Internet Draft
draft-ietf-usefor-msg-id-alt-00

Claus Andre Faerber

1998-09-06
Expires: 1999-03-06

Guidelines for the Generation of Message IDs and Similar Unique Identifiers

Status of this Memo

Copyright Notice

Abstract

[RFC822] and [RFC1036] define so-called 'Message-IDs' that represent a unique identifier for email and netnews messages. A similar identifier is also used by [RFC2045] for the 'Content-ID' label.

For each of these protocols, uniqueness of the identifiers generated is more or less essential. Unfortunately, the original Message-ID specification requires that the generator have an own, non-temporary full qualified domain name available, which does not allow hosts that are connected via dialup lines and get dynamically assigned IP addresses (and hostnames) to generate unique IDs offline.

This memo provides recommendations for the generation of such IDs without risking non-uniqueness.

Table of Contents

1 Format And Use of Message-IDs
  1.1 The Message-ID and Content-ID headers
  1.2 Syntax for IDs
  1.3 Uniqueness of Message-IDs and Content-IDs

2 Message ID format
   2.1 Message ID namespaces
      2.1.1 Based on the Host's Full Qualified Domain Name
      2.1.2 Based on an Email Address
      2.1.3 Based on Login Name and FQDN
      2.1.4 Obsolete methods
         2.1.4.1 Based on a UUCP name
         2.1.4.2 Based on an IP Address
      2.1.5 Non-Acceptable Methods

   2.2 Generating the "unique" part
      2.2.1 Current Date and Time
      2.2.2 Process and Thread ID
      2.2.3 Sequence Number
      2.2.4 Software name
      2.2.5 Other Unique Data Sources

3 Security considerations
   3.1 Namespace Invasions
   3.2 Revealing Information about the Generating system

Definitions

This memo uses the Augmented BNF defined in [RFC2234] as well as some definitions from [RFC822].
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1 Format And Use of Message-IDs

1.1 The Message-ID and Content-ID headers

The Message-ID is defined in [RFC822], to which [RFC1036] and [RFC2045] reference, as follows:

```
msg-id      =  "<" addr-spec ">"            ; Unique message id
```

NOTE: These IDs are used in special message headers, whose format is not important for this specification, which only deals with the part between the "<>"s.

In this specification, the terms "Message-ID" and "Content-ID" or just "ID" refer to the part between the "<" and ">", which are considered delimiters.

NOTE: This differs from [RFC822] and [RFC1036], where the angle brackets are considered part of the ID. However, the definition used herein is consistent with the use of such identifiers in other protocols, such as URIs.

1.2 Syntax for IDs

The syntax of IDs is defined in [RFC822] as being the same as that of domain-based Internet email addresses:

```
addr-spec   =  local-part "@" domain
```

1.3 Uniqueness of Message-IDs and Content-IDs

According to [RFC822], Message-IDs "are" unique, i.e. not reused for other email messages.

[RFC1036] RECOMMENDS Message-IDs to be unique for at least two years.

[RFC2045] says Content-IDs are world-unique "like Message-ID values".

This memo RECOMMENDS that IDs are generated in a way that guarantees uniqueness for an unlimited period of time. The methods presented here fulfil this recommendation.

None of the specifications above says whether reusing an IDs of one type as the ID of another type (e.g. using the same ID as a Message-ID and as a Content-ID) is allowed. As any ID generator must be able to generate an arbitrary number of unique IDs, reusing IDs of one type for other types of IDs is PROHIBITED.

As a special exception, for messages sent via both email and news, both copies may use the same Message-ID, provided both copies are considered the same.

NOTE: The exact definition of being "the same" is beyond the scope of this memo.

2 Message ID format

To guarantee uniqueness, IDs consist of two elements:

namespace: An identifier derived from managed databases such as the DNS. The owner of an ID namespace is the owner of the identifier assigned by that database.

unique: A more or less arbitrary string, whose uniqueness is guaranteed by the owner of the ID namespace.

Traditionally, the namespace was the right hand side of message ID. This is no longer true. See sections 2.1.2 and 2.1.3 for details.

2.1 Message ID namespaces

2.1.1 Based on the Host's Full Qualified Domain Name

If the host generating the ID has an own, not dynamically assigned

host name, this name MAY be used:

```
  fqdn-id      := unique "@" fqdn
  fqdn         := <name of the host generating the id>
```

The host name MUST be a full qualified domain name of the host. CNAMEs are ALLOWED. It is not required (OPTIONAL) that the FQDN be visible in the DNS as long as it has been assigned to a host owner by the authority of the domain it belongs to and it is syntactically valid.

NOTE: This allows providers that use dialup lines to assign "host names" solely for the purpose of offline ID creation to their customers.

The person who is responsible for running software that creates IDs of this type SHOULD have the explicit permission of the host owner. This may be the owner him/herself, the administrator or a privileged user.

Examples:

```
   <6$klsd0kfdl@mail.example.com>
   <A2D634B3.432534C0.9392@news0.example.com>
   <6334C0.9A4B3.4325392@j.smith.example.com>
```

```
Illegal Examples:                    Reason:
   <A23DF2343409@mycomputer>            Not full qualified.
   <fslkd0394lkdf0i203kl@dialup-23.example.com> Not static (assuming
                                        that dialup-23.example.com is
                                        a name assigned to one of the
                                        IP addresses for dynamic
                                        allocation).
```

2.1.2 Based on an Email Address

If messages are prepared offline, a static FQDN or IP address may not be available. For this case, IDs may be generated from the owner's email address:

```
  email-id     := unique "%" addr-spec
```

NOTE that addr-spec contains the required "@".

The local part MAY be passed through a non-reversible hash function, such as the standard POSIX crypt(). In this case, the delimiter SHOULD be "%%" or "%<hash function>%" instead of "%":

```
  emailhash-id := unique "%" [hash-name] "%" local-part "@" domain
  hash-name    := "crypt" / "md5" / "sha1" / ...
```

NOTE that hash functions do not guarantee uniqueness. Account managers should warn users if two local-parts in the same domain

produce the same result with one commonly used hash function.

NOTE There is currently no registry for hash function names. This is not a problem as the name is only used to avoid clashes.

Examples:
<dfsxl3kc03kl%claus@faerber.muc.de>
<dkfskflskf%md5%4e32df22da@faerber.muc.de>

2.1.3 Based on Login Name and FQDN

Similar to based on email address (section 2.1.2), however the email address is replaced by the login name of the current user and the FQDN of the host. This MAY or MAY NOT be a valid email address.

```
login-id     := unique "%" login "@" fqdn
login        := <login name of the current user
                or numeric user id>
```

For the full qualified domain names, the rules from section 2.1.1 apply except that the generator need not be under the control of the host owner, but MUST be run by or on behalf of by the person owning the login name used.

This method is preferred over using the FQDN only (section 2.1.1) for implementations not run by the host administrator or owner.

Hash functions MAY be used as for email addresses:

```
loginhash-id := unique "%" [hash-name] "%" login "@" fqdn
```

2.1.4 Obsolete methods

These methods for obtaining a unique identifier are still valid but deprecated. One of the methods above SHOULD be used instead.

2.1.4.1 Based on a UUCP name

Similar to based on the FQDN (section 2.1.1), however the FQDN is replaced by the UUCP name of the host in the top level domain "uucp".

The UUCP name MUST be reserved in the UUCP Worldmap. UUCP names MUST NOT be used without the trailing ".uucp".

Examples:
<kdkfjlsfjsdjf@uunet.uucp>

Illegal Examples:	Reason:
<slskdfdfslfld@uunet>	No ".uucp" suffix.
<lkflklfksldkf@mycomputer>	Not a valid UUCP name / not reserved in the Worldmap.

2.1.4.2 Based on an IP Address

Similar to based on the FQDN (section 2.1.1), however the FQDN is replaced by the IP address of the host in square brackets.

```
ip-id       := unique "@" ip-literal
ip-literal  := "[" <the numeric IP-address> "]"
```

The IP address used MUST be static, i.e. it MUST NOT be assigned dynamically (cf. section 2.1 about static host names).

The IP address MUST NOT be from on of the areas reserved for private use (e.g. 192.168.*.*).

Example:

```
<6ASDFLKF3409SFKLDK@[123.45.67.89]>
```

Illegal Example:	Reason:
<23043klksf034sdfs@[10.0.0.1]>	Private IP

2.1.5 Non-Acceptable Methods

The following methods are not acceptable and SHOULD NOT be used:

* Using only the domain part of the email address.
* Using a computer name that is for use in private networks (LANs) and not guaranteed to be world-unique.

* Methods that violate one or more of the restrictions introduced in sections 2.1.1 to 2.1.4.

2.2 Generating the "unique" part

The "unique" part is the part that changes from ID to ID generated by the same entity. The uniqueness is guaranteed by the owner of the ID namespace it is used in. It is important to use a scheme that makes it very unlikely, better impossible, that the same "unique" string will be generated twice.

The "unique" part is often derived from a combination of the following data:

* the current date and time
* the process and thread id of the generating service
* a sequence number
* the name of the software generating the ID

The data MAY be encoded in some way, to make IDs shorter and to make clear that the unique part of IDs SHOULD NOT be interpreted. Examples for such encodings are MIME Base64 [RFC2045] (preferably of the binary representation of the data), Hex (Base 16), etc.

Because IDs can occur very frequently in some protocols, particularly in References lines of mail and news messages, all efforts should be made to make them compact, as long as they remain unique.

The full set of characters that MAY be freely used for the encoded form is:

```
unique-chars := "A".."Z" / "a".."z" / "1".."9"
              / "!" / "#" / "$" / "&" / "'" / "*" / "+" / "-"
              / "/" / "=" / "?" / "^" / "_" / "`" / "{" / "|"
              / "}" / "~"                          ; 80 characters
```

The dot (".") MAY be used everywhere except the beginning and the end, e.g. as a field separator.

```
unique-word := 1*unique-chars
unique      := unique-word *( "." unique-word )
```

The percent sign ("%") is NOT RECOMMENDED due to its use as a separator for the email and login+FQDN namespaces (sections 2.1.3 and 2.1.4). The use of "quoted-string"s is NOT RECOMMENDED for compatibiliby with buggy software.

NOTE that protocols that use IDs MAY allow additional syntax elements within the IDs, such as comments or line breaks. These are not considered part of the ID.

Generators MAY encrypt IDs with reversible methods. Non-reversible hash functions SHOULD NOT be used, as they usually do not guarantee uniqueness.

Example for ID generating schemes:

<microseconds since 1970-01-01, base 36-encoded> "." <process id, base 36-encoded>

<seconds since 1960-01-01, hex-encoded> "." <internal counter>

<number of 100ns intervals since 1600-01-01, base 80-encoded> "." <process and thread id, base 80-encoded> "." <LAN computer name>

(These are provided as examples, not as a recommendation.)

2.2.1 Current Date and Time

The current date and time MUST be used, as it is the only data that changes regularly and most reliably. The highest available resolution is RECOMMENDED.

To make it impossible that two IDs are generated by the same process within the lowest measurable time on the system, the generating process may either sleep until the system time changes or simulate a higher resolution by incrementing an internal counter.

2.2.2 Process and Thread ID

As different processes might generate IDs at exactly the same time, the time alone may not be sufficient on multitasking systems. To avoid clashes, the process ID and - if the generating process is multithreaded - the thread ID MAY be used.

2.2.3 Sequence Number

A sequence number that is incremented for every ID and stored in a file MAY be used to make clashes less likely.

However, an ID generator MUST NOT, repeat MUST NOT use such counters as the only source for the generation of the unique part, as this will result in clashes in case the file is deleted and/or restored from a backup.

2.2.4 Software name

The name of the software MAY be used to avoid that users switching their software will accidentally have the same "unique" string created from different input data processed differently.

Note: The software name should be as short as possible, there should not be a version number of the software, unless the generation algorithm changes in a way that would make clashes possible.

2.2.5 Other Unique Data Sources

Other sources that may be used to guarantee or further enhance the probability of uniqueness include:

* The host's Ethernet MAC address.
* The host's LAN name (not full qualified).
* Data entered by the user told to enter a unique string.

It might be wise to use these together with email based (section 2.1.2) namespaces, as the same user can work on different machines.

3 Security considerations

3.1 Namespace Invasions

Message IDs are traditionally insecure. There is currently no method to prevent Message IDs from being faked.

Accidental or intentional clashes of IDs have different impact depending on the protocol they are used for. Each protocol specification using IDs MUST address the security issues raised and SHOULD provide methods to prevent abuse of ID namespaces owned by others.

Future standards MAY associate ID namespaces with public/private key pairs and require authentication on a per-namespace basis.

Using encryption of the unique part (see section 2.2) makes it harder to guess the next IDs generated for denial of service attacks.

3.2 Revealing Information about the Generating system

Implementers and users are warned that the following information might be derived by the analysis of message IDs:

* host the ID was generated on
* email address of the user (also as combination of login@fqdn)
* date and time when the ID was generated
* resolution of the system's clock
* operating system (from thread and process IDs)
* software used (also from structure of the unique string)

By encrypting the unique part (see section 2.2) or using hash functions (see section 2.1.2), this can be made nearly impossible.

References

[RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", August 1982, STD 11, RFC 822.

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

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

[RFC2111] Levinson, E., "Content-ID and Message-ID Uniform Resource Locators," March 1997, RFC 2111.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997, RFC 2119.

[RFC2234] Crocker, D., Overell, P., "Augmented BNF for Syntax Specifications: ABNF", November 1997, RFC 2234.

Author

Claus Andre Faerber
Mitterfeldstrasse 20
83043 Bad Aibling
Germany

Fax: +49-8061-3361

E-Mail: cfaerber@muc.de

Note: Please write the author's name with the correct diacritic marks where possible, i.e. Claus André Färber in HTML.

Internet Draft
Expires 1999-03-06