Network Working Group
Request for Comments: 4865
Updates: 3463, 3464
Category: Standards Track

G. White
Independent
G. Vaudreuil
Alcatel-Lucent

May 2007

SMTP Submission Service Extension for Future Message Release

Status of This Memo

Copyright Notice

Abstract

This memo defines an extension to the SMTP submission protocol for a client to indicate a future time for the message to be released for delivery. This extension permits a client to use server-based storage for a message that should be held in queue until an appointed time in the future. This is useful for clients which do not have local storage or are otherwise unable to release a message for delivery at an appointed time.

1. Introduction

There is a widely used feature within the voice messaging community to compose and send a message for delivery in the future. This is useful for sending announcements to be heard at the beginning of a work day, to send birthday greetings a day or so ahead, or to use as a lightweight facility to build a personal reminder service.

This extension uses the SMTP submission protocol [n3] to allow a client, when submitting a message, to indicate a future time for the message to be released for delivery.

2. Terminology

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 [n1].

3. Framework

The Future Message Release service extension for SMTP submission uses the SMTP service extension mechanism [n4] to extend the SMTP submission protocol [n3]. The following SMTP submission service extension is hereby defined:

The name of the SMTP submission service extension is "Future Message Release".

1) The Extended Hello (EHLO) keyword associated with this service extension is "FUTURERELEASE".

2) Two required parameters, the max-future-release-interval and the max-future-release-date-time, are combined with the EHLO keyword in the manner specified in [n4].

The max-future-release-interval is a positive integer indicating the maximum amount of time for which the message submission server (MSA) will hold messages for future release.

Using ABNF [n2], the syntax of this parameter is as follows:

```
future-release-integer = %x31-39 *8DIGIT
                         ; integer in the range 1-999999999
                         ; measured in seconds

max-future-release-interval = future-release-integer
```

The max-future-release-date-time is a timestamp, normalized to Universal Coordinated Time (UTC), indicating the most remote date and time in the future until which the MSA will hold messages for future release.

Using ABNF [n2], the syntax of this parameter is as follows:

```
max-future-release-date-time = date-time
```

where the format of date-time is defined in [n10].

3) When forming the portion of the EHLO reply containing the FUTURERELEASE keyword, the keyword is followed by the max-future-release-interval, and then the max-future-release-date-time. The keyword and two values are delimited by spaces.

   For example, the ABNF for a continuation line in the EHLO response that contains the FUTURERELEASE keyword is:

```
line = "250-FUTURERELEASE" SP max-future-release-interval
                           SP max-future-release-date-time
```

4) One required parameter, the hold-param, is added to the MAIL command using either the keyword "HOLDFOR" or the keyword "HOLDUNTIL".

   The HOLDFOR parameter value is a future-release-interval, which is a positive integer indicating the amount of time the message is to be held by the MSA before release.

   The HOLDUNTIL parameter value is a future-release-date-time, which is a timestamp, normalized to UTC, indicating the future date and time until which the message is to be held by the MSA before release.

   Using ABNF [n2], the syntax of this parameter is as follows:

```
future-release-interval = future-release-integer

future-release-date-time = Internet-style-date-time-UTC

hold-for-param = "HOLDFOR=" future-release-interval

hold-until-param = "HOLDUNTIL=" future-release-date-time

hold-param = hold-for-param / hold-until-param
```

   The absence of this parameter on the MAIL command does not imply a default value for this parameter.

5) The maximum length of a MAIL command is increased by 34 characters by the possible addition of the hold-param.

6) No additional SMTP verbs are defined by this extension.

7) This service extension is appropriate only for the SMTP submission protocol [n3]. This service extension is not appropriate for standard SMTP [n4].

4. Behavior

It is unfortunate to define two seemingly identical ways to indicate a future message release time. When the client has both accurate time and accurate time zone information, either interval or date-time can be trivially calculated from the other. However, in the current world of clients, there are clients with accurate local time but no indication of their time zone, and clients without a suitably accurate clock. Based on the limited facilities available to these time-challenged clients, it is likely that only one or the other of these mechanisms will be useful.

It is believed that servers will have accurate time, and can trivially convert between these mechanisms. It is also accepted that the protocol and implementation overhead of offering these two mechanisms is low, and that few interoperability challenges are anticipated.

4.1. SMTP Client

1) An SMTP client preparing to use Future Message Release MUST first verify that the MSA supports this extension.

2) An SMTP client using Future Message Release MUST include one, and only one, hold-param with the MAIL command.

3) An SMTP client using Future Message Release with the "for" option of the hold-param MUST ensure that the future-release-interval is less than or equal to the max-future-release-interval advertised by the MSA.

4) An SMTP client using Future Message Release with the "until" option of the hold-param MUST ensure that the future-release-date-time is earlier than or equal to the max-future-release-date-time advertised by the MSA.

4.2. MSA

1) An MSA supporting Future Message Release MUST comply with the SMTP submission protocol as described in [n3].

2) An MSA supporting Future Message Release MUST NOT advertise this support (i.e. include the FUTURERELEASE keyword in its EHLO reply) on any port other than the submission port.

3) An MSA supporting Future Message Release MUST include the FUTURERELEASE keyword, and associated max-future-release-interval and max-future-release-date-time parameters, in its reply to the EHLO command.

4) An MSA supporting Future Message Release MUST accept a MAIL command containing a valid hold-param, given that the MAIL command contains no other errors.

5) An MSA that accepts a message with a request for Future Message Release indicating the "for" option MUST NOT release the message until the amount of time specified in the future-release-interval elapses.

6) An MSA that accepts a message with a request for Future Message Release indicating the "until" option MUST NOT release the message until the date and time indicated by the future-release-date-time occurs.

7) An MSA supporting Future Message Release MUST reject a MAIL command containing the "for" option specifying a value that is greater than the advertised max-future-release-interval, or otherwise invalid.

8) An MSA supporting Future Message Release MUST reject a MAIL command containing the "until" option specifying a value that is later than the advertised max-future-release-date-time, or otherwise invalid.

9) An MSA supporting Future Message Release MUST reject a MAIL command containing more than one hold-param.

10) An MSA supporting Future Message Release, when rejecting a MAIL command per items 7, 8, or 9, above, SHOULD supply the reply code 501 (syntax error in parameters or arguments [n4]) in the reply.

11) An MSA supporting Future Message Release, when rejecting a MAIL command per items 7, 8, or 9, above, SHOULD supply the Enhanced Mail System Status Code 5.5.4 (invalid command arguments [i1]) in the reply.

5. Protocol Interactions

5.1. Interaction with the DSN SMTP Service Extensions

The Delivery Status Notification (DSN) service extension is described in [n7], and DSN message format is described in [n8].

5.1.1. SMTP Client Interaction with DSN

1) An SMTP client MUST NOT request Future Message Release when sending a DSN to the MSA.

5.1.2. MSA Interaction with DSN

1) If an MSA generates a DSN for a message that includes a Future Message Release request, the MSA MUST include an Arrival-Date field in the machine-readable body part of the DSN.

2) If an MSA generates a DSN for a message that includes a Future Message Release request, the MSA MUST include a Future-Release-Request field in the machine-readable body part of the DSN. The value of this field is the value of the HOLD parameter contained in the MAIL command of the original message.

   The Future-Release-Request field is an extension to the set of DSN per-message fields described in [n8]. Using ABNF [n2], the syntax of this new field is as follows:

```
orig-hold-param-value = ("for;" future-release-interval) /
                        ("until;" future-release-date-time)
                    ; this is the value of the HOLD param from
                    ; the MAIL command of the original message

future-release-request-field = "Future-Release-Request:"
                               orig-hold-param-value
```

5.2. Interaction with the DELIVERBY SMTP Service Extension

If an MSA supports the Future Message release and Deliver By service extensions, it is possible for an SMTP client to make simultaneous requests for future message release and deliver-by times when submitting a message. A problem will occur if the future message release time is farther in the future than the deliver-by time. In order to honor the deliver-by request, the future message release request has to be ignored. In order to honor the future message release request, the deliver-by request has to be ignored. This section addresses that problem. The Deliver By extension is described in [n6].

5.2.1. SMTP Client Interaction with DELIVERBY

1) When an SMTP client wishes to use the Future Message Release and Deliver By extensions with the same message, the client MUST ensure that the specified deliver-by time is farther in the future than the specified ("until" option) or implied ("for" option) future message release time.

5.2.2. MSA Interaction with DELIVERBY

1) If an MSA supports Future Message Release and Deliver By extensions, and receives a message requesting the use of both extensions, the MSA MUST reject the MAIL command if it determines that the future message release time is farther in the future than the deliver-by time.

2) When an MSA is rejecting a MAIL command per item 1, above, it SHOULD supply the reply code 501 (syntax error in parameters or arguments [n4]) in the reply.

3) When an MSA is rejecting a MAIL command per item 1, above, it SHOULD supply the Enhanced Mail System Status Code 5.5.4 (invalid command arguments [i1]) in the reply.

5.3. Interaction with the MDN Function

The Message Disposition Notification (MDN) function is described in [n9].

5.3.1. SMTP Client Interaction with MDN

1) An SMTP client MUST NOT request Future Message Release when sending an MDN to the MSA.

6. Security Considerations

The Future Message Release service extension presents a number of security considerations:

1) Unauthorized future-release messages provide a means to overwhelm the storage of an MSA. The authorization mechanisms required for the base mail submission protocol [n3] are expected to provide appropriate defense against such attacks.

2) Authorized future message release without a per-user quota may also provide a way to overwhelm an MSA’s storage. An MSA’s future release message storage SHOULD be subject to a per-user quota.

3) If an MSA is imposing a per-user quota on future-release message storage, and detects that an incoming future-release message will exceed the user's future-release message storage quota, the MSA MUST reject the MAIL command.

4) When an MSA is rejecting a MAIL command per 5.3, it SHOULD supply the reply code 552 (requested mail action aborted: exceeded storage allocation [n4]) in the reply.

5) When an MSA is rejecting a MAIL command per 5.3, it SHOULD supply the new Enhanced Mail System Status Code defined for this purpose. This new status code updates [i1].

   X.7.16 Future release per-user message quota exceeded

   There is insufficient per-user quota to queue the message for future release. This code suggests the client can submit again only after the per-user queue has drained.

   X.7.17 Future release system message quota exceeded

   There is insufficient system quota to queue the message for future release. This code suggests the client can submit again after the system queue has drained.

6) Inaccurate time on the MSA may result in premature or delayed release of messages. Both HOLDUNTIL and HOLDFOR request mechanisms are sensitive to inaccurate or changing clocks on the MSA.

7) Some element of deception is inherent in the future message release concept. The message release time is intentionally delayed past the time it would otherwise be released; hence, the message delivery time is delayed past the time it would otherwise be delivered. This extension provides no mechanism for hiding this from the message recipient. The RFC 2822 [n5] message header, and specifically the Date field, remain unchanged after submission. While a sending client MAY elect to place the future-message-release-time as the date in the Date field, there is no requirement or expectation that the Received fields and other trace information be modified by the transport system to further this deception.

7. IANA Considerations

This extension has been added to the list of SMTP Service Extensions on the Mail Parameters Web page.

8. Acknowledgments

Much of the credit for this document is due to the LEMONADE working group. Through many revisions, the discussion resulted in fundamental new understandings of this protocol and corresponding refinement of the implied requirements and protocol. Special thanks to those who patiently lead the WG to understand that doing both interval and date-time was the pragmatically correct approach to the needs of diverse clients.

9. Normative References

[n1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[n2] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[n3] Gellens, R. and J. Klensin, "Message Submission for Mail", RFC 4409, April 2006.

[n4] Klensin, J., "Simple Mail Transfer Protocol", RFC 2821, April 2001.

[n5] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[n6] Newman, D., "Deliver By SMTP Service Extension", RFC 2852, June 2000.

[n7] Moore, K., "Simple Mail Transfer Protocol (SMTP) Service Extension for Delivery Status Notifications (DSNs)", RFC 3461, January 2003.

[n8] Moore, K. and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications", RFC 3464, January 2003.

[n9] Hansen, T. and G. Vaudreuil, "Message Disposition Notification", RFC 3798, May 2004.

[n10] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002

10. Informative References

[i1] Vaudreuil, G., "Enhanced Mail System Status Codes", RFC 3463, January 2003.

Authors' Addresses

Gregory A. White
6519 Camille Ave.
Dallas, TX 75252
USA
EMail: g.a.white@tx.rr.com

Gregory M. Vaudreuil
Alcatel-Lucent
9489 Bartgis Ct
Frederick, MD 21702
USA
EMail: GregV@ieee.org

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.