

Network Working Group
Internet-Draft
Intended status: Best Current
Practice
Expires: May 11, 2008

J. Peterson
NeuStar
November 8, 2007

Normative Language and References
draft-peterson-informational-normativity-01

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on May 11, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document explores the use of normative language and references with a focus on reducing unnecessary normative references in IETF specifications.

Table of Contents

1.	Introduction	3
2.	What is 'normative'?	4
2.1.	Normative Pseudo-Keywords	5
3.	Normative references	6
3.1.	Examples and Ambiguities	8
4.	Normative Language off the (Standards) Track	9
4.1.	Informational Publication of Protocols	11
5.	IANA Considerations	12
6.	Security Considerations	12
7.	Acknowledgements	12
8.	Informational References	12
	Author's Address	13
	Intellectual Property and Copyright Statements	14

1. Introduction

[RFC2119](#) [1] provides a set of familiar directives to readers of IETF specifications, specifically the imperatives: "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL". This set of normative keywords, as they shall be known in this document, consists of a number of grammatical variations which ultimately describe three degrees of normative compliance: mandatory, recommended, and optional. The first two degrees may be used in either prescriptive or proscriptive contexts (e.g. "MUST" and "MUST NOT", "SHOULD" and "SHOULD NOT"), while for the third only prescriptive statements are permitted (there is a "MAY" but no "MAY NOT", something can be "OPTIONAL", but not "NOT OPTIONAL").

The use of normative keywords is one of the defining characteristics of IETF specifications. Normative keywords remain an indispensable tool for evaluating interoperability as specifications advance on the standards track, and moreover for pruning unimplemented features as protocols mature through deployment and usage. The application of normative keywords to these functions is predicated largely on the text of [RFC2026](#) [2].

[RFC2119](#) does not, however, contain the word 'normative', and nor does [RFC2026](#). The idea that a statement or reference can be 'normative' or 'informational' (let alone the requirement that the References section of an Internet-Draft be divided between the two) dates from a much later time, as does the term 'normative language'. The conditions that render a particular reference or statement 'normative' have never been specified; although there is a good understanding in the community of the common distinctions, practices can be very erratic in corner-cases.

An example of the resulting confusion is the use of normative keywords in requirements documents, which here are to be understood as Informational documents that apply constraints to future protocol specification work, as opposed to actual implementation work. Authors of standards-track protocol specifications intended to satisfy these requirements sometimes include such requirements documents in their "Normative References" sections, precisely because they are referring to statements containing normative keywords. This sort of downward reference is of course formally prohibited in [RFC2026](#), and thus must be corrected, but the whole situation arises needlessly. In the absence of some clarification, similar misconceptions will continue to arise.

This document therefore attempts to provide a stronger account of the classification designated by the term 'normative', and to detail

various conditions under which a reference need not be considered 'normative'. [RFC3967](#) [3] wisely notes that normative references resist definition, and the text in this memo does not claim to have articulated all of the associated subtleties and implications; however, in order to reduce overuse and misapplication of normative terms, a more substantive account of the term 'normative' appears here than has in [RFC3967](#) or previous works.

[RFC3967](#) and more recently [RFC4897](#) [4] have revised the guidance of [RFC2026](#) regarding the advancement of standards-track documents which refer to documents at a lower maturity level (or those not on the standards track at all). The present document is entirely compatible with the useful amendments introduced in those documents.

2. What is 'normative'?

Normative keywords are 'normative' in so far as they establish the norms that are the foundation of interoperability. Implementations of a particular specification can be considered to be a sort of community, and that community has practices that are mandatory and prohibited, recommended and counterrecommended, or simply optional - hence, they are norms.

'Normative language' or 'normative statements' are, broadly, passages of text in IETF documents which contain normative keywords that direct implementers, with varying degrees of stringency, to incorporate particular features in order to foster interoperability.

Normative language, as originally described in [RFC2119](#), is tooled solely to describe how implementations are intended to behave. As [RFC2119 Section 6](#) states, in reference to normative keywords:

In particular, they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)

Ironically, this normative statement is not internally consistent. It urges authors of specifications to use normative keywords only in reference to matters of implementation, but in order to amplify its point from mere urging to absolute dictum, it relies on a normative keyword. Therein lies the source of the confusion. Normative keywords are used commonly, but incorrectly, in precisely this fashion: for emphasis, in passages of descriptive text that in no way could be construed to address implementations.

When authors of subsequent specifications see such normative keywords used in an purely descriptive passage in an RFC, they may assume that

the document containing those normative keywords should be referenced normatively. This can cause an unnecessary apparent need for a downward reference.

Any statement that is non-normative is by definition purely informational. Informational or descriptive statements play a large role in IETF documents, providing context that is useful to implementers or authors of future specifications but which does not, strictly speaking, detail implementation behavior that will subsequently be measured for compliance or interoperability.

2.1. Normative Pseudo-Keywords

The use of the normative keywords has never been compulsory in the IETF. Numerous documents, both before and after the publication of [RFC2119](#), describe protocol behavior without relying on normative keywords. Normative keywords are a way of explicitly designating the degree of compliance associated with a behavioral prescription for implementations. It is equally possible, and sometimes more readable, to write precise text which is semantically identical to passages employing normative keywords. Constructions with that property are herein said to contain "normative pseudo-keywords": prescriptive behavioral keywords that could potentially be paraphrased with normative keywords.

In order for such a piece of text to qualify as normative, it must contain pseudo-keyword text which designates the degree of compliance levied on the behavior (i.e. terms that recognizably signify 'mandatory', 'recommended', or 'optional'). Furthermore, following the definition of the [RFC2119](#) keywords, these statements must pertain to implementation behavior. Statements can be tested by paraphrase; for example:

Messages are challenged using Digest authentication [[RFC2617](#)].

The passage above uses a normative pseudo-keyword (in this case, the verb 'to be') which can be paraphrased using a normative keyword as follows:

Messages **MUST** be challenged using Digest authentication [[RFC2617](#)].

However, it is equally possible to generate behavioral statements which do not satisfy this test, for example:

Messages might be challenged using Digest authentication [[RFC2617](#)] when a weaker form of authentication would be inappropriate.

Is this a recommendation, a counterrecommendation, or what? Without

sufficient supporting text to suggest when weaker forms of authentication might be appropriate, one could even read this as a mandate for the use of Digest. In the absence of a recognizable degree of compliance, this statement cannot be normative. In the interests of promoting interoperability it would ideally be replaced with a clearer sentence containing a normative keyword.

The use of pseudo-keywords to form normative statements is never completely unambiguous, and is therefore discouraged. It is not at all uncommon today for reviewers, at the IESG review phase or earlier, to provide blocking comments on Internet-Drafts of the form "is this 'must' supposed to be a 'MUST'?" The use of normative keywords, since they have accepted definitions within the community, are the most precise way of designating a degree of compliance. That much said, authors of specifications must also recognize that normative keywords are not a panacea, and that gibberish can be written just as easily with uppercase words as with lowercase. Some examples of this are given in [Section 3.1](#).

However, for the purposes of normative references and the evaluation of features and options, passages containing normative pseudo-keywords are treated as equivalent to passages containing normative keywords. Without this allowance, it would be impossible for new documents to refer normatively to many, if not most, existing RFCs. It is the semantics, not the syntax, of statements that is crucial to determining their normative status.

3. Normative references

This document follows the terminology of [RFC4897](#) for a 'source document' (a document in which the reference to another document is embedded) and a 'target document' (the document so referenced). It furthermore defines the 'referencing statement' as the statement in the source document which invokes the reference to the target document. A 'normative statement' is understood to be a statement which uses normative keywords (or pseudo-keywords) to associate a specific degree of normative compliance with a particular implementation behavior.

For the benefit of specification authors, the following is a list of conditions in which a reference to a document need not be normative:

1. if the source document is itself Informational (not a standards-track document, BCP or an Experimental document).
2. if the referencing statement is not a normative statement; i.e., does not prescribe some degree of normative compliance with the target document.

3. if the target document, and in particular any subset scope designated by the referencing statement (a section, or what have you), contains no normative statements.

If any of the above conditions apply, then the reference in question need not be normative. One additional possible condition, that the target document have an equal or greater standards maturity level to the source document, is not strictly speaking a necessary condition for a normative reference; however, normative references made when this condition prevails must successfully invoke the downref exception procedures defined in [RFC3976](#) in order to advance on the IETF standards-track.

One source of ambiguity in determining whether or not a reference is normative is the status of Best Current Practices (BCPs, as defined in [Section 5 of RFC2026](#)). The BCP designation is a bit of a catch-all in the IETF standards process. A BCP can prescribe practices varying from operations, which are indeed critical to the interoperability of the Internet, to IETF process, which is of a non-technical nature. As such, it is entirely appropriate, in some cases, to provide a normative reference to a BCP, and for a BCP to contain normative keywords.

In the case of IETF process BCPs, it is less clear that they should be understood normatively, and moreover less clear that it is appropriate for process documents to employ normative keywords. A precedent for using normative language in those documents was set by [RFC2418](#) [5] (see especially the last paragraph of [Section 1](#); this is also discussed further below in [Section 4](#)). When process documents do employ normative keywords, as [RFC2119](#) does in the citation above, it is almost always inconsistent with the definition of those terms in [RFC2119](#) and their intended use in [RFC2026](#). This in turn further contributes to the perception that it is appropriate for non-technical documents in general (such as requirements documents) to employ normative keywords. Unfortunately, this appropriateness of using normative language in BCPs must be assessed on a case-by-case basis.

A good rule of thumb, and a corollary of the guidance in [RFC3967](#), for whether a reference should be normative or not is the following: if the target document were lost, such that an implementer could not read, access or implement it, would implementations of the source document still interoperate for the functionality described in the statement? If not, then the reference must be normative.

3.1. Examples and Ambiguities

Even so strict an account of a normative reference cannot be entirely free from ambiguities that are grounded in the degrees of compliance themselves, especially regarding conditionals and conditional prohibitions (e.g. "SHOULD NOT"). Ambiguities also arise when degrees of compliance are associated with the use of a feature rather than its implementation. Consider the following referencing statement:

Implementations SHOULD NOT use 3DES [[1](#)].

If this had said that implementations "MUST NOT" use 3DES, it would be unambiguous that implementations were not required to implement 3DES, and thus the reference to 3DES would not be normative. You would never need to implement 3DES in order to be able to interoperate with another implementation of this specification. But what about the conditional prohibition "SHOULD NOT use 3DES"? Is it necessary to support a referenced specification in order to obey a conditional prohibition against using it? This is difficult to answer on its own, but in fact, that referencing statement entails this one:

Implementations MAY use 3DES [[1](#)].

Should 3DES be considered a Normative or Informational reference if it is OPTIONAL? Normative, clearly, by the litmus test. Both this statement and the one above allow, but do not require, implementations to support 3DES. And because this statement is entailed by the conditional prohibition above, both of these are making a normative reference to 3DES. This may appear to be something of a paradox, since it seems intuitively that counterrecommended behavior shouldn't require a normative reference, but optional behavior should - one just needs to bear in mind that everything counterrecommended is necessarily optional.

Given all that, better specmanship would make the exact implementation needs more clear. A closely analogous but infinitely superior phrasing would be:

While 3DES [[1](#)] is unsuitable for use in most environments, for backwards compatibility reasons implementations MUST support it.

In this case it is clear that support for the 3DES protocol is mandatory, even though its use is discouraged (non-normatively).

There are also classes of statements that employ normative keywords,

and contains references, but do so in a way that does not actual form a normative referencing statement. Consider this:

The application MUST implement an underlying transport which can provide integrity and confidentiality properties, for example TLS [3].

The citation of TLS above is merely exemplary; the referencing statement does not actually require application developers to implement TLS. Rather, it requires that any underlying transport that is implemented have certain properties, though not terribly specific ones. As such, this reference cannot be considered normative - it suggests no specific underlying transport to the implementation community. Precisely for this reason, it is an example of weak specmanship. Statements of this general form often seem attractive, however, to specification authors who hope to reference work-in-progress or Informational documents. The fix for this sort of specmanship is not to require TLS to appear in the Normative references section of the document (it shouldn't on the strength of this statement), but rather to encourage the authors to make a stronger referencing statement, one actually conducive to establishing implementation norms.

Another similar example is the use of disjunctive references like the following:

Implementations MUST provide this capability by implementing either PGP [4] or S/MIME [6].

Is this a normative reference to both PGP and S/MIME, or neither? It seems to read that implementers would only need to implement one in order to be compliant, so perhaps only one of them is actually a normative reference... but if so, which one? Ultimately, this is another instance of weak specmanship, in which the statement itself needs to be amended before its clear what the normative requirements are.

These examples are hardly exhaustive, but they at least serve to motivate the need for a strict understanding of 'normative'.

4. Normative Language off the (Standards) Track

This section examines the use of normative keywords in Informational documents which are not protocol specifications. Some Informational RFCs are in fact protocol specifications; this will be the subject of [Section 4.1](#).

Despite the text of [RFC2119](#), it is commonplace for normative keywords to appear in Informational requirements documents today, in statements that are intended to constrain the authoring of future specifications. The laudable intent of requirements documents is of course to establish consensus on the needs of the implementation community prior to the evaluation of candidate protocol specifications that might satisfy these needs. The requirements document becomes a measuring stick of the 'compliance' of a candidate protocol.

Undoubtedly some confusion arises from an accident of the language in [RFC2119](#). The Abstract of 2119 says that the normative keywords are "are used to signify the requirements in the specification", which could be read to suggest that Informational requirements that will be used to constrain further protocol specifications should use normative keywords. In fact, that interpretation clearly contradicts the previously-cited dictum that normative keywords are to be used only when required for "interoperation or to limit [implementation] behavior."

Text at the end of [Section 1 of RFC2418](#) furthermore suggests that normative keywords might be applied by analogy to non-protocol operations, in that case IETF process, in order to "reduce the chance for confusion about the process", but it isn't clear how such an analogy would operate. Were we to grant hypothetically that normative keywords apply to requirements by analogy, the interpretation of normative keywords in this context would remain problematic. How are we to understand the "SHOULD" keyword for protocol requirements, as opposed to protocols. What does it mean for a protocol that satisfies a given set of protocol requirements to be merely "conditionally compliant"?

Along these lines, it might seem compelling to imagine that the selection of two protocols X and Y, which were invented to satisfy a set of requirements A, might be decided by a single "SHOULD" statement specified in A which is support by X but not Y. But of course, if that "SHOULD" in A were instead a "MUST", the same selection would be made. The true utility of a "SHOULD" emerges when we instead consider two protocol implementations, X and Y, which have been implemented to specification A and are attempting to interoperate. In this case, if Y fails to implement a "MUST", a very different result can occur than if Y fails to implement a "SHOULD". In short, the normative keywords are designed to encourage cooperation, not decide competition. Using them in the latter context is a strained analogy, and the resulting strain rests on the IETF's standards process.

It is moreover critical to appreciate that the use of normative

keywords is tied to the functions of 2026: that is, the pruning of unused features of a protocol specification. From the guidance in 4.1.2 (where we understand 'features' to be at the mandatory degree of compliance, and 'options' to be at the recommended or optional degrees of compliance):

The requirement for at least two independent and interoperable implementations applies to all of the options and features of the specification. In cases in which one or more options or features have not been demonstrated in at least two interoperable implementations, the specification may advance to the Draft Standard level only if those options or features are removed.

Normative keywords exist to ensure interoperability; by contrast, a requirements document will never be interoperable with anything.

More rarely, normative keywords appear in Informational frameworks that describe high-level or abstract architectures. In this context they are primarily used for rhetorical emphasis. This practice can still lead authors of future specifications to improper referencing.

Finally, it is also possible for an Informational document to redefine normative keywords in lieu of any reference to [RFC2119](#). This practice only adds further misery to the confusion surrounding the use of normative keywords, and should be avoided. If there is a genuine need for terminology to characterize adherence to a set of requirements in the context of specification authoring, those terms should be clearly defined and explicitly distinguished, semantically and syntactically, from the [RFC2119](#) normative keywords. A similar direction should be taken regarding the use of normative keywords in process statements. Further consideration is left as a possible subject for future study.

4.1. Informational Publication of Protocols

There are a variety of circumstances in which protocol specifications are published as Informational RFCs. Sometimes authors request Informational publication of protocol specifications which were rejected as candidates in a working group process in order to preserve an historical record. Parties who do not participate directly in the IETF may similarly request publication of their designs as an Informational RFC. New ciphersuites designed outside the IETF are typically documented, in strict procedural language, within Informational RFCs as a convenient reference for protocol designers; these latter are frequently a target of legitimate downward references (see [RFC3967](#)). Some exceptional IETF procedures, for example MIBs or the SIP change ([RFC3427](#) [6]) process, may stipulate a lower bar of review and Informational publication for

certain protocol work.

These Informational documents often contain normative keywords, as their authors aspire to specify something that will yield interoperable implementations. One need not anticipate, or even understand, the eventual intended status of a specification in order to invoke [RFC2119](#) and use the normative keywords therein. The distinctions between such Informational documents and standards-track documents lie more in the implications about the level of review and community consensus which the standards-track entails than in any consideration about the use of normative keywords.

Because such documents exist, it is not reasonable to bar Informational specifications from containing [RFC2119](#) normative keywords. Indeed, the downref exception procedures of [RFC3967](#) exist so that it is possible to refer to such documents, under the proper conditions and with the required oversight.

Instead, we should prevent the casual or inappropriate use of normative keywords that to refer to matters other the proper implementation of protocols.

5. IANA Considerations

This document contains no considerations for the IANA.

6. Security Considerations

This is a IETF process document which does not impact the security of IETF protocols.

7. Acknowledgements

Harald Alvestrand, Brian Carpenter, Scott Bradner and Jonathan Rosenberg have provided valuable input to this document.

8. Informational References

- [1] Bradner, S., "Key words for use in RFCs to indicate requirement levels", [RFC 2119](#), March 1997.
- [2] Bradner, S., "The Internet Standards Process -- Revision 3", [RFC 2026](#), October 1996.

- [3] Bush, R. and T. Narten, "Clarifying when Standards Track Documents may Refer Normatively to Documents at a Lower Level", [RFC 3967](#), December 2004.
- [4] Klensin, J. and S. Hartman, "Handling Normative References to Standards-Track Documents", [RFC 4897](#), June 2007.
- [5] Bradner, S., "IETF Working Group Guidelines and Procedures", [RFC 2418](#), September 1998.
- [6] Mankin, A., Bradner, S., Mahy, R., Willis, D., Ott, J., and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)", [RFC 3427](#), December 2002.

Author's Address

Jon Peterson
NeuStar, Inc.
1800 Sutter St
Suite 570
Concord, CA 94520
USA

Phone: +1 925/363-8720
Email: jon.peterson@neustar.biz
URI: <http://www.neustar.biz/>

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

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

Intellectual Property

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

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

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

Acknowledgment

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

