

**RFC 9051, "Internet  
Message Access Protocol  
(IMAP) - Version 4rev2"**

Reported RFC Errata

## RFC 9051, "Internet Message Access Protocol (IMAP) - Version 4rev2", August 2021

Source of RFC: extra (art)

Errata ID: 6826

Status: Reported

Type: Technical

Publication Format(s) : HTML

Reported By: Hontvári Levente

Date Reported: 2022-02-02

Throughout the document, when it says:

```
18. RFC822, RFC822.HEADER, and RFC822.TEXT FETCH data items were
   deprecated. Clients should use the corresponding BODY[]
   variants instead.
```

It should say:

```
18. RFC822, RFC822.HEADER, and RFC822.TEXT FETCH data items were
   removed. Clients should use the corresponding BODY[]
   variants instead.
```

- Notes:

Contrary to the original text, these data items are not deprecated but they were completely removed from the text of the RFC. As far as I see other deprecated items are not removed completely but moved to a '...obsolete...' token in the formal syntax.

Errata ID: 7343

Status: Reported

Type: Technical

Publication Format(s) : TEXT

Reported By: Nicholas Evans

Date Reported: 2023-02-13

Section 9 says:

```
resp-cond-auth = ("OK" / "PREAUTH") SP resp-text
                ; Authentication condition

resp-cond-bye  = "BYE" SP resp-text

resp-cond-state = ("OK" / "NO" / "BAD") SP resp-text
                 ; Status condition SP

...

resp-text      = ["[" resp-text-code "]" SP] [text]
```

- Notes:

Appendix E, Changes from RFC 3501 / IMAP4rev1, item 23 states that "resp-text ABNF non-terminal was updated to allow for empty text."

In the spirit of Appendix E. 23, resp-text should allow "[" resp-text-code "]" without a trailing SP. Similarly, resp-cond-auth, resp-cond-bye, and resp-cond-state should also allow the trailing SP to be dropped when resp-text is empty. In all of these cases, the missing SP does not change the semantics of the response, so clients should not require it to be present. However, for compatibility with clients that strictly adhere to the formal syntax as-is, servers should conservatively send the trailing SP even when text or resp-text is empty.

This aligns with existing practice, as some existing IMAP4rev1 servers already fail to send these trailing spaces, and some liberal clients already allow them to be missing.

It should say:

```
resp-cond-auth = ("OK" / "PREAUTH") [SP resp-text]
                ; Authentication condition
                ;
                ; Servers SHOULD send the trailing SP when resp-text
                ; is empty.

resp-cond-bye  = "BYE" [SP resp-text]
                ; Servers SHOULD send the trailing SP when resp-text
                ; is empty.

resp-cond-state = ("OK" / "NO" / "BAD") [SP resp-text]
                 ; Status condition
                 ;
                 ; Servers SHOULD send the trailing SP when resp-text
                 ; is empty.

...

resp-text      = ["[" resp-text-code "]" [SP [text]]] / [text]
                ; Servers SHOULD send then trailing SP when
                ; resp-text-code is present but text is empty.
```

Errata ID: [7246](#)

Status: Reported

Type: Technical

Publication Format(s) : TEXT

Reported By: Mechiel Lukkien

Date Reported: 2022-11-12

Section 7.5.2. says:

```
BINARY[<section-binary>]<<number>>
```

```
An <nstring> or <literal8> expressing the content of the specified section after removing any encoding specified in the corresponding Content-Transfer-Encoding header field. If <number> is present, it refers to the offset within the DECODED section data.
```

It should say:

```
BINARY[<section-binary>]
```

```
An <nstring> or <literal8> expressing the content of the specified section after removing any encoding specified in the corresponding Content-Transfer-Encoding header field.
```

- Notes:

While a `FETCH_request` can be "partial" with `<...>` for both `BODY[]` and `BINARY[]`, only a `FETCH_response` for `BODY[]` can have an optional offset. A `FETCH_response` for `BINARY[]` cannot have an optional offset. At least according to the ABNF, which I believe is leading.

See lines 6756 and 6757: `msg-att-static = "BODY" section ["<" number ">"] SP nstring / "BINARY" section-binary SP (nstring / literal8)`  
And line 6987: `section-binary = "[" [section-part] "]"`

RFC 3516, IMAP4 Binary Content Extension, from which the original text was probably copied, appears to have the same issue but no errata.

Errata ID: [7323](#)

Status: Reported

Type: Technical

Publication Format(s) : TEXT

Reported By: Nicholas Evans

Date Reported: 2023-01-26

Section 6.4.4.4. says:

```
S: B283 NO [BADCHARSET UTF-8] KOI8-R is not supported
```

It should say:

```
S: B283 NO [BADCHARSET (UTF-8)] KOI8-R is not supported
```

- Notes:

The BADCHARSET response code is described in 7.1 as "Optionally followed by a parenthesized list of charsets", and in the formal syntax as:

```
resp-text-code =/ "BADCHARSET" [SP "(" charset *(SP charset) "]" ]
```

Although a client's parser might use a generic resp-text-code (atom [SP 1\*<any TEXT-CHAR except "]">]) as a fallback, the server should parenthesize even when only one charset is sent.

Errata ID: 7518

Status: Reported

Type: Technical

Publication Format(s) : TEXT

Reported By: Joó Ádám

Date Reported: 2023-05-18

Section 4.1.1. says:

A set of messages can be referenced by a sequence set containing either message sequence numbers or unique identifiers. See Section 9 for details. A sequence set can contain ranges of sequence numbers (such as "5:50"), an enumeration of specific sequence numbers, or a combination of the above. A sequence set can use the special symbol "\*" to represent the maximum sequence number in the mailbox. A sequence set never contains unique identifiers.

It should say:

Why am I required to fix someone else's sloppy writing?

Notes:

The first and last sentences are in direct contradiction. Clarification is needed.