widespread newsgroups with 14-character components in
their names. Implementors are warned that it is intended that the
successor to this Draft will increase the 14-character limit, and
they are urged to fix their software to handle longer names
gracefully (if such fixes are necessary, given the intended domain
of application of the particular software).

NOTE: The requirement that the first character of a name be a
letter accommodates existing software that assumes it can tell the
difference between a newsgroup name and other possible syntactic
entities by inspecting the first character. Similar
considerations motivate excluding "+", "-", and "_" from coming
first in a component, and the preference for components that do
not begin with digits. The "all" sequence is used as a wildcard
symbol in much existing software, and the "ctl" sequence was
involved in an obsolete historical mechanism for marking control
messages, so they are best avoided.

NOTE: Possibly newsgroup names should have been case-insensitive,
but all existing software treats them as case-sensitive.
([RFC977] claims that they are case-insensitive in NNTP, but
existing implementations are believed to ignore this.) The
simplest solution is just to ban use of uppercase letters, since
no widespread newsgroup name uses them anyway; this avoids any
possibility of confusion.

NOTE: The syntax has the disadvantage of containing no white
space, making it impossible to continue a Newsgroups header across
several lines. Implementors of relayers and reading agents are
warned that it is intended that the successor to this Draft will
change the definition of ng-delim to:

ng-delim = "," [space]

and are urged to fix their software to handle (i.e., ignore) white
space following the commas. Meanwhile, posters must avoid
inserting such space (despite the natural-language convention that
permits it), and posting agents should strip it out.

NOTE: Encoded words as components are somewhat problematic but are
clearly desirable for use in non-English-speaking nations. They
are not subject to the 14-character limit, and this (plus the
possibility of "/" within them) may require special handling in
news software.

Encoded words are allowed in newsgroup names ONLY where non-ASCII
characters are necessary to the name, and they must use the "b"
encoding [RFC2045] and the first suitable character set in the MIME
order of preferred character sets [RFC2047] {ASCII before ISO-8859-*
before anything else}.

> NOTE: Since the newsgroup name is the encoded form, NOT the
> underlying non-ASCII form, there is room for terrible confusion
> here if the choice of encoding for a particular name is not fully
> standardized.

Posters SHOULD use only the names of existing newsgroups in the
Newsgroups header, because newsgroups are NOT created simply by being
posted to. However, it is legitimate to cross-post to newsgroup(s)
that do not exist on the posting agent's host, provided that at least
one of the newsgroups DOES exist there, and followup agents MUST
accept this (posting agents MAY accept it, but SHOULD at least alert
the poster to the situation and request confirmation). Relayers MUST
NOT rewrite Newsgroups headers in any way, even if some or all of the
newsgroups do not exist on the relayer's host.

> NOTE: Early experience with news software that created newsgroups
> when they were mentioned in a Newsgroups header was thoroughly
> negative: posters frequently mistype newsgroup names.

> NOTE: While it is legitimate for some of an article's newsgroups
> not to exist on the host where it is posted, this IS a rather
> unusual situation except in followups (which should go to all
> newsgroups the precursor was posted to, even if not all of them
> reach the site where the followup is being posted).

> NOTE: Rewriting Newsgroups headers to strip locally unknown
> newsgroups is superficially attractive. However, early experience
> with exactly that policy was thoroughly negative: news propagation
> is more redundant and much less orderly than many people imagine,
> and in particular it is not unheard of for the (sometimes) fastest
> path between two (say) University of Toronto sites to pass outside
> the University of Toronto, in which case newsgroup stripping can
> cause incomplete propagation. Having an article's set of
> newsgroups change as it propagates can also result in followups
> not achieving the same propagation as the original. It's been
> tried; it's more trouble than it's worth; don't do it.

> NOTE: In particular, newsgroup stripping superficially looks like
> a solution to the problem of duplicate regional newsgroup names.
> For example, both the University of Toronto and the University of
> Texas have "ut.general" newsgroups, and material cross-posted to
> that name and a global newsgroup appears in both universities'

local newsgroups. However, the side effects of stripping are
sufficiently unacceptable to disqualify it for this purpose.
Don't do it.

Cross-posting an article to several relevant newsgroups is far
superior to posting separate articles with duplicated content to each
newsgroup, because reading agents can detect the situation and show
the article to a reader only once. Posters SHOULD cross-post rather
than duplicate-post.

   NOTE: On the other hand, cross-posting to a large number of
   newsgroups usually indicates that the poster has not thought about
   his audience; articles are rarely pertinent to more than (say)
   half a dozen newsgroups. Posting agents might wish to request
   confirmation when the number of newsgroups exceeds (say) five in
   the presence of a Followup-To header, or (say) two in the absence
   of such a header.

   NOTE: One problem with cross-postings is what to do with an
   article cross-posted to a set of newsgroups including both
   moderated and unmoderated ones. Posters tend to expect such an
   article to show up immediately in the unmoderated newsgroups,
   especially if they do not realize that one or more of the
   newsgroups is moderated. However, since it is not possible for a
   moderator to retroactively add an already-posted article to a
   moderated newsgroup, the only correct action is to mail such an
   article to one (and only one) of the moderators for action. It is
   probably best for the posting agent to detect this situation and
   ask the poster what action is preferred. The acceptable choices
   are to alter the newsgroup list or to mail to a moderator of the
   poster's choice; the posting agent should NOT offer duplicate-
   posting as an easy-to-request option (if only because many
   moderators will reject a submission that has already been posted
   to unmoderated newsgroups).

   NOTE: An article cross-posted to multiple moderated newsgroups
   really should have approval from all of the moderators involved.
   In practice, the only straightforward way to do this is to send
   the article to one of them and have him consult the others.

A newsgroup SHOULD NOT appear more than once in the Newsgroups
header.

Newsgroup names having only one component are reserved for newsgroups
whose propagation is restricted to a single host (or the
administrative equivalent). It is inadvisable to name a newsgroup

"poster" because that word has special meaning in the Followup-To
header (see Section 6.1). The names "control" and "junk" are
frequently used for pseudo-newsgroups internal to relayer
implementations, and hence are also best avoided.

> NOTE: Beware of the duplicate-regional-newsgroup-names problem
> mentioned above. In particular, there are many, many hosts with a
> newsgroup named "general", and some surprising things show up in
> such newsgroups when people cross-post. It is probably better to
> use multi-component names, which are less likely to be duplicated.
> Fred's Widget House should use "fwh.general" rather than just
> "general" as its in-house general-topics newsgroup.

It is conventional to reserve newsgroup names beginning with "to."
for test messages sent on an essentially point-to-point basis (see
also the ihave/sendme protocol described in Section 7.2); newsgroup
names beginning with "to." SHOULD NOT be used for any other purpose.
The second (and possibly later) components of such a name should,
together, comprise the relayer name (see Section 5.6) of a relayer.
The newsgroup exists only at the named relayer and its neighbors.
The neighbors all pass that newsgroup to the named relayer, while the
named relayer does not pass it to anyone.

The order of newsgroup names in the Newsgroups header is not
significant.

5.6. Path

The Path header's content indicates which relayers the article has
already visited, so that unnecessary redundant transmission can be
avoided:

```
Path-content    = [ path-list path-delimiter ] local-part
path-list       = relayer-name *( path-delimiter relayer-name )
relayer-name    = 1*rn-char
rn-char         = letter / digit / "." / "-" / "_"
path-delimiter  = "!"
```

The Path content is a list of relayer names, separated by path
delimiters, followed (after a final delimiter) by the local part of a
mailing address. Each relayer MUST prepend its name, and a
delimiter, to the Path content in all articles it processes. A
relayer MUST NOT pass an article to a neighboring relayer whose name
is already mentioned in an article's path list, unless this is
explicitly requested by the neighbor in some way. The Path content
is case-sensitive.

NOTE: The Path header supplied by a posting agent should normally
contain only the local part. The relayer that the posting agent
passes the article to for posting will prepend its relayer name to
get the path list started.

NOTE: Observe that the trailing local part is NOT part of the path
list. This Path header:

      Path: fee!fie!foe!fum

contains three relayer names: "fee", "fie", and "foe". A relayer
named "fum" is still eligible to be sent this article.

NOTE: This syntax has the disadvantage of containing no white
space, making it impossible to continue a Path header across
several lines. Implementors of relayers and reading agents are
warned that it is intended that the successor to this Draft will
change the definition of path delimiter to:

      path-delimiter = "!" [space]

and are urged to fix their software to handle (i.e., ignore) white
space following the exclamation points. They are urged to hurry;
some ill-behaved systems reportedly already feel free to add such
white space.

NOTE: [RFC1036] allows considerably more flexibility in choice of
delimiter, in theory, but this flexibility has never been used,
and most news software does not implement it properly. The
grammar reflects the current reality. Note, in particular, that
[RFC1036] treats "_" as a delimiter, but in fact it is known to
appear in relayer names occasionally.

Because an article will not propagate to a relayer already mentioned
in its path list, the path list MUST NOT contain any names other than
those of relayers the article has passed through AS NEWS. This is
trivially obvious for normal news articles but requires attention
from the moderators of moderated newsgroups and the implementors and
maintainers of gateways.

NOTE: For the same reason, a relayer and its neighbors need to
agree on the choice of relayer name, and names should not be
changed without notifying neighbors.

Relayer names need to be unique among all relayers that will ever see
the articles using them. A relayer name is normally either an
"official" name for the host the relayer runs on, or some other
"official" name controlled by the same organization. Except in

cooperating subnets that agree to some other convention and don't let
articles using it escape beyond the subnet, a relayer name MUST be
either a UUCP name registered in the UUCP maps (without any domain
suffix such as ".UUCP") or a complete Internet domain name. Use of a
(registered) UUCP name is recommended, where practical, to keep the
length of the path list down.

The use of Internet domain names in the path list presents one
problem: domain names are case-insensitive, but the path list is
case-sensitive. Relayers using domain names as their relayer names
MUST pick a standard form for the name and use that form consistently
to the exclusion of all others. The preferred form for this purpose,
which relayers SHOULD use, is the all-lowercase form.

    NOTE: It is arguably unfortunate that the path list is case-
    sensitive, but it is much too late to change this. Most Internet
    sites do, in any event, use one standardized form of their name
    almost everywhere.

In the ordinary case, where the poster is the author of the article,
the local part following the path list SHOULD be the local part of
the poster's full Internet domain mailing address.

    NOTE: It should be just the local part, not the full address. The
    character "@" does not appear in a Path header.

The Path content somewhat resembles a mailing address, particularly
in the UUCP world with its manual routing and "!" address syntax.
Historically, this resemblance was important, and the Path content
was often used as a reply address. This practice has always been
somewhat unreliable, since news paths are not always mail paths and
news relayer names are not always recognized by mail handlers, and
its reliability has generally worsened in recent times. The
widespread use of and recognition of Internet domain addresses, even
outside the actual Internet, has largely eliminated the problem.
Readers SHOULD NOT use the Path content as a reply address. On the
other hand, relayer administrators are urged not to break this usage
without good reason; where practical, paths followed by news SHOULD
be traversable by mail, and mail handlers SHOULD recognize relayer
names as host names.

It will typically be difficult or impractical for gateways and
moderators to supply a Path content that is useful as a reply address
for the author, bearing in mind that the path list they supply will
normally be empty. (To reiterate: the path list MUST NOT contain any
names other than those of relayers the article has passed through AS
NEWS.) They SHOULD supply a local part that will result in replies

   to a Path-derived address being returned to the sender with a brief
   explanation. Software permitting, the local part "not-for-mail" is
   recommended.

      NOTE: A moderator or gateway administrator who supplies a local
      part that delivers such mail to an administrative mailbox will
      quickly discover why it should be bounced automatically! It is
      best, however, for the returned message to include an explanation
      of what has probably happened, rather than just a mysterious
      "undeliverable mail" complaint, since the sender may not be aware
      that his/her software is unwisely using the Path content as a
      reply address. Reply software might wish to question attempts to
      reply to a Path-derived address ending in "not-for-mail" (which is
      why a specific name is being recommended here).

6. Optional Headers

   Many MAIL headers, and many of those specified in present and future
   MAIL extensions, are potentially applicable to news. Headers
   specific to MAIL's point-to-point transmission paradigm, e.g., To and
   Cc, SHOULD NOT appear in news articles. (Gateways wishing to
   preserve such information for debugging probably SHOULD hide it under
   different names; prefixing "X-" to the original headers, resulting in
   forms like "X-To", is suggested.)

   The following optional headers are either specific to news or of
   particular note in news articles; an article MAY contain some or all
   of them. (Note that there are some circumstances in which some of
   them are mandatory; these are explained under the individual
   headers.) An article MUST NOT contain two or more headers with any
   one of these header names.

      NOTE: The ban on duplicate header names does not apply to headers
      not specified in this Draft, such as "X-" headers. Software
      should not assume that all header names in a given article are
      unique.

6.1. Followup-To

   The Followup-To header contents specify to which newsgroup(s)
   followups should be posted:

      Followup-To-content = Newsgroups-content / "poster"

The syntax is the same as that of the Newsgroups content, with the
exception that the magic word "poster" means that followups should be
mailed to the article's reply address rather than posted. In the
absence of Followup-To, the default newsgroup(s) for a followup are
those in the Newsgroups header.

   NOTE: The way to request that followups be mailed to a specific
   address other than that in the From line is to supply
   "Followup-To: poster" and a Reply-To header. Putting a mailing
   address in the Followup-To line is incorrect; posting agents
   should reject or rewrite such headers.

   NOTE: There is no syntax for "no followups allowed" because
   "Followup-To: poster" accomplishes this effect without extra
   machinery.

Although it is generally desirable to limit followups to the smallest
reasonable set of newsgroups, especially when the precursor was
cross-posted widely, posting agents SHOULD NOT supply a Followup-To
header except at the poster's explicit request.

   NOTE: In particular, it is incorrect for the posting agent to
   assume that followups to a cross-posted article should be directed
   to the first newsgroup only. Trimming the list of newsgroups
   should be the poster's decision, not the posting agent's.
   However, when an article is to be cross-posted to a considerable
   number of newsgroups, a posting agent might wish to SUGGEST to the
   poster that followups go to a shorter list.

6.2. Expires

The Expires header content specifies a date and time when the article
is deemed to be no longer useful and should be removed ("expired"):

   Expires-content = Date-content

The content syntax is the same as that of the Date content. In the
absence of Expires, the default is decided by the administrators of
each host the article reaches, who MAY also restrict the extent to
which the Expires header is honored.

The Expires header has two main applications: removing articles whose
utility ends on a specific date (e.g., event announcements that can
be removed once the day of the event has passed) and preserving
articles expected to be of prolonged usefulness (e.g., information
aimed at new readers of a newsgroup). The latter application is
sometimes abused. Since individual hosts have local policies for
expiration of news (depending on available disk space, for instance),

posters SHOULD NOT provide Expires headers for articles unless there
is a natural expiration date associated with the topic. Posting
agents MUST NOT provide a default Expires header. Leave it out and
allow local policies to be used unless there is a good reason not to.
Expiry dates are properly the decision of individual host
administrators; posters and moderators SHOULD set only expiry dates
with which most administrators would agree.

   NOTE: A poster preparing an Expires header for an article whose
   utility ends on a specific day should typically specify the NEXT
   day as the expiry date. A meeting on July 7th remains of interest
   on the 7th.

6.3. Reply-To

   The Reply-To header content specifies a reply address different from
   the author's address given in the From header:

      Reply-To-content = From-content

   In the absence of Reply-To, the reply address is the address in the
   From header.

   Use of a Reply-To header is preferable to including a similar request
   in the article body, because reply-preparation software can take
   account of Reply-To automatically.

6.4. Sender

   The Sender header identifies the poster, in the event that this
   differs from the author identified in the From header:

      Sender-content = From-content

   In the absence of Sender, the default poster is the author (named in
   the From header).

   NOTE: The intent is that the Sender header have a fairly high
   probability of identifying the person who really posted the
   article. The ability to specify a From header naming someone
   other than the poster is useful but can be abused.

   If the poster supplies a From header, the posting agent MUST ensure
   that a Sender header is present, unless it can verify that the
   mailing address in the From header is a valid mailing address for the
   poster. A poster-supplied Sender header MAY be used, if its mailing
   address is verifiably a valid mailing address for the poster;

otherwise, the posting agent MUST supply a Sender header and delete
(or rename, for example, to X-Unverifiable-Sender) any poster-
supplied Sender header.

> NOTE: It might be useful to preserve a poster-supplied Sender
> header so that the poster can supply the full-name part of the
> content. The mailing address, however, must be right, hence, the
> posting agent must generate the Sender header if it is unable to
> verify the mailing address of a poster-supplied one.

> NOTE: NNTP implementors, in particular, are urged to note this
> requirement (which would eliminate the need for ad hoc headers
> like NNTP-Posting-Host), although there are admittedly some
> implementation difficulties. A user name from an [RFC1413] server
> and a host name from an inverse mapping of the address, perhaps
> with a "full name" comment noting the origin of the information,
> would be at least a first approximation:

>   Sender: fred@zoo.toronto.edu (RFC-1413@reverse-lookup;
>   not verified)

While this does not completely meet the specs, it comes a lot closer
than not having a Sender header at all. Even just supplying a
placeholder for the user name:

    Sender: somebody@zoo.toronto.edu (user name unknown)

would be better than nothing.

6.5. References

The References header content lists message IDs of precursors:

    References-content = message-id *(space message-id)

A followup MUST have a References header, and an article that is not
a followup MUST NOT have a References header. The References-content
of a followup MUST be the precursor's References-content (if any)
followed by the precursor's message ID.

> NOTE: Use the See-Also header (Section 6.16) for interconnection
> of articles that are not in a followup relationship to each other.

> NOTE: In retrospect, RFCs 850 and 1036, and the implementations
> whose practice they represented, erred here. The proper MAIL
> header to use for references to precursors is In-Reply-To, and the
> References header is meant to be used for the purposes here
> ascribed to See-Also. This incompatibility is far too solidly

established to be fixed, unfortunately. The best that can be done
is to provide a clear mapping between the two and urge gateways to
do the transformation. The news usage is (now) a deliberate
violation of the MAIL specifications; articles containing news
References headers are technically not valid MAIL messages,
although it is unlikely that much MAIL software will notice
because the incompatibility is at a subtle semantic level that
does not affect the syntax.

UNRESOLVED ISSUE: Would it be better to just give up and admit
that news uses References for both purposes?

UNRESOLVED ISSUE: Should the syntax be generalized to include URLs
as alternatives to message IDs? Perhaps not; too many things know
about References already. And non-articles can't be precursors of
articles, not really.

Followup agents SHOULD NOT shorten References headers. If it is
absolutely necessary to shorten the header, as a desperate last
resort, a followup agent MAY do this by deleting some of the message
IDs. However, it MUST NOT delete the first message ID, the last
three message IDs (including that of the immediate precursor), or any
message ID mentioned in the body of the followup. If it is possible
for the followup agent to determine the Subject content of the
articles identified in the References header, it MUST NOT delete the
message ID of any article where the Subject content changed (other
than by prepending of a back reference). The followup agent MUST NOT
delete any message ID whose local part ends with "_-_" (underscore
(ASCII 95), hyphen (ASCII 45), underscore); followup agents are urged
to use this form to mark subject changes and to avoid using it
otherwise.

NOTE: As software capable of exploiting References chains has
grown more common, the random shortening permitted by [RFC1036]
has become increasingly troublesome. ANY shortening is
undesirable, and software should do it only in cases of dire
necessity. In such cases, these rules attempt to limit the
damage.

NOTE: The first message ID is very important as the starting point
of the "thread" of discussion and absolutely should not be
deleted. Keeping the last three message IDs gives thread-
following software a fighting chance to reconstruct a full thread
even if an article or two is missing. Keeping message IDs
mentioned in the body is obviously desirable.

NOTE: Subject changes are difficult to determine, but they are
significant as possible beginnings of new threads. The "_-_"
convention is provided so that posting agents (which have more
information about subjects) can flag articles containing a subject
change in a way that followup agents can detect without access to
the articles themselves. The sequence is chosen as one that is
fairly unlikely to occur by accident.

UNRESOLVED ISSUE: Is "_-_" really worth having?

When a References header is shortened, at least three blanks SHOULD
be left between adjacent message IDs at each point where deletions
were made. Software preparing new References headers SHOULD preserve
multiple blanks in older References content.

NOTE: It's desirable to have some marker of where deletions
occurred, but the restricted syntax of the header makes this
difficult. Extra white space is not a very good marker, since it
may be deleted by software that ill-advisedly rewrites headers,
but at least it doesn't break existing software.

To repeat: followup agents SHOULD NOT shorten References headers.

NOTE: Unfortunately, reading agents and other software analyzing
References patterns have to be prepared for the worst anyway. The
worst includes random deletions and the possibility of circular
References chains (when References is misused in place of See-Also
(Section 6.16)).

6.6. Control

The Control header content marks the article as a control message and
specifies the desired actions (other than the usual ones of filing
and passing on the article):

    Control-content = verb *(space argument)
    verb   = 1*(letter / digit)
    argument   = 1*<ASCII printable character>

The verb indicates what action should be taken, and the argument(s)
(if any) supply details. In some cases, the body of the article may
also contain details. Section 7 describes the standard verbs. See
also the Also-Control header (Section 6.15).

NOTE: Control messages are often processed and filed rather
differently than normal articles.

NOTE: The restriction of verbs to letters and digits is new but is
consistent with existing practice and potentially simplifies
implementation by avoiding characters significant to command
interpreters. Beware that the arguments are under no such
restriction in general.

NOTE: Two other conventions for distinguishing control messages
from normal articles were formerly in use: a three-component
newsgroup name ending in ".ctl" or a subject beginning with
"cmsg " was considered to imply that the article was a control
message. These conventions are obsolete. Do not use them.

An article with a Control header MUST NOT have an Also-Control or
Supersedes header.

6.7. Distribution

The Distribution header content specifies geographic or
organizational limits on an article's propagation:

```
Distribution-content  = distribution *( dist-delim distribution )
dist-delim            = ","
distribution          = plain-component
```

A distribution is syntactically identical to a one-component
newsgroup name and must satisfy the same rules and restrictions. In
the absence of Distribution, the default distribution is "world".

NOTE: This syntax has the disadvantage of containing no white
space, making it impossible to continue a Distribution header
across several lines. Implementors of relayers and reading agents
are warned that it is intended that the successor to this Draft
will change the definition of dist delimiter to:

```
dist-delim = "," [ space ]
```

and are urged to fix their software to handle (i.e., ignore) white
space following the commas.

A relayer MUST NOT pass an article to another relayer unless
configuration information specifies transmission to that other
relayer of BOTH (a) at least one of the article's newsgroup(s), and
(b) at least one of the article's distribution(s). In effect, the
only role of distributions is to limit propagation, by preventing
transmission of articles that would have been transmitted had the
decision been based solely on newsgroups.

A posting agent might wish to present a menu of possible
distributions, or suggest a default, but normally SHOULD NOT supply a
default without giving the poster a chance to override it. A
followup agent SHOULD initially supply the same Distribution header
as found in the precursor, although the poster MAY alter this if
appropriate.

Despite the syntactic similarity and some historical confusion,
distributions are NOT newsgroup names. The whole point of putting a
distribution on an article is that it is DIFFERENT from the
newsgroup(s). In general, a meaningful distribution corresponds to
some sort of region of propagation: a geographical area, an
organization, or a cooperating subnet.

   NOTE: Distributions have historically suffered from the completely
   uncontrolled nature of their name space, the lack of feedback to
   posters on incomplete propagation resulting from use of random
   trash in Distribution headers, and confusion with newsgroups
   (arising partly because many regions and organizations DO have
   internal newsgroups with names resembling their internal
   distributions). This has resulted in much garbage in Distribution
   headers, notably the pointless practice of automatically supplying
   the first component of the newsgroup name as a distribution (which
   is MOST unlikely to restrict propagation!). Many sites have opted
   to maximize propagation of such ill-formed articles by essentially
   ignoring distributions. This unfortunately interferes with
   legitimate uses. The situation is bad enough that distributions
   must be considered largely useless except within cooperating
   subnets that make an organized effort to restrain propagation of
   their internal distributions.

   NOTE: The distributions "world" and "local" have no standard magic
   meaning (except that the former is the default distribution if
   none is given). Some pieces of software do assign such meanings
   to them.

6.8. Keywords

The Keywords header content is one or more phrases intended to
describe some aspect of the content of the article:

    Keywords-content = plain-phrase *("," [space] plain-phrase)

Keywords, separated by commas, each follow the <plain-phrase> syntax
defined in Section 5.2. Encoded words in keywords MUST NOT contain
characters other than letters (of either case), digits, and the
characters "!", "*", "+", "-", "/", "=", and "_".

NOTE: Posters and posting agents are asked to take note that
keywords are separated by commas, not by white space. The
following Keywords header contains only one keyword (a rather
unlikely and improbable one):

Keywords: Thompson Ritchie Multics Linux

and should probably have been written:

Keywords: Thompson, Ritchie, Multics, Linux

This particular error is unfortunately rather widespread.

NOTE: Reading agents and archivers preparing indexes of articles
should bear in mind that user-chosen keywords are notoriously poor
for indexing purposes unless the keywords are picked from a
predefined set (which they are not in this case). Also, some
followup agents unwisely propagate the Keywords header from the
precursor into the followup by default. At least one news-based
experiment has found the contents of Keywords headers to be
completely valueless for indexing.

6.9. Summary

The Summary header content is a short phrase summarizing the
article's content:

    Summary-content = nonblank-text

As with the subject, no restriction is placed on the content since it
is intended solely for display to humans.

NOTE: Reading agents should be aware that the Summary header is
often used as a sort of secondary Subject header, and (if present)
its contents should perhaps be displayed when the subject is
displayed.

The summary SHOULD be terse. Posters SHOULD avoid trying to cram
their entire article into the headers; even the simplest query
usually benefits from a sentence or two of elaboration and context,
and not all reading agents display all headers.

6.10. Approved

The Approved header content indicates the mailing addresses (and
possibly the full names) of the persons or entities approving the
article for posting:

```
Approved-content = From-content *( "," [ space ] From-content )
```

An Approved header is required in all postings to moderated
newsgroups; the presence or absence of this header allows a posting
agent to distinguish between articles posted by the moderator (which
are normal articles to be posted normally) and attempted
contributions by others (which should be mailed to the moderator for
approval). An Approved header is also required in certain control
messages, to reduce the probability of accidental posting of same;
see the relevant parts of Section 7.

   NOTE: There is, at present, no way to authenticate Approved
   headers to ensure that the claimed approval really was bestowed.
   Nor is there an established mechanism for even maintaining a list
   of legitimate approvers (such a list would quickly become out of
   date if it had to be maintained by hand). Such mechanisms,
   presumably relying on cryptographic authentication, would be a
   worthwhile extension to this Draft, and experimental work in this
   area is encouraged. (The problem is harder than it sounds because
   news is used on many systems that do not have real-time access to
   key servers.)

   NOTE: Relayer implementors, please note well: it is the POSTING
   AGENT that is authorized to distinguish between moderator postings
   and attempted contributions, and to mail the latter to the
   moderator. As discussed in Section 9.1, relayers MUST NOT, repeat
   MUST NOT, send such mail; on receipt of an unApproved article in a
   moderated newsgroup, they should discard the article, NOT
   transform it into a mail message (except perhaps to a local
   administrator).

   NOTE: [RFC1036] restricted Approved to a single From-content.
   However, multiple moderation is no longer rare, and multi-
   moderator Approved headers are already in use.

6.11. Lines

The Lines header content indicates the number of lines in the body of
the article:

```
Lines-content = 1*digit
```

The line count includes all body lines, including the signature (if
any) and including empty lines (if any) at the beginning or end of
the body. (The single empty separator line between the headers and
the body is not part of the body.) The "body" here is the body as
found in the posted article, AFTER all transformations such as MIME
encodings.

Reading agents SHOULD NOT rely on the presence of this header, since
it is optional (and some posting agents do not supply it). They MUST
NOT rely on it being precise, since it frequently is not.

   NOTE: The average line length in article bodies is surprisingly
   consistent at about 40 characters, and since the line count
   typically is used only for approximate judgements ("is this too
   long to read quickly?"), dividing the byte count of the body by 40
   gives an estimate of the body line count that is adequate for
   normal use. This estimate is NOT adequate if the body has been
   MIME encoded, but neither is the Lines header: at least one major
   relayer will add a Lines header to an article that lacks one,
   without considering the possibility of MIME encodings when
   computing the line count.

   NOTE: It would be better to have a Content-Size header as part of
   MIME, so that body parts could have their own sizes, and so that
   the units used could be appropriate to the data type (line count
   is not a useful measure of the size of an encoded image, for
   example). Doing this is preferable to trying to fix Lines.

   UNRESOLVED ISSUE: Update on Content-Size?

Relayers SHOULD discard this header if they find it necessary to
re-encode the article in such a way that the original Lines header
would be rendered incorrect.

6.12. Xref

The Xref header content indicates where an article was filed by the
last relayer to process it:

    Xref-content   = relayer 1*(space location)
    relayer   = relayer-name
    location   = newsgroup-name ":" article-locator
    article-locator = 1*<ASCII printable character>

The relayer's name is included so that software can determine which
relayer generated the header (and specifically, whether it really was
the one that filed the copy being examined). The locations specify
what newsgroups the article was filed under (which may differ from
those in the Newsgroups header) and where it was filed under them.
The exact form of an article locator is implementation-specific.

   NOTE: Reading agents can exploit this information to avoid
   presenting the same article to a reader several times. The
   information is sometimes available in system databases, but having
   it in the article is convenient. Relayers traditionally generate

an Xref header only if the article is cross-posted, but this is
not mandatory, and there is at least one new application
("mirroring": keeping news databases on two hosts identical) where
the header is useful in all articles.

NOTE: The traditional form of an article locator is a decimal
number, with articles in each newsgroup numbered consecutively
starting from 1. NNTP [RFC977] demands that such a model be
provided, and there may be other software that expects it, but it
seems desirable to permit flexibility for unorthodox
implementations.

A relayer inserting an Xref header into an article MUST delete any
previous Xref header. A relayer that is not inserting its own Xref
header SHOULD delete any previous Xref header. A relayer MAY delete
the Xref header when passing an article on to another relayer.

NOTE: [RFC1036] specified that the Xref header was not transmitted
when an article was passed to another relayer, but the major news
implementations have never obeyed this rule, and applications like
mirroring depend on this disobedience.

A relayer MUST use the same name in Xref headers as it uses in Path
headers. Reading agents MUST ignore an Xref header containing a
relayer name that differs from the one that begins the path list.

6.13. Organization

The Organization header content is a short phrase identifying the
poster's organization:

    Organization-content = nonblank-text

This header is typically supplied by the posting agent. The
Organization content SHOULD mention geographical location (e.g., city
and country) when it is not obvious from the organization's name.

NOTE: The motive here is that the organization is often difficult
to guess from the mailing address, is not always supplied in a
signature, and can help identify the poster to the reader.

NOTE: There is no "s" in "Organization".

The Organization content is provided for identification only and does
not imply that the poster speaks for the organization or that the
article represents organization policy. Posting agents SHOULD permit
the poster to override a local default Organization header.

6.14. Supersedes

   The Supersedes header content specifies articles to be cancelled on
   arrival of this one:

      Supersedes-content = message-id *(space message-id)

   Supersedes is equivalent to Also-Control (Section 6.15) with an
   implicit verb of "cancel" (Section 7.1).

      NOTE: Supersedes is normally used where the article is an updated
      version of the one(s) being cancelled.

      NOTE: Although the ability to use multiple message IDs in
      Supersedes is highly desirable (see Section 7.1), posters are
      warned that existing implementations often do not correctly handle
      more than one.

      NOTE: There is no "c" in "Supersedes".

   An article with a Supersedes header MUST NOT have an Also-Control or
   Control header.

6.15. Also-Control

   The Also-Control header content marks the article as being a control
   message IN ADDITION to being a normal news article and specifies the
   desired actions:

      Also-Control-content = Control-content

   An article with an Also-Control header is filed and passed on
   normally, but the content of the Also-Control header is processed as
   if it were found in a Control header.

      NOTE: It is sometimes desirable to piggyback control actions on a
      normal article, so that the article will be filed normally but
      will also be acted on as a control message. This header is
      essentially a generalization of Supersedes.

      NOTE: Be warned that some old relayers do not implement
      Also-Control.

   An article with an Also-Control header MUST NOT have a Control or
   Supersedes header.

6.16. See-Also

   The See-Also header content lists message IDs of articles that are
   related to this one but are not its precursors:

      See-Also-content = message-id *(space message-id)

   See-Also resembles References, but without the restrictions imposed
   on References by the followup rules.

      NOTE: See-Also provides a way to group related articles, such as
      the parts of a single document that had to be split across
      multiple articles due to its size, or to cross-reference between
      parallel threads.

      NOTE: See the discussion (in Section 6.5) on MAIL compatibility
      issues of References and See-Also.

      NOTE: In the specific case where it is desired to essentially make
      another article PART of the current one, e.g., for annotation of
      the other article, MIME's "message/external-body" convention can
      be used to do so without actual inclusion. "news-message-ID" was
      registered as a standard external-body access method, with a
      mandatory NAME parameter giving the message ID and an optional
      SITE parameter suggesting an NNTP site that might have the article
      available (if it is not available locally), by IANA 22 June 1993.

      UNRESOLVED ISSUE: Could the syntax be generalized to include URLs
      as alternatives to message IDs? Here it makes much more sense
      than in References.

6.17. Article-Names

   The Article-Names header content indicates any special significance
   the article may have in particular newsgroups:

      Article-Names-content = 1*(name-clause space)
      name-clause   = newsgroup-name ":" article-name
      article-name   = letter 1*(letter / digit / "-")

   Each name clause specifies a newsgroup (which SHOULD be among those
   in the Newsgroups header) and an article name local to that
   newsgroup. Article names MAY be used by relayers to file the article
   in special ways, or they MAY just be noted for possible special
   attention by reading agents. Article names are case-sensitive.

   NOTE: This header provides a way to mark special postings, such as
   introductions, frequently-asked-question lists, etc., so that
   reading agents have a way of finding them automatically. The
   newsgroup name is specified for each article name because the
   names may be newsgroup-specific; for example, many frequently-
   asked-question lists are posted to "news.answers" in addition to
   their "home" newsgroup, and they would not be known by the same
   name(s) in both newsgroups.

The Article-Names header SHOULD be ignored unless the article also
contains an Approved header.

   NOTE: This stipulation is made in anticipation of the possibility
   that Approved headers will be involved in cryptographic
   authentication.

The presence of an Article-Names header does not necessarily imply
that the article will be retained unusually long before expiration,
or that previous article(s) with similar Article-Names headers will
be cancelled by its arrival. Posters preparing special postings
SHOULD include appropriate other headers, such as Expires and
Supersedes, to request such actions.

Different networks MAY establish different sets of article names for
the special postings they deem significant; it is preferable for
usage to be standardized within networks, although it might be
desirable for individual newsgroups to have different naming
conventions in some situations. Article names MUST be 14 characters
or less. The following names are suggested but are not mandatory:

intro   Introduction to the newsgroup for newcomers.

charter   Charter, rules, organization, moderation policies, etc.

background Biographies of special participants, history of the
            newsgroup, notes on related newsgroups, etc.

subgroups   Descriptions of sub-newsgroups under this newsgroup,
            e.g., "sci.space.news" under "sci.space".

facts   Information relating to the purpose of the newsgroup,
            e.g., an acronym glossary in "sci.space".

references Where to get more information: books, journals, FTP
            repositories, etc.

faq   Answers to frequently asked questions.

    menu   If present, a list of all of the other article names
                 local to this newsgroup, with brief descriptions of their
                 contents.

   Such articles may be divided into subsections using the MIME
   "multipart/mixed" conventions. If size considerations make it
   necessary to split such articles, names ending in a hyphen and a part
   number are suggested; for example, a three-part frequently-asked-
   questions list could have article names "faq-1", "faq-2", and
   "faq-3".

      NOTE: It is somewhat premature to attempt to standardize article
      names, since this is essentially a new feature with no experience
      behind it. However, if reading agents are to attach special
      significance to these names, some attempt at standard conventions
      is imperative. This is a first attempt at providing some.

6.18. Article-Updates

   The Article-Updates header content indicates what previous articles
   this one is deemed (by the poster) to update (i.e., replace):

      Article-Updates-content = message-id *(space message-id)

   Each message ID identifies a previous article that this one is deemed
   to update. This MUST NOT cause the previous article(s) to be
   cancelled or otherwise altered, unless this is implied by other
   headers (e.g., Supersedes); Article-Updates is merely an advisory
   that MAY be noted for special attention by reading agents.

      NOTE: This header provides a way to mark articles that are only
      minor updates of previous ones, containing no significant new
      information and not worth reading if the previous ones have been
      read.

      NOTE: If suitable conventions using MIME multipart bodies and the
      "message/external-body" body-part type can be developed, a
      replacing article might contain only differences between the old
      text and the new text, rather than a complete new copy. This is
      the motivation for not making Article-Updates also function as
      Supersedes does: the replacing article might depend on the
      continued presence of the replaced article.

7. Control Messages

   The following sections document the currently defined control
   messages. "Message" is used herein as a synonym for "article" unless
   context indicates otherwise.

Posting agents are warned that since certain control messages require
article bodies in quite specific formats, signatures SHOULD NOT be
appended to such articles, and it may be wise to take greater care
than usual to avoid unintended (although perhaps well-meaning)
alterations to text supplied by the poster. Relayers MUST assume
that control messages mean what they say; they MAY be obeyed as is or
rejected, but MUST NOT be reinterpreted.

The execution of the actions requested by control messages is subject
to local administrative restrictions, which MAY deny requests or
refer them to an administrator for approval. The descriptions below
are generally phrased in terms suggesting mandatory actions, but any
or all of these MAY be subject to local administrative approval
(either as a class or case-by-case). Analogously, where the
description below specifies that a message or portion thereof is to
be ignored, this action MAY include reporting it to an administrator.

   NOTE: The exact choice of local action might depend on what action
   the control message requests, who it claims to come from, etc.

Relayers MUST propagate even control messages they do not understand.

In the following sections, each type of control message is defined
syntactically by defining its arguments and its body. For example,
"cancel" is defined by defining cancel-arguments and cancel-body.

7.1. cancel

   The cancel message requests that one or more previous articles be
   "cancelled":

      cancel-arguments = message-id *(space message-id)
      cancel-body   = body

   The argument(s) identify the articles to be cancelled, by message ID.
   The body is a comment, which software MUST ignore, and SHOULD contain
   an indication of why the cancellation was requested. The cancel
   message SHOULD be posted to the same newsgroup(s), with the same
   distribution(s), as the article(s) it is attempting to cancel.

      NOTE: Using the same newsgroups and distributions maximizes the
      chances of the cancel message propagating everywhere the target
      articles went.

      NOTE: [RFC1036] permitted only a single message-id in a cancel
      message. Support for cancelling multiple articles is highly
      desirable, especially for use with Supersedes (see Section 6.14).
      If several revisions of an article appear in fast succession, each

using Supersedes to cancel the previous one, it is possible for a
middle revision to be destroyed by cancellation before it is
propagated onward to cancel its predecessor. Allowing each
article to cancel several predecessors greatly alleviates this
problem. (Posting agents preparing a cancel of an article that
itself cancels other articles might wish to add those articles to
the cancel-arguments.) However, posters should be aware that much
old software does not implement multiple cancellation properly and
should avoid using it when reliable cancellation is vitally
important.

When an article (the "target article") is to be cancelled, there are
four cases of interest: the article hasn't arrived yet, it has
arrived and been filed and is available for reading, it has expired
and been archived on some less-accessible storage medium, or it has
expired and been deleted. The next few paragraphs discuss each case
in turn (in reverse order, which is convenient for the explanation).

EXPIRED AND DELETED. Take no action.

EXPIRED AND ARCHIVED. If the article is readily accessible and can
be deleted or made unreadable easily, treat as under AVAILABLE below.
Otherwise, treat as under EXPIRED AND DELETED.

   NOTE: While it is desirable for archived articles to be
   cancellable, this can easily involve rewriting an entire archive
   volume just to get rid of one article, perhaps with manual actions
   required to arrange it. It is difficult to envision a situation
   so dire as to require such measures from hundreds or thousands of
   administrators, or for that matter one in which widespread
   compliance with such a request is likely.

AVAILABLE. Compare the mailing addresses from the From lines of the
cancel message and the target article, bearing in mind that local
parts (except for "postmaster") are case-sensitive and domains are
case-insensitive. If they do not match, either refer the issue to an
administrator for a case-by-case decision, or treat as if they
matched.

   NOTE: It is generally trivial to forge articles, so nothing short
   of cryptographic authentication is really adequate to ensure that
   a cancel came from the original article's author. Moreover, it is
   highly desirable to permit authorities other than the author to
   cancel articles, to allow for cases in which the author is
   unavailable, uncooperative, or malicious, and in which damage
   and/or legal problems may be minimized by prompt cancellation.

Reliable authentication that would permit such administrative
cancels would be a worthwhile extension to this Draft, and
experimental work in this area is encouraged.

NOTE: Meanwhile, a simple check of addresses is useful accident
prevention and catches at least the most simple-minded forgers.
Since the intent is accident prevention rather than ironclad
security, use of the From address is appropriate, all the more so
because in the presence of gateways (especially redundant multiple
gateways), the author may not have full control over Sender
headers.

NOTE: The "refer... or treat as if they matched" rule is intended
to specifically forbid quietly ignoring cancels with mismatched
addresses.

If the addresses match, then if technically possible, the relayer
MUST delete the target article completely and immediately. Failing
that, it MUST make the target article unreadable (preferably to
everyone, minimally to everyone but the administrator) and either
arrange for it to be deleted as soon as possible or notify an
administrator at once.

NOTE: To allow for events such as criminal actions, malicious
forgeries, and copyright infringements, where damage and/or legal
problems may be minimized by prompt cancellation, complete removal
is strongly preferred over merely making the target article
unreadable. The potential for malice is outweighed by the
importance of really getting rid of the target article in some
legitimate cases. (In cases of inadvertent copyright violation in
particular, the ability to quickly remedy the violation is of
considerable legal importance.) Failing that, making it
unreadable is better than nothing.

NOTE: Merely annotating the article so that readers see an
indication that the author wanted it cancelled is not acceptable.
Making the article unreadable is the minimum action.

NOTE: There have been experiments with making cancelled articles
unreadable, so that local news administrators could reverse
cancellations. In practice, administrators almost never find
cause to do so. Removal appears to be clearly preferable where
technically feasible.

NOT ARRIVED YET. If practical, retain the cancel message until the
target article does arrive, or until there is no further possibility
of it arriving and being accepted (see Section 9.2), and then treat
as under AVAILABLE. Failing that, arrange for the target article to
be rejected and discarded if it does arrive.

   NOTE: It may well be impractical to retain the control message,
   given uncertainty about whether the target article will ever
   arrive. Existing practice in such cases is to assume that
   addresses would match and arrange the equivalent of deletion.
   This is often done by making a spurious entry in a database of
   already-seen message IDs (see Section 9.3), so that if the article
   does arrive, it will be rejected as a duplicate.

The cancel message MUST be propagated onward in the usual fashion,
regardless of which of the four cases applied, so that the target
article will be cancelled everywhere even if cancellation and target
article follow different routes.

   NOTE: [RFC1036] appeared to require stopping cancel propagation in
   the NOT ARRIVED YET case, although the wording was somewhat
   unclear. This appears to have been an unwise decision; there are
   known cases of important cancellations (in situations of
   inadvertent copyright violation, for example) achieving rather
   poorer propagation than the target article. News propagation is
   often a much less orderly process than the authors of [RFC1036]
   apparently envisioned. Modern implementations generally propagate
   the cancellation regardless.

Posting agents meant for use by ordinary posters SHOULD reject an
attempt to post a cancel message if the target article is available
and the mailing address in its From header does not match the one in
the cancel message's From header.

   NOTE: This, again, is primarily accident prevention.

7.2. ihave, sendme

The ihave and sendme control messages implement a crude batched
predecessor of the NNTP [RFC977] protocol. They are largely obsolete
in the Internet but still see use in the UUCP environment, especially
for backup feeds that normally are active only when a primary feed
path has failed.

   NOTE: The ihave and sendme messages defined here have ABSOLUTELY
   NOTHING TO DO WITH NNTP, despite similarities of terminology.

The two messages share the same syntax:

```
ihave-arguments   = *( message-id space ) relayer-name
sendme-arguments  = ihave-arguments
ihave-body        = *( message-id eol )
sendme-body       = ihave-body
```

Message IDs MUST appear in either the arguments or the body, but not
both. Relayers SHOULD generate the form putting message IDs in the
body, but the other form MUST be supported for backward
compatibility.

> NOTE: [RFC1036] made the relayer name optional, but difficulties
> could easily ensue in determining the origin of the message, and
> this option is believed to be unused nowadays. Putting the
> message IDs in the body is strongly preferred over putting them in
> the arguments because it lends itself much better to large numbers
> of message IDs and avoids the empty-body problem mentioned in
> Section 4.3.1.

The ihave message states that the named relayer has filed articles
with the specified message IDs, which may be of interest to the
relayer(s) receiving the ihave message. The sendme message requests
that the relayer receiving it send the articles having the specified
message IDs to the named relayer.

These control messages are normally sent essentially as point-to-
point messages, by using "to." newsgroups (see Section 5.5) that are
sent only to the relayer for which the messages are intended. The
two relayers MUST be neighbors, exchanging news directly with each
other. Each relayer advertises its new arrivals to the other using
ihave messages, and each uses sendme messages to request the articles
it lacks.

> NOTE: Arguably these point-to-point control messages should flow
> by some other protocol, e.g., mail, but administrative and
> interfacing issues are simplified if the news system doesn't need
> to talk to the mail system.

To reduce overhead, ihave and sendme messages SHOULD be sent
relatively infrequently and SHOULD contain substantial numbers of
message IDs. If ihave and sendme are being used to implement a
backup feed, it may be desirable to insert a delay between reception
of an ihave and generation of a sendme, so that a slightly slow
primary feed will not cause large numbers of articles to be requested
unnecessarily via sendme.

7.3. newgroup

   The newgroup control message requests that a new newsgroup be
   created:

        newgroup-arguments = newsgroup-name [space moderation]
        moderation   = "moderated" / "unmoderated"
        newgroup-body   = body
                            / [body] descriptor [body]
        descriptor   = descriptor-tag eol description-line eol
        descriptor-tag   = "For your newsgroups file:"
        description-line   = newsgroup-name space description
        description   = nonblank-text [" (Moderated)"]

   The first argument names the newsgroup to be created, and the second
   one (if present) indicates whether it is moderated. If there is no
   second argument, the default is "unmoderated".

      NOTE: Implementors are warned that there is occasional use of
      other forms in the second argument. It is suggested that such
      violations of this Draft, which are also violations of [RFC1036],
      cause the newgroup message to be ignored. [RFC1036] was slightly
      vague about how second arguments other than "moderated" were to be
      treated (specifically, whether they were illegal or just ignored),
      but it is thought that all existing major implementations will
      handle "unmoderated" correctly, and it appears desirable to
      tighten up the specs to make it possible for other forms to be
      used in future.

   The body is a comment, which software MUST ignore, except that if it
   contains a descriptor, the description line is intended to be
   suitable for addition to a list of newsgroup descriptions. The
   description cannot be continued onto later lines but is not
   constrained to any particular length. Moderated newsgroups have
   descriptions that end with the string " (Moderated)" (note that this
   string begins with a blank).

      NOTE: It is unfortunate that the description line is part of the
      body, rather than being supplied in a header, but this is
      established practice. Newsgroup creators are cautioned that the
      descriptor tag must be reproduced exactly as given above, must be
      alone on a line, and that it is case-sensitive. (To reduce errors
      in this regard, posting agents might wish to question or reject
      newgroup messages that do not contain a descriptor.) Given the
      desire for short lines, description writers should avoid content-
      free phrases like "discussion of" and "news about", and stick to
      defining what the newsgroup is about.

The remainder of the body SHOULD contain an explanation of the
purpose of the newsgroup and the decision to create it.

>   NOTE: Criteria for newsgroup creation vary widely and are outside
>   the scope of this Draft, but if formal procedures of one kind or
>   another were followed in the decision, the body should mention
>   this. Administrators often look for such information when
>   deciding whether to comply with creation/deletion requests.

A newgroup message that lacks an Approved header MUST be ignored.

>   NOTE: It would also be desirable to ignore a newgroup message
>   unless its Approved header names a person who is authorized (in
>   some sense) to create such a newsgroup. A cooperating subnet with
>   sufficiently strong coordination to maintain a correct and current
>   list of authorized creators might wish to do so for its internal
>   newsgroups. It also (or alternatively) might wish to ignore a
>   newgroup message for an internal newsgroup that was posted (or
>   cross-posted) to a non-internal newsgroup.

>   NOTE: As mentioned in Section 6.10, some form of (cryptographic?)
>   authentication of Approved headers would be highly desirable,
>   especially for control messages.

It would be desirable to provide some way of supplying a moderator's
address in a newgroup message for a moderated newsgroup, but this
will cause problems unless effective authentication is available, so
it is left for future work.

>   NOTE: This leaves news administrators stuck with the annoying
>   chore of arranging proper mailing of moderated-newsgroup
>   submissions. On Usenet, this can be simplified by exploiting a
>   forwarding facility that some major sites provide: they maintain
>   forwarding addresses, each the name of a moderated newsgroup with
>   all periods (".", ASCII 46) replaced by hyphens ("-", ASCII 45),
>   which forward mail to the current newsgroup moderators. More
>   advice on the subject of forwarding to moderators can be found in
>   the document titled "How to Construct the Mailpaths File", posted
>   regularly to the Usenet newsgroups news.lists, news.admin.misc,
>   and news.answers.

A newgroup message naming a newsgroup that already exists is
requesting a change in the moderation status or description of the
newsgroup. The same rules apply.

7.4. rmgroup

   The rmgroup message requests that a newsgroup be deleted:

      rmgroup-arguments = newsgroup-name
      rmgroup-body   = body

   The sole argument is the newsgroup name. The body is a comment,
   which software MUST ignore; it SHOULD contain an explanation of the
   decision to delete the newsgroup.

      NOTE: Criteria for newsgroup deletion vary widely and are outside
      the scope of this Draft, but if formal procedures of one kind or
      another were followed in the decision, the body should mention
      this. Administrators often look for such information when
      deciding whether to comply with creation/deletion requests.

   A rmgroup message that lacks an Approved header MUST be ignored.

      NOTE: It would also be desirable to ignore a rmgroup message
      unless its Approved header names a person who is authorized (in
      some sense) to delete such a newsgroup. A cooperating subnet with
      sufficiently strong coordination to maintain a correct and current
      list of authorized deleters might wish to do so for its internal
      newsgroups. It also (or alternatively) might wish to ignore a
      rmgroup message for an internal newsgroup that was posted (or
      cross-posted) to a non-internal newsgroup.

   Unexpected deletion of a newsgroup being a disruptive action,
   implementations are strongly advised to refer rmgroup messages to an
   administrator by default, unless perhaps the message can be
   determined to have originated within a cooperating subnet whose
   members are considered trustworthy. Abuses have occurred.

7.5. sendsys, version, whogets

   The sendsys message requests that a description of the relayer's news
   feeds to other relayers be mailed to the article's reply address:

      sendsys-arguments = [relayer-name]
      sendsys-body   = body

   If there is an argument, relayers other than the one named by the
   argument MUST NOT respond. The body is a comment, which software
   MUST ignore; it SHOULD contain an explanation of the reason for the
   request.

The version message requests that the name and version of the relayer
software be mailed to the reply address:

```
version-arguments  =
version-body       = body
```

There are no arguments. The body is a comment, which software MUST
ignore; it SHOULD contain an explanation of the reason for the
request.

The whogets message requests that a description of the relayer and
its news feeds to other relayers be mailed to the article's reply
address:

```
whogets-arguments  = newsgroup-name [ space relayer-name ]
whogets-body       = body
```

The first argument is the name of the "target newsgroup", specifying
the newsgroup for which propagation information is desired. This
MUST be a complete newsgroup name, not the name of a hierarchy or a
portion of a newsgroup name that is not itself the name of a
newsgroup. If there is a second argument, only the relayer named by
that argument should respond. The body is a comment, which software
MUST ignore; it SHOULD contain an explanation of the reason for the
request.

   NOTE: Whogets is intended as a replacement for sendsys (and
   version) with a precisely specified reply format. Since the
   syntax for specifying what newsgroups get sent to what other
   relayers varies widely between different forms of relayer
   software, the only practical way to standardize the reply format
   is to indicate a specific newsgroup and ask where THAT newsgroup
   propagates. The requirement that it be a complete newsgroup name
   is intended to (largely) avoid the problem of having to answer
   "yes and no" in cases where not all newsgroups in a hierarchy are
   sent.

Any of these messages lacking an Approved header MUST be ignored.
Response to any of these messages SHOULD be delayed for at least
24 hours, and no response should be attempted if the message has been
cancelled in that time. Also, no response SHOULD be attempted unless
the local part of the destination address is "newsmap". News
administrators SHOULD arrange for mail to "newsmap" on their systems
to be discarded (without reply) unless legitimate use is in progress.

   NOTE: Because these messages can cause many, many relayers to send
   mail to one person, such messages, specifying mailing to an
   innocent person's mailbox, have been forged as a half-witted

practical joke. A delay gives administrators time to notice a
fraudulent message and act (by cancelling the message, preparing
to divert the flood of mail into the bit bucket, or both).
Restriction of the destination address to "newsmap" reduces the
appeal of fraud by making it impossible to use it to harass a
normal user. (A site that does NOT discard mail to "newsmap", but
rather bounces it back, may incur higher communications costs than
if the mail had been accepted into a user's mailbox, but a
malicious forger could accomplish this anyway, by using an address
whose local part is very unlikely to be a legitimate mailbox
name.)

NOTE: [RFC1036] did not require the Approved header for these
control messages. This has been added because of the possibility
that cryptographic authentication of Approved headers will become
available.

The body of the reply to a sendsys message SHOULD be of the form:

```
sendsys-reply       = responder 1*sys-line
responder           = "Responding-System:" space domain eol
sys-line            = relayer-name ":" newsgroup-patterns
                          [ ":" text ] eol
newsgroup-patterns = newsgroup-name *( "," newsgroup-name )
```

The first line identifies the responding system, using a syntax
resembling a header (but note that it is part of the BODY).
Remaining lines indicate what newsgroups are sent to what other
systems. The syntax of newsgroup patterns is not well standardized;
the form described is common (often with newsgroup names only
partially given, denoting all names starting with a particular set of
components) but not universal. The whogets message provides a
better-defined alternative.

The reply to a version message is of somewhat ill-defined form, with
a body normally consisting of a single line of text that somehow
describes the version of the relayer software. The whogets message
provides a better-defined alternative.

The body of the reply to a whogets message MUST be of the form:

```
whogets-reply      = responder-domain responder-relayer
                       response-date responding-to arrived-via
                       responder-version whogets-delimiter
                       *pass-line
responder-domain   = "Responding-System:" space domain eol
responder-relayer  = "Responding-Relayer:" space relayer-name eol
response-date      = "Response-Date:" space date eol
responding-to      = "Responding-To:" space message-id eol
arrived-via        = "Arrived-Via:" path-list eol
responder-version  = "Responding-Version:" space nonblank-text eol
whogets-delimiter  = eol
pass-line          = relayer-name [ space domain ] eol
```

The first six lines identify the responding relayer by its Internet
domain name (use of the ".uucp" and ".bitnet" pseudo-domains is
permissible, for registered hosts in them, but discouraged) and its
relayer name; specify the date when the reply was generated and the
message ID of the whogets message being replied to; give the path
list (from the Path header) of the whogets message (which MAY, if
absolutely necessary, be truncated to a convenient length, but MUST
contain at least the leading three relayer names); and indicate the
version of relayer software responding. Note that these lines are
part of the BODY even though their format resembles that of headers.
Despite the apparently fixed order specified by the syntax above,
they can appear in any order, but there must be exactly one of each.

After those preliminaries, and an empty line to unambiguously define
their end, the remaining lines are the relayer names (which MAY be
accompanied by the corresponding domain names, if known) of systems
to which the responding system passes the target newsgroup. Only the
names of news relayers are to be included.

   NOTE: It is desirable for a reply to identify its source by both
   domain name and relayer name because news propagation is governed
   by the latter but location in a broader context is best determined
   by the former. The date and whogets message ID should, in
   principle, be present in the MAIL headers but are included in the
   body for robustness in the presence of uncooperative mail systems.
   The reason for the path list is discussed below. Adding version
   information eliminates the need for a separate message to gather
   it.

NOTE: The limitation of pass lines to contain only names of news
relayers is meant to exclude names used within a single host (as
identifiers for mail gateways, portions of ihave/sendme
implementations, etc.), which do not actually refer to other
hosts.

A relayer that is unaware of the existence of the target newsgroup
MUST NOT reply to a whogets message at all, although this MUST NOT
influence decisions on whether to pass the article on to other
relayers.

NOTE: While this may result in discontinuous maps in cases where
some hosts have not honored requests for creation of a newsgroup,
it will also prevent a flood of useless responses in the event
that a whogets message intended to map a small region "leaks" out
to a larger one. The possibility of discontinuous recognition of
a newsgroup does make it important that the whogets message itself
continue to propagate (if other criteria permit). This is also
the reason for the inclusion of the whogets message's path list,
or at least the leading portion of it, in the reply: to permit
reconstruction of at least small gaps in maps.

Different networks set different rules for the legitimacy of these
messages, given that they may reveal details of organization-internal
topology that are sometimes considered proprietary.

NOTE: On Usenet, in particular, willingness to respond to these
messages is held to be a condition of network membership: the
topology of Usenet is public information. Organizations wishing
to belong to such networks while keeping their internal topology
confidential might wish to organize their internal news software
so that all articles reaching outsiders appear to be from a single
"gatekeeper" system, with the details of internal topology hidden
behind that system.

UNRESOLVED ISSUE: It might be useful to have a way to set some
sort of hop limit for these.

7.6. checkgroups

   The checkgroups control message contains a supposedly authoritative
   list of the valid newsgroups within some subset of the newsgroup name
   space:

      checkgroups-arguments =
      checkgroups-body   = [invalidation] valid-groups
                             / invalidation
      invalidation   = "!" plain-component
                               *("," plain-component) eol
      valid-groups   = 1*(description-line eol)

   There are no arguments. The body lines (except possibly for an
   initial invalidation) each contain a description line for a
   newsgroup, as defined under the newgroup message (Section 7.3).

      NOTE: Some other, ill-defined, forms of the checkgroups body were
      formerly used. See Appendix A.

   The checkgroups message applies to all hierarchies containing any of
   the newsgroups listed in the body. The checkgroups message asserts
   that the newsgroups it lists are the only newsgroups in those
   hierarchies. If there is an invalidation, it asserts that the
   hierarchies it names no longer contain any newsgroups.

   Processing a checkgroups message MAY cause a local list of newsgroup
   descriptions to be updated. It SHOULD also cause the local lists of
   newsgroups (and their moderation statuses) in the mentioned
   hierarchies to be checked against the message. The results of the
   check MAY be used for automatic corrective action or MAY be reported
   to the news administrator in some way.

      NOTE: Automatically updating descriptions of existing newsgroups
      is relatively safe. In the case of newsgroup additions or
      deletions, simply notifying the administrator is generally the
      wisest action, unless perhaps the message can be determined to
      have originated within a cooperating subnet whose members are
      considered trustworthy.

      NOTE: There is a problem with the checkgroups concept: not all
      newsgroups in a hierarchy necessarily propagate to the same set of
      machines. (Notably, there is a set of newsgroups known as the
      "inet" newsgroups, which have relatively limited distribution but
      coexist in several hierarchies with more widely distributed
      newsgroups.) The advice of checkgroups should always be taken
      with a grain of salt and should never be followed blindly.

8. Transmission Formats

   While this Draft does not specify transmission methods, except to
   place a few constraints on them, there are some data formats used
   only for transmission that are unique to news.

8.1. Batches

   For efficient bulk transmission and processing of news articles, it
   is often desirable to transmit a number of them as a single block of
   data, i.e., a "batch". The format of a batch is:

       batch   = 1*(batch-header article)
       batch-header = "#! rnews " article-size eol
       article-size = 1*digit

   A batch is a sequence of articles, each prefixed by a header line
   that includes its size. The article size is a decimal count of the
   octets in the article, counting each EOL as one octet regardless of
   how it is actually represented.

      NOTE: A relayer might wish to accept either a single article or a
      batch as input. Since "#" cannot appear in a header name,
      examination of the first octet of the input will reveal its
      nature.

      NOTE: In the header line, there is exactly one blank before
      "rnews", there is exactly one blank after "rnews", and the EOL
      immediately follows the article size. Beware that some software
      inserts non-standard trash after the size.

      NOTE: Despite the similarity of this format to the executable-
      script format used by some operating systems, it is EXTREMELY
      unwise to just feed incoming batches to a command interpreter in
      the anticipation that it will run a command named "rnews" to
      process the batch. Unless arrangements are made to very tightly
      restrict the range of commands that can be executed by this means,
      the security implications are disastrous.

8.2. Encoded Batches

   When transmitting news, especially over communications links that are
   slow or are billed by the bit, it is often desirable to batch news
   and apply data compression to the batches. Transmission links
   sending compressed batches SHOULD use out-of-band means of
   communication to specify the compression algorithm being used. If
   there is no way to send out-of-band information along with a batch,
   the following encapsulation for a compressed batch MAY be used:

        ec-batch   = "#! " compression-keyword eol
                               compressed-batch
        compression-keyword = "cunbatch"

   A line containing a keyword indicating the type of compression is
   followed by the compressed batch. The only truly widespread
   compression keyword at present is "cunbatch", indicating compression
   using the widely distributed "compress" program. Other compression
   keywords MAY be used by mutual agreement between the hosts involved.

      NOTE: An encapsulated compressed batch is NOT, in general, a text
      file, despite having an initial text line. This combination of
      text and non-text data is often awkward to handle; for example,
      standard decompression programs cannot be used without first
      stripping off the initial line, and that in turn is painful to do
      because many text-handling tools that are superficially suited to
      the job do not cope well with non-text data, hence the
      recommendation that out-of-band communication be used instead when
      possible.

      NOTE: For UUCP transmission, where a batch is typically
      transmitted by invoking the remote command "rnews" with the batch
      as its input stream, a plausible out-of-band method for indicating
      a compression type would be to give a compression keyword in an
      option to "rnews", perhaps in the form:

      rnews -d decompressor

      where "decompressor" is the name of a decompression program (e.g.,
      "uncompress" for a batch compressed with "compress" or "gunzip"
      for a batch compressed with "gzip"). How this decompression
      program is located and invoked by the receiving relayer is
      implementation-specific.

      NOTE: See the notes in Section 8.1 on the inadvisability of
      feeding batches directly to command interpreters.

NOTE: There is exactly one blank between "#!" and the compression
keyword, and the EOL immediately follows the keyword.

8.3. News within Mail

It is often desirable to transmit news as mail, either for the
convenience of a human recipient or because that is the only type of
transmission available on a restrictive communication path.

Given the similarity between the news format and the MAIL format, it
is superficially attractive to just send the news article as a mail
message. This is typically a mistake: mail-handling software often
feels free to manipulate various headers in undesirable ways (in some
cases, such as Sender, such manipulation is actually mandatory), and
mail transmission problems, etc. MUST be reported to the
administrators responsible for the mail transmission rather than to
the article's author. In general, news sent as mail should be
encapsulated to separate the MAIL headers and the news headers.

When the intended recipient is a human, any convenient form of
encapsulation may be used. Recommended practice is to use MIME
encapsulation with a content type of "message/news", given that news
articles have additional semantics beyond what "message/rfc822"
implies.

NOTE: "message/news" was registered as a standard subtype by IANA
22 June 1993.

When mail is being used as a transmission path between two relayers,
however, a standard method is desirable. Currently the standard
method is to send the mail to an address whose local part is "rnews",
with whatever MAIL headers are necessary for successful transmission.
The news article (including its headers) is sent as the body of the
mail message, with an "N" prepended to each line.

NOTE: The "N" reduces the probability of an innocent line in a
news article being taken as a magic command to mail software and
makes it easy for receiving software to strip off any lines added
by mail software (e.g., the trailing empty line added by some UUCP
mail software).

This method has its weaknesses. In particular, it assumes that the
mail transmission channel can transmit nearly arbitrary body text
undamaged. When mail is being used as a transmission path of last
resort, however, the mail system often has inconvenient preconceived
notions about the format of message bodies. Various ad hoc encoding
schemes have been used to avoid such problems. The recommended
method is to send a news article or batch as the body of a MIME mail

message, using content type "application/news-transmission" and
MIME's "base64" encoding (which is specifically designed to survive
all known major mail systems).

   NOTE: In the process, MIME conventions could be used to fragment
   and reassemble an article that is too large to be sent as a single
   mail message over a transmission path that restricts message
   length. In addition, the "conversions" parameter to the content
   type could be used to indicate what (if any) compression method
   has been used. Also, the Content-MD5 header [RFC1544] can be used
   as a "checksum" to provide high confidence of detecting accidental
   damage to the contents.

   UNRESOLVED ISSUE: The "conversions" parameter no longer exists.
   What should be done about this, if anything?

   NOTE: It might look tempting to use a content type such as
   "message/X-netnews", but MIME bans non-trivial encodings of the
   entire body of messages with content type "message". The intent
   is to avoid obscuring nested structure underneath encodings. For
   inter-relayer news transmission, there is no nested structure of
   interest, and it is important that the entire article (including
   its headers, not just its body) be protected against the vagaries
   of intervening mail software. This situation appears to fit the
   MIME description of circumstances in which "application" is the
   proper content type.

   NOTE: "application/news-transmission", with a "conversions"
   parameter, was registered as a standard subtype by IANA
   22 June 1993.

   UNRESOLVED ISSUE: The "conversions" parameter no longer exists in
   MIME. What should we do about this?

8.4. Partial Batches

   UNRESOLVED ISSUE: The existing batch conventions assemble
   (potentially) many articles into one batch. Handling very large
   articles would be substantially less troublesome if there was also
   a fragmentation convention for splitting a large article into
   several batches. Is this worth defining at this time?

9. Propagation and Processing

  Most aspects of news propagation and processing are implementation-
  specific. The basic propagation algorithms, and certain details of
  how they are implemented, nevertheless need to be standard.

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.

However, in the case of news there is an even more important
principle, derived from a much older code of practice, the
Hippocratic Oath (we will thus call this the Hippocratic Principle):

   First, do no harm.

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

9.1. Relayer General Issues

Relayers MUST NOT alter the content of articles unnecessarily. Well-
intentioned attempts to "improve" headers, in particular, typically
do more harm than good. It is necessary for a relayer to prepend its
own name to the Path content (see Section 5.6) and permissible for it
to rewrite or delete the Xref header (see Section 6.12). Relayers
MAY delete the thoroughly obsolete headers described in Appendix A.3,
although this behavior no longer seems useful enough to encourage.
Other alterations SHOULD be avoided at all costs, as per the
Hippocratic Principle.

   NOTE: As discussed in Section 2.3, tidying up the headers of a
   user-prepared article is the job of the posting agent, not the
   relayer. The relayer's purpose is to move already-compliant
   articles around efficiently without damaging them. Note that in
   existing implementations, specific programs may contain both
   posting-agent functions and relayer functions. The distinction is
   that posting-agent functions are invoked only on articles posted
   by local posters, never on articles received from other relayers.

   NOTE: A particular corollary of this rule is that relayers should
   not add headers unless truly necessary. In particular, this is
   not SMTP; do not add Received headers.

Relayers MUST NOT pass non-conforming articles on to other relayers,
except perhaps in a cooperating subnet that has agreed to permit
certain kinds of non-conforming behavior. This is a direct
consequence of the Internet Robustness Principle.

The two preceding paragraphs may appear to be in conflict. What is
to be done when a non-conforming article is received? The Robustness
Principle argues that it should be accepted but must not be passed on
to other relayers while still non-conforming, and the Hippocratic
Principle strongly discourages attempts at repair. The conclusion
that this appears to lead to is correct: a non-conforming article MAY
be accepted for local filing and processing, or it MAY be discarded
entirely, but it MUST NOT be passed on to other relayers.

A relayer MUST NOT respond to the arrival of an article by sending
mail to any destination, other than a local administrator, except by
explicit prearrangement with the recipient. Neither posting an
article (other than certain types of control messages; see
Section 7.5) nor being the moderator of a moderated newsgroup
constitutes such prearrangement. UNDER NO CIRCUMSTANCES WHATSOEVER
may a relayer attempt to send mail to either an article's originator
or a moderator.

   NOTE: Reporting apparent errors in message composition is the job
   of a posting agent, not a relayer. The same is true of mailing
   moderated-newsgroup postings to moderators. In networks of
   thousands of cooperating relayers, it is simply unacceptable for
   there to be any circumstance whatsoever that causes any
   significant fraction of them to simultaneously send mail to the
   same destination. (Some control messages are exceptions, although
   perhaps ill-advised ones.) What might, in a smaller network, be a
   useful notification or forwarding becomes a deluge of nearly
   identical messages that can bring mail software to its knees and
   severely inconvenience recipients. Moderators, in particular,
   historically have suffered grievously from this.

Notification of problems in incoming articles MAY go to local
administrators, or at most (by prearrangement!) to the
administrators of the neighboring relayer(s) that passed on the
problematic articles.

   NOTE: It would be desirable to notify the author that his posting
   is not propagating as he expects. However, there is no known
   method for doing this that will scale up gracefully. (In
   particular, "notify only if within N relayers of the originator"
   falls down in the presence of commercial news services like UUNET:
   there may be hundreds or thousands of relayers within a couple of
   hops of the originator.) The best that can be done right now is
   to notify neighbors, in hopes that the word will eventually
   propagate up the line, or organize regional monitoring at major
   hubs.

If it is necessary to alter an article, e.g., translate it to another
character set or alter its EOL representation, strenuous efforts
should be made to ensure that such transformations are reversible,
and that relayers or other software that might wish to reverse them
know exactly how to do so.

   NOTE: For example, a cooperating subnet that exchanges articles
   using a non-ASCII character set like EBCDIC should define a
   standard, reversible ASCII-EBCDIC mapping and take pains to see
   that it is used at all points where the subnet meets the outside.
   If the only reason for using EBCDIC is that the readers typically
   employ EBCDIC devices, it would be more robust to employ ASCII as
   the interchange format and do the transformation in the reading
   and posting agents.

9.2. Article Acceptance and Propagation

   When a relayer first receives an article, it must decide whether to
   accept it. (This applies regardless of whether the article arrived
   by itself or as part of a batch, and in principle regardless of
   whether it originated as a local posting or as traffic from another
   relayer.) In a cooperating subnet with well-controlled propagation
   paths, some of the tests specified here MAY be delegated to centrally
   located relayers; that is, relayers that can receive news ONLY via
   one of the central relayers might simplify acceptance testing based
   on the assumption that incoming traffic has already passed the full
   set of tests at a central relayer.

   The wording that follows is based on a model in which articles arrive
   on a relayer's host before acceptance tests are done. However,
   depending on the degree of integration of the transport mechanisms
   and the relayer, some or all of these tests MAY be done before the
   article is actually transmitted, so that articles that definitely
   will not be accepted need not be transmitted at all.

   The wording that follows also specifies a particular order for the
   acceptance tests. While this order is the obvious one, the tests MAY
   be done in any order.

   First, the relayer MUST verify that the article is a legal news
   article, with all mandatory headers present with legal contents.

   NOTE: This check in principle is done by the first relayer to see
   an article, so an article received from another relayer should
   always be legal, but there is enough old software still
   operational that this cannot be taken for granted; see the
   discussion of the Internet Robustness Principle in Section 9.1.

Second, the relayer MUST determine whether it has already seen this
article (identified by its message ID). This is normally done by
retaining a history of all article message IDs seen in the last
N days, where the value of N is decided by the relayer's
administrator but SHOULD be at least 7. Since N cannot practically
be infinite, articles whose Date content indicates that they are
older than N days are declared "stale" and are deemed to have been
seen already.

     NOTE: This check is important because news propagation topology is
     typically redundant, often highly so, and it is not at all
     uncommon for a relayer to receive the same article from several
     neighbors. The history of already-seen message IDs can get quite
     large, hence, the desire to limit its length, but it is important
     that it be long enough that slowly propagating articles are not
     classed as stale. News propagation within the Internet is
     normally very rapid, but when UUCP links are involved, end-to-end
     delays of several days are not rare, so a week is not a
     particularly generous minimum.

     NOTE: Despite generally more rapid propagation in recent times, it
     is still not unheard of for some propagation paths to be very
     slow. This can introduce the possibility of old articles arriving
     again after they are gone from the history, hence the "stale"
     rule.

Third, the relayer MUST determine whether any of the article's
newsgroups are "subscribed to" by the host, i.e., fit a description
of what hierarchies or newsgroups the site wants to receive.

     NOTE: This check is significant because information on what
     newsgroups a relayer wishes to receive is often stored at its
     neighbors, who may not have up-to-date information or may simplify
     the rules for implementation reasons. As a hedge against the
     possibility of missed or delayed newgroup control messages,
     relayers may wish to observe a notion of a newsgroup subscription
     that is independent of the list of newsgroups actually known to
     the relayer. This would permit reception and relaying of articles
     in newsgroups that the relayer is not (yet) aware of, subject to
     more general criteria indicating that they are likely to be of
     interest.

Once an article has been accepted, it may be passed on to other
relayers. The fundamental news propagation rule is a flooding
algorithm: on receiving and accepting an article, send it to all
neighboring relayers not already in its path list that are sent its
newsgroup(s) and distribution(s).

   NOTE: The path list's role in loop prevention may appear
   relatively unimportant, given that looping articles would
   typically be rejected as duplicates anyway. However, the path
   list's role in preventing superfluous transmissions is not
   trivial. In particular, the path list is the only thing that
   prevents relayer X, on receiving an article from relayer Y, from
   sending it back to Y again. (Indeed, the usual symptom of
   confusion about relayer names is that incoming news loops back in
   this manner.) The looping articles would be rejected as
   duplicates, but doubling the communications load on every news
   transmission path is not to be taken lightly!

   In general, relayers SHOULD NOT make propagation decisions by
   "anticipation": relayer X, noting that the article's path list
   already contains relayer Y, decides not to send it to relayer Z
   because X anticipates that Z will get the article by a better path.
   If that is generally true, then why is there a news feed from X to Z
   at all? In fact, the "better path" may be running slowly or may be
   down. News propagation is very robust precisely because some
   redundant transmission is done "just in case". If it is imperative
   to limit unnecessary traffic on a path, use of NNTP [RFC977] or
   ihave/sendme (see Section 7.2) to pass articles only when necessary
   is better than arbitrary decisions not to pass articles at all.

   Anticipation is occasionally justified in special cases. Such cases
   should involve both (1) a cooperating subnet whose propagation paths
   are well-understood and well-monitored, with failures and slowdowns
   noticed and dealt with promptly, and (2) a persistent pattern of
   heavy unnecessary traffic on a path that is either slow or costly.
   In addition, there should be some reason why neither NNTP nor
   ihave/sendme is suitable as a solution to the problem.

9.3. Administrator Contact

   It is desirable to have a standardized contact address for a
   relayer's administrators, in the spirit of the "postmaster" address
   for mail administrators. Mail addressed to "newsmaster" on a
   relayer's host MUST go to the administrator(s) of that relayer. Mail
   addressed to "usenet" on the relayer's host SHOULD be handled
   likewise. Mail addressed to either address on other hosts using the
   same news database SHOULD be handled likewise.

   NOTE: These addresses are case-sensitive, although it would be
   desirable for sequences equivalent to them using case-insensitive
   comparison to be handled likewise. While "newsmaster" seems the
   preferred network-independent address, by analogy to "postmaster",
   there is an existing practice of using "usenet" for this purpose,

and so "usenet" should be supported if at all possible (especially
on hosts belonging to Usenet!). The address "news" is also
sometimes used for purposes like this, but less consistently.

10. Gatewaying

Gatewaying of traffic between news networks using this Draft and
those using other exchange mechanisms can be useful but must be done
cautiously. Gateway administrators are taking on significant
responsibilities and must recognize that the consequences of error
can be quite serious.

10.1. General Gatewaying Issues

This section will primarily address the problems of gatewaying
traffic INTO news networks. Little can be said about the other
direction without some specific knowledge of the network(s) involved.
However, the two issues are not entirely independent: if a non-news
network is gatewayed into a news network at more than one point,
traffic injected into the non-news network by one gateway may appear
at another as a candidate for injection back into the news network.

This raises a more general principle, the single most important issue
for gatewaying:

   Above all, prevent loops.

The normal loop prevention of news transmission is vitally dependent
on the Message-ID header. Any gateway that finds it necessary to
remove this header, alter it, or supersede it (by moving it into the
body) MUST take equally effective precautions against looping.

   NOTE: There are few things more effective at turning news readers
   into a lynch mob than a malfunctioning gateway, or pair of
   gateways, that takes in news articles, mangles them just enough to
   prevent news relayers from recognizing them as duplicates, and
   regurgitates them back into the news stream. This happens rather
   too often.

Gateway implementors should realize that gateways have all of the
responsibilities of relayers, plus the added complications introduced
by transformations between different information formats. Much of
the discussion in Section 9 about relayer issues is relevant to
gateways as well. In particular, gateways SHOULD keep a history of
recently seen articles, as described in Section 9.2, and not assume
that articles will never reappear. This is particularly important
for networks that have their own concept analogous to message IDs: a
gateway should keep a history of traffic seen from BOTH directions.

If at all possible, articles entering the non-news network SHOULD be
marked in some way so that they will NOT be re-gatewayed back into
news. Multiple gateways obviously must agree on the marking method
used; if it is done by having them know each others' names, name
changes MUST be coordinated with great care. If marking cannot be
done, all transformations MUST be reversible so that a re-gatewayed
article is identical to the original (except perhaps for a longer
Path header).

Gateways MUST NOT pass control messages (articles containing Control,
Also-Control, or Supersedes headers) without removing the headers
that make them control messages, unless there are compelling reasons
to believe that they are relevant to both sides and that conventions
are compatible. If it is truly desirable to pass them unaltered,
suitable precautions MUST be taken to ensure that there is NO
POSSIBILITY of a looping control message.

   NOTE: The damage done by looping articles is multiplied a
   thousandfold if one of the affected articles is something like a
   sendsys message (see Section 7.5) that requests multiple automatic
   replies. Most gateways simply should not pass control messages at
   all. If some unusual reason dictates doing so, gateway
   implementors and administrators are urged to consider bulletproof
   rate-limiting measures for the more destructive ones like sendsys,
   e.g., passing only one per hour no matter how many are offered.

Gateways, like relayers, SHOULD make determined efforts to avoid
mangling articles unnecessarily. In the case of gateways, some
transformations may be inevitable, but keeping them to a minimum and
ensuring that they are reversible is still highly desirable.

Gateways MUST avoid destroying information. In particular, the
restrictions of Section 4.2.2 are best taken with a grain of salt in
the context of gateways. Information that does not translate
directly into news headers SHOULD be retained, perhaps in "X-"
headers, both because it may be of interest to sophisticated readers
and because it may be crucial to tracing propagation problems.

Gateway implementors should take particular note of the discussion of
mailed replies, or more precisely the ban on same, in Section 9.1.
Gateway problems MUST be reported to the local administration, not to
the innocent originator of traffic. "Gateway problems" here includes
all forms of propagation anomaly on the non-news side of the gateway,
e.g., unreachable addresses on a mailing list. Note that this
requires consideration of possible misbehavior of "downstream" hosts,
not just the gateway host.

10.2. Header Synthesis

   News articles prepared by gateways MUST be legal news articles. In
   particular, they MUST include all of the mandatory headers (see
   Section 5) and MUST fully conform to the restrictions on said
   headers. This often requires that a gateway function not only as a
   relayer but also partly as a posting agent, aiding in the synthesis
   of a conforming article from non-conforming input.

      NOTE: The full-conformance requirement needs particularly careful
      attention when gatewaying mailing lists to news, because a number
      of constructs that are legal in MAIL headers are NOT permissible
      in news headers. (Note also that not all mail traffic fully
      conforms to even the MAIL specification.) The rest of this
      section will be phrased in terms of mail-to-news gatewaying, but
      most of it is more generally applicable.

   The mandatory headers generally present few problems.

   If no date information is available, the gateway should supply a Date
   header with the gateway's current date. If only partial information
   is available (e.g., date but not time), this should be fleshed out to
   a full Date header by adding default values, not by mixing in parts
   of the gateway's current date. (Defaults should be chosen so that
   fleshed-out dates will not be in the future!) It may be necessary to
   map time zone information to the restricted forms permitted in the
   news Date header. See Section 5.1.

      NOTE: The prohibition of mixing dates is on the theory that it is
      better to admit ignorance than to lie.

   If the author's address as supplied in the original message is not
   suitable for inclusion in a From header, the gateway MUST transform
   it so it is (for example, by use of the "% hack" and the domain
   address of the gateway). The desire to preserve information is NOT
   an excuse for violating the rules. If the transformation is drastic
   enough that there is reason to suspect loss of information, it may be
   desirable to include the original form in an "X-" header, but the
   From header's contents MUST be as specified in Section 5.2.

   If the message contains a Message-ID header, the contents should be
   dealt with as discussed in Section 10.3. If there is no message ID
   present, it will be necessary to synthesize one, following the news
   rules (see Section 5.3).

   Every effort should be made to produce a meaningful Subject header;
   see Section 5.4. Many news readers select articles to read based on
   Subject headers, and inserting a placeholder like "<no subject

available>" is considered highly objectionable. Even synthesizing a
Subject header by picking out the first half-dozen nouns and
adjectives in the article body is better than using a placeholder,
since it offers SOME indication of what the article might contain.

The contents of the Newsgroups header (Section 5.5) are usually
predetermined by gateway configuration, but a gateway to a network
that has its own concept of newsgroups or discussions might have to
make transformations. Such transformations should be reversible;
otherwise, confusion is likely on both sides.

It will rarely be possible for gateways to provide a Path header that
is both an accurate history of the relayers the article has passed
through AS NEWS and a usable reply address. The history function
MUST be given priority; see the discussion in Section 5.6. It will
usually be necessary for a gateway to supply an empty path list,
abandoning the reply function.

It is desirable for gatewayed articles to convey as much useful
information as possible, e.g., by use of optional news headers (see
Section 6) when the relevant information is available. Synthesis of
optional headers can generally follow similar rules.

Software synthesizing References headers should note the discussion
in Section 6.5 concerning the incompatibility between MAIL and news.
Also of interest is the possibility of incorporating information from
In-Reply-To headers and from attribution lines in the body; an
incomplete or somewhat conjectural References header is much better
than none at all, and reading agents already have to cope with
incomplete or slightly erroneous References lists.

10.3. Message ID Mapping

   This section, like the previous one, is phrased in terms of mail
   being gatewayed into news, but most of the discussion should be more
   generally applicable.

   A particularly sticky problem of gatewaying mail into news is
   supplying legal news message IDs. Note, in particular, that not all
   MAIL message IDs are legal in news; the news syntax (specified in
   Section 5.3, with related material in Section 5.2) is more
   restrictive. Generating a fully conforming news article from a mail
   message may require transforming the message ID somewhat.

   Generation and transformation of message IDs assumes particular
   importance if a given mailing list (or whatever) is being handled by
   more than one gateway. It is highly desirable that the same article
   contents not appear twice in the same newsgroup, which requires that

they receive the same message ID from all gateways. Gateways SHOULD
use the following algorithm (possibly modified by the later
discussion of gatewaying into more than one newsgroup) unless local
considerations dictate another:

   1. Separate message ID from surroundings, if necessary. A
      plausible method for this is to start at the first "<", end at
      the next ">", and reject the message if no ">" is found or a
      second "<" is seen before the ">". Also reject the message if
      the message ID contains no "@" or more than one "@", or if it
      contains no ".". Also reject the message if the message ID
      contains non-ASCII characters, ASCII control characters, or
      white space.

         NOTE: Any legitimate domain will include at least one ".".
         [RFC822], Section 6.2.2, forbids white space in this context
         when passing mail on to non-MAIL software.

   2. Delete the leading "<" and trailing ">". Separate message ID
      into local part and domain at the "@".

   3. In both components, transliterate leading dots (".", ASCII 46),
      trailing dots, and dots after the first in sequences of two or
      more consecutive dots, into underscores (ASCII 95).

   4. In both components, transliterate disallowed characters other
      than dots (see the definition of <unquoted-char> in
      Section 5.2) to underscores (ASCII 95).

   5. Form the message ID as

         "<" local-part "@" domain ">"

   NOTE: This algorithm is approximately that of Rich Salz's
   successful gatewaying package.

Despite the desire to keep message IDs consistent across multiple
gateways, there is also a more subtle issue that can require a
different approach. If the same articles are being gatewayed into
more than one newsgroup, and it is not possible to arrange that all
gateways gateway them to the same cross-posted set of newsgroups,
then the message IDs in the different newsgroups MUST be DIFFERENT.

   NOTE: Otherwise, arrival of an article in one newsgroup will
   prevent it from appearing in another, and which newsgroup a
   particular article appears in will be an accident of which
   direction it arrives from first. It is very difficult to maintain
   a coherent discussion when each participant sees a randomly

selected 50% of the traffic. The fundamental problem here is that
the basic assumption behind message IDs is being violated: the
gateways are assigning the same message ID to articles that differ
in an important respect (Newsgroups header).

In such cases, it is suggested that the newsgroup name, or an agreed-
on abbreviation thereof, be prepended to the local part of the
message ID (with a separating ".") by the gateway. This will ensure
that multiple gateways generate the same message ID, while also
ensuring that different newsgroups can be read independently.

   NOTE: It is preferable to have the gateway(s) cross-post the
   article, avoiding the issue altogether, but this may not be
   feasible, especially if one newsgroup is widespread and the other
   is purely local.

10.4. Mail to and from News

Gatewaying mail to news, and vice versa, is the most obvious form of
news gatewaying. It is common to set up gateways between news and
mail rather too casually.

It is hard to go very wrong in gatewaying news into a mailing list,
except for the non-trivial matter of making sure that error reports
go to the local administration rather than to the authors of news
articles. (This requires attention to the "envelope address" as well
as to the message headers.) Doing the reverse connection correctly
is much harder than it looks.

   NOTE: In particular, just feeding the mail message to "inews -h"
   or the equivalent is NOT, repeat NOT, adequate to gateway mail to
   news. Significant gatewaying software is necessary to do it
   right. Not all headers of mail messages conform to even the MAIL
   specifications, never mind the stricter rules for news.

It is useful to distinguish between two different forms of
mail-to-news gatewaying: gatewaying a mailing list into a newsgroup,
and operating a "post-by-mail" service in which individual articles
can be posted to a newsgroup by mailing them to a specific address.
In the first case, the message is already being "broadcast", and the
situation can be viewed as gatewaying one form of news into another.
The second case is closer to that of a moderator posting submissions
to a moderated newsgroup.

In either case, the discussions in the preceding two sections are
relevant, as is the Hippocratic Principle of Section 9. However,
some additional considerations are specific to mail-to-news
gatewaying.

As mentioned in Section 6, point-to-point headers like To and Cc
SHOULD NOT appear as such in news, although it is suggested that they
be transformed to "X-" headers, e.g., X-To and X-Cc, to preserve
their information content for possible use by readers or
troubleshooters. The Received header is entirely specific to MAIL
and SHOULD be deleted completely during gatewaying, except perhaps
for the Received header supplied by the gateway host itself.

The Sender header is a tricky case, one where mailing-list and post-
by-mail practice should differ. For gatewaying mailing lists, the
mailing-list host should be considered a relayer, and the From and
Sender headers supplied in its transmissions left strictly untouched.
For post-by-mail, as for a moderator posting a mailed submission, the
Sender header should reflect the poster rather than the author. If a
post-by-mail gateway receives a message with its own Sender header,
it might wish to preserve the content in an X-Sender header.

It will generally be necessary to transform between mail's
In-Reply-To/References convention and news's References/See-Also
convention, to preserve correct semantics of cross references. This
also requires attention when going the other way, from news to mail.
See the discussion of the difference in Section 6.5.

10.5. Gateway Administration

Any news system will benefit from an attentive administrator,
preferably assisted by automated monitoring for anomalies. This is
particularly true of gateways. Gateway software SHOULD be
instrumented so that unusual occurrences, such as sudden massive
surges in traffic, are reported promptly. It is desirable, in fact,
to go further: gateway software SHOULD endeavor to limit damage in
the event that the administrator does not respond promptly.

   NOTE: For example, software might limit the gatewaying rate by
   queueing incoming traffic and emptying the queue at a finite
   maximum rate (well below the maximum that the host is capable of!)
   that is set by the administrator and is not raised automatically.

Traffic gatewayed into a news network SHOULD include a suitable
header, perhaps X-Gateway-Administrator, giving an electronic address
that can be used to report problems. This SHOULD be an address that
goes directly to a human, and not to a "routine administrative
issues" mailbox that is examined only occasionally, since the point
is to be able to reach the administrator quickly in an emergency.
Gateway administrators SHOULD arrange substitutes to cover gateway
operation (with suitable redirection of mail) when they are on
vacation, etc.

11. Security and Related Issues

   Although the interchange format itself raises no significant security
   issues, the wider context does.

11.1. Leakage

   The most obvious form of security problem with news is "leakage" of
   articles that are intended to have only restricted circulation. The
   flooding algorithm is EXTREMELY good at finding any path by which
   articles can leave a subnet with supposedly restrictive boundaries.
   Substantial administrative effort is required to ensure that local
   newsgroups remain local, unless connections to the outside world are
   tightly restricted.

   A related problem is that the sendme control message can be used to
   ask for any article by its message ID. The usefulness of this has
   declined as message-ID generation algorithms have become less
   predictable, but it remains a potential problem for "secure"
   newsgroups. Hosts with such newsgroups may wish to disable the
   sendme control message entirely.

   The sendsys, version, and whogets control messages also allow
   "outsiders" to request information from "inside", which may reveal
   details of internal topology (etc.) that are considered
   confidential. (Note that at least limited openness about such
   matters may be a condition of membership in such networks, e.g.,
   Usenet.)

   Organizations wishing to control these forms of leakage are strongly
   advised to designate a small number of "official gateway" hosts to
   handle all news exchange with the outside world, so that a bounded
   amount of administrative effort is needed to control propagation and
   eliminate problems. Attempts to keep news out entirely, by refusing
   to support an official gateway, typically result in large numbers of
   unofficial partial gateways appearing over time. Such a
   configuration is much more difficult to troubleshoot.

   A somewhat related problem is the possibility of proprietary material
   being disclosed unintentionally by a poster who does not realize how
   far his words will propagate, either from sheer misunderstanding or
   because of errors made (by human or software) in followup
   preparation. There is little that can be done about this except
   education.

11.2. Attacks

   Although the limitations of the medium restrict what can be done to
   attack a host via news, some possibilities exist, most of them
   problems news shares with mail.

   If reading agents are careless about transmitting non-printable
   characters to output devices, malicious posters may post articles
   containing control sequences ("letterbombs") meant to have various
   destructive effects on output devices. Possible effects depend on
   the device, but they can include hardware damage (e.g., by repeated
   writing of values into configuration memories that can tolerate only
   a limited number of write cycles) and security violation (e.g., by
   reprogramming function keys potentially used by privileged readers).

   A more sophisticated variation on the letterbomb is inclusion of
   "Trojan horses" in programs. Obviously, readers must be cautious
   about using software found in news, but more subtly, reading agents
   must also exercise care. MIME messages can include material that is
   executable in some sense, such as PostScript documents (which are
   programs!), and letterbombs may be introduced into such material.

   Given the presence of finite resources and other software
   limitations, some degree of system disruption can be achieved by
   posting otherwise-innocent material in great volume, either in single
   huge articles (see Section 4.6) or in a stream of modest-sized
   articles. (Some would say that the steady growth of Usenet volume
   constitutes a subtle and unintentional attack of the latter type;
   certainly it can have disruptive effects if administrators are
   inattentive.) Systems need some ability to cope with surges, because
   single huge articles occur occasionally as the result of software
   error, innocent misunderstanding, or deliberate malice; and downtime
   at upstream hosts can cause droughts, followed by floods, of
   legitimate articles. (There is also a certain amount of normal
   variation; for example, Usenet traffic is noticeably lighter on
   weekends and during Christmas holidays, and rises noticeably at the
   start of the school term of North American universities.) However, a
   site that normally receives little traffic may be quite vulnerable to
   "swamping" attack if its software is insufficiently careful.

   In general, careless implementation may open doors that are not
   intrinsic to news. In particular, implementation of control messages
   (see Sections 6.6 and 7) and unbatchers (see Sections 8.1 and 8.2)
   via a command interpreter requires substantial precautions to ensure
   that only the intended capabilities are available. Care must also be
   taken that article-supplied text is not fed to programs that have
   escapes to command interpreters.

   Finally, there is considerable potential for malice in the sendsys,
   version, and whogets control messages. They are not harmful to the
   hosts receiving them as news, but they can be used to enlist those
   hosts (by the thousands) as unwitting allies in a mail-swamping
   attack on a victim who may not even receive news. The precautions
   discussed in Section 7.5 can reduce the potential for such attacks
   considerably, but the hazard cannot be eliminated as long as these
   control messages exist.

11.3. Anarchy

   The highly distributed nature of news propagation, and the lack of
   adequate authentication protocols (especially for use over the less-
   interactive transport mechanisms such as UUCP), make article forgery
   relatively straightforward. It may be possible to at least track a
   forgery to its source, once it is recognized as such, but clever
   forgers can make even that relatively difficult. The assumption that
   forgeries will be recognized as such is also not to be taken for
   granted; readers are notoriously prone to blindly assuming
   authenticity. If a forged article's initial path list includes the
   relayer name of the supposed poster's host, the article will never be
   sent to that host, and the alleged author may learn about the forgery
   secondhand or not at all.

   A particularly noxious form of forgery is the forged "cancel" control
   message. Notably, it is relatively straightforward to write software
   that will automatically send out a (forged) cancel message for any
   article meeting some criterion, e.g., written by a specific author.
   The authentication problems discussed in Section 7.1 make it
   difficult to solve this without crippling cancel's important
   functionality.

   A related problem is the possibility of disagreements over newsgroup
   creation, on networks where such things are not decided by central
   authorities. There have been cases of "rmgroup wars", where one
   poster persistently sends out newgroup messages to create a newsgroup
   and another, equally persistently, sends out rmgroup messages asking
   that it be removed. This is not particularly damaging, if relayers
   are configured to be cautious, but it can cause serious confusion
   among innocent third parties who just want to know whether or not
   they can use the newsgroup for communication.

11.4. Liability

   News shares the legal uncertainty surrounding other forms of
   electronic communication: what rules apply to this new medium of
   information exchange? News is a particularly problematic case

because it is a broadcast medium rather than a point-to-point one
like mail, and analogies to older forms of communication are
particularly weak.

Are news-carrying hosts common carriers, like the phone companies,
providing communications paths without having either authority over
or responsibility for content? Or are they publishers, responsible
for the content regardless of whether they are aware of it or not?
Or something in between? Such questions are particularly significant
when the content is technically criminal, e.g., some types of
sexually oriented material in some jurisdictions, in which case
ignorance of its presence may not be an adequate defense.

Even in milder situations such as libel or copyright violation, the
responsibilities of the poster, his host, and other hosts carrying
the traffic are unclear. Note, in particular, the problems arising
when the article is a forgery, or when the alleged author claims it
is a forgery but cannot prove this.

12. References

[ISO/IEC9899] "Information technology - Programming Language C",
               ISO/IEC 9899:1990 {more recently 9899:1999}, 1990.

[Metamail]   Borenstein, N.,
               <http://ftp.funet.fi/pub/unix/mail/metamail/ANNOUNCE>,
               February 1994.

[RFC821]   Postel, J., "Simple Mail Transfer Protocol", STD 10,
               RFC 821, August 1982.

[RFC822]   Crocker, D., "STANDARD FOR THE FORMAT OF ARPA INTERNET
               TEXT MESSAGES", STD 11, RFC 822, August 1982.

[RFC850]   Horton, M., "Standard for interchange of Usenet
               messages", RFC 850, June 1983.

[RFC977]   Kantor, B. and P. Lapsley, "Network News Transfer
               Protocol - A Proposed Standard for the Stream-Based
               Transmission of News", RFC 977, February 1986.

[RFC1036]   Horton, M. and R. Adams, "Standard for interchange of
               USENET Messages", RFC 1036, December 1987.

[RFC1123]   Braden, R., Ed., "Requirements for Internet Hosts -
               Application and Support", STD 3, RFC 1123,
               October 1989.

   [RFC1341]   Borenstein, N. and N. Freed, "MIME (Multipurpose
                    Internet Mail Extensions): Mechanisms for Specifying
                    and Describing the Format of Internet Message Bodies",
                    RFC 1341, June 1992.

   [RFC1342]   Moore, K., "Representation of Non-ASCII Text in
                    Internet Message Headers", RFC 1342, June 1992.

   [RFC1345]   Simonsen, K., "Character Mnemonics and Character
                    Sets", RFC 1345, June 1992.

   [RFC1413]   St. Johns, M., "Identification Protocol", RFC 1413,
                    February 1993.

   [RFC1456]   Vietnamese Standardization Working Group, "Conventions
                    for Encoding the Vietnamese Language", RFC 1456,
                    May 1993.

   [RFC1544]   Rose, M., "The Content-MD5 Header Field", RFC 1544,
                    November 1993.

   [RFC1896]   Resnick, P. and A. Walker, "The text/enriched MIME
                    Content-type", RFC 1896, February 1996.

   [RFC2045]   Freed, N. and N. Borenstein, "Multipurpose Internet
                    Mail Extensions (MIME) Part One: Format of Internet
                    Message Bodies", RFC 2045, November 1996.

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

   [RFC2047]   Moore, K., "MIME (Multipurpose Internet Mail
                    Extensions) Part Three: Message Header Extensions for
                    Non-ASCII Text", RFC 2047, November 1996.

   [RFC2049]   Freed, N. and N. Borenstein, "Multipurpose Internet
                    Mail Extensions (MIME) Part Five: Conformance Criteria
                    and Examples", RFC 2049, November 1996.

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

   [RFC3977]   Feather, C., "Network News Transfer Protocol (NNTP)",
                    RFC 3977, October 2006.

   [RFC5322]   Resnick, P., Ed., "Internet Message Format", RFC 5322,
                   October 2008.

   [RFC5536]   Murchison, K., Ed., Lindsey, C., and D. Kohn, "Netnews
                   Article Format", RFC 5536, November 2009.

   [RFC5537]   Allbery, R., Ed., and C. Lindsey, "Netnews
                   Architecture and Protocols", RFC 5537, November 2009.

   [Sanderson]   David Sanderson, Smileys, O'Reilly & Associates Ltd.,
                   1993.

   [UUCP]   Tim O'Reilly and Grace Todino, Managing UUCP and
                   Usenet, O'Reilly & Associates Ltd., January 1992.

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

Appendix A. Archaeological Notes

A.1. "A News" Article Format

   The obsolete "A News" article format consisted of exactly five lines
   of header information, followed by the body. For example:

      Aeagle.642
      news.misc
      cbosgd!mhuxj!mhuxt!eagle!jerry
      Fri Nov 19 16:14:55 1982
      Usenet Etiquette - Please Read
      body
      body
      body

   The first line consisted of an "A" followed by an article ID
   (analogous to a message ID and used for similar purposes). The
   second line was the list of newsgroups. The third line was the path.
   The fourth was the date, in the format above (all fields fixed
   width), resembling an Internet date but not quite the same. The
   fifth was the subject.

   This format is documented for archaeological purposes only. Do not
   generate articles in this format.

A.2. Early "B News" Article Format

   This obsolete pseudo-Internet article format, used briefly during the
   transition between the A News format and the modern format, followed
   the general outline of a MAIL message but with some non-standard
   headers. For example:

      From: cbosgd!mhuxj!mhuxt!eagle!jerry (Jerry Schwarz)
      Newsgroups: news.misc
      Title: Usenet Etiquette -- Please Read
      Article-I.D.: eagle.642
      Posted: Fri Nov 19 16:14:55 1982
      Received: Fri Nov 19 16:59:30 1982
      Expires: Mon Jan 1 00:00:00 1990

      body
      body
      body

   The From header contained the information now found in the Path
   header, plus possibly the full name now typically found in the From
   header. The Title header contained what is now the Subject content.

The Posted header contained what is now the Date content. The
Article-I.D. header contained an article ID, analogous to a message
ID and used for similar purposes. The Newsgroups and Expires headers
were approximately as they are now. The Received header contained
the date when the latest relayer to process the article first saw it.
All dates were in the above format, with all fields fixed width,
resembling an Internet date but not quite the same.

This format is documented for archaeological purposes only. Do not
generate articles in this format.

A.3. Obsolete Headers

Early versions of news software following the modern format sometimes
generated headers like the following:

    Relay-Version: version B 2.10 2/13/83; site cbosgd.UUCP
    Posting-Version: version B 2.10 2/13/83; site eagle.UUCP
    Date-Received: Friday, 19-Nov-82 16:59:30 EST

Relay-Version contained version information about the relayer that
last processed the article. Posting-Version contained version
information about the posting agent that posted the article. Date-
Received contained the date when the last relayer to process the
article first saw it (in a slightly nonstandard format).

These headers are documented for archaeological purposes only. Do
not generate articles using them.

A.4. Obsolete Control Messages

There once was a senduuname control message, resembling sendsys but
requesting transmission of the list of hosts to which the receiving
host had UUCP connections. This rapidly ceased to be of much use,
and many organizations consider information about their internal
connectivity to be confidential.

Historically, a checkgroups body consisting of one or two lines, the
first of the form "-n newsgroup", caused checkgroups to apply to only
that single newsgroup. This form is documented for archaeological
purposes only; do not use it.

Historically, an article posted to a newsgroup whose name had exactly
three components of which the third was "ctl" signified that article
was to be taken as a control message. The Subject header specified
the actions in the same way the Control header does now. This form
is documented for archaeological purposes only; do not use it; do not
implement it.

Appendix B. A Quick Tour of MIME

   (The editor wishes to thank Luc Rooijakkers; most of this appendix is
   a lightly edited version of a summary he kindly supplied.)

   MIME (Multipurpose Internet Mail Extensions) is an upward-compatible
   set of extensions to [RFC822], currently documented in [RFC2045],
   [RFC2046], and [RFC2047]. This appendix summarizes these documents.
   See the MIME RFCs for more information; they are very readable.

      UNRESOLVED ISSUE: These RFC numbers (here and elsewhere in this
      Draft) need updating when the new MIME RFCs come out {now
      resolved!}.

   MIME defines the following new headers:

      MIME-Version
      Content-Type
      Content-Transfer-Encoding
      Content-ID
      Content-Description

   The MIME-Version header is mandatory for all messages conforming to
   the MIME specification and carries the version number of the MIME
   specification. Example:

      MIME-Version: 1.0

   The Content-Type header indicates the content type of the message.
   Content types are split into a top-level type and a subtype,
   separated by a slash. Auxiliary information can also be supplied,
   using an attribute-value notation. Example:

      Content-Type: text/plain; charset=us-ascii

   (In the absence of a Content-Type header this is in fact the default
   content type.)

   Important type/subtype combinations are:

   text/plain   Plain text, possibly in a non-ASCII character
                         set.

   text/enriched   A very simple wordprocessor-like language
                         supporting character attributes (e.g.,
                         underlining), justification control, and
                         multiple character sets. (This proposal has

                         gone through several iterations and has
                         recently split off from the main MIME RFCs
                         into a separate document [RFC1896].)

   message/rfc822   A mail message conforming to a slightly
                         relaxed version of [RFC822].

   message/partial   Part of a message (supporting the transparent
                         splitting and joining of messages when they
                         are too large to be handled by some transport
                         agent).

   message/external-body A message whose body is external. Possible
                         access methods include via mail, FTP, local
                         file, etc.

   multipart/mixed   A message whose body consists of multiple
                         parts, possibly of different types, intended
                         to be viewed in serial order. Each part
                         looks like an [RFC822] message, consisting of
                         headers and a body. Most of the [RFC822]
                         headers have no defined semantics for body
                         parts.

   multipart/parallel   Likewise, except that the parts are intended
                         to be viewed in parallel (on user agents that
                         support it).

   multipart/alternative Likewise, except that the parts are intended
                         to be semantically equivalent such that the
                         part that best matches the capabilities of
                         the environment should be displayed. For
                         example, a message may include plain-text,
                         enriched-text, and postscript versions of
                         some document.

   multipart/digest   A variant of multipart/mixed especially
                         intended for message digests (the default
                         type of the parts is message/rfc822 instead
                         of text/plain, saving on the number of
                         headers for the parts).

   application/postscript A PostScript document. (PostScript is a
                         trademark of Adobe.)

   Other top-level types exist for still images, audio, and video
   samples.

   Some of the above types require the ability to transport binary data.
   Since the existing message systems usually do not support this, MIME
   provides a Content-Transfer-Encoding header to indicate the kind of
   encoding used.   The possible encodings are:

   7bit   No encoding; the data consists of short (less than
                      1000 characters) lines of 7-bit ASCII data,
                      delimited by EOL sequences. This is the default
                      encoding.

   8bit   Like 7bit, except that bytes with the high-order
                      bit set may be present. Many transmission paths
                      are incapable of carrying messages that use this
                      encoding.

   binary   No encoding; any sequence of bytes may be present.
                      Many transmission paths are incapable of carrying
                      messages that use this encoding.

   base64   The data is encoded by representing every group of
                      3 bytes as 4 characters from the alphabet
                      "A-Za-z0-9+/", which was chosen for its high
                      robustness through mail gateways (the alphabet used
                      by uuencode does not survive ASCII-EBCDIC-ASCII
                      translations). In the final group of 4 characters,
                      "=" is used for those characters not representing
                      data bytes. Line length is limited, and EOLs in
                      the encoded form are ignored.

   quoted-printable   Any byte can be represented by a three-character
                      "=XX" sequence where the X's are uppercase
                      hexadecimal digits. Bytes representing printable
                      7-bit US-ASCII characters except "=" may be
                      represented literally. Tabs and blanks may be
                      represented literally if not at the end of a line.
                      Line length is limited, and an EOL preceded by "="
                      was inserted for this purpose and is not present in
                      the original.

   The base64 and quoted-printable encodings are applied to data in
   Internet canonical form, which means that any EOL encoded as anything
   but EOL must be an Internet canonical EOL: CR followed by LF.

   The Content-Description header allows further description of a body
   part, analogous to the use of Subject for messages.

Finally, the Content-ID header can be used to assign an
identification to body parts, analogous to the assignment of
identifications to messages by Message-ID.

Note that most of these headers are structured header fields, as
defined in [RFC822]. Consequently, comments are allowed in their
values. The following is a legal MIME header:

```
Content-Type: (a comment) text (yeah)   /
        plain    (and now some params:) ; charset= (guess what)
    iso-8859-1 (we don't have iso-10646 yet, pity)
```

NOTE: Although the MIME specification was developed for mail,
there is nothing precluding its use for news as well. While it
might simplify implementation to restrict the MIME headers
somewhat, in the same way that other news headers (e.g., From) are
restricted subsets of the [RFC822] originals, this would add yet
another divergence between two formats that ought to be as
compatible as possible. In the case of the MIME headers, there is
no body of existing code posing compatibility concerns. A full-
featured MIME reading agent needs a full [RFC822] parser anyway,
to properly handle body parts of types like message/rfc822, so
there is little gain from restricting MIME headers. Adopting the
MIME specification unchanged seems best. However, article-level
MIME headers must still comply with the overall news header syntax
given in Section 4, so that news software that is NOT interested
in MIME need not contain a full [RFC822] parser.

"MIME (Multipurpose Internet Mail Extensions) Part Three: Message
Header Extensions for Non-ASCII Text" [RFC2047] addresses the problem
of non-ASCII characters in headers. An example of a header using the
[RFC2047] mechanism is

```
From: =?ISO-8859-1?Q?Andr=E9_?= Pirard <PIRARD@vm1.ulg.ac.be>
```

Such encodings are allowed in selected headers, subject to the
restrictions listed in [RFC2047].

The MIME effort has also produced an RFC defining a Content-MD5
header [RFC1544] containing an MD5-based "checksum" of the contents
of an article or body part, giving high confidence of detecting
accidental modifications to the contents.

The "metamail" software package [Metamail] helps provide MIME support
with minimal changes to mailers and may also be relevant to news
reading agents.

The PEM (Privacy Enhanced Mail) effort is pursuing analogous
facilities to offer stronger guarantees against malicious
modifications, unauthorized eavesdropping, and forgery. This work
too may be applicable to news, once it is reconciled with MIME (by
efforts now underway).

Appendix C. Summary of Changes Since RFC 1036

   This Draft is much longer than [RFC1036], so there is obviously much
   change in content. Much of this is just increased precision and
   rigor. Noteworthy changes and additions include:

      + restrictions on article bodies (Section 4.3)

      + all references to MIME facilities

      + size limits on articles

      + precise specification of Date-content syntax

      + message IDs must never be re-used, ever

      + "!" is the only Path delimiter

      + multiple moderators in the Approved header

      + rules on References trimming, and the _-_ mechanism

      + generalization of the Xref rules

      + multiple message IDs in Cancel and Supersedes

      + Also-Control

      + See-Also

      + Article-Names

      + Article-Updates

      + more precise rules for cancellation

      + cancellation authorization based on From, not Sender

      + "unmoderated" and descriptors in newgroup messages

      + restrictive rules on handling of sendsys and version messages

      + the whogets control message

      + precise specification of checkgroups messages

      + compression type preferably specified out-of-band

        + rules for encapsulating news in MIME mail

        + tighter specification of relayer functioning (Section 9.1)

        + the "newsmaster" contact address

        + rules for gatewaying (Section 10)

        + discussion of security issues (Section 11)

Appendix D. Summary of Completely New Features

   Most of this Draft merely documents existing practice, preferred
   versions thereof, or straightforward generalizations of it, but there
   are a few outright inventions. These are:

      + the _-_ mechanism for References trimming

      + Also-Control

      + See-Also

      + Article-Names

      + Article-Updates

      + the whogets control message

Appendix E. Summary of Differences from RFCs 822 and 1123

   The following are noteworthy differences between this Draft's
   articles and MAIL messages:

      + generally less-permissive header syntax

      + notably, limited From syntax

      + MAIL header comments allowed in only a few contexts

      + slightly more restricted message-ID syntax

      + several more mandatory headers

      + duplicate headers forbidden

      + References/See-Also versus In-Reply-To/References (Section 6.5)

      + case sensitivity in some contexts

      + point-to-point headers, e.g., To and Cc, forbidden (Section 6)

      + several new headers

Author's Address

   Henry Spencer
   SP Systems
   Box 280 Stn. A
   Toronto, Ontario M5W1B2
   Canada

   EMail: henry@zoo.utoronto.ca