

IMAP & Perl

Wil Cooley <wcooley@nakedape.cc>

Summary

- Overview of IMAP
- On the wire: the protocol itself
- Scripting on-line mail processing with Perl

Overview: What is IMAP?

- Internet mail layers
 - MTA - Mail Transfer Agent
 - Postfix, Sendmail
 - MUA - Mail User Agent
 - Thunderbird, Evolution, Mutt
 - MP[AL] - Mail Presentation Agent/Layer
 - Direct file access (bleh)
 - IMAP Server
 - Cyrus, Dovecot, UW
 - POP Server
 - cucipop, most IMAP servers also
- Internet standard for mail access
- Supported by most MUAs

Overview: IMAP Features

- 3 Interaction Models
 - On-line
 - Off-line
 - Disconnected
- Mail Stored on Server
 - Access mail anywhere
 - Persistent message flags
 - Concurrent access
 - Folders on server
- Server-side searching (w/indexes!)
- Server-side filtering
- Shared mailboxes
- Store/post message to mailbox
- Extensible!

IMAP on the Wire

- IMAP is plain-text, line-oriented
- Commands begin with unique tag
- 4 states: non-auth, auth, (mailbox) selected, logout
- Non-authenticated state:
 - Capabilities
 - Authenticate
 - Logout
- Authenticated state - work with mailboxes:
 - Create, delete, rename
 - Subscribe, unsubscribe
 - ACLs, quotas, other attributes
- Mailbox selected state - work with messages:
 - Identified by sequence numbers & UIDs
 - Flags
 - Body & header: whole or in parts
 - MIME body structure

On the Wire - Authenticating

```
$ nc localhost 143
* OK [CAPABILITY IMAP4 IMAP4rev1 LITERAL+ ID STARTTLS
AUTH=PLAIN AUTH=LOGIN SASL-IR] localhost Cyrus IMAP4
v2.3.11 server ready
. capability
* CAPABILITY IMAP4 IMAP4rev1 LITERAL+ ID STARTTLS
AUTH=PLAIN AUTH=LOGIN SASL-IR ACL RIGHTS=kxte QUOTA
MAILBOX-REFERRALS NAMESPACE UIDPLUS NO_ATOMIC_RENAME
UNSELECT CHILDREN MULTIAPPEND BINARY SORT SORT=MODSEQ
THREAD=ORDEREDSUBJECT THREAD=REFERENCES ANNOTATEMORE
CATENATE CONDSTORE LISTEXT LIST-SUBSCRIBED X-NETSCAPE
URLAUTH
. OK Completed
. login jensenb ****
. OK [CAPABILITY IMAP4 IMAP4rev1 ... ] User logged in
```

- Could auth w/ SASL (Kerberos, CRAM-MD5, etc)
- Could start TLS encryption

On the Wire: Mailboxes, Before Selected

- Listing subscribed mailboxes:

```
. lsub "INBOX" "*"
* LSUB (\Noinferiors) "." "INBOX"
. OK Completed (0.000 secs 2 calls)
. lsub "" "*"
* LSUB (\Noinferiors) "." "INBOX"
* LSUB () "." "Drafts"
* LSUB () "." "Sent"
* LSUB (\HasChildren) "." "lists"
. OK Completed (0.000 secs 5 calls)
```
- Get mailbox status:

```
. status "INBOX" (MESSAGES UNSEEN)
* STATUS INBOX (MESSAGES 2043 UNSEEN 4)
. OK Completed
```

On the Wire: Selecting Mailbox

```
. select "INBOX"  
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen $Label15  
$Label14 $Label11 $Label12 $Label13 Junk Old)  
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted  
\Seen $Label15 $Label14 $Label11 $Label12 $Label13 Junk Old  
\*)]  
* 2043 EXISTS  
* 4 RECENT  
* OK [UNSEEN 2040]  
* OK [UIDVALIDITY 1131996718]  
* OK [UIDNEXT 19148]  
* OK [NOMODSEQ] Sorry, modsequences have not been enabled  
on this mailbox  
* OK [URLMECH INTERNAL]  
. OK [READ-WRITE] Completed
```

On the Wire: Fetching Messages

- Fetch flags for several messages

```
. fetch 1816:1818 flags
```

```
* 1816 FETCH (FLAGS (\Seen $Label1))
```

```
* 1817 FETCH (FLAGS (\Seen $Label2))
```

```
* 1818 FETCH (FLAGS (\Answered \Seen $Label2))
```

- Fetch certain headers

```
. fetch 1818 (body.peek[header.fields (subject message-id)] rfc822.size uid)
```

```
* 1818 FETCH (UID 18832 RFC822.SIZE 6594
```

```
BODY[HEADER.FIELDS (subject message-id)] {94}
```

```
Subject: RE: more raw devices
```

```
Message-Id: <1208881489.6207.6.camel@foobar.example.com>
```

```
)
```

```
. OK Completed (0.000 sec)
```

Scripting IMAP with Perl: Basics

```
use Mail::IMAPClient;
```

```
my $im = Mail::IMAPClient->new(  
    Server      => 'localhost',  
    User        => 'tester',  
    Password    => '***',  
    Debug       => 1,  
    ) or die "Error connecting: $EVAL_ERROR";
```

- List all folders and get message count:

```
for my $f ( $im->folders() ) {  
    printf "%-20s %-20d\n", $f, $im->message_count($f);  
}
```

- List subscribed folders only:

```
for my $f ( $im->subscribed() ) { ... }
```

Scripting IMAP: Searching

- First select mailbox:

```
$im->select("INBOX");
```

- Can search full text:

```
my @ids = $im->search('TEXT "IMAP"');  
print "Found ", scalar @ids, " matches\n";
```

- Search for message with flags set

```
@ids = $im->search('KEYWORD $Label12');  
print "Found ", scalar @ids, " matches\n";
```

- Search based on time

```
my $expire_secs = 30 * 24 * 60 * 60;  
my $imdate = strftime("%d-%b-%Y", localtime(time-  
$expire_secs));  
@ids = $im->search("SENTBEFORE $imdate");  
print "Found ", scalar @ids, " matches\n";
```

Scripting IMAP: Expiring Old, Large Messages

```
$im->select("INBOX");
my $imdate = strftime("%d-%b-%Y", localtime(time-
daytosec(90)));

# Search for read messages, sent more than 90 days ago and
# greater than 2MiB
@ids = $im->search(SENTBEFORE => $imdate,
                  'SEEN',
                  LARGER => MiBtoOctet(2),
                  );
print "Found ", scalar @ids, " matches\n";

sub daytosec { $_[0] * 24 * 60 * 60; }
sub MiBtoOctet { $_[0] * 1024 * 1024; }
```

Scripting IMAP: Deleting & Expunging

- Delete a bunch of message IDs at once w/an array ref

```
$im->delete_message(\@ids);
```

- Messages are flagged "\Deleted" but not actually deleted until expunged (and recently, closed).

```
$im->expunge();
```

```
$im->close();
```

Sieve: Server-side Filtering

- Special language designed solely for filtering mail
- Standard: RFC 3028 + others
- Includes file-transfer protocol to manage scripts
- Extensible but cannot call out or shell out, such as to SpamAssassin or pipe filters
- Won't lose mail
 - Scripts compiled when uploaded
 - Run-time failure results in delivery to INBOX
- Supported by Cyrus, Dovecot, etc.

More Information

- General IMAP information: <http://www.imap.org/>
- Listing of RFCs:
<http://cyrusimap.web.cmu.edu/imapd/specs.html>
- Mail::IMAPClient on CPAN:
<http://search.cpan.org/~markov/Mail-IMAPClient-3.07/>
- *Managing IMAP*:
<http://www.oreilly.com/catalog/9780596000127/index.html>
 - Badly out of date wrt Cyrus IMAP but still useful