Yahoo/Aol IMAP Pagination and Mail Sync

Introduction

There have been several extensions on IMAP to support multiple clients operating efficiently on a mailbox and reduce the client-server payload. However, clients are still sending requests to servers that are increasingly expensive as the mailbox keeps increasing in size.

IMAP servers use two different ids to uniquely identify messages in a mailbox. UID, unique identifiers persist across sessions while message sequence numbers are assigned based on the message's relative position in a mailbox. In order to compute these message sequence numbers and associations to messages in the mailbox, the server and client create a snapshot of the mailbox in memory. Both client and server will operate on the in-memory mailbox model during a connected session. Computation of message sequence numbers is quite onerous on servers for users with large mailboxes as servers need to cache the entire message list. In most cases, users rarely access their older messages but due to the IMAP protocol design & implementation, clients are continuously seeking updates on all the messages. IMAP CONDSTORE was created to reduce the payload retrieved from the servers. However, CONDSTORE works only for existing messages and clients perform a full mailbox synchronization to identify the deleted messages. CONDSTORE needs to return sequence numbers on updates to existing messages that require the server to compute the message's relative position.

In this document, we are proposing a few changes that will let clients walk through a user's entire mailbox while allowing servers to operate more efficiently. We also demonstrate a message-limiting change to mailboxes with messages more than a pre-configured limit at the end.

Staging Endpoints for Yahoo/Aol IMAP

New IMAP changes will be available for all Yahoo/Aol accounts at the following endpoints:

1. **Yahoo IMAP**: imaps://staging.imap.mail.yahoo.com:993/
2. **AOL IMAP**: imaps://staging.imap.mail.aol.com:993/

IMAP Protocol Changes

IMAP clients currently use the message sequence numbers to walk through a mailbox when working with a defined set of messages or sometimes to reduce payload from the server. To replace this request, we will be using IMAP PARTIAL RFC that will allow clients to walk through the entire mailbox using only UIDs. The following section uses the IMAP PARTIAL capability for clients to determine the maximum number of messages that can be retrieved per imap command. We also provide examples using IMAP PARTIAL syntax for adding pagination arguments to imap commands. We explain the change in IMAP server responses due to the introduction of the new IMAP PARTIAL capability.

Message Limits

The IMAP server will expose the maximum number of messages in a response as the capability, MESSAGELIMIT. IMAP server advertises PARTIAL and MESSAGELIMIT in its CAPABILITY list as shown below:

```
C: 02 CAPABILITY
S: * CAPABILITY IMAP4rev1 ENABLE OBJECTID CONDSTORE QRESYNC UIDONLY PARTIAL
MESSAGELIMIT=500
S: 02 OK Completed
```

An IMAP client should send the "CAPABILITY" command **post authentication** to retrieve the message limit, MESSAGELIMIT, for that session. IMAP clients should keep all IMAP commands to return messages below the server's announced limit. IMAP clients requesting more than the limit on any IMAP message commands will receive only partial message responses. The IMAP response changes to a failure with partial message responses in UidOnly mode or when the client uses the pagination arguments as described in those sections.

Examples below demonstrate the IMAP server responses to requests that exceed the posted message page limit.
Sample selected mailbox for the operations below:

```
C: 03 CAPABILITY
S: * CAPABILITY IMAP4rev1 ENABLE OBJECTID CONDSTORE QRESYNC UIDONLY PARTIAL
MESSAGELIMIT=1000
S: 03 OK Completed
C: 04 SELECT TestMailbox
S: * 14595 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 26911] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk S:
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1631839303426]
S: * OK [MAILBOXID (133)] Ok
S: 04 OK [READ-WRITE] SELECT completed; now in selected state
```

1. Fetch UID & Flags in the 22000 to 25000 uid range. The total message count exceeds the server's published 1000 messages limit.

```
C: 05 UID FETCH 22000:25000 (UID FLAGS EMAILID)
S: * 14589 FETCH (FLAGS (\Seen) UID 25000 EMAILID (ALzfLRpZADIfXlXHSQzc-N4UyoV))
```

```
S: * 14588 FETCH (FLAGS (\Answered) UID 24998 EMAILID (APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: … 997 fetch responses
S: * 14489 FETCH (FLAGS () UID 23221 EMAILID (AEV7JVJoeCGTXlXHSQdmiHw0823))
S: 05 OK [MESSAGELIMIT 1000 23221] FETCH completed with 1000 partial results
```

2. Search for UNDELETED UIDs between 22000:25000. The total message count exceeds the server's published 1000 messages limit.

```
C: 06 UID SEARCH UID 22000:25000 UNDELETED
S: * SEARCH 25000 24998 (... 997 UIDs ...) 23221
S: 06 OK [MESSAGELIMIT 1000 23221] SEARCH completed with 1000 partial results
```

3. Copy UIDs between 18000:21000. The total message count exceeds the server's published 1000 messages limit.

```
C: 07 UID COPY 18000:21000 "Trash"
S: * NO [MESSAGELIMIT 1000 20376] Too many messages to copy
S: 07 OK [COPYUID 1397597919 20376:21000 21363:22362] COPY completed for 1000 messages
```

4. Move UIDs between 18000:21000. The total message count exceeds the server's published 1000 messages limit.

```
C: 08 UID MOVE 18000:21000 "Trash"
S: * OK [COPYUID 1397597919 20376:21000 22363:23362] Some messages were not copied
S: * 12336 EXPUNGE
S: * 12335 EXPUNGE
...
S: * 11335 EXPUNGE
S: 08 OK [MESSAGELIMIT 1000 20376] MOVE completed for 1000 messages
```

5. Update flags on UIDs between 18000:20000. The total message count exceeds the server's published 1000 messages limit.

```
C: 09 UID STORE 18000:20000 +FLAGS (\Deleted)
S: * 11215 FETCH (FLAGS (\Seen \Deleted) UID 20000)
S: * 11214 FETCH (FLAGS (\Seen \Answered \Deleted) UID 19998)
...
S: * 10216 FETCH (FLAGS (\Deleted) UID 19578)
S: * 11215 EXPUNGE
S: * 11214 EXPUNGE
...
S: * 10216 EXPUNGE
S: 09 OK [MESSAGELIMIT 1000 19578] STORE completed for 1000 messages
```

Pagination Changes

IMAP clients can walk efficiently through a mailbox only through the use of message sequence numbers as they do not have gaps. Maintaining message sequence numbers on the server and client is quite expensive as server & client need to load the entire mailbox into memory. While some IMAP clients make all their requests for message discovery and updates using UIDs and avoiding the use of message sequence numbers completely, the IMAP server still has to compute MSNs for all message responses. Without message sequence numbers, message synchronization or discovery that requires a client to walk through the entire mailbox will be quite expensive for a client as UIDs are sparse.

Using pagination, an IMAP client can start from any valid UID and request a page of UIDs going all the way to the latest message; UIDNEXT - 1, or to the earliest message; 1. An IMAP client can use the pagination changes on all servers that publish the PARTIAL capability. Pagination support is only added on FETCH & SEARCH commands that are used for message discovery. COPY, MOVE, STORE & EXPUNGE are expected to operate on previously downloaded messages known to the IMAP clients and the client is expected to paginate using a sequence of individual UIDs.

FETCH Command Pagination

An IMAP client can make a FETCH request with pagination by passing a single UID range and a message count that is less than or equal to the announced PARTIAL capability. IMAP requests with multiple ranges or lists of uids will be rejected. Message sequence number requests with pagination will also be rejected. The following pagination examples illustrate the request and responses on any selected mailbox:

1. Fetch UID & Flags for 10 oldest in the 25900 to 26600 range in batches of 5

```
C: 10 UID FETCH 25900:26600 (UID FLAGS) (PARTIAL 1:5)
S: * 12595 FETCH (FLAGS (\Seen) UID 25900)
S: * 12594 FETCH (FLAGS (\Flagged) UID 25902)
S: * 12593 FETCH (FLAGS (\Answered) UID 26310)
S: * 12592 FETCH (FLAGS () UID 26311)
S: * 12591 FETCH (FLAGS (\Answered) UID 26498)
S: 10 OK FETCH completed

C: 11 UID FETCH 26499:26600 (UID FLAGS) (PARTIAL 1:5)
S: * 12595 FETCH (FLAGS () UID 26500)
S: * 12594 FETCH (FLAGS (\Seen) UID 26503)
S: * 12593 FETCH (FLAGS (\Seen) UID 26515)
S: * 12592 FETCH (FLAGS () UID 26540)
S: * 12591 FETCH (FLAGS (\Answered \Seen) UID 26541)
S: 11 OK FETCH completed
```

2. Fetch UID, FLAGS, INTERNALDATE, RFC822.SIZE, EMAILID, THREADID, X-POP-UIDL, X-MSG-DECOS fields from 25900:* and the 3 newest in the range

```
C: 12 UID FETCH 25900:* (UID FLAGS INTERNALDATE RFC822.SIZE EMAILID THREADID
X-POP-UIDL X-MSG-DECOS) (PARTIAL -1:-3)
S: * 12590 FETCH (FLAGS (\Seen) UID 26901 X-POP-UIDL "ALzfLRpZ5DIfXlXHSQzc+N4UxnU"
INTERNALDATE "26-Feb-2020 01:18:01 +0000" RFC822.SIZE 553 EMAILID
(ALzfLRpZ5DIfXlXHSQzc-N4UxnU) THREADID (20756) X-MSG-DECOS ("EML" "F24"))
```

```
S: * 12591 FETCH (FLAGS (\Answered) UID 26498 X-POP-UIDL "APS/n1oo8m6gXlXHSQo9GB2GJhA"
INTERNALDATE "26-Feb-2020 01:18:01 +0000" RFC822.SIZE 553 EMAILID
(APS_n1oo8m6gXlXHSQo9GB2GJhA) THREADID (20755) X-MSG-DECOS ("EML" "F24"))
S: * 12592 FETCH (FLAGS () UID 26311 X-POP-UIDL "AEV7JVJoe4GTXlXHSQdmiHw0798"
INTERNALDATE "26-Feb-2020 S: 01:18:01 +0000" RFC822.SIZE 553 EMAILID
(AEV7JVJoe4GTXlXHSQdmiHw0798) THREADID (20754) X-MSG-DECOS ("EML" "F24"))
S: 12 OK FETCH completed
```

3. Fetch UID, Flags, Headers between 25900:26500 and 3 oldest in the range

```
C: 13 UID FETCH 25900:26500 (UID FLAGS BODY[HEADER.FIELDS (FROM TO SUBJECT
MESSAGE-ID)) (PARTIAL 1:3)
S: * 12595 FETCH (FLAGS (\Seen) UID 25900 BODY[HEADER.FIELDS (FROM TO SUBJECT
MESSAGE-ID)] {140}
From: Fred Michael <freddie@michaels.com>
Subject: TestMessage - 2899
To: abcd@yahoo.com
Message-Id: <B27397-0100000@michaels.com>

)
S: * 12594 FETCH (FLAGS (\Flagged) UID 25902 BODY[HEADER.FIELDS (FROM TO SUBJECT
MESSAGE-ID)] {140}
From: Justin Hyde <justin@hydebar.com>
Subject: TestMessage - 2900
To: abcd@yahoo.com
Message-Id: <B27397-0100023@hydebar.com>

)
S: * 12593 FETCH (FLAGS (\Answered) UID 26310 BODY[HEADER.FIELDS (FROM TO SUBJECT
MESSAGE-ID)] {140}
From: Jeff Dreamroll <jeffdream@disney.com>
Subject: TestMessage - 2901
To: abcd@yahoo.com
Message-Id: <B27397-0100034@disney.com>

)
S: 13 OK FETCH completed
```

4. Sequence number fetch UID, Flags between 11900:12500 and 10 oldest in the range

```
C: 14 FETCH 11900:12500 (UID FLAGS) (PARTIAL 1:10)
S: 14 NO [CANNOT] Pagination is not supported with sequence numbers
```

5. Fetch UID, Flags, EmailId between 22000:25000. The total message count exceeds the 1000 messages limit.

```
C: 15 UID FETCH 22000:25000 (UID FLAGS EMAILID)
S: * 12589 FETCH (FLAGS (\Seen) UID 25000 EMAILID (ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: * 12588 FETCH (FLAGS (\Answered) UID 24998 EMAILID (APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: … 997 fetch responses
S: * 12489 FETCH (FLAGS () UID 23221 EMAILID (AEV7JVJoeCGTXlXHSQdmiHw0823))
S: 15 OK [MESSAGELIMIT 1000 23221] FETCH completed with 1000 partial results
```

6. Fetch UID, Flags, EmailId between 22000:25000 for the 1500 newest in the range. The 1500 count exceeds the 1000 messages limit.

```
C: 16 UID FETCH 22000:25000 (UID FLAGS EMAILID) (PARTIAL -1:-1500)
S: 16 NO [MESSAGELIMIT 1000] FETCH exceeds the maximum 1000 message limit
```

7. Fetch UID, Flags, EmailId between 22000:23000 and 24000:25000 for the 700 newest in the range. Multiple UID ranges are requested here.

```
C: 17 UID FETCH 22000:23000,24000:25000 (UID FLAGS EMAILID) (PARTIAL -1:-700)
S: 17 NO [CANNOT] FETCH Multiple sequence ranges are not valid for pagination
```

SEARCH Command Pagination

Similar to FETCH, the SEARCH request with pagination will work with a single UID range. SEARCH command pagination requests and responses are described below:

1. Search UIDs between 25900:26500 and oldest 5 uids in the range

```
C: 18 UID SEARCH RETURN (PARTIAL 1:5) UID 25900:26500
S: * ESEARCH (TAG "18") UID PARTIAL (1:5 25900,25902,26310,26311,26498)
S: 18 OK SEARCH completed
```

2. Search UIDs between 25900:26500 to "gmail.com" and the newest 3 in the range

```
C: 19 UID SEARCH RETURN (PARTIAL -1:-3) UID 25900:26500 TO "gmail.com"
S: * ESEARCH (TAG "19") UID PARTIAL (-1:-3 26498,26311,26310)
S: 19 OK SEARCH completed
```

3. Search UNDELETED UIDs between 25900:26500 and oldest 3 in the range

```
C: 20 UID SEARCH RETURN (PARTIAL 1:3) UID 25900:26500 UNDELETED
S: * ESEARCH (TAG "20") UID PARTIAL (1:3 25900,25902,26310)
S: 20 OK SEARCH completed
```

4. Search SEEN msns between 11900:12500 and newest 10 in the range

```
C: 21 UID SEARCH RETURN (PARTIAL -1:-10) 11900:12500 SEEN
S: 21 NO [CANNOT] Pagination is not supported with sequence numbers
```

5. Search for UNDELETED UIDS between 22000:25000. The total message count exceeds the 1000 messages limit.

```
C: 22 UID SEARCH UID 22000:25000 UNDELETED
S: * SEARCH 25000 24998 (... 997 UIDs ...) 23221
S: 22 OK [MESSAGELIMIT 1000 23221] SEARCH completed with 1000 partial results
```

6. Search for UNDELETED UIDS between 22000:25000 and the newest 1001 in the range. The 1001 message count exceeds the 1000 messages limit.

```
C: 23 UID SEARCH RETURN (PARTIAL -1:-1001) UID 22000:25000 UNDELETED
S: 23 NO [MESSAGELIMIT 1000] SEARCH exceeds the maximum 1000 message limit
```

"All Mail" Mailbox

"All Mail" mailbox represents a virtual IMAP mailbox that provides access to all the user's mailboxes using a unified view as described in RFC 6154. An IMAP client can interact with this virtual mailbox similar to other IMAP mailboxes.

All interactions between server and client must follow the standard IMAP protocol specification unless documented here. Support for IMAP ENABLE RFC is a prerequisite for supporting the "All Mail" mailbox.

Once enabled, IMAP clients can use LIST, STATUS, SELECT, EXAMINE commands to access the "All Mail" mailbox.

On the LIST response, the "All Mail" mailbox is identified using the LIST special use mailbox attribute, "\All". The CREATE or RENAME or DELETE of the "All Mail" mailbox will fail as the name is reserved and there is no real mailbox to delete.

The "All Mail" mailbox includes "Trash" and "Spam" messages.

As UIDs by definition have gaps, the IMAP server has pagination **PARTIAL** support on FETCH and SEARCH for clients to walk through a mailbox.

The following changes apply to all imap responses in the "All Mail" mailbox:

- The IMAP server will enable **QRESYNC** along with "All Mail". CONDSTORE, OBJECTID, PARTIAL are available for all mailboxes.
- The IMAP server will switch to the UIDONLY RFC after "All Mail" is enabled on all mailboxes. All message queries and updates should be performed using UIDs.
- The IMAP server will include MAILBOXID(s), from OBJECTID RFC, in every message FETCH response to identify the mailboxes a message belongs to without an explicit client request for the MAILBOXID.
- The UID MOVE command executed in the "All Mail" mailbox will update the message's MAILBOXID. The UID MOVE will return an untagged FETCH response with the updated MAILBOXID.
- The UID STORE command can also be used to update a message's MAILBOXID. Similar to UID MOVE, the UID STORE will return an untagged FETCH response with the updated MAILBOXID.

Due to server implementation limitations, **QRESYNC** on non "All Mail" mailboxes will not return deleted messages for CHANGEDSINCE queries.

UID Only Capability

IMAP servers that support UID only mode should announce using UIDONLY on the CAPABILITY response as specified in the UIDONLY RFC.

```
C: 24 CAPABILITY
S: * CAPABILITY IMAP4rev1 ENABLE OBJECTID CONDSTORE QRESYNC UIDONLY PARTIAL
MESSAGELIMIT=500
S: 24 OK Completed
```

Enabling UID Only Mode

IMAP servers and clients need to support the ENABLE and CONDSTORE RFCs to start using the "All Mail" mailbox. Client should check if the server supports ENABLE, CONDSTORE & UIDONLY capabilities and only then the client enables the UID only extension.

The following interaction shows a client listing the server capabilities followed by enabling the UID only mode.

```
C: 25 CAPABILITY
S: * CAPABILITY IMAP4rev1 ENABLE OBJECTID CONDSTORE QRESYNC UIDONLY PARTIAL
MESSAGELIMIT=500
S: 25 OK CAPABILITY completed
C: 26 ENABLE UIDONLY QRESYNC
S: * ENABLED UIDONLY QRESYNC
S: 26 OK ENABLE completed
C: 27 CAPABILITY
S: * CAPABILITY IMAP4rev1 ENABLE OBJECTID CONDSTORE QRESYNC UIDONLY PARTIAL
MESSAGELIMIT=1000
S: 27 OK Completed
```

LIST Response

The IMAP server will include a mailbox with an "\All" tag after client requests exposing the virtual "All Mail" mailbox. IMAP clients will be able to perform LIST and STATUS calls similar to other IMAP mailboxes. Every STATUS & LIST-STATUS will include MAILBOXID in the response, refer to OBJECTID RFC.

In the example below, the client requests a listing of all mailboxes after "UIDONLY" was enabled.

```
C: 28 LIST "" "*" RETURN (SPECIAL-USE STATUS (MESSAGES UNSEEN))
S: * LIST (\All \NoInferiors) "/" "All Mail"
S: * STATUS "All Mail" (MESSAGES 12818 UNSEEN 8778 MAILBOXID (A1) HIGHESTMODSEQ
1631839303426)
S: * LIST (\HasNoChildren) "/" "Inbox"
S: * STATUS "Inbox" (MESSAGES 213 UNSEEN 128 MAILBOXID (1) HIGHESTMODSEQ
1631839303426)
S: * LIST (\Sent \HasNoChildren) "/" "Sent"
S: * STATUS "Sent" (MESSAGES 0 UNSEEN 0 MAILBOXID (2) HIGHESTMODSEQ 1631839303426)
```

```
S: * LIST (\Drafts \HasNoChildren) "/" "Draft"
S: * STATUS "Draft" (MESSAGES 10 UNSEEN 0 MAILBOXID (3) HIGHESTMODSEQ 1631839303426)
S: * LIST (\Trash \HasNoChildren) "/" "Trash"
S: * STATUS "Trash" (MESSAGES 0 UNSEEN 0 MAILBOXID (4) HIGHESTMODSEQ 1631839303426)
S: * LIST (\Junk \HasNoChildren) "/" "Bulk"
S: * STATUS "Bulk" (MESSAGES 0 UNSEEN 0 MAILBOXID (6) HIGHESTMODSEQ 1631839303426)
S: * LIST (\Archive \HasNoChildren) "/" "Archive"
S: * STATUS "Archive" (MESSAGES 0 UNSEEN 0 MAILBOXID (21) HIGHESTMODSEQ 1631839303426)
S: * LIST (\HasNoChildren) "/" "TestMailbox"
S: * STATUS "TestMailbox" (MESSAGES 12595 UNSEEN 8766 MAILBOXID (133) HIGHESTMODSEQ
1631839303426)
S: 28 OK LIST completed
```

UID as MSN

After the UID only mode is enabled using the UIDONLY RFC extension, the client and server interact using only UIDs. We are making the following changes to the IMAP server operation to completely remove MSNs in the protocol:

- The IMAP server will reject all IMAP MSN commands.
- The IMAP server will respond with UID in place of MSN for all responses.
- The IMAP server will return a MESSAGELIMIT failure to clients requesting messages exceeding the PARTIAL value.
- The IMAP server will return a VANISHED response for deleted messages.

Rejecting MSN Commands

After UID only mode is enabled, the IMAP server will switch all MSN responses to be UID on all the mailboxes. IMAP clients must not use message sequence number requests on any mailbox for reading or updating messages.

The transaction below shows the response from the server for a client attempting to use sequence numbers in the "All Mail" mailbox.

```
C: 29 SELECT "All Mail"
S: * 12818 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 28527] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1631839303426]
S: * OK [MAILBOXID (A1)] Ok
S: 29 OK [READ-WRITE] SELECT completed; now in selected state

C: 30 FETCH 10001:10003 (UID FLAGS EMAILID)
S: 30 NO [CANNOT] Message sequence numbers not supported in UIDONLY mode
```

LIMIT Response on FETCH & SEARCH

After the UID only mode is enabled, the IMAP server will respond with a LIMIT failure on FETCH & SEARCH for requests exceeding the PARTIAL. Below are some of the requests that will fail and the server response.

1. Fetch UID, Flags, EmailId between 22000:25000. Number of messages requested exceeds the 1000 messages limit.

```
C: 31 UID FETCH 22000:25000 (UID FLAGS EMAILID)
S: * 25000 UIDFETCH (FLAGS (\Seen) UID 25000 MAILBOXID (1) EMAILID
(ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: * 24998 UIDFETCH (FLAGS (\Answered) UID 24998 MAILBOXID (1) EMAILID
(APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: … 997 fetch responses
S: * 23221 UIDFETCH (FLAGS () UID 23221 MAILBOXID (2) EMAILID
(AEV7JVJoeCGTXlXHSQdmiHw0823))
S: 31 NO [MESSAGELIMIT 1000 23221] FETCH completed with 1000 partial results
```

2. Fetch UID, Flags, EmailId between 22000:25000 for the 1500 newest in the range. The 1500 count exceeds the 1000 messages limit.

```
C: 32 UID FETCH 22000:25000 (UID FLAGS EMAILID) (PARTIAL 1:1500)
S: 32 NO [MESSAGELIMIT 1000] FETCH exceeds the maximum 1000 message limit
```

3. Search for UNDELETED UIDS between 22000:25000. The total message count exceeds the 1000 messages limit.

```
C: 33 UID SEARCH UID 22000:25000 UNDELETED
S: * SEARCH 25000 24998 (... 997 UIDs ...) 23221
S: 33 NO [MESSAGELIMIT 1000 23221] SEARCH completed with 1000 message results
```

4. Search for UNDELETED UIDS between 22000:25000 and the newest 1001 in the range. The 1001 message count exceeds the 1000 messages limit.

```
C: 34 UID SEARCH RETURN (PARTIAL -1:-1001) UID 22000:25000 UNDELETED
S: 34 NO [MESSAGELIMIT 1000] SEARCH exceeds the maximum 1000 message limit
```

MOVE Command to update Mailbox

The "All Mail" mailbox is virtual and the messages listed belong to the mailboxes returned by the LIST command. The MOVE command can be used to update the mailbox of a message.

The example below shows a client moving a message from "Draft" mailbox to "Sent" mailbox with "All Mail" mailbox selected.

```
C: 35 UID MOVE 28309 "Sent"
S: * 28309 FETCH (FLAGS (\Seen) UID 28309 MAILBOXID (2) EMAILID
(AF_lbZlR7J6KYTqZpACA6P8p7Xp))
```

```
S: 35 MOVE Completed
```

STORE Command to update Mailbox

As the "All Mail" mailbox is virtual, IMAP clients can use the STORE command to update the mailbox for a message. The following examples show the different variations of UID STORE to update a message’s mailbox:

1. Updating a message from "Draft" mailbox to "Sent" mailbox with "All Mail" mailbox selected.

```
C: 36 UID STORE 28411 MAILBOXID (2)
S: * 28411 UIDFETCH (FLAGS () UID 28411 MAILBOXID (2) EMAILID
(AG_lbZlR7J6KYTqZpACA6P8p8Yq))
S: 36 STORE Completed
```

2. Moving a message to the "Inbox" mailbox with "All Mail" mailbox selected using "+".

```
C: 37 UID STORE 24998 +MAILBOXID (1)
S: * 24998 UIDFETCH (FLAGS (\Answered) UID 24998 MAILBOXID (1) EMAILID
(APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: 37 STORE Completed
```

3. Removing a message from the "Inbox" mailbox with "All Mail" mailbox selected. Messages will be moved to the "Archive" mailbox.

```
C: 38 UID STORE 24998 -MAILBOXID (1)
S: * 24998 UIDFETCH (FLAGS (\Answered) UID 24998 MAILBOXID (21) EMAILID
(APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: 38 STORE Completed
```

The above operation on an "Archive" mailbox message will be noop or the message will remain in the "Archive" mailbox.

Querying Message Updates

IMAP clients can query for message updates in connected or offline mode.

NOOP or IDLE Responses

Connected IMAP clients can retrieve mailbox updates using NOOP or IDLE commands. Included below is a sample set of IMAP responses that will be sent due to mailbox updates.

```
C: 39 NOOP
S: * 12820 EXISTS
S: * OK [HIGHESTMODSEQ 1631840680837]
S: * VANISHED 23221
S: * 28530 UIDFETCH (FLAGS () UID 28530 MODSEQ (1631840680830) MAILBOXID (1) EMAILID
(AH_lbZlR8J7KYTqZpACA6P8pBct)
```

```
S: * 28527 UIDFETCH (FLAGS () UID 28527 MODSEQ (1631840680820) MAILBOXID (1) EMAILID
(AH_lbZlR7J7KYTqZpACA6P8pAbs)
S: * 25000 UIDFETCH (FLAGS (\Seen $Forwarded) UID 25000 MODSEQ (1631840680810)
MAILBOXID (133) EMAILID (ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: 39 NOOP Completed
```

SELECT or FETCH Responses using CONDSTORE

In offline mode, IMAP clients can retrieve mailbox updates using SELECT or FETCH commands. Clients must use the HIGHESTMODSEQ from the previous session to retrieve all the message updates. Using message MODSEQ will be unreliable to get updates.

1. Retrieving changes since the last mailbox access using FETCH command

```
C: 110 SELECT "All Mail"
S: * 12822 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 28537] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1631844942121]
S: * OK [MAILBOXID (A1)] Ok
S: 110 OK [READ-WRITE] SELECT completed; now in selected state

C: 111 UID FETCH 1:* (UID FLAGS EMAILID INTERNALDATE) (CHANGEDSINCE 1631840680837
VANISHED)
S: * VANISHED (EARLIER) 28530 28532 28535
S: * 28534 UIDFETCH (FLAGS () UID 28534 MODSEQ (1631844880343) MAILBOXID (1) EMAILID
(AH_lbZlR8J7KYTqZpACA6P8pEgw)
S: * 28531 UIDFETCH (FLAGS (\Seen $Forwarded) UID 28531 MODSEQ (1631844780230)
MAILBOXID (1) EMAILID (AH_lbZlR8J7KYTqZpACA6P8pDev)
S: * 28527 UIDFETCH (FLAGS (\Seen) UID 28527 MODSEQ (1631844680820) MAILBOXID (1)
EMAILID (AH_lbZlR7J7KYTqZpACA6P8pCdu)
S: 111 OK FETCH Completed
```

2. Retrieving changes since the last mailbox access using QRESYNC parameters

```
C: 210 SELECT "All Mail" (QRESYNC (1585615561 1631844942121))
S: * 12823 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 28540] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1632168028647]
S: * OK [MAILBOXID (A1)] Ok
S: * VANISHED (EARLIER) 28527, 28531
S: * 28539 UIDFETCH (FLAGS ($NotJunk) UID 28539 MODSEQ (1632168028647) MAILBOXID (1)
EMAILID(AK_lbZlR8J7KYTqZpACA6PAsHiy))
```

```
S: * 28538 UIDFETCH (FLAGS (\Seen $NotJunk) UID 28538 MODSEQ (1632124828333) MAILBOXID
(1) EMAILID(AJ_lbZlR8J7KYTqZpACA6P9rGgx))
S: * 28537 UIDFETCH (FLAGS (\Seen $NotJunk) UID 28537 MODSEQ (1632124818333) MAILBOXID
(2) EMAILID(AI_lbZlR8J7KYTqZpACA6P8qFhv))
S: * 28534 UIDFETCH (FLAGS (\Seen $Forwarded) UID 28534 MODSEQ (1632157028538)
MAILBOXID (133) EMAILID(AH_lbZlR8J7KYTqZpACA6P8pEgw))
S: * 25000 UIDFETCH (FLAGS (\Flagged \Seen $Forwarded) UID 25000 MODSEQ
(1632157228549) MAILBOXID (1) EMAILID(ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: 210 OK [READ-WRITE] SELECT completed; now in selected state
```

3. Retrieving more than PARTIAL changes using FETCH since last mailbox access

```
C: 310 SELECT "All Mail"
S: * 12824 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 28541] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1632211228486]
S: * OK [MAILBOXID (A1)] Ok
S: 310 OK [READ-WRITE] SELECT completed; now in selected state

C: 311 UID UIDFETCH 1:* (UID FLAGS EMAILID INTERNALDATE) (CHANGEDSINCE 1632168028647
VANISHED)
S: * 28540 UIDFETCH (FLAGS ($NotJunk) UID 28540 MODSEQ (1632211228486) MAILBOXID (1)
EMAILID(AL_mcZlR8J7KYTqZpACA6PAtIjz))
S: * 25000 UIDFETCH (FLAGS (\Flagged) UID 25000 MODSEQ (1632181252543) MAILBOXID (133)
EMAILID(ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: * 24998 UIDFETCH (FLAGS (\Answered) UID 24998 MODSEQ (1632181252543) MAILBOXID
(133) EMAILID(APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: … 996 fetch responses
S: * 23221 UIDFETCH (FLAGS () UID 23221 MODSEQ (1632181252543) MAILBOXID (133)
EMAILID(AEV7JVJoeCGTXlXHSQdmiHw0823))
S: 311 NO [MESSAGELIMIT 1000 23221] FETCH completed with 1000 partial results
```

4. Retrieving more than PARTIAL changes using since last mailbox access using pagination

```
C: 410 SELECT "All Mail"
S: * 12824 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 28541] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1632211228486]
S: * OK [MAILBOXID (A1)] Ok
S: 410 OK [READ-WRITE] SELECT completed; now in selected state
C: 411 UID FETCH 1:* (UID FLAGS EMAILID INTERNALDATE) (CHANGEDSINCE 1632168028647
VANISHED PARTIAL -1:-1000)
```

```
S: * 28540 UIDFETCH (FLAGS ($NotJunk) UID 28540 MODSEQ (1632211228486) MAILBOXID (1)
EMAILID(AL_mcZlR8J7KYTqZpACA6PAtIjz))
S: * 25000 UIDFETCH (FLAGS (\Flagged) UID 25000 MODSEQ (1632181252543) MAILBOXID (133)
EMAILID(ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: * 24998 UIDFETCH (FLAGS (\Answered) UID 24998 MODSEQ (1632181252543) MAILBOXID
(133) EMAILID(APS_n1oo8mBgXlXHSQo9GB2GKiB))
S: … 996 fetch responses
S: * 23221 UIDFETCH (FLAGS () UID 23221 MODSEQ (1632181252543) MAILBOXID (133)
EMAILID(AEV7JVJoeCGTXlXHSQdmiHw0823))
S: 411 OK FETCH Completed
C: 412 UID FETCH 1:23220 (UID FLAGS EMAILID INTERNALDATE) (CHANGEDSINCE 1632168028647
VANISHED PARTIAL -1:-1000)
S: * 23220 UIDFETCH (FLAGS (\Seen) UID 23220 MODSEQ (1632181252543) MAILBOXID (133)
EMAILID(AM1_MSpZADIfXlXHSQzc-N4UzpW))
S: * 23218 UIDFETCH (FLAGS (\Seen \Flagged) UID 23218 MODSEQ (1632181252543) MAILBOXID
(133) EMAILID(AJ1_MSpZADIfXlXHSQzc-N4VaqV))
S: … 997 fetch responses
S: * 22114 UIDFETCH (FLAGS (\Seen \Answered) UID 22114 MODSEQ (1632177652455)
MAILBOXID (1) EMAILID(AJ1_MSpZADIfXlXHSQzc-N4VaqV))
S: 412 OK FETCH Completed
```

5. Deleting messages using the STORE command. NOTE: Please use STORE or MOVE to move messages to the "Trash" mailbox.

```
C: 413 UID STORE 28540, 28539 +FLAGS (\Deleted)
S: * VANISHED 28540, 28539
S: 413 OK STORE Completed
```

Limited View Mailbox

By default, Yahoo/Aol IMAP servers will limit the maximum number of messages accessible on the mailbox. Users can configure the maximum number of messages through their mail settings on the web. Limited view will support IMAP protocol compliant message sequence numbers. Although clients can use both MSN and UID based commands in this mode, they are encouraged to use UIDs as much as possible. For large mailboxes, an IMAP client in the limited view will only have access to the top of the mailbox or the largest UIDs in the mailbox. The following examples will demonstrate the server responses to various client requests.

Sample selected mailbox for the operations below:

```
C: 510 SELECT TestMailbox
S: * 1000 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 26911] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk S:
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1631839303426]
S: * OK [MAILBOXID (133)] Ok
S: 510 OK [READ-WRITE] SELECT completed; now in (LIMITED) mode selected state
```

1. Search for all UIDs in the mailbox

```
C: 511 UID SEARCH UID 1:* UNDELETED
S: * SEARCH 26901 26498 26311 (... 995 UIDs ...) 25000 24998
S: 511 OK SEARCH completed
```

2. Update message flags using UIDs

```
C: 512 UID STORE 26498,25000 +FLAGS (\SEEN)
S: * 999 FETCH (UID 26498 FLAGS (\Seen) MAILBOXID (133) EMAILID
(APS_n1oo8m6gXlXHSQo9GB2GJhA))
S: * 2 FETCH (UID 25000 FLAGS (\Seen) MAILBOXID (133) EMAILID
(ALzfLRpZADIfXlXHSQzc-N4UyoV))
S: 512 OK STORE Completed
```

3. Move a message to a different mailbox using UIDs

```
C: 513 UID MOVE 26311 "Trash"
S: * OK [COPYUID 1397597919 26311 18134]
S: * 998 EXPUNGE
S: * 999 EXISTS
S: 513 OK MOVE Completed
```

4. Copy a message to the selected mailbox using UIDs

```
C: 514 UID COPY 26498 TestMailbox
S: * 998 EXPUNGE
S: * 999 EXISTS
S: 514 OK [COPYUID 1585615561 26498 26911] COPY Completed
```

5. Append a message to the selected mailbox

```
C: 515 APPEND TestMailbox (\SEEN) {2538+}
C: From: …
C: …
S: * 1000 EXISTS
S: 515 OK [APPENDUID 1585615561 26912] APPEND Completed
```

6. NOOP after a message was delivered to the mailbox

```
C: 516 NOOP
S: * 1 EXPUNGE
S: * 1000 EXISTS
S: * 1000 FETCH (UID 26913 FLAGS () MAILBOXID (133) EMAILID
(ALzfLRpZADIfXlYIRQzc-O5VzpW))
S: 516 OK NOOP Completed
```

7. Update "\Deleted" flag on a message using UID

```
C: 517 UID STORE 26901 +FLAGS (\DELETED)
S: * 997 EXPUNGE
S: * 999 EXISTS
S: 517 OK STORE Completed
```

8. Accessing the same mailbox after connecting again

```
C: 518 SELECT TestMailbox
S: * 999 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 1585615561] UIDs valid
S: * OK [UIDNEXT 26914] Predicted next UID
S: * FLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk $NotJunk)
S: * OK [PERMANENTFLAGS (\Answered \Deleted \Draft \Flagged \Seen $Forwarded $Junk S:
$NotJunk)] Permanent flags
S: * OK [HIGHESTMODSEQ 1631839503426]
S: * OK [MAILBOXID (133)] Ok
S: 518 OK [READ-WRITE] SELECT completed; now in (LIMITED) mode selected state
C: 519 UID SEARCH UID 1:* UNDELETED
S: * SEARCH 26913 26912 26911 (... 995 UIDs ...) 25000
S: 519 OK SEARCH completed
```

Change History

S.No.	Date	Comments
1.	September 28th, 2021	Initial document
2.	October 25th, 2021	Updated with new pagination changes IMAP Partial RFC draft
3.	November 10th, 2021	New default limited view
4.	November 30th, 2021	Updated MESSAGELIMIT response codes
5.	March 02nd, 2022	Updated to use IMAP UIDONLY RFC draft