

Network Working Group  
Internet Draft: IMAP4 Channel Transport Mechanism  
Document: [draft-nerenberg-imap-channel-01.txt](#)

S. Hole  
L. Nerenberg  
ACI Worldwide  
B. Leiba  
IBM Research  
November 2001

## 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

Lines prefixed with ">>>" are meta-comments and will not appear in the final document.

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)

### [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.

## [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 comma-seperated list of schemes supported by the CHANNEL extension.

### [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 implementation defined, however every URI MUST contain at least a scheme.

The syntax of the CHANNEL command is:

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

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

```
>>> the intent is that the client can indicate a list of
>>> services in descending order of usefulness/quality.
>>> Also, there is no guarantee that a server can express a
>>> particular body section through all of its advertised
>>> schemes, thus the list provides fallback for the server
>>> as well as the client.
```

channel-set is a list of message body sections to be retrieved

through the specified URI.

```
>>> example syntax:
>>> 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 method is RTSP, however if
>>> RTSP isn't available/usable, try IMAP. In either case,
>>> the server will fill in the connection end-point
>>> information. You cannot ask for .HEADERS or .MIME data
>>> with CHANNEL.
```

### [3.3.](#) CHANNEL Response

The CHANNEL response returns connection status and location information to the client. One untagged response is returned for each body section requested.

```
>>> example response to above command:
>>>
>>> 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 through the schemes,
>>> or because some resource was unavailable.
```

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

#### [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 [\[URI\]](#): scheme, URI-reference.

```
capability      =/ "CHANNEL=" scheme *["," scheme]
channel         =  "CHANNEL" SP uri-list SP channel-set
channel-data    =  "CHANNEL" nz-number (URI-reference / nil)
channel-set     =  1*( "(" nz-number SP section-part ")" )
```

```
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 IMAP4rev1
                  ; <response-data>
uri-list        =  "(" URI-reference *[SP URI-reference] ")"
```

#### [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

## [7.](#) Authors' Addresses

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

<lyndon@atg.aciworldwide.com>

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

<steve.hole@messagingdirect.com>

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