Network Working Group
Request for Comments: 3516
Category: Standards Track

L. Nerenberg
Orthanc Systems

April 2003

IMAP4 Binary Content Extension

Status of this Memo

Copyright Notice

Abstract

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

1. 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.

2. Introduction

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.

3. 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 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 expressible 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 eight bit 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.)

4. Framework for the IMAP4 Binary Extension

This memo defines the following extensions to [IMAP4rev1].

4.1. CAPABILITY Identification

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

4.2. FETCH Command Extensions

This extension defines three new FETCH command data items.

BINARY<section-binary>[<partial>]

Requests that the specified section be transmitted after performing CTE-related decoding.

The <partial> argument, if present, requests that a subset of the data be returned. The semantics of a partial FETCH BINARY command are the same as for a partial FETCH BODY command, with the exception that the <partial> arguments refer to the DECODED section data.

BINARY.PEEK<section-binary>[<partial>]

An alternate form of FETCH BINARY that does not implicitly set the \Seen flag.

BINARY.SIZE<section-binary>

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 might 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.

4.3. FETCH Response Extensions

This extension defines two new FETCH response data items.

BINARY<section-binary>[<<number>>]

An <nstring> or <literal8> expressing the content of the specified section after removing any CTE-related encoding. If <number> is present it refers to the offset within the DECODED section data.

If the domain of the decoded data is "8bit" and the data does not contain the NUL octet, 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 octet without having to explicitly scan the data stream for 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-binary>

The size of the section after removing any CTE-related encoding. The value returned MUST match the size of the <nstring> or <literal8> 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.

4.4. 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 destination 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.

5. 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.

6. 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. 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.

7. 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-binary [partial]
                    / "BINARY.SIZE" section-binary

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

msg-att-static =/  "BINARY" section-binary SP (nstring / literal8)
                    / "BINARY.SIZE" section-binary SP number

partial         =   "<" number "." nz-number ">"

resp-text-code =/  "UNKNOWN-CTE"

section-binary =   "[" [section-part] "]"
```

8. Normative References

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

[IMAP4rev1] Crispin, M., "Internet Message Access Protocol Version 4rev1", RFC 3501, March 2003.

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

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

[MIME-MHE] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

9. Security Considerations

There are no known additional security issues with this extension beyond those described in the base protocol described in [IMAP4rev1].

10. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

11. Author's Address

Lyndon Nerenberg
Orthanc Systems
1606 - 10770 Winterburn Road
Edmonton, Alberta
Canada T5S 1T6

EMail: lyndon@orthanc.ab.ca

12. Full Copyright Statement

Acknowledgement

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