

# EPITA's Netiquette

Assistants C/Unix

September 3, 2016

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>General rules</b>	<b>2</b>
2.1	Message header . . . . .	3
2.1.1	Subject . . . . .	3
2.1.2	Header fields related to transmission . . . . .	4
2.1.3	Identification . . . . .	5
2.2	Message body . . . . .	6
2.2.1	Content . . . . .	6
2.2.2	General formatting . . . . .	7
2.2.3	Reply . . . . .	8
2.3	Signature . . . . .	12
<b>3</b>	<b>Tickets</b>	<b>12</b>
<b>4</b>	<b>Appendices</b>	<b>13</b>
4.1	ASCII characters notation . . . . .	13
4.2	Definitions . . . . .	13
4.3	Additional definitions for the EBNF notation . . . . .	14

## 1 Introduction

This document takes an inventory of the rules that any form of written communication must respect, in the following circumstances:

- assistants are involved
- the communication channel used is under responsibility of assistants

Some rules or recommendations described here are deliberately stricter than those from the original Netiquette (RFC 1855<sup>1</sup>). The aim is to prepare you to

---

<sup>1</sup><https://tools.ietf.org/html/rfc1855> “RFC 1855, Netiquette Guidelines”

communicate with the whole computing community by following the standards and usages that it stands by. Indeed, during any electronic communication, following the Netiquette shows how educated and respectful you are to your interlocutors, which is essential. However, keep in mind that rules described in this document may be overruled by common sense.

Although this Netiquette has specific rules for EPITA, if you cannot identify these, you can safely follow the standards described in this document, in your future electronic communications (both professional and personal) — since, outside EPITA where these are rules, they *at least* constitute a set of good practices.

**You must know and apply accurately the principles described in this document.** Of course, you are strongly advised to visit other links and to deepen the various concepts and protocols mentioned: after all, *knowledge is power*.

This document relies on RFC 1855<sup>2</sup>, that you should refer to if any doubt arises. It sometimes uses the EBNF notation defined by RFC 822<sup>3</sup>, as well as certain definitions summed up in another section<sup>4</sup> of the same RFC. Additional definitions are described in the appendix. In order to quickly identify those, all of these definitions' names are written using monospaced letters, such as `DIGIT` or `subject`.

In case of error, question or any suggestion related to this publication, you are encouraged to contact us using [qualites@tickets.assistants.epita.fr](mailto:qualites@tickets.assistants.epita.fr).

## 2 General rules

Rules described in this section apply to all situations mentioned in this document. Details or exceptions could be related to these, in more specialized sections. In any case, you must follow (by decreasing order of priority) specialized rules, then general rules (presented in this section) and finally, RFC 1855<sup>5</sup>; if any conflict arises, comply with the rule with the highest priority.

---

<sup>2</sup><https://tools.ietf.org/html/rfc1855> “RFC 1855, Netiquette Guidelines”

<sup>3</sup><https://tools.ietf.org/html/rfc822#section-2> “RFC 822, Standard for the Format of ARPA Internet Text Messages - 2. Notational Conventions”

<sup>4</sup><https://tools.ietf.org/html/rfc822#appendix-D> “RFC 822, Standard for the Format of ARPA Internet Text Messages - Appendix D: Alphabetical Listing of Syntax Rules”

<sup>5</sup><https://tools.ietf.org/html/rfc1855> “RFC 1855, Netiquette Guidelines”

## 2.1 Message header

### 2.1.1 Subject

As soon as your message can be archived, whether it is in essence (e.g. news) or from convenience (e.g. mails), you must associate a short text to it. It allows an efficient and quick identification. This ability is generally brought by a header field named *Subject*. If the used communication channel does not handle such field, you must use its equivalent or (if you fail to identify this latter) ask for an advice from assistants.

The value of this header field must be a combination of two parts, **tags** and **summary**:

`subject = tags SPACE summary`

#### 2.1.1.1 Tags

**tags** are identifiers that allow a quick classification and an easier automatic filtering. These are sequences of characters (capital letters, numbers, dash, underscore, plus sign or slash) enclosed by two brackets:

`tag-id = 1*( UPPER / DIGIT / "-" / "_" / "+" / "/" )`

`tags = 2( "[" tag-id "]" )`

Note that tags are not preceded nor separated by spaces. To preserve their efficiency more than for aesthetics, we will avoid **tag-id** whose length exceeds ten characters. Also, those with at least half of their characters that do not match UPPER (cf. appendix) will also be avoided.

The first tag provides the **context of the message** (thread topic, project in question, etc.) whereas the second one sums up its *cause* or its *aim* (password retrieval, etc.). For example: [APARTMENT] [RENT].

If your message cannot be seen in any well-defined context, you are then allowed to use [MISC] as the first tag; however, this should only be occasional and this usage certainly cannot become the explanation for a lazy reflection. Its abuse could, in this case, be sanctioned.

#### 2.1.1.2 Summary

The **summary** must not repeat information provided by tags, it must complete these. It describes the subject of the message into a few words, a sequence of words (generally a *nominal group*) that sums up the **issue of the message** and must follow these constraints:

- it must not be a sentence (determiners and other non-essential elements of the statement must be removed), but it must not be an illogical sequence of keywords neither.
- the length of the subject, including both tags and summary, must not exceed 80 characters.
- without having to read the message, we can guess its content by reading the summary; after reading it, we must be able to find it quickly in search results.

Therefore, the following summaries are **invalid**:

Git problem

The Windows partition of my rack does not boot, once again!

On the contrary, these ones are **valid**:

Git quota exceeded, reset request

Windows partition (rack) damaged

### 2.1.1.3 Response' subject

According to RFC 5322, about the *Subject* header field:

*When used in a reply, the field body MAY start with the string "Re: " (from the Latin "res", in the matter of) followed by the contents of the "Subject:" field body of the original message. If this is done, only one instance of the literal string "Re: " ought to be used since use of other strings or more than one instance can lead to undesirable consequences.*

### 2.1.2 Header fields related to transmission

Communication protocols always define a certain number of metadata, associated to the message you want to send. In SMTP protocols (for emails; cf. RFC 5322<sup>6</sup>) and NNTP (for news; cf. RFC 5536<sup>7</sup>), these metadata are represented by header fields located at the beginning of the message's header and separated from the message's body by CRLF CRLF bytes (an empty line).

Some metadata probably sound familiar to you: for example, the sender (*From*) or, in SMTP, the receiver (*To*). It is vital to pay attention to those: your receivers will wait for your messages to respect that standard and will reply to

<sup>6</sup><https://tools.ietf.org/html/rfc5322> "RFC 5322, Internet Message Format"

<sup>7</sup><https://tools.ietf.org/html/rfc5536> "RFC 5536, Netnews Article Format"

those consequently. If you have, for instance, used an invalid *Reply-To*, their reaction might be inappropriate.

In your communications, check carefully:

**Cc (SMTP)** Specifies **secondary message recipients**. These addresses are visible by any recipient of the message. While replying, check that every recipient is concerned and that every one of them allows you to share their addresses to others.

**Reply-To (SMTP, NNTP)** Specifies *the mailbox for replies to message* from your interlocutors. Their email client will probably select this address without them knowing. If your message specifies an address other than from the one used to send the mail in the *Reply-To* field, this means you expect an answer at this address over the other one. Quoting RFC 1855<sup>8</sup>:

*Be careful when you reply to messages or postings. Frequently replies are sent back to the address which originated the post - which in many cases is the address of a list or group! You may accidentally send a personal response to a great many people, embarrassing all involved. It's best to type in the address instead of relying on "reply."*

**In-Reply-To (SMTP)** Specifies the **original message** (using its *Message-Id*<sup>9</sup>) to which the current message is a reply. It is essential to stick to this definition: if you specify messages to which you do not answer directly, you will bother your interlocutors as their messaging clients could display the evolution of a discussion thread in a form of a structure tree. You should rather use the header field *References* (SMTP, NNTP) to specify all messages from a given discussion thread.

### 2.1.3 Identification

The sender of a message must be able to be identified by any person from school. The *From* header field must therefore contain, at least, one of these information related to the sender:

- the *login*
- the surname **and** the first name
- an address using the **epita.fr** domain

---

<sup>8</sup><https://tools.ietf.org/html/rfc1855> "RFC 1855, Netiquette Guidelines"

<sup>9</sup>**Message-ID** (SMTP, NNTP) is a header field that helps to uniquely identify a message, for instance: <51DE8431.9030507@epita.fr>. It is generally composed of a timestamp and the message's original domain.

## 2.2 Message body

### 2.2.1 Content

#### 2.2.1.1 Clarity and common courtesy

Be **intelligible** in your remarks: your interlocutors might not have as much information and time to give to your issue, as you.

Do not hesitate to bring **context information**, but generally not more than a short paragraph. Be careful of the enunciation of your issue: use **simple, suggestive and non-ambiguous terms**.

Especially if you ask for an advice, **greet** the people who read you (using, for instance, “Good morning”) and take care of ending your message by thanking them for their time and their help: a **salutation** is thereby required.

Beyond any other expectation, you must show a flawless courtesy when you communicate using communication channels mentioned in this document. If clumsy writing mistakes might be tolerated, any misuse of language is unacceptable and would be severely sanctioned.

Keep also in mind that you must separate the main content of your message from greetings and salutations with empty lines (one above, one below).

Instead of writing:

```
Hello,  
Sometimes, I will start a sentence and I do not even know where it is  
going. I just hope I find it along the way.  
Sincerly yours,
```

--

Michael Scott

You must write:

```
Hello,  
  
Sometimes, I will start a sentence and I do not even know where it is  
going. I just hope I find it along the way.  
  
Sincerly yours,
```

--

Michael Scott

**Recommended reading:** *How To Ask Questions The Smart Way*<sup>10</sup>.

---

<sup>10</sup><http://www.catb.org/~esr/faqs/smart-questions.html> “How To Ask Questions The Smart Way”

## 2.2.2 General formatting

### 2.2.2.1 Width

Each line of your message (such as displayed on the screen of its recipients) must consist of 80 columns<sup>11</sup> or less, in addition to the carriage-return and line-feed.

This limit of 78 or 80-column width, whose origin goes back to the screens' size adjusted to text mode, is nowadays preserved to avoid the catastrophic effect of automatic line breaks in a terminal or when editing text.

As it is possible that the content of your message might be *quoted* (cf. Reply), you will pay attention that every line of your own message (opposed to quotes) will not see its length exceed **72 columns**; this way, they will be able to be quoted multiple times, in a reasonable way, without exceeding the 80-column limit.

However, we will pay attention to the width of multi-line paragraphs: the wrapping must not step in before 60 columns.

### 2.2.2.2 MIME Type

Sometimes, some people wish to use, while writing a message, more than what *plain text* allows, such as an italic or bold composition: they will therefore use, for example, the HTML format. However, by taking in consideration compatibilities issues, it is essential to have an equivalent version of the message in **plain text**: indeed, some messaging clients are unable to process anything else. Therefore, we include multiple version of a content in a single message, using the **multipart/alternative** as MIME type.

However, in the context of this document, the use of **multipart/alternative** is **not allowed**: exchanged messages must only contain a single version, that is *plain text*. In other words, the *Content-Type* header field can only contain one of these values:

- **text/plain**
- **multipart/mixed** with the first part in **text/plain** if there is any attached file
- **application/pgp-signature** with the first part in **text/plain** if the message is to be cryptographically signed. If a signature is present, it **must** be valid.

This allows the use of parameters **Format** and **DelSp** in *Content-Type*, as well as described by RFC 3676.

---

<sup>11</sup>The unit measure used here is the *column* of a terminal (rather using *characters*) as some characters are displayed on more or less a single column: for instance, consider the case of the tab character `\t`.

### 2.2.2.3 Length

The number of lines that a message can contain is not limited. However, please note that the more lines a message contains, the less likely it is that its recipients would read it soon or even entirely, not to say: at all. Therefore, it is clever to begin a message that has more than 20 lines by a sentence that sums up the points of discussion: this way, the recipients can understand quickly its purpose and, if necessary, start planning its actual reading.

### 2.2.2.4 Paragraphs

Even if it might be interesting to *lighten* long messages (for example, by writing short paragraphs and adding multiple line breaks between each of those), it is considered as a bad practice. The text formatting by paragraphs must follow the **logical division of a speech**, more than its aesthetics.

However, we will pay attention to **avoid**, if possible, paragraphs containing only 1 or 2 lines unless the short length of the message (of, at most, 2 or 3 paragraphs) allows it.

### 2.2.2.5 Trailing Whitespace

The addition of trailing whitespace characters is considered inelegant, is useless noise, does not serve to convey your message, and is therefore forbidden.

## 2.2.3 Reply

In this section, we will consider the message  $B$  as the one you are writing as a reply to a message  $A$ . For any number  $i$ ,  $B_i$  is the part of  $B$  that replies to the part  $A_i$  of  $A$  (messages  $A$  and  $B$  can also contain only one part each).

### 2.2.3.1 Quoting order

It is generally desirable to **remind the context** related to your answer. In order to do this, we can mention important parts of  $A_i$  **before**  $B_i$ ; that way, it is not necessary to read again  $A$  to understand your answer. Beware of “**before**”; if  $A_i$  appears,  $B_i$  is always placed **below**  $A_i$  (*bottom-post*):

$$\begin{array}{l} A_1 \\ B_1 \\ A_2 \\ B_2 \\ \dots \end{array}$$

rather than the overly used:



*B*  
*A*

Answering *above* the quoted message (*top-post*) makes no sense, as illustrated by this part of *Jargon File's FAQ* :

A: No.

Q: Should I include quotations after my reply?

Please beware of the use of *top-post*, even though it is commonly used outside EPITA, as it is most likely to represent one of the worst infringements we can do to this Netiquette and the most obvious monstrosity you can show to interlocutors educated to electronic communications.

### 2.2.3.2 Quote formatting

We quote by preceding each line concerned by a quote mark (>, ASCII 0x3E, also called *angle bracket*) followed by a space:

On Wed, 29 May 2013 05:42:27 -0700, "Will Coleda via RT" wrote:

> How are you running the spec tests? - it looks like you've included some  
> that are not marked as runnable by rakudo.

I think I did a "gmake spectest\_full". Shall I run just the spectest target again or can you just ignore the tests that are not marked as runnable?

If your quote contains another one, each line of this latter must be preceded by *two* quote marks and a space (and so on, for even more nested quotes, though this case should rarely happen); see the example in Quote attribution. Note that no space should be inserted between two successive quote marks.

Quoted material must be separated from the text nearby with empty lines (one above, one below). However, a block of quoted text nested at a level *n* must only be separated from its parent block with a line containing *n-1* quote marks.

If the quote section is too long, it can be wise to only quote its beginning and its end, separated with [...]:

On 8/18/13, James Bowery <jabowery@gmail.com> wrote:

> Of the two key conceptual gaps in current programming language philosophy  
> -- commensurability and change propagation -- commensurability, if filled  
> with due rigor, has the greatest potential for clearing up confusion by  
> recasting other core features as derivative.

>  
> [...]

>  
> Well, for starters, the core laws of quantum mechanics fall out as  
> completely general theorems of relation arithmetic generalized to include

> "negative" relationships (opt cit). Among the implications of this are  
> core programming language constructs for quantum information systems.

Let's get the basics nailed down and working so that we can learn  
them, before wandering any further into theoretical CS.

### 2.2.3.3 Quote attribution

RFC 1855<sup>12</sup> specifies:

*If you are forwarding or re-posting a message you've received, do not  
change the wording. If the message was a personal message to you  
and you are re-posting to a group, you should ask permission first.  
You may shorten the message and quote only relevant parts, but be  
sure you give proper attribution.*

This attribution is made by inserting, right before the quote, a line that indicates  
the author and date of the quoted message. For example:

```
On Thu, Jul 4, 2013 at 10:10 AM, Paul Phillips <paulp@improving.org> wrote:
> On Thu, Jul 4, 2013 at 12:40 AM, Lukas Rytz <lukas.rytz@epfl.ch> wrote:
>> That explains why it works that way, but not why we do the inlining at all.
>> The only use case I know of is annotation arguments for java defined
>> annotations, which are required to be compile-time constants.
>
> Performance. Conditional compilation, for instance - constant false allows
> for losing code entirely where a field containing false would not.
```

Without disagreeing, I would expect that this is the kind of optimization  
that the VM is able to handle reliably.

Modifying a quoted part is inappropriate unless this brings necessary and  
minimalist clarifications (placed between brackets):

> Without disagreeing, [Lukas Rytz] would expect that this is the kind of ...

No specific format is established for this attribution line; we will simply pay  
attention to its conciseness, clarity and that it allows to correctly identify the  
author of the quote. A standard attribution line generally contains:

- the date
- the name of the sender
- the address of the sender

of the quoted message.

Specifying the attribution is only required on the *first time* of a given message.

---

<sup>12</sup><https://tools.ietf.org/html/rfc1855> "RFC 1855, Netiquette Guidelines"

On Wed, Jul 24, 2013 at 11:24:25PM +0200, Tobias Leich wrote:  
> But what speaks for having our own configure system: What if you already  
> have libapr/libuv installed with their dev-packages?

Currently, it looks like MoarVM just ignores an installed libapr, so that's currently not an issue.

> For the moment I would love to see a pull request for solaris 11.

Well, I don't know anything about pull requests, but I've appended the patch I used below.

However, if quoted parts of *various* messages are nested, we will remind, between each intertwining, the corresponding attribution.

#### 2.2.3.4 Quote content

According to RFC 1855<sup>13</sup>:

*Content of a follow-up post should exceed quoted content.*

Avoid to include more than enough necessary text from  $A_i$  to understand  $B_{\sim i}$ : this would probably obstruct your message and is probably the reason why *top-post* was widely spread among ordinary people. This rule implies to *never* quote salutations nor signatures (unless it is the topic of the discussion). Therefore, this would be invalid:

On 8/18/13, Walter White <heisenberg@gmail.com> wrote:  
> Hi Jesse,

Hi Mr. White,

> Did you get the flower I sent you?

I did. Thank you.

> Yours sincerely,

Yours sincerely,

You should rather write:

Hi Mr. White,

On 8/18/13, Walter White <heisenberg@gmail.com> wrote:  
> Did you get the flower I sent you?

---

<sup>13</sup><https://tools.ietf.org/html/rfc1855> "RFC 1855, Netiquette Guidelines"

I did. Thank you.

Yours sincerely,

This message conveys the same meaning but is freed from visual interferences, compared to the first one.

## 2.3 Signature

The *signature* is a short text block inserted at the end of a message and it contains information that should allow the identification and contact of its author, even without the message's header (cf. RFC 1849<sup>14</sup>). It is automatically inserted at the end of the message by many messaging clients, using the file `~/.signature` (on UNIX systems).

The signature block must not exceed 80 columns nor contain more than 4 lines. It must not start with an empty line neither.

In order to be easily detected by programs and humans, it must be separated from the last paragraph with an empty line, followed by a line containing only two dashes (ASCII 0x2D) and a space:

```
<content> CRLF
CRLF
"--" SPACE CRLF
<text signature>
```

Note the following rules:

- The sequence of characters CRLF "--" SPACE CRLF must not appear more than once in a message
- Most messaging clients are able to convert LF to CRLF, which spares the sender any additional precautions
- While quoting, the signature must **never** be included (unless, of course, if it is specifically targeted; however, we will not quote the line containing "--" that is useless in this context)
- Beware of text editors that remove automatically but blindly spaces at the end of lines.

## 3 Tickets

Tickets are a specific kind of email: these are sent to a program, the *tickets manager*, that allows to follow the evolution of a request. Therefore, it is a process that is partly automated. Also, when communicating through tickets,

---

<sup>14</sup><http://tools.ietf.org/html/rfc1849#page-22> "RFC 1849, "Son of 1036": News Article Format and Transmission"

pay attention to **never** modify the header field **Subject**: the system would then lose track of your message.

To send a ticket to the assistants, write to:

- acu@tickets.assistants.epita.fr if the current team is the ACU one
- yaka@tickets.assistants.epita.fr if the current team is the YAKA one

You will receive an acknowledgment of receipt once the ticket is created. If you did not receive anything after a reasonable period of time (few hours at most), your message probably never made it to the assistants.

The header field *Subject* of your message must always respect the following format:

`subject = tags SPACE summary`

But in this context, `summary` must respect the following format:

`summary = login " - " description`

where `login` is:

- if this is a group project, the chief's *login*
- the sender's *login* otherwise

and `description` must be replaced by the usual summary.

**Example:** [PISCINE-D3] [TRACES] login - Clarification request for ex3

## 4 Appendices

### 4.1 ASCII characters notation

When we write ASCII  $k$  (where  $k$  is a number between 0 and 0x7F), it refers to the character n° $k$  of the ASCII table; thus, ASCII 0x40 represents @.

### 4.2 Definitions

This documents uses some terms with a specific meaning; the following definitions are meant to indicate how to understand these.

**space** The ASCII character, ASCII 0x20 (SPACE). Other kind of spaces, which usually depend on Unicode, are not taken into account. In a negative statement (“no space”), this term refers however to **all** whitespace characters (e.g. \t), including Unicode’s ones and could even, according to the context, also include end-of-line characters (\n, etc.).

### 4.3 Additional definitions for the EBNF notation

**UPPER** Every ASCII character between A (ASCII 0x41) and Z (ASCII 0x5A).