IMAP PARTIAL
(paged Search and Fetch) and
MESSAGELIMIT extensions

draft-melnikov-imap-partial-04.txt
Alexey Melnikov <alexey.melnikov@isode.com>
and Yahoo team
Problem Statement

• How to handle large IMAP mailboxes in an efficient way?

• How “large” is “large”?  
  • 50k+ messages

• What is exactly the problem?
  • Memory/resource usage of keeping msgno-to-UID map on the server
  • Some clients can’t even provide user access to that many messages
  • Make SEARCH more efficient - no need for the server to search the whole mailbox, if only subset is sufficient
  • Clients blindly doing [UID] FETCH 1:* FLAGS every so often, even if there are no changes
    • CONDSTORE/QRESYNC are your friends, but more can be done
Proposed PARTIAL extension

• New SEARCH return option for returning a "page" of search results at a time

  • This can reduce the amount of work a server has to do, as the server can stop processing SEARCH once the requested page is filled in

  • Also reduces the amount of data sent by the server over IMAP

• Similarly, a new UID FETCH modifier that can restrict the number of messages processed
Proposed PARTIAL extension: details (1 of 2)

• Paged SEARCH/UID SEARCH using PARTIAL search result option

• Originally defined in RFC 5267 (part of CONTEXT=SEARCH)

• Extended to allow Python-like negative ranges (e.g. "-1:-3" -- "the last 3" resulting messages)

Example:

A01 UID SEARCH RETURN (PARTIAL -1:-100) UNDELETED UNKEYWORD $Junk
Proposed PARTIAL extension: details (2 of 2)

• Paged UID FETCH using PARTIAL FETCH modifier
  
  • Same syntax as for the SEARCH
  
  • Can be used to limit the number of messages in a UID set, when the number of messages is not known

Example:

10 UID FETCH 25900:26600 (UID FLAGS) (PARTIAL -1:-15)
Bonus Feature

• The draft clarifies interaction between PARTIAL and SAVE return options. They were unspecified before.
Changes since -00

- Clarified how PARTIAL and CONDSTORE can be used together
- The draft now also contains another extension: MESSAGELIMIT
  - MESSAGELIMIT limits the number of messages that can be operated upon in a single FETCH/SEARCH/STORE/COPY/MOVE
  - Can be implemented separately from PARTIAL, but there are some interactions. The document has examples of using just the MESSAGELIMIT and both together.
  - Clarified interaction of MESSAGELIMIT with SORT (can be used, but there are restrictions), and THREAD (can’t be used together if the mailbox has more than MESSAGELIMIT messages)
Open Issues: MESSAGELIMIT

• Should COPY/UID COPY be allowed to fail partially?
Next steps

• PARTIAL seems to be ready for Last Call

• MESSAGELIMIT part might need a bit more work. Any interest in implementing?