Network Working Group
Request for Comments: 2442
Category: Informational

N. Freed
D. Newman
Innosoft
J. Belissent
Sun Microsystems
M. Hoy
Mainbrace

November 1998

The Batch SMTP Media Type

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document defines a MIME content type suitable for tunneling an ESMTP [RFC-821, RFC-1869] transaction through any MIME-capable transport. This type can be used for a variety of purposes, including: Extending end-to-end MIME-based security services (e.g., [RFC-1847]) to cover message envelope information as well as message content. Making it possible to use specific SMTP extensions such as NOTARY [RFC-1891] over unextended SMTP transport infrastructure. Enabling the transfer of multiple separate messages in a single transactional unit.

Requirements Notation

This document occasionally uses terms that appear in capital letters. When the terms "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" appear capitalized, they are being used to indicate particular requirements of this specification. A discussion of the meanings of the terms "MUST", "SHOULD", and "MAY" appears in [RFC-1123]; the terms "MUST NOT" and "SHOULD NOT" are logical extensions of this usage.

The Application/batch-SMTP Content Type

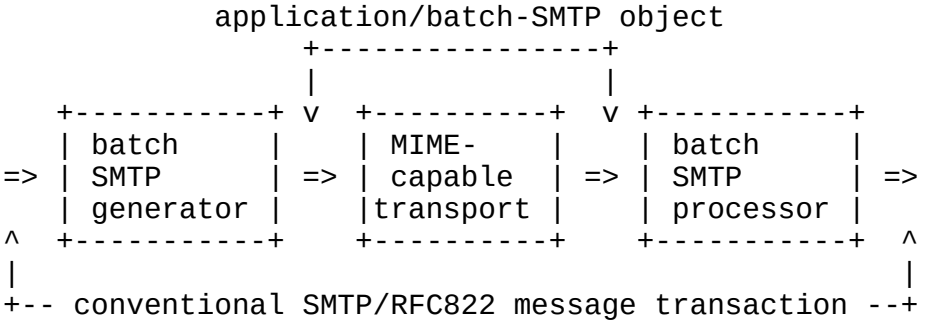
The "application/batch-SMTP" MIME content type is a container for the client side of an SMTP or ESMTP transaction. In keeping with traditional SMTP, the contents are line oriented and CRLF line terminators MUST be used.

The "application/batch-SMTP" type is defined as follows:

Media type name: application
Media subtype name: batch-SMTP
Required parameters: none
Optional parameters: required-extensions
Encoding considerations:
8bit material may appear, so quoted-printable or base64 encoding may be necessary on transports that do not support 8bit. While the content of this type is line-oriented and uses conventional CR/LF terminators, lines longer than 7bit and 8bit encodings allow (998 octets) may appear, hence quoted-printable or base64 encoding may be necessary even in conjunction with 8bit transports.
Security considerations:
Discussed in the Security Considerations Section.

How application/batch-SMTP is used

The following diagram illustrates how the application/batch-SMTP type is intended to be used:



A conventional SMTP message transaction is converted into an application/batch-SMTP object by the batch SMTP generator. This object is then carried over some type of MIME-capable transport. Once the destination is reached the object is presented to a batch SMTP processor, which converts the application/batch-SMTP object back into a conventional SMTP message transaction.

Generation of application/batch-SMTP material

Application/batch-SMTP material is generated by a specially modified SMTP client operating without a corresponding SMTP server. The client simply assumes a successful response to all commands it issues. The resulting content then consists of the collected output from the SMTP client.

Honoring SMTP restrictions

Most batch SMTP processors will be constructed by modifying and extending existing SMTP servers. As such, all of the restrictions on SMTP constructs imposed by RFC 821, RFC 1123, and RFC 1869 MUST be observed. In particular, restrictions on command and data line lengths, number of recipients, and so on still exist and apply to batch SMTP.

Use of SMTP Extensions

Since no SMTP server is present the client must be prepared to make certain assumptions about which SMTP extensions can be used. The generator MAY assume that ESMTP [RFC-1869] facilities are available, that is, it is acceptable to use the EHLO command and additional parameters on MAIL FROM and RCPT TO. If EHLO is used MAY assume that the 8bitMIME [RFC-1652], SIZE [RFC-1870], and NOTARY [RFC-1891] extensions are available. In particular, NOTARY SHOULD be used. MAY create private bilateral agreements which specify the availability of additional SMTP extensions. Additional SMTP extensions MUST NOT be used in the absence of such an agreement, and, perhaps more importantly, a conformant generation of application/batch-SMTP objects MUST be able to produce objects restricted to use of the extensions listed above.

The "required-extensions" content type parameter MAY be used to communicate a list of the extensions actually used, specified as a comma-separated list of EHLO responses. If absent it defaults to the list "8bitMIME,SIZE,NOTARY". Any use by private bilateral agreement of additional or different extensions MUST be noted in the "required-extensions" parameter.

Note that many SMTP extensions simply do not make sense in the context of batch SMTP. For example, the pipelining extension [RFC-2197] makes no sense in the absence of a network connection.

Handling Multiple Messages

Generators SHOULD attempt to minimize the number of messages placed in a single application/batch-SMTP object. Ideally a single application/batch-SMTP object will be created for each message. Note, however, that some uses of application/batch-SMTP (e.g., mail bagging) may exist solely to take advantage of the multiple messages in a single container capability of batch SMTP, so requiring one message per container is not possible.

DISCUSSION: The SMTP protocol provides for the transfer of a series of messages over a single connection. This extends in a natural way to batch SMTP. However, the issues in batch SMTP are somewhat different. Suppose, for example, that a batch SMTP processor receives an application/batch-SMTP object containing two messages but is unable to process the second message because of a storage allocation failure. But suppose that not only does this failure preclude processing of the second message, it also precludes recording that the first message has already been processed. Subsequent reprocessing of the application/batch-SMTP could then lead to duplication of the first message.

This issue is not materially different from the well-known problems with SMTP synchronization that in practice often lead to duplicated messages. Since this behavior is inherent in SMTP to begin with it is not incumbent on application/batch-SMTP to completely address the issue. Nevertheless, it seems prudent for application/batch-SMTP to try and not make matters even worse.

Transport of application/batch-SMTP objects

Application/batch-SMTP objects may be transported by any transport capable of preserving their MIME labelling, e.g., HTTP or SMTP.

Transports MUST remain cognizant of the special nature of application/batch-SMTP. An application/batch-SMTP object contains one or more "frozen" SMTP message transactions. SMTP message transactions typically carry with them various assumptions about quality of service, e.g., that messages will either be delivered successfully or a nondelivery notification will be returned, that a nondelivery notification will be returned if delivery cannot be accomplished in a timely fashion, and so on. It is vital that the encapsulation of these objects for carriage over other forms of transport not interfere with these capabilities.

Processing of application/batch-SMTP material

Processing of application/batch-SMTP material is considerably more complex than generating it. As might be expected, a modified SMTP/ESMTP processor is used. However, since it cannot return information to the client, it must handle all error conditions that arise itself. In other words, a batch SMTP processor assumes both the responsibilities of a traditional SMTP server as well as part of the responsibilities of a traditional SMTP client.

As such, a conforming processor: MUST check MIME content type information to insure that the material it has been presented with is labelled as application/batch-SMTP and doesn't specify any extensions the processor doesn't support in the "required-extensions" parameter. Application/batch-SMTP objects that employ an unsupported extension SHOULD be forwarded to the local postmaster for manual inspection and handling. MUST accept any syntactically valid EHLO or HELO command. MUST accept any syntactically valid MAIL FROM command. A conforming processor, MAY, if it so desires, note the unacceptability of some part of a given MAIL FROM command and use this information to subsequently generate non-delivery notifications for any or all recipients. MUST accept any syntactically valid RCPT TO command. A conforming processor SHOULD note the unacceptability of some part of a given RCPT TO command and subsequently use this information to generate a non-delivery notification for this recipient in lieu of actually delivering the message. MUST accept any of the additional parameters defined by the 8bitMIME, SIZE, and NOTARY SMTP extensions on the MAIL FROM and RCPT TO commands. MUST accept the DATA command even when no valid recipients are present. 8bit MIME messages MUST be accepted. MUST accept the RSET command and handle multiple messages in a single application/batch-SMTP object. Processors MUST process each message in an application/batch-SMTP object once and SHOULD take whatever steps are necessary to avoid processing a message more than once. For example, if processing of an application/batch-SMTP object containing multiple messages is interrupted at an intermediate point it should subsequently be restarted at the end of the last message that was completely processed. SHOULD forward any syntactically invalid application/batch-SMTP message to the local postmaster for manual inspection and handling.

Security Considerations

Application/batch-SMTP implements a tunneling mechanism. In general tunneling mechanisms are prone to abuse because they may provide a means of bypassing existing security restrictions. For example, an application/batch-SMTP tunnel implemented over an existing SMTP transport may allow someone to bypass relay restrictions imposed to block redistribution of spam.

Application/batch-SMTP processors SHOULD implement access restrictions designed to limit access to the processor to authorized generators only. (Note that this facility may be provided automatically if application/batch-SMTP is being used to secure message envelope information.)

Acknowledgements

The general concept of batch SMTP has been around for a long time. One particular type of batch SMTP was defined by Alan Crosswell and used on BITNET to overcome BITNET's native 8 character limit on user and host names. However, this form of batch SMTP differed from the current proposal in that it envisioned having the server return the status code responses to the client. In this case the client bore the burden of correlating responses with the original SMTP dialogue after the fact.

Unfortunately this approach proved not to work well in practice. BITNET eventually switched to the same basic form of batch SMTP that has been defined here. Unfortunately that definition was, to the best of the present authors' knowledge, never captured in a formal specification. It should also be noted that the definition given here also differs in that it takes SMTP extensions into account.

Einar Stefferud had previously considered the problem of carrying extended SMTP messages over unextended SMTP transports. He proposed that some form of "double enveloping" technology be developed to address this problem. The mechanism presented here effectively implements the type of solution he proposed.

References

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

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

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

[RFC-1652] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extension for 8bit-MIMEtransport", RFC 1652, July 1994.

[RFC-1847] Galvin, J., Murphy, S., Crocker, S. and N. Freed, "Security Multiparts for MIME: Multipart/Signed and Multipart/Encrypted", RFC 1847, October 1995.

[RFC-1869] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extensions", STD 10, RFC 1869, November 1995.

[RFC-1870] Klensin, J., Freed, N., Moore, K., "SMTP Service Extension for Message Size Declaration", STD 10, RFC 1870, November, 1995.

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

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

[RFC-2197] Freed, N. and A. Cargille, "SMTP Service Extension for Command Pipelining", RFC 2197, September 1997.

Authors' Addresses

Ned Freed
Innosoft International, Inc.
1050 Lakes Drive
West Covina, CA 91790
USA

Phone: +1 626 919 3600
Fax: +1 626 919 3614
EMail: ned.freed@innosoft.com

Dan Newman
Innosoft International, Inc.
1050 Lakes Drive
West Covina, CA 91790
USA

Phone: +1 626 919 3600
Fax: +1 626 919 3614
EMail: dan.newman@innosoft.com

Mark Hoy
Mainbrace Corporation
1136 West Evelyn Avenue
Sunnyvale, CA 94086

tel: +1 408 774 5265
fax: +1 408 774 5290
email: mark.hoy@mainbrace.com

Jacques Bellisent
SunSoft

Phone: +1 650 786 3630
EMail: jacques.belissent@eng.sun.com

Full Copyright Statement