
C

IMAP Commands

This appendix covers the IMAP commands as described in RFC 2060 (IMAP4rev1). Commands may only be valid in certain connection states. We'll list the commands by the state in which they're valid. Server response codes (untagged responses) are not listed here in detail, but a knowledge of the response codes is not required for general evaluation and trouble-shooting of IMAP sessions. If you require detailed knowledge of the response codes, see Section 7.1 of RFC 2060 (<http://www.ietf.org/rfc/rfc2060.txt>).

Commands Valid in Any State

CAPABILITY

The CAPABILITY command returns a list of the server's capabilities (e.g., supported authentication mechanisms and supported IMAP extensions).

Usage

CAPABILITY

Arguments

None

Untagged server response

An untagged CAPABILITY response

Results

OK (command completed successfully)

BAD (invalid arguments or command unknown)

Example

```
A00001 CAPABILITY
* CAPABILITY IMAP4REV1 MAILBOX-REFERRALS LOGIN-REFERRALS AUTH=CRAM-MD5
A00001 OK Completed
```

LOGOUT

The LOGOUT command tells the server to close the network connection.

Usage

LOGOUT

Arguments

None

Untagged server responses

An untagged BYE response

Results

OK (logout completed)
BAD (command failed or invalid arguments)

Example

```
A00003 logout
* BYE localhost IMAP4rev1 server terminating connection
A00003 OK Completed
```

NOOP

The NOOP command, besides resetting the server's inactivity autologout timer, also causes a protocol round-trip, which in turn may cause an announcement of new mail. NOOP is often used to poll the status of a mailbox when the client doesn't otherwise have anything to do.

Usage

NOOP

Arguments

None

Untagged server response

None

Command result

OK (NOOP completed successfully)
BAD (invalid arguments or command unknown)

Example

```
A00002 NOOP
A00002 OK Completed
```

Commands Valid in the Non-authenticated State

AUTHENTICATE

The AUTHENTICATE command authenticates the user to the server using the specified SASL authentication mechanism.

Usage

AUTHENTICATE *mechanism*

Arguments

The authentication mechanism to use

Untagged server responses

Optional server challenge (the server sends the untagged response + and the client replies with a BASE64 encoded response that is specific to the specified authentication mechanism).

Results

OK (successful authentication)
NO (authentication failed)
BAD (command unknown or invalid arguments)

Example

```
A00004 authenticate anonymous
+
A00004 OK Completed
```

LOGIN

The LOGIN command authenticates the user to the server using the plaintext password.

Usage

LOGIN *username password*

Arguments

Username and password

Untagged server responses

None

Results

OK (successful login)

NO (login failed)

BAD (command unknown or invalid arguments)

Example

```

A00005 LOGIN JOHNDOE "XXXXXXXX"
A00005 NO LOGIN failed
A00006 LOGIN JOHNDOE "YYYYYYYY"
A00006 OK Completed

```

Commands Valid in the Authenticated State

APPEND

The APPEND command appends a new message to the end of the specified mailbox. If flags or a date-time string is specified as an argument, the flags and date are set accordingly in the appended message.

Usage

APPEND mailbox (flags) date {message-size}

Arguments

Mailbox name

(Optional) Parenthesized list of flags to set in the appended message

(Optional) Date/time string; the internal data of the appended message will be set to this date/time

Message size (size of the message expressed in RFC 822 octets that will follow)

Untagged server responses

No specific response

Results

OK (successfully appended)

NO (APPEND command failed due to errors in *flags*, *date*, or message text)

BAD (command unknown or invalid arguments)

Example

```

A00159 APPEND tmp (\Seen \Draft) {1259}
+ Ready for argument

```

```

From: "Kevin W. Mullet" <kwm@unt.edu>
To: "'kwm@unt.edu'" <kwm@unt.edu>
Subject: Case HD0000000019310, Request Urgency: Cannot access network
Message-ID: <2767191894.951954964@kwm.unt.edu>
X-Mailer: Mulberry/2.0.0b9 (Win32)
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii; format=flowed
Content-Transfer-Encoding: 7bit

Case HD0000000019310 has been assigned to your group. Short
Description: HD: A problem not listed here. Open the Support Console
on remedy.unt.edu to view the case.
A00159 OK Completed

```

The APPEND command in this sample tells the server to append a message to the *tmp* mailbox with the *\Seen* and *\Draft* flags set, and that a message 1259 octets in size will follow. The client then sends the raw RFC 822 message text.

CREATE

The CREATE command creates a mailbox with the given name. When the mailbox name is given with a trailing hierarchy delimiter, the command creates just a directory and not a mailbox.

Usage

CREATE mailbox

Arguments

Mailbox name

Untagged server responses

None

Results

OK (mailbox was created successfully)
 NO (mailbox could not be created)
 BAD (command unknown or invalid arguments)

Example

```

A00009 CREATE saved-messages
A00009 OK Completed

```

DELETE

The DELETE command permanently removes the specified mailbox.

Usage

DELETE mailbox

Arguments

Mailbox name

Untagged server responses

None

Results

OK (successfully deleted)

NO (delete mailbox failed)

BAD (command unknown or invalid arguments)

Example

```
A00004 DELETE sent-mail.old
A00004 OK Completed
```

EXAMINE

The EXAMINE command is identical to the SELECT command, except that the mailbox is selected in read-only mode.

Usage

EXAMINE mailbox

Arguments

Mailbox name

Untagged server responses

The untagged responses FLAGS, EXISTS, and RECENT are required. The untagged responses UNSEEN and PERMANENTFLAGS are optional.

Results

OK (EXAMINE completed successfully, mailbox is selected)

NO (EXAMINE failed and mailbox is not selected)

BAD (command unknown or invalid arguments)

Example

```
A00008 EXAMINE inbox
* 0 EXISTS
* 0 RECENT
* OK [UIDVALIDITY 951200840] UID validity status
* OK [UIDNEXT 26] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS ()] Permanent flags
A00008 OK Completed
```

The protocol requires the server to send the untagged responses EXISTS and RECENT before returning the OK result. The FLAGS response refers to the flags, mentioned earlier in this appendix, that are defined in the mailbox. EXISTS

specifies the number of messages in the mailbox. RECENT specifies the number of messages that have the *\Recent* flag set (meaning that this is the first session that has observed the message). *\Recent* does not mean “unread.” Not all unread messages are recent, and not all recent messages are unread. New messages are unread *and* recent. Once read, they are still recent in that session. If not read in that session, then they are unread and not recent in subsequent sessions.

Each mailbox has a UIDVALIDITY value. UIDVALIDITY is a fail-safe mechanism to permit IMAP clients to make assumptions about whether the message UID numbers in a given mailbox have changed. If they have, the UIDVALIDITY value is increased by an arbitrary value. This attribute is of particular value to clients in disconnected mode—it lets them determine if the UID numbers for the constituent messages in a given mailbox are trustworthy from one session to the next.

LIST

The LIST command returns a list of mailbox names that match the arguments given.

Usage

LIST *basename mailbox*

Arguments

Base name of the part of namespace in which the *mailbox* lives
Mailbox name (wildcards permitted)

Untagged server responses

One or more untagged LIST responses

Results

OK (LIST command successful)
NO (LIST command failed)
BAD (command unknown or invalid arguments)

Example

```
A00001 LIST "" "~/mail/*"
* LIST (\NoSelect) "/" ~/mail
* LIST (\NoInferiors) "/" ~/mail/saved-mail
* LIST (\NoInferiors) "/" ~/mail/sent-mail
A00001 OK Completed
```

Usage of the LIST command is somewhat server-implementation-specific. If the reference name argument is left empty, then the mailbox name is interpreted in the same way as it is by the SELECT command. In the previous example, taken from a UW IMAP server, mailboxes are selected relative to the user’s home directory by default:

```
A00002 LIST "user.johndoe.sent-mail" ""
* LIST () "." "user.johndoe.sent-mail"
* LIST () "." "user.johndoe.sent-mail.oldmail"
A00002 OK Completed
```

If the reference name argument contains a string, it usually specifies a level of the mailbox hierarchy. In the example above, taken from a Cyrus IMAP server, the reference argument tells the server to list all mailboxes below *user.johndoe.sent-mail* in the hierarchy.

LSUB

The LSUB command returns a list of mailbox names to which the user is subscribed.

Usage

LSUB *basename mailbox*

Arguments

Base name of the part of the namespace in which the *mailbox* lives
Mailbox name (wildcards permitted)

Untagged server responses

One or more LSUB untagged responses

Results

OK (LSUB completed successfully)
NO (LSUB failed)
BAD (command unknown or invalid arguments)

Example

```
A00002 LSUB "~/mail" ""
* LSUB () "/" ~/mail/saved-messages
A00002 OK Completed
```

RENAME

The RENAME command changes the name of the specified mailbox.

Usage

RENAME *oldmailbox newmailbox*

Arguments

Old mailbox name
New mailbox name

Untagged server responses

None

Results

OK (mailbox successfully renamed)
NO (rename failed)
BAD (command unknown or invalid arguments)

Example

```
A00020 RENAME sent-mail sent-mail.old
A00020 OK Completed
```

SELECT

The SELECT command selects the specified mailbox so messages in that mailbox can be accessed. SELECT attempts to acquire read-write access but will succeed even if it can only get read-only access.

Usage

SELECT mailbox

Arguments

Mailbox name

Untagged server responses

The untagged responses FLAGS, EXISTS, and RECENT are required. UNSEEN and PERMANENTFLAGS are optional untagged responses.

Results

OK (SELECT completed successfully)
NO (can't access mailbox or mailbox does not exist)
BAD (command unknown or invalid arguments)

Example

```
A00003 SELECT INBOX
* 14 EXISTS
* 5 RECENT
* OK [UIDVALIDITY 951200840] UID validity status
* OK [UIDNEXT 22] Predicted next UID
* FLAGS (\Answered \Flagged \Deleted \Draft \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Draft \Seen)] Permanent flags
A00003 OK Completed
```

The protocol requires the server to send the untagged responses EXISTS and RECENT before returning the OK result. The FLAGS response refers to the flags, mentioned earlier in this appendix, that are defined in the mailbox. EXISTS speci-

fies the number of messages in the mailbox. RECENT specifies the number of messages that have the *\Recent* flag set (meaning that this is the first session that has observed the message). *\Recent* does not mean “unread.” Not all unread messages are recent, and not all recent messages are unread. New messages are unread *and* recent. Once read, they are still recent in that session. If not read in that session, then they are unread and not recent in subsequent sessions.

Each mailbox has a UIDVALIDITY value. UIDVALIDITY is a fail-safe mechanism to permit IMAP clients to make assumptions about whether the message UID numbers in a given mailbox have changed. If they have, the UIDVALIDITY value is increased by an arbitrary value. This attribute is of particular value to clients in disconnected mode—it lets them determine if the UID numbers for the constituent messages in a given mailbox are trustworthy from one session to the next.

STATUS

The STATUS command returns the requested status data items associated with the specified mailbox. The STATUS command is used to get information about a mailbox other than the currently selected mailbox, without having to select it first. Note that you never need to run a STATUS command on the currently selected mailbox because IMAP sends (and updates) that information automatically for the selected mailbox.

Usage

STATUS mailbox item

Arguments

Mailbox name

Status data item names (see Table C-1)

Untagged server responses

One or more untagged STATUS responses

Results

OK (STATUS successfully completed)

NO (STATUS command failed)

BAD (command unknown or invalid arguments)

Example

```
A00004 STATUS "~/mail/sent-mail" (MESSAGES)
* STATUS ~/mail/test3 (MESSAGES 122)
A00004 OK Completed
```

Table C-1. STATUS Data Items

Item	Meaning
MESSAGES	Number of messages in the mailbox
RECENT	Number of messages that have the \Recent flag set (i.e., new messages)
UIDNEXT	Next UID that will be assigned to a new message in the mailbox
UIDVALIDITY	Unique identifier of the mailbox
UNSEEN	Number of messages that do not have the \Seen flag set (i.e., unread messages)

SUBSCRIBE

The SUBSCRIBE command adds the specified mailbox to the user's list of “subscribed” mailboxes. A “subscribed mailbox” is the IMAP equivalent of a bookmarked URL in a web browser.

Usage

SUBSCRIBE mailbox

Arguments

Mailbox name

Untagged server responses

None

Results

OK (successfully subscribed to mailbox)

NO (subscribe failed)

BAD (command unknown or invalid arguments)

Example

A00032 SUBSCRIBE saved-messages

A00032 OK Completed

UNSUBSCRIBE

The UNSUBSCRIBE command removes the specified mailbox from the user's “subscribed” list of mailboxes.

Usage

UNSUBSCRIBE mailbox

Arguments

Mailbox name

Untagged server responses

None

Results

OK (successfully unsubscribed)

NO (unsubscribe failed)

BAD (command unknown or invalid arguments)

Example

```
A00033 UNSUBSCRIBE saved-messages
A00033 OK Completed
```

Commands Valid in the Selected State

CHECK

The CHECK command performs server- or mailstore-dependent mailbox house-keeping not normally performed as part of each IMAP command (also known as a *checkpoint*).

Usage

CHECK

Arguments

None

Untagged server responses

None

Results

OK (CHECK completed)

BAD (command unknown or invalid arguments)

Example

```
A00002 CHECK
A00002 OK Completed
```

CLOSE

The CLOSE command removes from the currently selected mailbox all messages that have the *\Deleted* flag set and returns the user from the selected state back to the authenticated state. Unlike the EXPUNGE command, no untagged EXPUNGE responses are sent for the removed messages by the CLOSE command.

Usage

CLOSE

Arguments

None

Untagged server responses

None

Results

OK (CLOSE completed successfully)

NO (No mailbox selected: CLOSE command failed)

BAD (command unknown or invalid arguments)

Example

```
A00008 CLOSE
A00008 OK Completed
```

COPY

The COPY command copies the specified messages from the selected mailbox into the specified mailbox.

Usage

COPY message-set mailbox

Arguments

Set of messages to copy

Name of target mailbox

Untagged server responses

None

Results

OK (successfully copied)

NO (COPY command failed)

BAD (command unknown or invalid arguments)

Example

```
A00322 COPY 1:5 saved-messages
A00322 OK Completed
```

EXPUNGE

The EXPUNGE command permanently removes all messages that have the *\Deleted* flag set from the selected mailbox.

Usage

EXPUNGE

Arguments

None

Untagged server responses

One or more untagged EXPUNGE responses

Results

OK (EXPUNGE completed successfully)

NO (EXPUNGE failed)

BAD (command unknown or invalid arguments)

Example

```
A00010 EXPUNGE
* 1 EXPUNGE
* 4 EXPUNGE
* 0 EXISTS
* 0 RECENT
A00010 OK Expunged 2 messages
```

FETCH

The FETCH command returns message data belonging to a message or set of messages. The data items defined for the FETCH command are shown in Table C-2.

Usage

FETCH message-set data-items

Arguments

Set of messages

Names of message data items to retrieve

Untagged server responses

One or more untagged FETCH responses

Results

OK (FETCH completed successfully)

NO (FETCH command failed)

BAD (command unknown or invalid arguments)

Example

```
A00003 FETCH 2:4 (FLAGS)
* 2 FETCH (FLAGS (\Flagged))
* 3 FETCH (FLAGS (\Flagged \Seen))
* 4 FETCH (FLAGS ())
A00003 OK Completed
```

Table C-2. FETCH Data Items

Data Item	Meaning
ALL	Shortcut for specifying the data items FLAGS, ENVELOPE, INTERNALDATE, and RFC 822.SIZE
BODY	Older form of BODYSTRUCTURE that was not extensible
BODY [section] (MIME part specifier) <partial>	Text contained in <i>section</i> of the message body. <partial>, if specified, is the octet offset and desired octet count of a portion of the desired text
BODY.PEEK [section] <partial>	Same as BODY [section], but this command does not set the \Seen message flag
BODYSTRUCTURE	The representation of the MIME information contained in the message
ENVELOPE	The parsed representation of the RFC 822 envelope
FAST	Shortcut for specifying the data items FLAGS, INTERNALDATE, and RFC 822.SIZE
FLAGS	Flags that are set for the message
FULL	Shortcut for specifying the data items FLAGS, INTERNALDATE, RFC 822.SIZE, ENVELOPE, and BODY
INTERNALDATE	Date the message was received by the IMAP server
RFC822	Fetches the entire RFC 822 message
RFC822.HEADER	Fetches the RFC 822 message header
RFC822.SIZE	Fetches the RFC 822 message size
RFC822.TEXT	Fetches the RFC 822 message body
UID	The message's unique identifier

SEARCH

The SEARCH command returns a list of the message numbers of messages that match the given search criteria. The defined search criteria are shown in Table C-3.

Usage

SEARCH (CHARSET) criteria

Arguments

(Optional) CHARSET specification
Search criteria

Untagged server responses

The untagged SEARCH response is required.

Results

OK (SEARCH completed successfully)
NO (SEARCH failed)
BAD (command unknown or invalid arguments)

Example

```
00000007 SEARCH ALL 1:80 SUBJECT "key largo"
* SEARCH 69 70 71
00000007 OK Completed
```

Table C-3. SEARCH Criteria

SEARCH Key	Meaning
message-set (e.g., 1:5)	Message numbers in the given set of sequence numbers only
ALL	All messages in the selected mailbox
ANSWERED	Messages that have the \Answered flag set
BCC <i>string</i>	Messages with BCC header field containing <i>string</i>
BEFORE <i>date</i>	Messages received before internal date <i>date</i>
BODY <i>string</i>	Messages that contain <i>string</i> in the message body
CC <i>string</i>	Messages that contain <i>string</i> in the CC header field
DELETED	Messages that have the \Deleted flag set
DRAFT	Messages that have the \Draft flag set
FLAGGED	Messages that have the \Flagged flag set
FROM <i>string</i>	Messages that contain <i>string</i> in the FROM header field
HEADER <i>field-name string</i>	Messages that contain <i>string</i> in the <i>field-name</i> header field
KEYWORD <i>flag</i>	Messages with <i>flag</i> set
LARGER <i>n</i>	Messages larger than <i>n</i> 8-bit bytes in size
NEW	Messages that have the \Recent but not the \Seen flag set
NOT <i>searchkey</i>	Messages that do not match <i>searchkey</i>
OLD	Messages that do not have the \Recent flag set
ON <i>date</i>	Messages received on internal date <i>date</i>
OR <i>searchkey1 searchkey2</i>	Messages that match either <i>searchkey1</i> or <i>searchkey2</i>
RECENT	Messages that have the /Recent flag set
SEEN	Messages that have the /Seen flag set
SENTBEFORE <i>date</i>	Messages whose RFC 822 Date: header is dated earlier than <i>date</i>
SENTON <i>date</i>	Messages whose RFC 822 Date: header matches <i>date</i>
SENTSINCE <i>date</i>	Messages whose RFC 822 Date: header is dated later than <i>date</i>
SINCE <i>date</i>	Messages with internal date later than <i>date</i>
SMALLER <i>n</i>	Messages smaller than <i>n</i> 8-bit bytes in size
SUBJECT <i>string</i>	Messages that contain <i>string</i> in the header's SUBJECT field
TEXT <i>string</i>	Messages that contain <i>string</i> in the message header or body

Table C-3. SEARCH Criteria (continued)

SEARCH Key	Meaning
TO <i>string</i>	Messages that contain <i>string</i> in the header's TO field
UID <i>message-set</i>	Messages whose unique identifiers match those specified in <i>message-set</i>
UNANSWERED	Messages that do not have the \Answered flag set
UNDELETED	Messages that do not have the \Deleted flag set
UNDRAFT	Messages that do not have the \Draft flag set
UNFLAGGED	Messages that do not have the \Flagged flag set
UNKEYWORD	Messages that do not have the given keyword flag set
UNSEEN	Messages that do not have the \Seen flag set

STORE

The STORE command is used to set or unset message flags.

Usage

STORE message data

Arguments

Set of messages
 Message data item name
 Message data item value

Untagged server responses

One or more untagged FETCH responses

Results

OK (STORE completed successfully)
 NO (STORE command failed)
 BAD (command unknown or invalid arguments)

Example

```
A00009 STORE 1:3 +FLAGS (\Deleted)
* 1 FETCH (FLAGS (\Seen \Deleted \Flagged))
A00009 OK Completed
```

UID

The UID command uses UID numbers for a set of messages requested by the COPY, FETCH, STORE, or SEARCH commands. Normally those commands return message sequence numbers, not UID numbers.

Usage

UID command command-args

Arguments

Command name (may take one of the values COPY, FETCH, STORE, or SEARCH)

Command arguments (options arguments to the given command)

Untagged server responses

Server responds with the untagged responses FETCH and SEARCH

Results

OK (command successfully completed)

NO (command error)

BAD (command unknown or invalid arguments)

Example

```
A000002 UID SEARCH ALL 1:80 SUBJECT "key largo"
* SEARCH 151 152 153
A000002 OK Completed
```

