

# EMAILCORE WG

## Online Prague 2021

Chairs:

Alexey Melnikov <[alexey.melnikov@isode.com](mailto:alexey.melnikov@isode.com)>

Todd Herr <[todd.herr@valimail.com](mailto:todd.herr@valimail.com)>

# Note Well

- This is a reminder of IETF policies in effect on various topics such as patents or code of conduct. It is only meant to point you in the right direction. Exceptions may apply. The IETF's patent policy and the definition of an IETF "contribution" and "participation" are set forth in BCP 79; please read it carefully.
- As a reminder:
  - By participating in the IETF, you agree to follow IETF processes and policies.
  - If you are aware that any IETF contribution is covered by patents or patent applications that are owned or controlled by you or your sponsor, you must disclose that fact, or not participate in the discussion.
  - As a participant in or attendee to any IETF activity you acknowledge that written, audio, video, and photographic records of meetings may be made public.
  - Personal information that you provide to IETF will be handled in accordance with the IETF Privacy Statement.
  - As a participant or attendee, you agree to work respectfully with other participants; please contact the ombudsteam (<https://www.ietf.org/contact/ombudsteam/>) if you have questions or concerns about this.

# Note Well

## (continued)

- Definitive information is in the documents listed below and other IETF BCPs. For advice, please talk to WG chairs or ADs:
  - BCP 9 (Internet Standards Process)
  - BCP 25 (Working Group processes)
  - BCP 25 (Anti-Harassment Procedures)
  - BCP 54 (Code of Conduct)
  - BCP 78 (Copyright)
  - BCP 79 (Patents, Participation)
  - <https://www.ietf.org/privacy-policy/> (Privacy Policy)

# Administrivia

- This Meetecho session is being recorded
- Meetecho:
  - <https://meetings.conf.meetecho.com/ietf110/?group=emailcore&short=&item=1>
- Jabber room (discussions/back channel):
  - emailcore@jabber.ietf.org
- Shared note taking:
  - <https://codimd.ietf.org/notes-ietf-110-emailcore>
- Note taker?

# Agenda

- Agenda bashing, administrivia, note well (chairs) - 5 mins
- #7 (Better definition for trace header fields) - both 5321bis and 5322bis are affected <<https://trac.ietf.org/trac/emailcore/ticket/7>>
- #8 (Need a new IANA registry of header fields that are Ok to add during submission/relay/delivery) <<https://trac.ietf.org/trac/emailcore/ticket/8>>
- #19 (G.7.6. Requirements for domain name and/or IP address in EHLO) 5321bis EHLO argument requirements (domain name, domain name validity, IP address) <<https://trac.ietf.org/trac/emailcore/ticket/19>>
  - Note interaction with #1 (G.1. IP address literals in EHLO) and #9 (G.7.3. Definition of domain name in Section 2.3.5))
- Quoted strings in 5321bis and 5322bis: #21 (G.9. Revisiting Quoted Strings), #35 (Erratum 3135: quoted-string definition allows for semantically invalid empty strings) <<https://trac.ietf.org/trac/emailcore/ticket/21>>, <<https://trac.ietf.org/trac/emailcore/ticket/35>>
- #46 (Clarify the behaviour of SMTP clients when the server closes the connection) <<https://trac.ietf.org/trac/emailcore/ticket/46>>
- #5 (G.5. Remove or deprecate the work-around from code 552 to 452?) 5321bis <<https://trac.ietf.org/trac/emailcore/ticket/5>>
- #16 (G.7.12. Review Timeout Specifications) <<https://trac.ietf.org/trac/emailcore/ticket/16>>

# RFC 5322/5321

Better definition for trace header fields  
- problem statement

<https://trac.ietf.org/trac/emailcore/ticket/7>

- Various documents define trace header fields which can be added during SMTP relay and/or final delivery. RFC 5322 defines ABNF (and list 2 header fields) in Section 3.6.7 ("Trace Fields"). Other RFCs added other trace header fields, e.g. Authentication-Results (RFC 7601) and more esoteric SIO-Label-History (RFC 7444).
- Also, neither RFC 8098 nor RFC 3461 say that Original-Recipient is a trace header field.

# RFC 5322/5321

## Better definition for trace header fields (1 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/7>

- Proposed 5322bis Section 3.6.7 text regarding semantics seemed to be uncontroversial:
- **OLD:**
- A full discussion of the Internet mail use of trace fields is contained in [I-D.klensin-rfc5321bis]. For the purposes of this specification, the trace fields are strictly informational, and any formal interpretation of them is outside of the scope of this document.
- **NEW:**
- The trace fields document actions taken as a message moves through the transport system. A full discussion of the Internet mail use of the "Return-Path:" and "Received:" trace fields is contained in [I-D.klensin-rfc5321bis]; other specifications describe the use of other fields that are to be interpreted as trace fields. For the purposes of this specification, the trace fields are strictly informational, and any formal interpretation of them is outside of the scope of this document.

# RFC 5322/5321

## Better definition for trace header fields (1 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/7>

- Proposed rfc5322bis Section 3.6.7 text regarding syntax seems OK **except for “lone Return-Path:”**

### •OLD:

- The trace fields are a group of header fields consisting of an optional "Return-Path:" field, and one or more "Received:" fields. The "Return-Path:" header field contains a pair of angle brackets that enclose an optional addr-spec. The "Received:" field contains a (possibly empty) list of tokens followed by a semicolon and a date-time specification. Each token must be a word, angle-addr, addr-spec, or a domain. Further restrictions are applied to the syntax of the trace fields by specifications that provide for their use, such as [I-D.klensin-rfc5321bis].

### •NEW:

- The trace fields are a group of header fields consisting of an optional "Return-Path:" field, and one or more "Received:" fields or other fields (indicated by "optional-field" below) that are defined by other specifications as belonging within the trace fields grouping. The "Return-Path:"...



# RFC 5322/5321

## Better definition for trace header fields (1 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/7>

•Syntax currently in 3.6.7:

•trace = [return]  
• 1\*received

•Proposal on list:

•trace = [return]  
• 1\*(received / optional-field)

•Possible (but not pretty) suggestion:

•trace = return /  
• ([return]  
• 1\*(received / optional-field))

# RFC 5322/5321

## Better definition for trace header fields (1 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/7>

•Syntax currently in 3.6:

- fields = \*(trace
- \***optional-field** /
- \*(resent-date /
- resent-from /
- *[...other resent-\**
- resent-msg-id))
- \*(orig-date /
- from /
- sender /
- [truncated for brevity]

# RFC 5322/5321

Better definition for trace header fields (2 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/7>

- Remaining open issues:
  - Meaning of trace header fields in rfc5321bis
  - Should registry of trace header fields be mentioned in rfc5321bis? (ticket #8 is about the registry itself, next slide)

# RFC 5321 => A/S

Need a registry of header fields that are Ok to add during and after submission

<https://trac.ietf.org/trac/emailcore/ticket/8>

## Strawman:

Add to IANA Considerations (text in ***italics+bold*** is what has changed since the last proposal):

IANA is requested to create a new subregistry for email header fields that can be added to a message header section by a ***MSA/MTA/MDA***. The new subregistry would show whether a header field can be added by a ***"message submission", "relay", "delivery" system or some combination of them***. Headers appearing in this ***subregistry SHOULD also be registered in*** <<https://www.iana.org/assignments/message-headers/message-headers.xhtml>> (whether it is registered as a Permanent Message Header Field Name or as a Provisional Message Header Field Name). Registration policy for this new subregistry is ***"Expert Review" (FCFS?)***.

# RFC 5321

## G.7.6. Requirements for domain name and/or IP address in EHLO (1 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/19>

### 4.1.4. Order of Commands

**An SMTP server MAY verify that the domain name argument in the EHLO command actually corresponds to the IP address of the client.** However, if the verification fails, the server **MUST NOT** refuse to accept a message on that basis.

**Proposed change to the 1st sentence** by David MacQuigg <david\_macquigg@yahoo.com> on 2009.01.30 06:37 -0700 to:

An SMTP server MAY verify that the domain name argument in the EHLO command has an address record matching the IP address of the client.

Clarify what is "**address record**"? A / AAAA

# RFC 5321

## G.7.6. Requirements for domain name and/or IP address in EHLO (2 of 2)

<https://trac.ietf.org/trac/emailcore/ticket/19>

### 4.1.4. Order of Commands

An SMTP server MAY verify that the domain name argument in the EHLO command actually corresponds to the IP address of the client. **However, if the verification fails, the server MUST NOT refuse to accept a message on that basis.**

**Feedback on the 2nd sentence** by Sam Varshavchik:

I think that the "MUST NOT", in there, should be, at the most, a "MAY NOT". The only situation where MUST NOT makes sense in my eyes would be someone who's already authenticated; a mail client on port 587.

Todd: Forward-confirmed reverse DNS check on EHLO is a well established anti-abuse practice in MTAs.

Further feedback suggests some agreement on **SHOULD NOT** instead + some explanation. **Should explanation live in the A/S draft?**

# RFC 5321

## G.1. IP address literals in **EHLO**

<https://trac.ietf.org/trac/emailcore/ticket/1>

Example: EHLO [1.2.3.4]

The specification is unclear about whether IP address literals, particularly IP address literals used as arguments to the EHLO command, are required to be accepted or whether they are allowed to be rejected as part of the general "operational necessity" exception.

Some have suggested that rejection of them is so common as an anti-spam measure that the use of such ***literals should be deprecated*** entirely in the specification, others that they are still useful and used and/or that, whatever is said about IP address literals within an SMTP session (e.g., in MAIL or RCPT commands), they ***should continue to be allowed (and required) in EHLO.***

Todd: Another case for putting in the A/S if it's not already there that **IP address literals might be syntactically valid, but mail from such hosts might be rejected on anti-spam grounds**

# RFC 5321

## G.9. Revisiting Quoted Strings - problem statement

<https://trac.ietf.org/trac/emailcore/ticket/21>

Which of the following are valid in the address local-parts?

"ab cd ef"

"ab\ cd ef"

ab\ cd ef

"ab cd ef\""

"\a\b\ c\c\ e\f"

"\ ab cd ef"

Similar questions for the following:

"\\ \ \ \ \"

"\ \"

""

"\""



# RFC 5321

## G.9. Revisiting Quoted Strings

<https://trac.ietf.org/trac/emailcore/ticket/21>

Which of the following are valid in the address local-parts?

"\\ \\ \\ \"

"\ \"

""

"\""

**Possible resolution** for this ticket: clarify whether ""@example.com and/or "\"@example.com and a few variations are valid, clarify that white space reduction **MUST NOT** be applied by intermediate systems, and then put a paragraph into the A/S that says that any mail administrators who allow those sorts of things as mailboxes on their system, whether it tries to canonicalize them or not, is looking for bad karma to descend upon them.

Also possibly clarify how local-part comparison for equality should be done, i.e. "canonicalize before comparison".

# RFC 5322

## Disallow empty quoted string

<https://trac.ietf.org/trac/emailcore/ticket/35>

<https://www.rfc-editor.org/errata/eid3135>

Section 3.2.4 says:

quoted-string = [CFWS]

DQUOTE \*([FWS] qcontent) [FWS] DQUOTE

[CFWS]

**Strawman proposal** is not to change ABNF to disallow empty quoted string, but add some text to A/S to talk about where empty quoted strings are Ok and where they are not.

# RFC 5322 => A/S

## Disallow empty quoted string

Notes from Ashley Willis:

The text following this definition states that a "quoted-string is identical to atom, semantically." "Semantically, neither the optional CFWS outside of the quote characters nor the quote characters themselves are part of the quoted-string; the quoted-string is what is contained between the two quote characters."

Where are quoted strings used?

- "local-part" → **can't be empty**. What about `"".bar` or `""foo.`?"
- "received-token" (uses "word") → **can't be empty**
- "keywords" (uses "phrase") → **can't be empty?**
- "name-addr" and "group" use "display-name". "display-name" (uses "phrase", which uses "word")
  - Display-name in "name-addr" **can be empty!**
  - ***But not the name of the group?***

# RFC 5321

Clarify the behaviour of SMTP clients when the server closes the connection

<https://trac.ietf.org/trac/emailcore/ticket/46>

A couple of issues in Section 3.8 (Terminating Sessions and Connections):

After detecting the need to shut down the SMTP service and returning a 421 reply code. This reply code can be issued after the server receives any command or, if necessary, asynchronously from command receipt (***on the assumption that the client will receive it after the next command is issued***).

Does the part in parenthesis really occur in practice, notably when the closure is asynchronous from command receipt? Maybe it should be reworded this way: "on the assumption that the client will receive it after the next command is issued **or read it before closing the connection at its side**"?

# RFC 5321

Clarify the behaviour of SMTP clients when the server closes the connection

<https://trac.ietf.org/trac/emailcore/ticket/46>

Strawman proposal, part 2:

SMTP clients that experience a connection close, reset, or other communications failure due to circumstances not under their control (in violation of the intent of this specification but sometimes unavoidable) **SHOULD**, to maintain the robustness of the mail system,  
**+ try to read a possible asynchronous reply from the SMTP server, and**  
**+ otherwise**  
treat the mail transaction as if a 421 response had been received and act accordingly.

Alternative proposal: **try to check if the server already sent an asynchronous reply**

# RFC 5321

**Clarify the behaviour** of SMTP clients when the server closes the connection

<https://trac.ietf.org/trac/emailcore/ticket/46>

Blocking TCP client:

- Time t0: Server writes "421 shutting down" and gracefully closes its end of the TCP connection
- Time t1: Client does blocking **write()** containing one or more SMTP command to the socket and this succeeds
- Time t2: Client does blocking **read()** of the response from the socket. The read succeeds, as "421 shutting down" is already in the TCP socket buffer.

# RFC 5321

***Clarify the behaviour*** of SMTP clients when the server closes the connection

<https://trac.ietf.org/trac/emailcore/ticket/46>

Similar for non blocking client:

- Time t0: Server writes "421 shutting down" and gracefully closes its end of the TCP connection
- Time t1: Client does non blocking **write()** containing one or more SMTP command to the socket
- Time t2: Client does **poll()** waiting for read/write/close events. The poll() completes immediately and returns data already in the TCP buffer.

# RFC 5321

Remove or deprecate the work-around from code 552 to 452?

<https://trac.ietf.org/trac/emailcore/ticket/5>

## 4.5.3.1.10. Too Many Recipients Code

RFC 821 [3] incorrectly listed the error where an SMTP server exhausts its implementation limit on the number of RCPT commands ("too many recipients") as having reply code 552. The correct reply code for this condition is 452. ***Clients SHOULD treat a 552 code in this case as a temporary, rather than permanent, failure so the logic below works.***

The suggestion in Section 4.5.3.1.10 may have outlived its usefulness and/or be inconsistent with current practice. Should it be removed and/or explicitly deprecated?

**Any feedback from SMTP server and client implementors?**



# RFC 5321

## G.7.12. Review Timeout Specifications

<https://trac.ietf.org/trac/emailcore/ticket/16>

- 4.5.3.2.1. Initial 220 Message: 5 Minutes
- 4.5.3.2.2. MAIL Command: 5 Minutes
- 4.5.3.2.3. RCPT Command: 5 Minutes
- 4.5.3.2.4. DATA Initiation: 2 Minutes
- 4.5.3.2.5. Data Block: 3 Minutes
- 4.5.3.2.6. DATA Termination: 10 Minutes.
- 4.5.3.2.7. Server Timeout: 5 Minutes.

RFC 5321 (and its predecessors going back to 821) specify minimum periods for client and server to wait before timing out. Are those intervals still appropriate in a world of faster processors and faster networks? Should they be updated and revised? Or should more qualifying language be added?

Extra slides, if needed

# **Suggested** scope for the “Core Email Applicability Statement”

- Best practices on use of SMTP, email format/MIME.
- Don't touch POP/IMAP/JMAP or Sieve
  - They IMAP/Sieve and JMAP have their own WGs (EXTRA and JMAP respectively)
- Don't touch SMTP Submission (RFC 6409) this time around
- Reference DMARC/DKIM/SPF?
  - Note that DMARC has its own WG, so not doing any work here

# RFC 5321

## G.7.3. Resolvable FQDN in SMTP and private domain names

<https://trac.ietf.org/trac/emailcore/ticket/9>

In Section 2.3.5 ("Domain Names"):

Only resolvable, fully-qualified domain names (FQDNs) are permitted when domain names are used in SMTP.

Editor's Note: does "in the public DNS" or equivalent need to be added to "resolvable"???

Strawman: "Resolvable" can mislead readers to think that they need to attempt resolve all domain names when received.  
Proposal to drop "resolvable". Possibly discuss "private domain names" in the Applicability Statement document.