Internet Draft
draft-ietf-usefor-message-id-01.txt
Category-to-be: Informational

Matt Curtin
The Ohio State University
Jamie Zawinski
Netscape Communications

July 1998
Expires: Six Months from above date

Recommendations for generating Message IDs

Status of this Memo

Abstract

This draft provides recommendations on how to generate globally unique Message IDs in client software.

Table of Contents

1. Introduction
2. Message-ID formatting
3. Message-ID generation
3.1 "Domain part"
3.2 "Local part"
3.2.1 Sequence number
3.2.2 Using a pseudorandom number generator
3.2.3 Using a hash
3.3 Bringing it all together
4. Acknowledgments
5. References
6. Authors' addresses

1. Introduction

Message-ID headers are used to uniquely identify Internet messages. Having a unique identifier for each message has many benefits, including ease in the following of threads and intelligent scoring of messages based on threads to which they belong.

It has been suggested that it is impossible for client software to be able to generate globally-unique Message-IDs. We believe this to be incorrect, and herein to offer suggestions for generating unique Message-IDs.

2. Message-ID formatting

As defined in [NEWS], a message ID consists of two parts, a local part and a domain, separated by an at-sign and enclosed in angle brackets:

```
message-id = "<" local-part "@" domain ">"
```

Practically, 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 software generating a Message-ID would be well-advised to avoid this pitfall.

It is also noted that some buggy software considers message IDs completely case-insensitive, in violation of the standards. It is therefore advised that one not generate IDs such that two IDs so generated can differ only in character case.

3. Message-ID generation

As shown above, the Message-ID is made up of two sections. We'll consider each seperately.

3.1. "Domain part"

On many client systems, it is not always possible to get the fully-qualified domain name (FQDN) of the local host. In that situation, a reasonable fallback course of action would be to use the domain-part of the user's return address. (Use of an unqualified hostname for the domain part of the Message-ID header would be foolish, and should never be done.)

Using the domain-part of the user's return address makes the generation of the "local part" be more important; in particular, it means that a process ID is probably not sufficient.

3.2. "Local part"

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.

A number of approaches here are possible. Each has its advantages and drawbacks. The importance of the local part's uniqueness increases with the frequency of messages being generated in a given domain. Using several of these methods together will produce a Message-ID that is longer, but significantly less likely to collide.

3.2.1. Sequence number

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 for servers, but requires careful design to cope properly with simultaneous posting attempts, and is not as robust in the presence of crashes and other malfunctions. For client Message-ID generation, particularly on hosts where the exact FQDN cannot be obtained, or is subject to change, this might not even be workable.

3.2.2. Using a psuedorandom number generator

One could take 64 bits from a good, well-seeded pseudorandom number generator [PRNG] in order to significantly increase the uniqueness of the Message-ID. The advantage of this method is that it is fast and generally effective. The disadvantage is that in a perfect random number generation scheme, the possibility of getting the same number twice in a row is exactly the same probability as getting any two numbers.

3.2.3. Using a hash

Another approach would be to generate a hash of the message and use that after the timestamp. If this is done well, this can also significantly reduce the opportunity for collision, and will generate a value that is relatively unique. Note that, in practice, this is more difficult than it sounds. It is recommended that a cryptographically secure hash function [SHA1, MD5] be used, as others, such as CRC, are likely to have higher instances of collision.

3.3. Bringing it all together

In summary, the approaches to generating a Message-ID that we'll consider here are in the following format:

1 Append "<".

2 Get the current time in the highest resolution to which you have access (at least seconds, though most systems will give you milliseconds) and generate a timestamp in the format yyyymmddHHMMSS.ss;

3 Generate additional data to prevent Message-ID collision on two messages processed by the same host at precisely the same moment. (See section 3.2.) Convert these two numbers to base 36 (0-9 and A-Z), and write the first number, then additional parts, each section seperated by a ".", and an "@".

5 Append the FQDN of the local host, or the host name in the user's return address.

6 Append ">".

4. Acknowledgments

This document is partially derived from an earlier, unrelated draft by Henry Spencer.

5. References

Ref.	Author, title	IETF status (June 1998)
[NEWS]	M.R. Horton, R. Adams: "Standard for interchange of USENET messages", RFC 1036, December 1987.	Non-standard (but still widely used as a de-facto standard).
[SHA1]	National Institute of Standards and Technology (NIST), "Announcement of Weakness in the Secure Hash Standard", May 1994. (Update of FIPS 180: "Secure Hash Standard".)	
[MD5]	R. Rivest: "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.	Informational (but (widely used as a de-facto standard).
[PRNG]	D. Eastlake, 3rd, S. Crocker, J. Schiller: "Randomness Recommendations for Security", RFC 1750, December 1994.	Informational.

6. Authors' Addresses

Matt Curtin
The Ohio State University
791 Dreese Laboratories
2015 Neil Ave
Columbus OH 43210
+1 614 292 7352
cmcurtin@cis.ohio-state.edu

Jamie Zawinski
Netscape Communications Corporation
501 East Middlefield Road
Mountain View, CA 94043
(650) 937-2620
jwz@netscape.com