

Network Working Group
Internet Draft: IMAP4 Channel Transport Mechanism
Document: [draft-nerenberg-imap-channel-02.txt](#)

S. Hole
L. Nerenberg
ACI Worldwide
B. Leiba
IBM Research
June 2002

IMAP4 Channel Transport Mechanism

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/lid-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.

[@](#). Administrivia

Discussion concerning this draft should be directed to the <ietf-imap-voice@imc.org> mailing list. (To subscribe: echo subscribe | mail ietf-imap-voice-request@imc.org)

Changes in -02:

Changed <channel-set> to use <section-spec> instead of <section-text>. This allows retrieval of headers and MIME structure.

<channel-data> returns a <section-spec>, not <nz-number> (to match <channel> syntax).

Add missing SP tokens to grammar.

Grammar fix to allow foo: as a valid URI in a request.

Add UID CHANNEL.

Clarify response when client issues a command with an unsupported scheme.

Add section on command sequencing.

Note arbitrary ordering of untagged responses.

Replace URI-reference with absoluteURI. The IMAP server can't maintain the state required to deal with relative URIs. This also solves an ambiguity between parsing "NIL" as <nil> or as a relative URI.

Outstanding Issues

Responses encode the URL as an <absoluteURI>. Does the syntax of <absoluteURI> conflict with the base IMAP grammar? There are enough punctuation characters available in a URL to put a protocol parser into an intractable state. Someone (besides the draft authors) needs to verify there are no conflicts between <absoluteURI> and the rest of IMAP.

Security considerations needs to be written.

1. Abstract

IMAP4 is being used to serve rich media content in environments that extend beyond traditional text-based e-mail. One example is a cellular telephone that can retrieve and send MIME-encoded audio data through IMAP4. While this type of content can be exchanged natively using IMAP4, some applications will work better if the message content can be manipulated using schemes external to the

IMAP4 connection. In our cellular telephone example, it might be preferable for the telephone client to retrieve the audio data using RTSP. This specifications defines a mechanism for an IMAP4 client to request message content from a server through an external scheme.

[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](#)].

In examples, "C:" and "S:" preface lines sent by the client and the server respectively.

The examples in this document do NOT form part of the specification. Where conflicts exist between the text and the formal grammar, the grammar is authoritative.

[3.](#) Protocol Framework

This memo defines the following extensions to [[IMAP4rev1](#)].

[3.1.](#) CAPABILITY Identification

IMAP4 servers that support this extension MUST include a CHANNEL capability response in the response list to the CAPABILITY command. This entry indicates the server supports the extension, and lists the schemes available to the CHANNEL command. The capability response consists of the string "CHANNEL=" followed by a list of schemes supported by the CHANNEL extension.

Example:

```
* CAPABILITY IMAP4rev1 AUTH=DIGEST-MD5 CHANNEL=imap,ftp
```

[3.2.](#) CHANNEL Command

The CHANNEL command requests that message data be retrieved through an external scheme. Clients may issue a partially-qualified URI, in which case the server will determine the final connection end-point. What constitutes a partially-qualified URI is implemen-

tation defined.

The syntax of the CHANNEL command is:

```
tag CHANNEL channel-uri-list channel-set
```

<channel-uri-list> is a list of URIs or schemes specifying how the client is willing to retrieve the message data. If <channel-uri-list> contains more than one element the server must enumerate the list in order and SHOULD return the message data via the first item in the list it is capable of using.

<channel-set> is a list of message-number/body-section pairs describing the content to be retrieved. The message number specifies the sequence number of the message to act on, or in the case of a UID CHANNEL command, the UID of the message.

Example:

```
C: 0 CHANNEL (rtsp: imap:) (1 2)(3 1)(3 9.1)
```

asks for [section 2](#) of message 1 and sections [1](#) and [9.1](#) of message 3. The preferred retrieval scheme is RTSP. If RTSP isn't available the IMAP scheme should be attempted. In either case the server will fill in the connection end-point information.

[3.3.](#) CHANNEL Response

An untagged CHANNEL response is returned for each message-number/body-section pair specified by the corresponding CHANNEL

command:

```
* message-number CHANNEL section-spec URI
```

The ordering of these responses is arbitrary. The message number and <section-spec> in the response mirror those in the corresponding request, therefore responses to UID CHANNEL commands report the message UID rather than the message sequence number.

Example:

The responses to the example command in the previous section might look like:

```
S: * 1 CHANNEL 2 rtsp://frobozz.example.com/144124
S: * 3 CHANNEL 1
      imap://user@example.com:/inbox;uidvalidity=2/;uid=33
S: * 3 CHANNEL 9.1 NIL
S: 0 OK done
```

The NIL response to the [section 9.1](#) request indicates that the part could not be retrieved via either of the requested schemes. This could be caused by the inability to convert or represent the content via the requested schemes, or because a resource was unavailable.

The server MUST NOT issue an untagged CHANNEL response containing a URL until such time as that URL is available for the client to dereference. The lifetime of the URL is implementation defined.

If any one of the schemes in the <channel-uri-list> does not match one of the schemes listed in the server channel capability list the server: 1) MUST NOT execute any part of the command, 2) MUST NOT return any untagged responses to the command, and 3) MUST issue only a tagged BAD completion response.

[3.4.](#) Command Sequencing

Since there is no way to distinguish between responses to CHANNEL and UID CHANNEL, clients MUST NOT issue a UID CHANNEL command while a CHANNEL command is in progress. Conversely, clients MUST NOT issue a CHANNEL command while a UID CHANNEL command is in progress. These restrictions are in addition to the normal sequencing rules that apply to UID-style commands.

[4.](#) Formal Protocol Syntax

The following syntax specification uses the augmented Backus-Naur Form (ABNF) notation as defined in [\[ABNF\]](#), and incorporates by reference the Core Rules from that document. This syntax extends the grammar specified in [\[IMAP4rev1\]](#).

The following tokens are incorporated from [\[URI\]](#): absoluteURI, scheme.

capability =/ "CHANNEL=" scheme *("," scheme)
 channel = ["UID" SP] "CHANNEL" SP channel-uri-list
 SP channel-set
 channel-data = "CHANNEL" SP section-spec SP
 (absoluteURI / nil)
 channel-set = 1*("(" nz-number SP section-spec ")")
 channel-uri-list = "(" channel-uri-reference
 1*(SP channel-uri-reference) ")"
 channel-uri-reference = absoluteURI / scheme ":"
 command-select =/ channel
 response-data = "*" SP (resp-cond-state / resp-cond-by /
 mailbox-data / message-data /
 capability-data / channel-data) CRLF
 ; adds <channel-data> to <response-data>

5. References

- [ABNF] Crocker, D., P. Overell, "Augmented BNF for Syntax Specifications: ABNF." [RFC2234](#), November 1997
 [IMAP4rev1] Crispin, M., "Internet Message Access Protocol - Version 4rev1," Work in Progress
 [KEYWORD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," [BCP 9](#), [RFC2119](#), March 1997
 [URI] Berners-Lee, T., et al, "Uniform Resource Identifiers (URI): Generic Syntax," [RFC2396](#), August 1998

6. Security Considerations

>>> TBD <<<

[7.](#) Authors' Addresses

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

<lyndon@atg.aciworldwide.com>
Phone: +1 780 424 4922

Steve Hole
ACI Worldwide
Suite 900
10117 - Jasper Avenue
Edmonton, Alberta
Canada T5J 1W8

<steve.hole@messagingdirect.com>
Phone: +1 780 424 4922

Barry Leiba
IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
<leiba@watson.ibm.com>
Phone: +1 914 784 7941

