

IMAP MailConnect 3 Report

Status of this Document

This document provides information for the Internet community. This document does not specify an Internet standard of any kind. Distribution of this document is unlimited.

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts.

Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress".

To learn the current status of any Internet-Draft, please check the `l1d-abstracts.txt` listing contained in the Internet-Drafts Shadow Directories on `ds.internic.net`, `nic.nordu.net`, `ftp.isi.edu`, or `munniari.oz.au`.

[1.](#) Abstract

In an effort to help increase the depth of interoperability in the Internet mail industry, over 80 people from 16 companies participated in MailConnect 3, a two-day event sponsored by the Internet Mail Consortium and held in San Jose, California, October 30 and 31, 1997. This document reports the findings of the MailConnect 3 event.

[2.](#) Conventions used in this document

In examples, "C:" indicates lines sent by a client that is connected to a server. "S:" indicates lines sent by the server to the client.

When the text calls out references, in [square brackets], see [section 7](#), "References".

Internet DRAFT

IMAP MailConnect 3 Report

November 1997

3. About the MailConnect Event

In an effort to help increase the depth of interoperability in the Internet mail industry, over 80 people from 16 companies participated in MailConnect 3, a two-day event sponsored by the Internet Mail Consortium (see <http://www.imc.org/> for information about the IMC and about the MailConnect events) and held in San Jose, California, October 30 and 31, 1997. The participant companies included:

AT&T Labs	Innosoft	Qualcomm
Control Data	i-Planet	Software.com
ESYS	Isocor	Sun
Hewlett-Packard	MetaInfo	Worldtalk
IBM/Lotus	Microsoft	
IRdg	Netscape	

The two primary focuses of the event were testing IMAP clients and servers, and testing end-to-end Internet mail security using S/MIME and PGP/MIME. The goal of the event was to enable participating vendors to test their software against that of other vendors so that each could find where they did and did not interoperate and conform to the protocols.

Unlike many marketing-based events, MailConnect 3 was exclusively for technical staff of the participants, and press and marketing staff were not invited. There were no formal tests that vendors had to participate in, and no pass or fail grades were given out. The purpose of the event was the testing itself, not grading the performance of any of the participants.

4. The IMAP Testing

Most of the participants at MailConnect 3 were testing IMAP clients and servers (or both). In general, people felt that IMAP interoperability had improved markedly in the six months since MailConnect 2, particularly for companies that had participated in each. At the wrap-up session at the end of the event, the consensus was that the typical user perception of interoperability in IMAP would be very good, although some felt that it would only be adequate.

Although fetching of body parts was not implemented in all clients, and disconnected mode was not implemented in any of the clients that were tested, most companies felt that the servers that were tested would be ready when the clients were.

Having a large number of IMAP vendors being in the same room caused

some discussions on important topics that were not being addressed in the market. The two biggest discussions during the event were about authentication and internationalization (see below).

4.1 IMAP Authentication [Auth] and SASL [SASL] (Chris Newman, Innosoft)

The customer requirements were perceived as:

- * Many customers are happy with a mechanism such as LOGIN (an experimental unspecified SASL mechanism which some have implemented) or PLAIN (a non-standard specified SASL mechanism [[Plain](#)]) when the SASL profile base64-encodes the password.
- * Most customers appear to be satisfied by 40-bit SSL + plaintext password.
- * Developers recognize neither of the above two options is acceptably secure, but they are adequate for most customer demand. A few customer markets want something stronger.

Developers, of course, had their own opinions. For some vendors, any mechanism which requires a new authentication database is very hard to deploy and will be ignored unless it becomes a very popular among customers. These include CRAM [[CRAM](#)], SCRAM [SCRAM], OTP, S/Key and Kerberos.

CRAM: Server vendors aren't fond of this mechanism. Some vendors don't like need for new authentication database, others don't like plaintext equivalent verifiers on server. Client vendors like simplicity.

SCRAM: Server vendors who don't mind deploying a new authentication

database like this due to simplicity.

OTP or S/Key: Some have implemented S/Key [Auth]. Not interesting to many due to authentication database requirements and speed with which one time passwords are used up.

Kerberos5: This is too hard to administrate at present and there are no current implementations in IMAP clients/servers (there are a few Kerberos4 implementations). This may become viable if WinNT 5.0 makes it simple enough to administrate, but time will tell.

Proposed SASL mechanism which does strong authentication of password but doesn't require authentication of rest of session: This is interesting to vendors as it should be exportable and will strongly protect the password. Will allow strong authentication without

change of backend database or slowdown of entire session. Chris agreed to produce specification soon with DSS/DH/3DES.

Other authentication issues were also discussed. Many participants were interested in password change protocol [IPCP], but some vendors satisfied with SSL web page. There was general agreement to switch to STARTTLS command [TLS] instead of using separate port for TLS. Some developers have started work on SASL API; discussion will be on the ietf-sasl@imc.org mailing list (subscribe at ietf-sasl-request@imc.org).

[4.2](#) IMAP Internationalization (John Myers, Netscape)

The group discussed the issue of what charsets to use with the SEARCH CHARSET facility of IMAP. It was agreed that the IMAP implementor's guide [Implement] include a recommendation that servers implement the UTF-8 charset with this command.

The group also discussed the proposed LANGUAGE extension [Language] and its interaction with IMAP namespaces. Servers implementing LANGUAGE are likely to want to localize the exposed NAMESPACE prefixes and the name of the INBOX. However, it is clear that this causes a problem with IMAP URLs. If we use LANGUAGE, the client will need some way to figure out the URL for a given folder. After much

talk, it was decided that the problem will need to be discussed on the IMAP list.

[4.3](#) IBM Research's Report (Barry Leiba, IBM Research)

The 4.3.x subsections contain reports from IBM Research's testing only. They do not represent a consensus of any kind. They do not necessarily even represent the general view of IBM or Lotus; indeed, there were developers from other parts of IBM and Lotus besides IBM Research, and those developers may have had different results from those stated here. Nevertheless, this author feels that the general community of IMAP developers will be well served by having this summary publicised.

The rules of the IMC MailConnect events do not permit the divulgence of specific results against specific products. Rather, this report is on the author's preception of the general state of interoperability between IMAP clients and IMAP servers at this time, and, as such, is intended to encourage and assist the development of clients and servers that operate well with each other.

Those interested in other recommendations for implementation of IMAP clients and servers should refer to [Implement], which reflects the

consensus of the IMAP community in many interoperability matters.

[4.3.1](#) Client Comments

IBM Research ran a server that is part of a research project, against which the client developers tested their clients. This author's comments about client behaviour are based solely on the logs kept on that server, from which we could see the protocol being used by the clients (after eliminating the obvious telnet sessions, in which testers were typing commands by hand). These comments, therefore, do not reflect what a user might actually have seen while using the clients.

There were, unfortunately, relatively few clients represented at the MailConnect 3 event - fewer than ten different clients were tested

there. The protocol used by the clients was generally valid protocol. One client used commands associated with a protocol extension whose presence was not advertised in the CAPABILITY response, and one client used the obsolete FIND command. Apart from that, all protocol used was legal, valid IMAP4rev1 protocol.

Clients varied widely in their discovery of mailbox lists. Most clients have now ceased to use the dangerous

C: 001 LIST "" *
method of mailbox discovery in favour of

C: 001 LIST "" %
and

C: 001 LSUB "" *
but there is a great variability in the mechanism used to go below that. Some clients search for all second-level inferences at startup, presumably so that they may correctly place "+" expansion icons on the mailbox list. This can be costly if the mailbox tree broadens quickly at the second level, and there is a new Internet Draft [Children] designed to make the accurate placement of the expansion icons easier and more efficient.

Some clients do not look for second-level mailboxes until the user opens a first-level mailbox, but some of those clients use the "refname" parameter of the LIST command for that discovery, a practice that assumes specific server behaviour with respect to the refname, and which is, therefore, inadvisable (the command

C: 002 LIST "XYZ" %
will, on some servers, result in the desired list of the inferences of mailbox "XYZ", but on other servers it will not).

Clients, at least those present at MailConnect 3, seem to be moving away from the practice of fetching entire message bodies (with [FETCH RFC822](#), for instance), and are, instead, fetching individual body

parts as needed. This is a welcome change from this author's earlier observations.

Clients are beginning to use some of the new FETCH options, including BODY[HEADER.FIELDS (...)], BODY[n.MIME], and partial fetches. This shows that clients are finding ways to take advantage of some of the advanced features of IMAP4rev1, but it is also uncovering some server

problems as servers are unprepared for these usages and are sometimes immature in their support of them (see below).

We do not know how many clients offer configuration of a mailbox prefix, but we cannot overstress the importance of providing this. Some servers, for example, do not allow creation of inferiors of "inbox", while others require that all user-created mailboxes be inferiors of "inbox". Until the Namespace extension [Namespace] is completed and widely deployed, there's no way to reconcile this sort of variation without making a configuration option available.

[4.3.2](#) Server Comments

Our group did server testing mostly with an automated test client, which ran predefined test scenarios against each server. These scenarios were designed to test different aspects of server behaviour with

- * commands that we expect to be common from clients,
- * commands that we expect to be uncommon, but reasonable,
- * commands that were designed to test the limits of the protocol, and
- * commands that are actually invalid.

The servers are generally returning syntactically valid responses to the clients, though there were some exceptions, especially in the areas of some of the less-used FETCH options. Particular problems seemed to show up in the BODY[HEADER.FIELDS (...)] responses; server developers must take particular care to follow the grammar for these responses.

On the receiving end, we found that quite a bit of valid IMAP4rev1 protocol was rejected or misinterpreted by some servers. One item was so bad that we had to change our test suite in order to get it to run on several servers, so it's worth mentioning here: the APPEND command takes the mailbox name before the data, and is usually issued in this form:

```
C: 004 APPEND MBOXNAME {nnn}
S: + send data
C: ..... RFC822 data .....
S: 004 OK done
```

The mailbox name may, however, also be sent as a literal, thus:

```
C: 004 APPEND {8}
S: + send data
C: MBOXNAME {nnn}
S: + send data
C: ..... RFC822 data .....
S: 004 OK done
```

Several servers did not support this syntax. Some of them saw the first literal, assumed it was to be the [RFC822](#) data, and complained that the mailbox name was missing. Others took the string "{8}" to be the mailbox name, failing to recognise it as a literal. This was the single most serious problem that we encountered. Server developers should be careful to accept literals anywhere they are valid. (We did not actually expect this to be a stress test; the reason our test sent mailbox names as literals is because it's easier to set it up that way in case the hierarchy delimiter turns out to be the backslash, which must either be escaped or sent in a literal.)

All servers maintained persistent UIDs and almost all servers supported IMAP4rev1, returned accurate [RFC822](#).SIZE, allowed dual-use mailboxes (mailboxes that can contain messages and inferiors at the same time), allowed the deletion of a non-empty mailbox, and allowed multiple sessions to select the same mailbox. But note that "almost all": clients MUST still deal with servers that do not allow these things.

On the other hand, servers were much more divided in whether they allow the deletion of the currently selected mailbox and in how they handle the refname parameter of the LIST command. It continues to be a good idea for clients to assume that they can't delete the selected mailbox and that they should not use the refname in an attempt to list inferiors.

There are also distressingly many servers that do not support subscriptions (they respond to LSUB either with all mailboxes or with none). We suspect that this will change over time, as users demand this feature.

There is still very little server support for authentication methods, and the LOGIN command remains the way one must gain access to most IMAP servers. A client that supported a wide array of authentication methods actually could often find one to use - many servers support one or another of them - but there is no one method (or even two methods) that could be recommended at this point as "the best one to start with".

Servers were generally weakest on SEARCH commands, giving various results and exhibiting various bugs and limitations. All servers

Internet DRAFT

IMAP MailConnect 3 Report

November 1997

handled flag searches fine, which is good: the flag searches are the ones most likely to be used internally by clients. Still, users who want to do date searching (SENDSINCE, and such) and those who want to search for substrings in message texts and header fields may be in for unpredictable results. Some servers did not correctly match substrings in header fields, some only allowed matching on certain "known" header fields, some had problems with the handling of date searches. These may be uncommon things to do, but they will also cause significant end-user confusion when they don't work consistently.

There were a number of miscellaneous protocol problems - where servers responded NO or BAD to valid protocol - and all developers were happy to find out about these problems and hurried off to fix them. There seems to be no long-term problem here, but just a bit of advice: carefully implement to the spec, and particularly to the formal grammar. Try to avoid the temptation to implement to someone else's implementation, because that leads to interoperability problems further along. It's one thing to say, "We want our server to work with Joe's client, so we'll test it against Joe's client first." It's another thing to say, "We want our server to work with Joe's client, so we'll design it based on what Joe's client does," and that approach is ill-advised.

Servers are about evenly divided in whether they do or don't permit the client to create a mailbox with a non-7-bit-ascii character in the name. The IMAP4rev1 spec advises the use of UTF-7 encoding for such names, but about half the servers allow the raw 8-bit character. This is arguably a client issue (client shouldn't do that), but from an interoperability standpoint it would be useful for servers to stop it at their end (otherwise one client might create a mailbox name that another client couldn't deal with).

There is still a wide variation on the response to a command such as

```
C: 008 FETCH 18 ENVELOPE
```

in a mailbox with fewer than 18 messages. We think that, because this is syntactically correct but semantically wrong, the correct response is NO. We see servers returning all three possible answers: OK, NO, and BAD. In reality this will not likely present an interoperability problem, since only a buggy client will issue such a

FETCH command.

A few servers modified APPENDED messages, adding, removing, or changing some message header fields. This might present interoperability problems for clients doing resynchronization after reconnecting - in order to determine the UID of the message that the client just APPENDED, it might compare certain header fields with its copy, and could get confused if these header fields changed.

Some servers did not make APPENDED messages available immediately. This could again be an interoperability problem for disconnected clients trying to reconnect and resynchronise. If the clients APPEND new messages to the selected mailbox and then expect to be able to retrieve information about them immediately after receiving a tagged OK from the APPEND command, they may become confused when the new message seems not yet to be there. Clients should be careful to wait for an EXISTS response announcing the availability of the new message, but this can be annoying when the client must periodically send NOOP commands to poll for this event. Happily, most servers do provide access to the new message immediately.

Some servers allow the deletion of a mailbox that's intermediate in the hierarchy - that is, if mailboxes exist as "x", "x/y", and "x/y/z", some servers allow "DELETE x/y" and will actually eliminate x/y from the list of mailboxes. When used with clients that only allow access to inferior mailboxes through the listed hierarchy, this will "widow" the inferior mailbox x/y/z, rendering it inaccessible to such clients (and this does seem to be the common way for clients to access mailboxes now). Instead, we think the server should mark the deleted mailbox "\Noselect" and let it remain in the mailbox list.

[4.3.3](#) What IBM Research Tested

The details of our automated test scenario might be useful to some, so we'll list the contents of the test here. Note that while we're showing a mailbox hierarchy delimiter of "/" and are showing mailbox names as quoted strings, the actual test used the proper hierarchy delimiter as return by the server, and sent all mailbox names as literals.

```

LOGIN ... ..
LIST "" "" [get hierarchy delimiter]
CREATE "Sent Mail" [mailbox names actually sent as
literals]
CREATE "Sent Mail/Broccoli" [substitute proper delimiter]
CREATE "Sent Mail/Spinach"
CREATE "inbox/Peaches" [see if we can create inbox
inferiors]
APPEND eight messages to "Sent Mail/Broccoli"
[some APPEND commands set specific message flags]
LIST "" %
LIST "" %/%
FIND ALL.MAILBOXES INBOX [test for FIND compatibility
support]
SELECT "Sent Mail/Broccoli"
STORE 8 FLAGS() [turn off flags on last message]
FETCH 1:* FULL

```

```

FETCH 8 BODY[1] [this one should return \Seen flag]
FETCH 8 BODY[1]
STORE 8 +FLAGS \Flagged [test support of flags and syntax]
FETCH 8 RFC822.SIZE [save size for next command]
FETCH 8 RFC822 [compare size returned with prior]
SEARCH ANSWERED
SEARCH SENTBEFORE 1-Aug-1996
SEARCH SENTON 5-Aug-1996
SEARCH SENTAFTER 31-FEB-1997 [test bad date]
SEARCH OR HEADER SUBJECT "JPEG" (NOT FLAGGED NOT DRAFT)
      NOT FROM "xyz" [test complicated search string]
Now comes a series of FETCH commands, fetching various attributes and
body parts of various messages. Then...
COPY 1 "Sent Mail/Spinach"
SELECT "Sent Mail/Spinach"
FETCH 1 FULL
STORE 1 +FLAGS (\Seen)
FETCH 1 BODY[1]<0.53> [test partial fetch]
STORE 1 +FLAGS (\Deleted)
CLOSE
FETCH 1:* FULL [test "wrong state" response]
SELECT "Sent Mail/Spinach" [make sure message count went down]

```

```

DELETE "Sent Mail/Spinach"      [test deletion of selected mailbox]
DELETE "Sent Mail/Broccoli" [test deletion of non-empty mailbox]
CREATE "Sent Mail/Bad*Folder1"  ["*" is actually 0xE0; test
                                creation of 8-bit mailbox name]
CREATE "Sent Mail/SubFolder1"
CREATE "Sent Mail/SubFolder1/SubFolder2"
LIST "" "Sent Mail/*"
LIST "Sent Mail/SubFolder1" %   [see what happens with refname]
LIST "Sent Mail/SubFolder1/" %  [second variation...]
Now test hierarchical rename and hierarchical delete.
RENAME "Sent Mail/SubFolder1" "Sent Mail/RenFolder1"
LIST "" "Sent Mail/*"
DELETE "Sent Mail/SubFolder1"   [should fail]
DELETE "Sent Mail/RenFolder1"   [see what happens here]
LIST "" "Sent Mail/*"
SELECT "Sent Mail"
APPEND three messages and look for EXISTS responses.
Send NOOP commands if needed to get EXISTS responses.
Fetch various combinations of items, including things like
FETCH 1 (BODY[1] BODY[1.MIME] BODY[2.MIME])
...to test fetching multiple body elements in one command.
STORE 1:* +FLAGS.SILENT (\Deleted}
CLOSE
Delete all created mailboxes, then log out.

```

[5.](#) End-to-End Security with S/MIME and PGP/MIME

MailConnect 3 was the first time that S/MIME and PGP/MIME vendors had gotten together to do interoperability testing since the two protocols had been established in the market. Only a few companies supporting each protocol participated, although it is expected that many more will want to test their products at MailConnect 4.

The three vendors with S/MIME products tested a wide variety of message types, using different encryption algorithms and types of certificates. They reported that they had almost universal success with the tests, with the exception in the area of self-signed certificates, which are not well documented in the S/MIME v2

specification. The three vendors testing PGP/MIME only tested a small number of messages, and reported that there were some problems with some implementations.

6. Post-Event Testing

The IMAP, S/MIME, and PGP/MIME testers agreed that having some post-event testing, before MailConnect 4, would be a good way to check whether or not they had in fact fixed bugs that were found at the event. The event's mailing list is always available to any vendor who wants to find other vendors to test with. In addition, IMC will set up a testing system for both S/MIME and PGP/MIME vendors to exchange messages easily.

7. References

[[RFC-2060](#)], Crispin, M., "Internet Message Access Protocol - Version 4rev1", [RFC 2060](#), University of Washington, December 1996.

[Auth], Myers, J., "IMAP4 Authentication Mechanisms", [RFC 1731](#), Carnegie Mellon, December 1994.

[SASL], Myers, J., "Simple Authentication and Security Layer (SASL)", [draft-myers-auth-sasl-13.txt](#), October 1997.

[IPCP], Newman & Gellens, "Internet Password Change Protocol (IPCP)", [draft-newman-pwchange-00.txt](#), July 1997.

[TLS], Newman, C., "Using TLS with IMAP4 and POP3", [draft-newman-tls-imappop-01.txt](#), Innosoft, November 1997.

[Plain] Newman, C., "Plaintext Password SASL Mechanism for Transitioning", [draft-newman-sasl-plaintrans-04.txt](#), Innosoft, November 1997.

[CRAM] Klensin, Catoe, Krumviede, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", [RFC 2195](#), MCI, September 1997.

[SCRAM], Newman, C., "Salted Challenge Response Authentication Mechanism (SCRAM)", [draft-newman-auth-scram-01.txt](#), Innosoft, October 1997.

[Implement], Leiba, B., "IMAP4 Implementation Recommendations", [draft-leiba-imap-implement-guide-04.txt](#), IBM Research, November 1997.

[Language], Gahrns & McCown, "IMAP4 Language Extension", [draft-gahrns-imap-language-00.txt](#), November 1997.

[Children], Gahrns & Cheng, "IMAP4 Child Mailbox Extension", [draft-gahrns-imap-child-mailbox-00.txt](#), Microsoft, November 1997.

[Namespace], Gahrns & Newman, "IMAP4 Namespace", [draft-gahrns-imap-namespace-05.txt](#), November 1997.

8. Authors' Addresses

Barry Leiba
IBM T.J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532 USA

Phone: 1-914-784-7941
Email: leiba@watson.ibm.com

Paul Hoffman, Director
Internet Mail Consortium
127 Segre Place
Santa Cruz, CA 95060 USA

Phone: 1-408-426-9827
Email: phoffman@imc.org