Network Working Group
Request for Comments: 2062
Category: Informational

M. Crispin
University of Washington

December 1996

Internet Message Access Protocol - Obsolete Syntax

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document describes obsolete syntax which may be encountered by IMAP4 implementations which deal with older versions of the Internet Mail Access Protocol. IMAP4 implementations MAY implement this syntax in order to maximize interoperability with older implementations.

This document repeats information from earlier documents, most notably RFC 1176 and RFC 1730.

Obsolete Commands and Fetch Data Items

The following commands are OBSOLETE. It is NOT required to support any of these commands or fetch data items in new server implementations. These commands are documented here for the benefit of implementors who may wish to support them for compatibility with old client implementations.

The section headings of these commands are intended to correspond with where they would be located in the main document if they were not obsoleted.

6.3.OBS.1. FIND ALL.MAILBOXES Command

Arguments: mailbox name with possible wildcards

Data: untagged responses: MAILBOX

Result: OK - find completed
NO - find failure: can't list that name
BAD - command unknown or arguments invalid

The FIND ALL.MAILBOXES command returns a subset of names from the complete set of all names available to the user. It returns zero or more untagged MAILBOX replies. The mailbox argument to FIND ALL.MAILBOXES is similar to that for LIST with an empty reference, except that the characters "%" and "?" match a single character.

Example: C: A002 FIND ALL.MAILBOXES *
S: * MAILBOX blurdybloop
S: * MAILBOX INBOX
S: A002 OK FIND ALL.MAILBOXES completed

6.3.OBS.2. FIND MAILBOXES Command

Arguments: mailbox name with possible wildcards

Data: untagged responses: MAILBOX

Result: OK - find completed
NO - find failure: can't list that name
BAD - command unknown or arguments invalid

The FIND MAILBOXES command returns a subset of names from the set of names that the user has declared as being "active" or "subscribed". It returns zero or more untagged MAILBOX replies. The mailbox argument to FIND MAILBOXES is similar to that for LSUB with an empty reference, except that the characters "%" and "?" match a single character.

Example: C: A002 FIND MAILBOXES *
S: * MAILBOX blurdybloop
S: * MAILBOX INBOX
S: A002 OK FIND MAILBOXES completed

6.3.OBS.3. SUBSCRIBE MAILBOX Command

Arguments: mailbox name

Data: no specific data for this command

Result: OK - subscribe completed
NO - subscribe failure: can't subscribe to that name
BAD - command unknown or arguments invalid

The SUBSCRIBE MAILBOX command is identical in effect to the SUBSCRIBE command. A server which implements this command must be able to distinguish between a SUBSCRIBE MAILBOX command and a SUBSCRIBE command with a mailbox name argument of "MAILBOX".

```
   Example:    C: A002 SUBSCRIBE MAILBOX #news.comp.mail.mime
               S: A002 OK SUBSCRIBE MAILBOX to #news.comp.mail.mime
               completed
               C: A003 SUBSCRIBE MAILBOX
               S: A003 OK SUBSCRIBE to MAILBOX completed
```

6.3.OBS.4. UNSUBSCRIBE MAILBOX Command

```
   Arguments:  mailbox name

   Data:       no specific data for this command

   Result:     OK - unsubscribe completed
               NO - unsubscribe failure: can't unsubscribe that name
               BAD - command unknown or arguments invalid
```

The UNSUBSCRIBE MAILBOX command is identical in effect to the UNSUBSCRIBE command. A server which implements this command must be able to distinguish between a UNSUBSCRIBE MAILBOX command and an UNSUBSCRIBE command with a mailbox name argument of "MAILBOX".

```
   Example:    C: A002 UNSUBSCRIBE MAILBOX #news.comp.mail.mime
               S: A002 OK UNSUBSCRIBE MAILBOX from #news.comp.mail.mime
               completed
               C: A003 UNSUBSCRIBE MAILBOX
               S: A003 OK UNSUBSCRIBE from MAILBOX completed
```

6.4.OBS.1 PARTIAL Command

```
   Arguments:  message sequence number
               message data item name
               position of first octet
               number of octets

   Data:       untagged responses: FETCH

   Result:     OK - partial completed
               NO - partial error: can't fetch that data
               BAD - command unknown or arguments invalid
```

The PARTIAL command is equivalent to the associated FETCH command, with the added functionality that only the specified number of octets, beginning at the specified starting octet, are returned. Only a single message can be fetched at a time. The first octet of a message, and hence the minimum for the starting octet, is octet 1.

The following FETCH items are valid data for PARTIAL: RFC822, RFC822.HEADER, RFC822.TEXT, BODY[<section>], as well as any .PEEK forms of these.

Any partial fetch that attempts to read beyond the end of the text is truncated as appropriate. If the starting octet is beyond the end of the text, an empty string is returned.

The data are returned with the FETCH response. There is no indication of the range of the partial data in this response. It is not possible to stream multiple PARTIAL commands of the same data item without processing and synchronizing at each step, since streamed commands may be executed out of order.

There is no requirement that partial fetches follow any sequence. For example, if a partial fetch of octets 1 through 10000 breaks in an awkward place for BASE64 decoding, it is permitted to continue with a partial fetch of 9987 through 19987, etc.

The handling of the \Seen flag is the same as in the associated FETCH command.

```
Example:    C: A005 PARTIAL 4 RFC822 1 1024
            S: * 1 FETCH (RFC822 {1024}
            S: Return-Path: <gray@cac.washington.edu>
            S: ...
            S: .........  FLAGS (\Seen))
            S: A005 OK PARTIAL completed
```

6.4.5.OBS.1 Obsolete FETCH Data Items

The following FETCH data items are obsolete:

BODY[<...>0] A body part number of 0 is the [RFC-822] header of the message. BODY[0] is functionally equivalent to BODY[HEADER], differing in the syntax of the resulting untagged FETCH data (BODY[0] is returned).

RFC822.HEADER.LINES <header_list>
Functionally equivalent to BODY.PEEK[HEADER.LINES <header_list>], differing in the syntax of the resulting untagged FETCH data (RFC822.HEADER is returned).

RFC822.HEADER.LINES.NOT <header_list>
Functionally equivalent to BODY.PEEK[HEADER.LINES.NOT <header_list>], differing in the syntax of the resulting untagged FETCH data (RFC822.HEADER is returned).

RFC822.PEEK Functionally equivalent to BODY.PEEK[], except for the syntax of the resulting untagged FETCH data (RFC822 is returned).

RFC822.TEXT.PEEK
Functionally equivalent to BODY.PEEK[TEXT], except for the syntax of the resulting untagged FETCH data (RFC822.TEXT is returned).

Obsolete Responses

The following responses are OBSOLETE. Except as noted below, these responses MUST NOT be transmitted by new server implementations. Client implementations SHOULD accept these responses.

The section headings of these responses are intended to correspond with where they would be located in the main document if they were not obsoleted.

7.2.OBS.1. MAILBOX Response

Data: name

The MAILBOX response MUST NOT be transmitted by server implementations except in response to the obsolete FIND MAILBOXES and FIND ALL.MAILBOXES commands. Client implementations that do not use these commands MAY ignore this response. It is documented here for the benefit of implementors who may wish to support it for compatibility with old client implementations.

This response occurs as a result of the FIND MAILBOXES and FIND ALL.MAILBOXES commands. It returns a single name that matches the FIND specification. There are no attributes or hierarchy delimiter.

Example: S: * MAILBOX blurdybloop

7.3.OBS.1. COPY Response

Data: none

The COPY response MUST NOT be transmitted by new server implementations. Client implementations MUST ignore the COPY response. It is documented here for the benefit of client implementors who may encounter this response from old server implementations.

In some experimental versions of this protocol, this response was returned in response to a COPY command to indicate on a per-message basis that the message was copied successfully.

Example: S: * 44 COPY

7.3.OBS.2. STORE Response

Data: message data

The STORE response MUST NOT be transmitted by new server implementations. Client implementations MUST treat the STORE response as equivalent to the FETCH response. It is documented here for the benefit of client implementors who may encounter this response from old server implementations.

In some experimental versions of this protocol, this response was returned instead of FETCH in response to a STORE command to report the new value of the flags.

Example: S: * 69 STORE (FLAGS (\Deleted))

Formal Syntax of Obsolete Commands and Responses

Each obsolete syntax rule that is suffixed with "_old" is added to the corresponding name in the formal syntax. For example, command_auth_old adds the FIND command to command_auth.

```
command_auth_old ::= find

command_select_old
                 ::= partial

date_year_old    ::= 2digit
                     ;; (year - 1900)

date_time_old    ::= <"> date_day_fixed "-" date_month "-" date_year
                     SPACE time "-" zone_name <">
```

```
   find            ::= "FIND" SPACE ["ALL."] "MAILBOXES" SPACE
                       list_mailbox

   fetch_att_old   ::= "RFC822.HEADER.LINES" [".NOT"] SPACE header_list /
                       fetch_text_old

   fetch_text_old  ::= "BODY" [".PEEK"] section_old /
                       "RFC822" [".HEADER" / ".TEXT" [".PEEK"]]

   msg_data_old    ::= "COPY" / ("STORE" SPACE msg_att)

   partial         ::= "PARTIAL" SPACE nz_number SPACE fetch_text_old SPACE
                       number SPACE number

   section_old     ::= "[" (number ["." number]) "]"

   subscribe_old   ::= "SUBSCRIBE" SPACE "MAILBOX" SPACE mailbox

   unsubscribe_old ::= "UNSUBSCRIBE" SPACE "MAILBOX" SPACE mailbox

   zone_name       ::= "UT" / "GMT" / "Z" /                ;; +0000
                       "AST" / "EDT" /                     ;; -0400
                       "EST" / "CDT" /                     ;; -0500
                       "CST" / "MDT" /                     ;; -0600
                       "MST" / "PDT" /                     ;; -0700
                       "PST" / "YDT" /                     ;; -0800
                       "YST" / "HDT" /                     ;; -0900
                       "HST" / "BDT" /                     ;; -1000
                       "BST" /                             ;; -1100
                       "A" / "B" / "C" / "D" / "E" / "F" / ;; +1 to +6
                       "G" / "H" / "I" / "K" / "L" / "M" / ;; +7 to +12
                       "N" / "O" / "P" / "Q" / "R" / "S" / ;; -1 to -6
                       "T" / "U" / "V" / "W" / "X" / "Y"   ;; -7 to -12
```

Security Considerations

Security issues are not discussed in this memo.

Author's Address

Mark R. Crispin
Networks and Distributed Computing
University of Washington
4545 15th Aveneue NE
Seattle, WA 98105-4527

Phone: (206) 543-5762
EMail: MRC@CAC.Washington.EDU