IMAP4 Binary Content Extension

Status of this memo

This document is an Internet Draft and is in full conformance with all provisions of Section 10 of RFC 2026.

Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

A revised version of this draft document will be submitted to the RFC editor as a Proposed Standard for the Internet Community. Discussion and suggestions for improvement are requested. Distribution of this draft is unlimited.

1. Abstract

This memo defines the BINARY extension to the Internet Message Access Protocol [IMAP4rev1]. It provides a mechanism for IMAP4 clients and servers to exchange message body data without using a MIME content-transfer-encoding.

2. Conventions Used in this Document

The key words "MUST," "MUST NOT," "SHOULD," "SHOULD NOT," and "MAY" in this document are to be interpreted as described in [KEYWORD].

The abbreviation "CTE" means content-transfer-encoding.
3. Overview

The MIME extensions to Internet messaging allow for the transmission of non-textual (binary) message content [MIME-IMB]. Since the traditional transports for messaging are not always capable of passing binary data transparently, MIME provides encoding schemes that allow binary content to be transmitted over transports that are not otherwise able to do so.

The overhead of MIME encoding this content can be considerable in some contexts (e.g. slow radio links, streaming multimedia). Reducing the overhead associated with CTE schemes such as base64 can give a noticeable reduction in resource consumption. The BINARY extension lets the server perform CTE decoding prior to transmitting message data to the client.

4. Content-Transfer-Encoding Considerations

Every IMAP4 body section has a MIME content-transfer-encoding. (Those without an explicit Content-Transfer-Encoding header are implicitly labeled as "7bit" content.) In the terminology of [MIME-IMB], the CTE specifies both a decoding algorithm and the domain of the decoded data. In the context of this memo, "decoding" refers to the CTE decoding step described in [MIME-IMB].

Certain CTEs use an identity encoding transformation. For these CTEs there is no decoding required, however the domain of the underlying data may not be expressable in the IMAP4 protocol (e.g. MIME "binary" content containing NUL octets). To accommodate these cases the BINARY extension introduces a new type of literal protocol element that is fully 8bit transparent.

Thus, server processing of the FETCH BINARY command involves two logical steps:

1) perform any CTE-related decoding
2) determine the domain of the decoded data

Step 2 is necessary to determine which protocol element should be used to transmit the decoded data. (See FETCH Response Extensions for further details.)

5. Framework for the IMAP4 Binary Extension

This memo defines the following extensions to [IMAP4rev1].

5.1. CAPABILITY Identification

IMAP4 servers that support this extension MUST include "BINARY" in the response list to the CAPABILITY command.
5.2. FETCH Command Extensions

This extension defines three new FETCH command data items.

BINARY[<section>]<partial>>
Requests that the specified section be transmitted after performing CTE-related decoding.

When performing a partial FETCH, the offset argument refers to the offset into the DECODED section.

BINARY.PEEK[<section>]<partial>>
An alternate form of BINARY[<section>] that does not implicitly set the \Seen flag.

BINARY.SIZE[<section>]
Requests the decoded size of the section (i.e. the size to expect in response to the corresponding FETCH BINARY request).

Note: client authors are cautioned that this may be an expensive operation for some server implementations. Needlessly issuing this request could result in degraded performance due to servers having to calculate the value every time the request is issued.

5.3. FETCH Response Extensions

This extension defines two new FETCH response data items.

BINARY[<section>]<origin_octet>>
A <string> or <literal8> expressing the content of the specified section after removing any CTE-related encoding.

If the domain of the decoded data is "8bit" and the data does not contain the NUL character, the server SHOULD return the data in a <string> instead of a <literal8>; this allows the client to determine if the "8bit" data contains the NUL character without having to explicitly scan the data stream for NULs.

If the server does not know how to decode the section’s CTE, it MUST fail the request and issue a "NO" response that contains the "UNKNOWN-CTE" extended response code.

BINARY.SIZE[<section>]
The size of the section after removing any CTE-related encoding. The value returned MUST match the <literal> or <literal8> size that will be returned by the corresponding FETCH BINARY request.

If the server does not know how to decode the section’s CTE, it MUST fail the request and issue a "NO" response that contains the "UNKNOWN-CTE" extended response code.
5.4. APPEND Command Extensions

The APPEND command is extended to allow the client to append data containing NULs by using the <literal8> syntax. The server MAY modify the CTE of the appended data, however any such transformation MUST NOT result in a loss of data.

If the specified mailbox does not support the storage of binary content, the server MUST fail the request and issue a "NO" response that contains the "UNKNOWN-CTE" extended response code.

6. MIME Encoded Headers

[MIME-MHE] defines an encoding that allows for non-US-ASCII text in message headers. This encoding is not the same as the content-transfer-encoding applied to message bodies, and the decoding transformations described in this memo do not apply to [MIME-MHE] encoded header text. A server MUST NOT perform any conversion of [MIME-MHE] encoded header text in response to any binary FETCH or APPEND request.

7. Implementation Considerations

Messaging clients and servers have been notoriously lax in their adherence to the Internet CRLF convention for terminating lines of textual data in Internet protocols. When sending data using the BINARY extension, servers MUST ensure that textual line-oriented sections are always transmitted using the IMAP4 CRLF line termination syntax, regardless of the underlying storage representation of the data on the server.

A server may choose to store message body binary content in a non-encoded format. Regardless of the internal storage representation used, the server MUST issue BODYSTRUCTURE responses that describe the message as though the binary-encoded sections are encoded in a CTE acceptable to the IMAP4 base specification (see [IMAP4rev1] section 4.3.1). Furthermore, the results of a FETCH BODY MUST return the message body content in the format described by the corresponding FETCH BODYSTRUCTURE response.

While the server is allowed to modify the CTE of APPENDED <literal8> data, this should only be done when it is absolutely necessary. Gratuitous encoding changes will render useless most cryptographic operations that have been performed on the message.

This extension provides an optimization that is useful in certain specific situations. It does not absolve clients from providing basic functionality (content transfer decoding) that should be available in all messaging clients. Clients supporting this extension SHOULD be prepared to perform their own CTE decoding operations.
8. Formal Protocol Syntax

The following syntax specification uses the augmented Backus-Naur Form (ABNF) notation as used in [ABNF], and incorporates by reference the Core Rules defined in that document.

This syntax augments the grammar specified in [IMAP4rev1].

append =/ "APPEND" SP mailbox [SP flag_list] [SP date_time] SP literal8

fetch_att =/ "BINARY" [".PEEK"] section_number ["<" number "." nz_number ">"] / "BINARY.SIZE" section_number

literal8 = "~{" number "}" CRLF *OCTET
; <number> represents the number of OCTETs in the response string.

msg_att = "{" 1#("ENVELOPE" SP envelope / "FLAGS" SP "(" #(flag / "\Recent") ")" / "INTERNALDATE" SP date_time / "RFC822" [".HEADER" / ".TEXT"] SP nstring / "RFC822.SIZE" SP number / "BODY" ["STRUCTURE"] SP body / "BODY" section ["<" number ">"] SP nstring / "UID" SP uniqueid / "BINARY" section_number ["<" number ">"] SP (nstring / literal8) / "BINARY.SIZE" section_number SP number) ")"

;; Append BINARY and BINARY.SIZE to <msg_att>.
;; This is not meant to be an absolute list;
;; it was done this way due to ABNF syntax restrictions.

resp_code_text =/ "UNKNOWN-CTE"

section_number = (nz_number *"." nz_number)

9. References


[KEYWORD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 9, RFC2119, March 1997


10. Security Considerations

There are no known security issues with this extension.

11. Authors’ Address

Lyndon Nerenberg
ACI Worldwide
Suite 900
10117 - Jasper Avenue
Edmonton, Alberta
Canada T5J 1W8

Email: lyndon@atg.aciworldwide.com