

# IMAP Extension: Status-Counters

(draft-neystadt-imap-  
status-counters-01.txt)

*IETF-55 Update*

John Neystadt

Alexey Melnikov

Ari Erev

November 21, 2002

# IMAP: Status-Counters

- ⌘ In a Multimedia / Multi-Context the message store contains messages of different “Contexts” (internet-draft (draft-ietf-vmim-hint-08.txt))
- ⌘ The draft implements a requirement from um-issues internet draft (draft-wong-umcli-01.txt) to extend the “Mailbox Summary” to have per-context counters.
- ⌘ Provides aggregate information based on message types/contexts.
- ⌘ Basically a performance/optimization to save on network and CPU resources.

# Status-Counters Overview

- ⌘ Extends the IMAP STATUS command.
- ⌘ A new meta-data item – ‘COUNTERS’
  - 📧 In addition to: ‘Recent’, ‘Messages’, ‘UIDNEXT”
- ⌘ Expresses a list of counter-names.
- ⌘ Provides the client with different numeric summaries of messages - grouped according to the value of Message-Context header field.
- ⌘ Uses “STATUS-COUNTERS” capability name.

# Status-Counters – use case\*

- ⌘ The main use case is:
- ⌘ Typical voice-mail sessions starts:  
"you have 2 new voice messages, 6 unheard messages, and 1 new fax message".
- ⌘ Currently - this information can be calculated using:
  - ⌘ **Option 1:**
    - ⌘ For a given mailbox, the client issues:
    - ⌘ a SELECT mailbox
    - ⌘ b FETCH 1:\* (BODY[HEADER (Message-Context)] FLAGS)
    - ⌘ Calculate the counters from FETCH responses

Or:

# Status-Counters – use case (cont.)\*

## ☒ Option 2:

1. a **SELECT** mailbox
2. b **SEARCH HEADER** "Message-Context" "Voice-Message"  
Construct message set <a> from the result; if is empty, skip till step 5.
3. c **SEARCH** <a> **UNSEEN**.  
Construct message set <b> from the result (is the list of all unseen voice messages); if is empty, skip till step 5.
4. d **SEARCH** <b> **KEYWORD** \$Important.  
Result is the list of all important unseen voice messages;
5. Repeat steps 2-4 for each message class the client is interested in.
6. 5. Calculate the counters from **SEARCH** responses

# Status-Counters Advantage

- ⌘ The proposed extension suggest to optimize performance (traffic, CPU) by standardizing a set of known "Queries".
- ⌘ Per-context counting is done on the server and only results are transferred to the client.
- ⌘ Much simpler for the client.
- ⌘ Possibly less resources on the server (which can cache/pre-compute the counters).

# Status-Counters example

```
☒ C: A CAPABILITY
  S: * CAPABILITY ... IMAP4rev1 STATUS-COUNTERS ...
☒ S: S: A OK CAPABILITY
...
C: B STATUS Inbox (UIDNEXT UIDVALIDITY
  COUNTERS (\Seen $Important "Unseen-Important" \Recent))
S: * STATUS Inbox (UIDNEXT 850 UIDVALIDITY 1234
  COUNTERS
  (All (100 \Seen 30 $Important 20
    "Unseen-Important" 10 \Recent 5)
    "Voice-Message" (10 \Seen 3 $Important 2
      "Unseen-Important" 1 \Recent 2)
    "Fax-Message" (10 \Seen 3 $Important 2
      "Unseen-Important" 1 \Recent 3)))
S: B OK STATUS completed.
```

# Status-Counters: \$Important

- ⌘ STATUS-COUNTERS relies on a special support of \$Important keyword defined in [IMAP-KEYWORDS] by the IMAP server.
- ⌘ The server MUST automatically set \$Important flag on injection of an "important" message as described in IMAP-KEYWORDS I-d (draft-melnikov-imap-keywords-00.txt). Basically:
  - ☒ If the root body-part of that message contains the header field "Importance" with the value "High".
  - ☒ If the root body-part contains either a header field "Priority" with the value of "urgent", or a header field "X-Priority" with the value "1" or "2".



# Status-Counters: Open Issues

- ⌘ 1. Poor selection syntax, use subset of SEARCH? (flags only, OR/NOT).
- ⌘ 2. \Flagged vs. \$Important flag.
- ⌘ 3. Do not repeat the criteria in each group count response.
- ⌘ 4. (Implementation) The syntax of the STATUS command is extended in a way that it is no longer possible to store STATUS items as a bitmap.

# Status-Counters: ToDo

- ⌘ Add a new command (similar to STATUS COUNTERS) that can be used in the SELECTED state, as IMAP4rev1 doesn't allow to use STATUS on the selected mailbox.

# For Further Information...

⌘ UM-LEMONADE mailing list

✉ send 'subscribe um'

to:

Majordomo@[mail.snowshore.com](mailto:mail.snowshore.com)

⌘ Contact us

✉ John Neystadt ([john@comverse.com](mailto:john@comverse.com))

✉ Alexey Melnikov ([mel@messagingdirect.com](mailto:mel@messagingdirect.com))