                A Mechanism for Content Indirection
              in Session Initiation Protocol (SIP) Messages

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   This document defines an extension to the URL MIME External-Body
   Access-Type to satisfy the content indirection requirements for the
   Session Initiation Protocol (SIP). These extensions are aimed at
   allowing any MIME part in a SIP message to be referred to indirectly
   via a URI.

Table of Contents

1. Introduction

   The purpose of the Session Initiation Protocol [9] (SIP) is to
   create, modify, or terminate sessions with one or more participants.
   SIP messages, like HTTP, are syntactically composed of a start line,
   one or more headers, and an optional body. Unlike HTTP, SIP is not
   designed as a general-purpose data transport protocol.

   There are numerous reasons why it might be desirable to specify the
   content of the SIP message body indirectly. For bandwidth-limited
   applications such as cellular wireless, indirection provides a means
   to annotate the (indirect) content with meta-data, which may be used
   by the recipient to determine whether or not to retrieve the content
   over a resource-limited link.

   It is also possible that the content size to be transferred might
   overwhelm intermediate signaling proxies, thereby unnecessarily
   increasing network latency. For time-sensitive SIP applications,
   this may be unacceptable. Indirect content can remedy this by moving
   the transfer of this content out of the SIP signaling network and
   into a potentially separate data transfer channel.

   There may also be scenarios where the session-related data (body)
   that needs to be conveyed does not directly reside on the endpoint or
   User Agent. In such scenarios, it is desirable to have a mechanism
   whereby the SIP message can contain an indirect reference to the
   desired content. The receiving party would then use this indirect
   reference to retrieve the content via a non-SIP transfer channel such
   as HTTP, FTP, or LDAP.

   The purpose of content indirection is purely to provide an
   alternative transport mechanism for SIP MIME body parts. With the
   exception of the transport mechanism, indirect body parts are

equivalent to, and should have the same treatment as, in-line body
parts.

Previous attempts at solving the content indirection problem made use
of the text/uri-list [6] MIME type. While attractive for its
simplicity (a list of URIs delimited by end-of-line markers), it
failed to satisfy a number of the requirements for a more general-
purpose content indirection mechanism in SIP. Most notably lacking
is the ability to specify various attributes on a per-URI basis.
These attributes might include version information, the MIME type of
the referenced content, etc.

RFC 2017 defines a strong candidate for a replacement for the
text/uri-list MIME type. RFC 2017 [1] defines an extension to the
message/external-body MIME type originally defined in RFC2046 [3].
The extension that RFC 2017 makes allows a generic URI to specify the
location of the content rather than protocol-specific parameters for
FTP, etc., as originally defined in RFC2046. Although it provides
most of the functionality needed for a SIP content indirection
mechanism, RFC 2017 by itself is not a complete solution. This
document specifies the usage of RFC 2017 necessary to fulfill the
requirements outlined for content indirection.

The requirements can be classified as applying either to the URI,
which indirectly references the desired content, or to the content
itself. Where possible, existing MIME parameters and entity headers
are used to satisfy those requirements. MIME (Content-Type)
parameters are the preferred manner of describing the URI, while
entity headers are the preferred manner of describing the (indirect)
content. See RFC 2045 [2] for a description of most of these entity
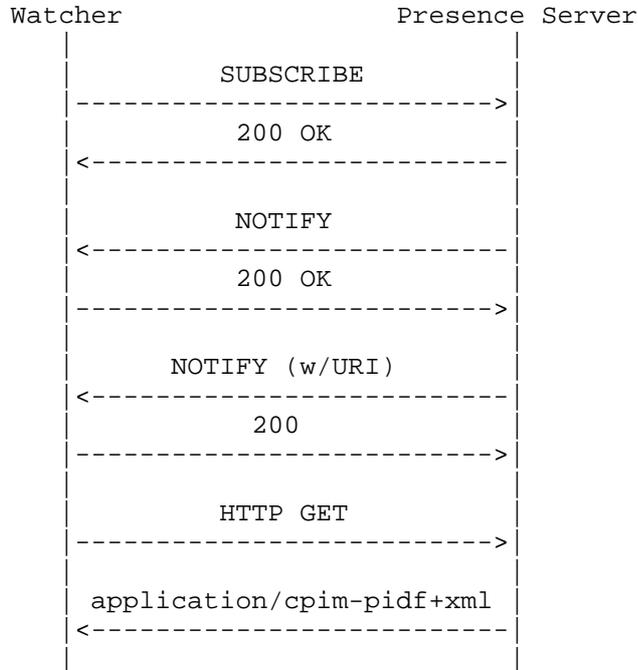headers and MIME parameters.

2. Terminology

   RFC 2119 [5] defines the keywords "MUST", "MUST NOT", "REQUIRED",
   "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY",
   and "OPTIONAL".

3. Use Case Examples

   There are several examples of using the content indirection
   mechanism. These are examples only and are not intended to limit the
   scope or applicability of the mechanism.
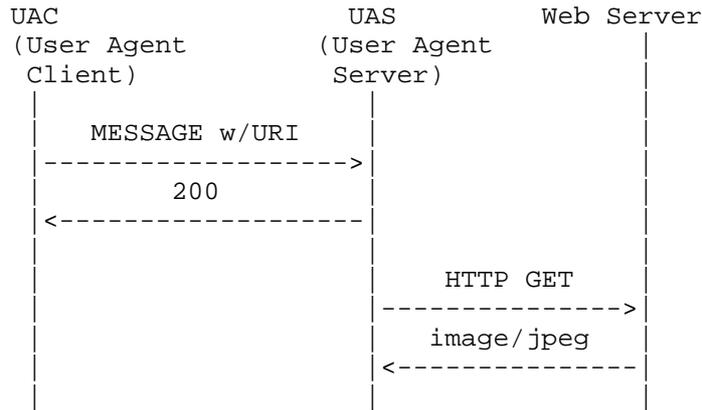
3.1. Presence Notification

   The information carried in a presence document could exceed the
   recommended size for a SIP (NOTIFY) request, particularly if the
   document carries aggregated information from multiple endpoints. In
   such a situation, it would be desirable to send the NOTIFY request
   with an indirect pointer to the presence document, which could then
   be retrieved by, for example, HTTP.

```
              Watcher                      Presence Server
                 |                            |
                 |         SUBSCRIBE          |
                 |--------------------------->|
                 |           200 OK           |
                 |<---------------------------|
                 |                            |
                 |           NOTIFY           |
                 |<---------------------------|
                 |           200 OK           |
                 |--------------------------->|
                 |                            |
                 |        NOTIFY (w/URI)       |
                 |<---------------------------|
                 |            200             |
                 |--------------------------->|
                 |                            |
                 |          HTTP GET          |
                 |--------------------------->|
                 |                            |
                 | application/cpim-pidf+xml  |
                 |<---------------------------|
                 |                            |
```

   In this example, the presence server returns an HTTP URI pointing to
   a presence document on the presence server, which the watcher can
   then fetch by using an HTTP GET.

3.2. Document Sharing

   During an instant messaging conversation, a useful service is
   document sharing, wherein one party sends an IM (MESSAGE request)
   with an indirect pointer to a document that is meant to be rendered
   by the remote party. Carrying such a document directly in the
   MESSAGE request is not an appropriate use of the signaling channel.
   Furthermore, the document to be shared may reside on a completely
   independent server from that of the originating party.

```
              UAC                    UAS           Web Server
              (User Agent           (User Agent         |
               Client)               Server)            |
               |                     |                  |
               |    MESSAGE w/URI    |                  |
               |-------------------->|                  |
               |        200          |                  |
               |<------------------- |                  |
               |                     |                  |
               |                     |     HTTP GET     |
               |                     |--------------->| |
               |                     |    image/jpeg    |
               |                     |<---------------| |
               |                     |                  |
```

   In this example, a user UAC wishes to exchange a JPEG image that she
   has stored on her web server with user UAS with whom she has an IM
   conversation. She intends to render the JPEG inline in the IM
   conversation. The recipient of the MESSAGE request launches an HTTP
   GET request to the web server to retrieve the JPEG image.

4. Requirements

   o It MUST be possible to specify the location of content via a URI.
      Such URIs MUST conform with RFC2396 [7].

   o It MUST be possible to specify the length of the indirect content.

   o It MUST be possible to specify the type of the indirect content.

   o It MUST be possible to specify the disposition of each URI
      independently.

   o It MUST be possible to label each URI to identify if and when the
      content referred to by that URI has changed. Applications of this
      mechanism may send the same URI more than once. The intention of
      this requirement is to allow the receiving party to determine
      whether the content referenced by the URI has changed, without
      having to retrieve that content. Examples of ways the URI could
      be labeled include a sequence number, timestamp, and version
      number. When used with HTTP, the entity-tag (ETAG) mechanism, as
      defined in RFC2068 [4], may be appropriate. Note that we are
      labeling not the URI itself but the content to which the URI
      refers, and that the label is therefore effectively "metadata" of
      the content itself.

   o It MUST be possible to specify the time span for which a given URI
      is valid. This may or may not be the same as the lifetime for the
      content itself.

   o It MUST be possible for the UAC and the UAS to indicate support of
      this content indirection mechanism. A fallback mechanism SHOULD
      be specified in the event that one of the parties is unable to
      support content indirection.

   o It MUST be possible for the UAC and UAS to negotiate the type of
      the indirect content when using the content indirection mechanism.

   o It MUST be possible for the UAC and UAS to negotiate support for
      any URI scheme to be used in the content indirection mechanism.
      This is in addition to the ability to negotiate the content type.

   o It SHOULD be possible to ensure the integrity and confidentiality
      of the URI when it is received by the remote party.

   o It MUST be possible to process the content indirection without
      human intervention.

   o It MUST allow for indirect transference of content in any SIP
      message that would otherwise carry that content as a body.

5. Application of RFC 2017 to the Content Indirection Problem

   The following text describes the application of RFC 2017 to the
   requirements for content indirection.

5.1. Specifying Support for Content Indirection

   A UAC/UAS indicates support for content indirection by including the
   message/external-body MIME type in the Accept header. The UAC/UAS
   MAY supply additional values in the Accept header to indicate the
   content types that it is willing to accept, either directly or
   through content indirection. User-Agents supporting content
   indirection MUST support content indirection of the application/sdp
   MIME type.

   For example:

           Accept: message/external-body, image/*, application/sdp

5.2. Mandatory support for HTTP URI

   Applications that use this content indirection mechanism MUST support
   the HTTP URI scheme. Additional URI schemes MAY be used, but a
   UAC/UAS MUST support receiving a HTTP URI for indirect content if it
   advertises support for content indirection.

   The UAS MAY advertise alternate access schemes in the schemes
   parameter of the Contact header in the UAS response to the UAC's
   session establishment request (e.g., INVITE, SUBSCRIBE), as described
   in RFC 3840 [11].

5.3. Rejecting Content Indirection

   If a UAS receives a SIP request that contains a content indirection
   payload and the UAS cannot or does not wish to support such a content
   type, it MUST reject the request with a 415 Unsupported Media Type
   response as defined in section 21.4.13 of SIP [9]. In particular,
   the UAC should note the absence of the message/external-body MIME
   type in the Accept header of this response to indicate that the UAS
   does not support content indirection, or the absence of the
   particular MIME type of the requested comment to indicate that the
   UAS does not support the particular media type.

5.4. Specifying the Location of the Content via a URI

   The URI for the indirect content is specified in a "URI" parameter of
   the message/external-body MIME type. An access-type parameter
   indicates that the external content is referenced by a URI. HTTP URI
   specifications MUST conform to RFC 2396 [7].

   For example:

           Content-Type: message/external-body; access-type="URL";
              URL="http://www.example.com/the-indirect-content"

5.5. Marking Indirect Content Optional

   Some content is not critical to the context of the communication if
   there is a fetch or conversion failure. The content indirection
   mechanism uses the Critical-Content mechanism described in RFC 3459
   [10]. In particular, if the UAS is unable to fetch or render an
   optional body part, then the server MUST NOT return an error to the
   UAC.

5.6. Specifying Versioning Information for the URI

   In order to determine whether the content indirectly referenced by
   the URI has changed, a Content-ID entity header is used. The syntax
   of this header is defined in RFC 2045 [2]. Changes in the underlying
   content referred to by a URI MUST result in a change in the Content-
   ID associated with that URI. Multiple SIP messages carrying URIs
   that refer to the same content SHOULD reuse the same Content-ID, to
   allow the receiver to cache this content and to avoid unnecessary
   retrievals. The Content-ID is intended to be globally unique and
   SHOULD be temporally unique across SIP dialogs.

   For example:

          Content-ID: <4232423424@www.example.com>

5.7. Specifying the URI Lifetime

   The URI supplied by the Content-Type header is not required to be
   accessible or valid for an indefinite period of time. Rather, the
   supplier of the URI MUST specify the time period for which this URI
   is valid and accessible. This is done through an "EXPIRATION"
   parameter of the Content-Type. The format of this expiration
   parameter is an RFC 1123 [12] date-time value. This is further
   restricted in this application to use only GMT time, consistent with
   the Date: header in SIP. This is a mandatory parameter. Note that
   the date-time value can range from minutes to days or even years.

   For example:

          Content-Type: message/external-body;
                         expiration="Mon, 24 June 2002 09:00:00 GMT"

5.8. Specifying the type of the Indirect Content

   To support existing SIP mechanisms for the negotiation of content
   types, a Content-Type entity header SHOULD be present in the entity
   (payload) itself. If the protocol (scheme) of the URI supports its
   own content negotiation mechanisms (e.g., HTTP), this header may be
   omitted. The sender MUST, however, be prepared for the receiving
   party to reject content indirection if the receiver is unable to
   negotiate an appropriate MIME type by using the underlying protocol
   for the URI scheme.

For example:

```
        Content-Type: message/external-body; access-type="URL";
            expiration="Mon, 24 June 2002 09:00:00 GMT";
            URL="http://www.example.com/the-indirect-content"
        <CRLF>
        Content-Type: application/sdp
        Content-Disposition: session
        <CRLF>
```

5.9.   Specifying the Size of the Indirect Content

   When known in advance, the size of the indirect content in bytes
   SHOULD be supplied via a size parameter on the Content-Type header.
   This is an extension of RFC 2017 but is in line with other access
   types defined for the message/external-body MIME type in RFC 2046.
   The content size is useful for the receiving party to make a
   determination about whether to retrieve the content. As with
   directly supplied content, a UAS may return a 513 error response in
   the event that the content size is too large. Size is an optional
   parameter.

   For example:

```
        Content-Type: message/external-body; access-type="URL";
            expiration="Mon, 24 June 2002 09:00:00 GMT";
            URL="http://www.example.com/the-indirect-content";
            size=4123
```

5.10. Specifying the Purpose of the Indirect Content

   A Content-Disposition entity header MUST be present for all indirect
   content.

   For example:

```
        Content-Type: message/external-body; access-type="URL";
            expiration="Mon, 24 June 2002 09:00:00 GMT";
            URL="http://www.example.com/the-indirect-content"
        <CRLF>
        Content-Type: image/jpeg
        Content-Disposition: render
```

5.11. Specifying Multiple URIs for Content Indirection

   If there is a need to send multiple URIs for content indirection, an
   appropriate multipart MIME type [3] should be used. Each URI MUST be
   contained in a single entity. Indirect content may be mixed with
   directly-supplied content. This is particularly useful with the
   multipart/alternative MIME type.

   NOTE: This specification does not change the meanings of the various
   multipart flavors, particularly multipart/related, as described in
   RFC 2387 [13].

   For example:

          MIME-Version: 1.0
          Content-Type: multipart/mixed; boundary=boundary42

          --boundary42
          Content-Type: text/plain; charset=us-ascii

          The company announcement for June, 2002 follows:
          --boundary42
          Content-Type: message/external-body;
              access-type="URL";
              expiration="Mon, 24 June 2002 09:00:00 GMT";
              URL="http://www.example.com/announcements/07242002";
              size=4123

          Content-Type: text/html
          Content-Disposition: render

          --boundary42--

5.12. Specifying a Hash Value for the Indirect Content

   If the sender knows the specific content being referenced by the
   indirection, and if the sender wishes the recipient to be able to
   validate that this content has not been altered from that intended by
   the sender, the sender includes a SHA-1 [8] hash of the content. If
   it is included, the hash is encoded by extending the MIME syntax [3]
   to include a "hash" parameter for the content type "message/
   external-body", whose value is a hexadecimal encoding of the hash.

For example:

```
        Content-Type: message/external-body;
            access-type="URL";
            expiration="Mon, 24 June 2002 09:00:00 GMT";
            URL="http://www.example.com/the-indirect-content.au";
            size=52723;
            hash=10AB568E91245681AC1B
        <CRLF>
        Content-Disposition: render
```

5.13. Supplying Additional Comments about the Indirect Content

   One MAY use the Content-Description entity header to provide
   optional, freeform text to comment on the indirect content. This
   text MAY be displayed to the end user but MUST NOT used by other
   elements to determine the disposition of the body.

   For example:

```
        Content-Type: message/external-body;
            access-type="URL";
            expiration="Mon, 24 June 2002 09:00:00 GMT";
            URL="http://www.example.com/the-indirect-content";
            size=52723
        <CRLF>
        Content-Description: Multicast gaming session
        Content-Disposition: render
```

5.14. Relationship to Call-Info, Error-Info, and Alert-Info Headers

   SIP [9] defines three headers that supply additional information with
   regard to a session, a particular error response, or alerting. All
   three of these headers allow the UAC or UAS to indicate additional
   information through a URI. They may be considered a form of content
   indirection. The content indirection mechanism defined in this
   document is not intended as a replacement for these headers. Rather,
   the headers defined in SIP MUST be used in preference to this
   mechanism, where applicable, because of the well-defined semantics of
   those headers.

6. Examples

6.1. Single Content Indirection

         INVITE sip:boromir@example.com SIP/2.0
         From: <sip:gandalf@example.net>;tag=347242
         To: <sip:boromir@example.com>
         Call-ID: 3573853342923422@example.net
         CSeq: 2131 INVITE
         Accept: message/external-body application/sdp
         Content-Type: message/external-body;
             ACCESS-TYPE=URL;
             URL="http://www.example.net/party/06/2002/announcement";
             EXPIRATION="Sat, 20 Jun 2002 12:00:00 GMT";
             size=231
         Content-Length: 105

         Content-Type: application/sdp
         Content-Disposition: session
         Content-ID: <4e5562cd1214427d@example.net>

6.2. Multipart MIME with Content Indirection

         MESSAGE sip:boromir@example.com SIP/2.0
         From: <sip:gandalf@example.net>;tag=34589882
         To: <sip:boromir@example.com>
         Call-ID: 9242892442211117@example.net
         CSeq: 388 MESSAGE
         Accept: message/external-body, text/html, text/plain,
                 image/*, text/x-emoticon
         MIME-Version: 1.0
         Content-Type: multipart/mixed; boundary=zz993453

         --zz993453
         Content-Type: message/external-body;
             access-type="URL";
             expiration="Mon, 24 June 2002 09:00:00 GMT";
             URL="http://www.example.net/company_picnic/image1.png";
             size=234422

         Content-Type: image/png
         Content-ID: <9535035333@example.net>
         Content-Disposition: render
         Content-Description: Kevin getting dunked in the wading pool

         --zz993453

```
            Content-Type: message/external-body;
                 access-type="URL";
                 expiration="Mon, 24 June 2002 09:00:00 GMT";
                 URL="http://www.example.net/company_picnic/image2.png";
                 size=233811

            Content-Type: image/png
            Content-ID: <1134299224244@example.net>
            Content-Disposition: render
            Content-Description: Peter on his tricycle

            --zz993453--
```

7. Security Considerations

   Any content indirection mechanism introduces additional security
   concerns. By its nature, content indirection requires an extra
   processing step and information transfer. There are a number of
   potential abuses of a content indirection mechanism:

   o Content indirection allows the initiator to choose an alternative
      protocol with weaker security or known vulnerabilities for the
      content transfer (for example, asking the recipient to issue an
      HTTP request that results in a Basic authentication challenge).

   o Content indirection allows the initiator to ask the recipient to
      consume additional resources in the information transfer and
      content processing, potentially creating an avenue for denial-of-
      service attacks (for example, an active FTP URL consuming 2
      connections for every indirect content message).

   o Content indirection could be used as a form of port-scanning
      attack where the indirect content URL is actually a bogus URL
      pointing to an internal resource of the recipient. The response
      to the content indirection request could reveal information about
      open (and vulnerable) ports on these internal resources.

   o A content indirection URL can disclose sensitive information about
      the initiator such as an internal user name (as part of an HTTP
      URL) or possibly geolocation information.

   Fortunately, all of these potential threats can be mitigated through
   careful screening of both the indirect content URIs that are received
   and those that are sent. Integrity and confidentiality protection of
   the indirect content URI can prevent additional attacks as well.

   For confidentiality, integrity, and authentication, this content
   indirection mechanism relies on the security mechanisms outlined in

RFC 3261. In particular, the usage of S/MIME as defined in section
23 of RFC 3261 provides the necessary mechanism to ensure integrity,
protection, and confidentiality of the indirect content URI and
associated parameters.

Securing the transfer of the indirect content is the responsibility
of the underlying protocol used for this transfer. If HTTP is used,
applications implementing this content indirection method SHOULD
support the HTTPS URI scheme for secure transfer of content and MUST
support the upgrading of connections to TLS, by using starttls. Note
that a failure to complete HTTPS or starttls (for example, due to
certificate or encryption mismatch) after having accepted the
indirect content in the SIP request is not the same as rejecting the
SIP request, and it may require additional user-user communication
for correction.

Note that this document does not advocate the use of transitive
trust. That is, just because the UAS receives a URI from a UAC that
the UAS trusts, the UAS SHOULD NOT implicitly trust the object
referred to by the URI without establishing its own trust
relationship with the URI provider.

Access control to the content referenced by the URI is not defined by
this specification. Access control mechanisms may be defined by the
protocol for the scheme of the indirect content URI.

If the UAC knows the content in advance, the UAC SHOULD include a
hash parameter in the content indirection. The hash parameter is a
hexadecimal-encoded SHA-1 [8] hash of the indirect content. If a
hash value is included, the recipient MUST check the indirect content
against that hash and indicate any mismatch to the user.

In addition, if the hash parameter is included and the target URI
involves setting up a security context using certificates, the UAS
MUST ignore the results of the certificate validation procedure, and
instead verify that the hash of the (canonicalized) content received
matches the hash presented in the content-indirection hash parameter.

If the hash parameter is NOT included, the sender SHOULD use only
schemes that offer message integrity (such as https:). When the hash
parameter is not included and security using certificates is used,
the UAS MUST verify any server certificates, by using the UAS's list
of trusted top-level certificate authorities.

If hashing of indirect content is not used, the content returned to
the recipient by exercise of the indirection might have been altered
from that intended by the sender.

8. Contributions

    Sean Olson, seanol@microsoft.com, provided the vast majority of the
    content of this document, including editorship through the first IESG
    review. Dean Willis touched it next.

    Eric Burger edited the document and addressed IESG comments,
    including the access protocol negotiation mechanism.

9. Acknowledgements

    Cullen Jennings and Nancy Greene provided a through review and
    valuable comments and suggestions.

10. References

10.1. Normative References

    [1]   Freed, N. and K. Moore, "Definition of the URL MIME External-
           Body Access-Type", RFC 2017, October 1996.

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

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

    [4]   Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T.
           Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC
           2068, January 1997.

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

    [6]   Daniel, R., "A Trivial Convention for using HTTP in URN
           Resolution", RFC 2169, June 1997.

    [7]   Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform
           Resource Identifiers (URI): Generic Syntax", STD 66, RFC 3986,
           January 2005.

    [8]   Eastlake, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)",
           RFC 3174, September 2001.

   [9]   Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A.,
         Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP:
         Session Initiation Protocol", RFC 3261, June 2002.

   [10] Burger, E., "Critical Content Multi-purpose Internet Mail
         Extensions (MIME) Parameter", RFC 3459, January 2003.

   [11] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating
         User Agent Capabilities in the Session Initiation Protocol
         (SIP)", RFC 3840, August 2004.

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

10.2. Informative Reference

   [13] Levinson, E., "The MIME Multipart/Related Content-type", RFC
         2387, August 1998.

Author's Address

   Eric Burger (editor)
   Cantata Technolgy, Inc.

   EMail: eburger@cantata.com
   URI: http://www.cantata.com

Full Copyright Statement

Intellectual Property

Acknowledgement