

# RFC 5322 : Internet Message Format

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 octobre 2008

Date de publication du RFC : Octobre 2008

<https://www.bortzmeyer.org/5322.html>

---

Vieux de désormais trente et une années (la première norme était le RFC 724<sup>1</sup> en mai 1977), le format des messages électroniques vient donc de recevoir une nouvelle spécification. Pas de grands changements mais beaucoup de petites erreurs corrigées, elle marque l'avancée de la norme au statut de « projet de norme ».

Le courrier électronique d'Internet repose sur deux piliers : un protocole d'échange des messages, SMTP, normalisé dans le RFC 5321 et un format des messages, que traite notre RFC 5322. Les deux sont relativement indépendants, notamment dans le sens où on peut transporter des messages sur d'autres protocoles que SMTP, comme par exemple UUCP. Autre exemple, les adresses traitées par les deux RFC sont différentes (même si leur syntaxe est la même), SMTP gérant l'enveloppe du message et le format gérant son contenu (section 1). Cela justifie deux RFC séparés.

Que contient le RFC 5322 qui vient de remplacer le RFC 2822 ? Moins qu'on ne pourrait le croire. Des pans énormes de l'utilisation du courrier sont en dehors de ce RFC comme l'important format MIME (RFC 2045) ou comme les extensions d'internationalisation (RFC 6532).

Il reste donc un RFC sur l'envoi de messages textuels en ASCII. Mais, sans même prendre en compte des extensions comme MIME, c'est déjà assez complexe. La section 2 définit les règles lexicales des messages. Les messages sont découpés en lignes composées de caractères. Le nombre maximal de caractères par ligne est de 998 caractères (section 2.1.1, qui recommande d'accepter également des longueurs plus grandes, mais sans générer soi-même de lignes plus longues).

La section 2.2 définit les **en-têtes** qui composent le début du message. Un message au format RFC 5322 ressemble, sur le câble, à :

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc724.txt>

---

Date: Tue, 9 Sep 2008 09:33:17 +0200  
 To: ubuntu-fr@lists.ubuntu.com  
 Message-ID: <20080909073317.GA23390@sources.org>  
 Subject: [Son] Silence complet sur mon ESS ES1978 Maestro 2E  
 From: Stephane Bortzmeyer <stephane@sources.org>

Le son, avec Linux, c'est vraiment p=E9nible.

J'ai un Dell Inspiron 7500 dont le son ne marche pas du tout. Silence complet.

...

Le début, avant la ligne vide, est composé de cinq en-têtes (il y en a bien d'autres, non affichés ici. Avec mutt, c'est la commande `h` qui permet d'afficher tous les en-têtes). Chaque en-tête a un nom, suivi du deux-points et d'un corps. Si les premiers MUA affichaient directement les noms des différents champs, cela ne se fait plus guère aujourd'hui. En général, ce que voit l'utilisateur est bien différent de ce qui passe sur le câble, notamment à des fins de localisation (afficher « *Objet* » et pas « *"Subject"* », par exemple). Le corps du champ peut être non structuré (section 2.2.1, on y met ce qu'on veut, c'est le cas du champ `Subject` : ) ou bien structuré (section 2.2.2, il obéit à une mini-grammaire spécialisée, c'est le cas de `Date` : ou de `From` : ). Les champs peuvent être très longs et il est donc prévu (section 2.2.3) de pouvoir les **plier** sur plusieurs lignes. C'est un des points les plus souvent oubliés des auteurs d'analyseurs de ce format. Il faut donc rappeler qu'on ne peut **pas** utiliser d'outil orienté ligne comme `grep` sur un message électronique. (Une solution est d'utiliser l'excellente commande `formail`, dans le paquetage `procmail`, avec son option `-c` qui replie les lignes.)

Enfin, il y a le **corps** du message, après les en-têtes. Précédé d'une ligne vide, ce corps n'a pratiquement pas de structure dans notre RFC 5322 (section 2.3) mais d'autres normes comme MIME (RFC 2045) se chargent de combler ce manque et permettent, par exemple, de transporter plusieurs fichiers, y compris binaires, dans un seul message.

La section 3 décrit la syntaxe d'un message. La grammaire est très riche et très complexe. Pire, en raison de l'ancienneté de ce format, un grand nombre de productions grammaticales sont désormais abandonnées mais doivent toujours être reconnues par un analyseur, au cas où un logiciel les génère encore. Elles ont un nom commençant par `obs-` pour qu'on puisse les reconnaître plus facilement. (La section 4 décrit cette grammaire abandonnée plus en détail. On y trouve, par exemple, un format pour les années sur deux chiffres, non compatible avec l'an 2000.)

Parmi les subtilités de cette grammaire :

- La syntaxe des dates, dans le champ (structuré) du même nom (section 3.3). Elle est très souvent violée : en pratique, on voit des dates de toutes les formes, même lorsque le logiciel est un programme payant et cher. Un exemple correct est `Date: Tue, 30 Sep 2008 11:49:07 +0200` (le format du courrier est bien plus ancien que la norme ISO 8601). Ce champ est décrit en 3.6.1.
- Les adresses (section 3.4). C'est, encore plus que les dates, un des aspects les plus mal compris du courrier. Le Web abonde de programmes écrits dans des langages qui encouragent l'approximation (comme PHP ou Javascript) et qui refusent des adresses parfaitement légales `<https://www.bortzmeyer.org/arreter-d-interdire-des-adresses-legales.html>`. À leur décharge, il faut dire que la grammaire de la section 3.4.1 n'est pas triviale ! Un exemple d'adresse (on les utilise notamment dans les champs `From` : et `To` : ) est `stephane+blog@bortzmeyer.org`. On notera que des normes comme NAI (RFC 7542) ou bien XMPP (RFC 7622) utilisent une syntaxe proche.

Enfin, la section 3.6 décrit tous les en-têtes. Citons entre autres :

---

<https://www.bortzmeyer.org/5322.html>

- L'expéditeur, `From:`, `Sender:` et `Reply-To:`, section 3.6.2.
- Les destinataires, `To:` et `Cc:` (ceux qui reçoivent une copie), section 3.6.3.
- `Message-ID:`, section 3.6.4, un champ structuré qui contient un identificateur unique du message, par exemple `<17b4fdcd0809290041i4323797a1964ccf5a344b1c47@mail.gmail.com>`.
- `Subject:`, section 3.6.5, qui indique l'objet du message et n'est pas structuré. Pour qu'il puisse contenir autre chose que des caractères ASCII, il faut utiliser l'encodage du RFC 2047 (ou le plus récent RFC 6532).
- `Received:`, section 3.6.7, qui sert à indiquer par quels relais est passé un message. Par exemple,  
`Received: from rv-out-0708.google.com (rv-out-0708.google.com [209.85.198.250])  
by mx1.nic.fr (Postfix) with ESMTP id 98D821714088 for <bortzmeyer@nic.fr>;  
Mon, 29 Sep 2008 09:41:22 +0200 (CEST)`. C'est un excellent outil de débogage pour l'administrateur réseau.

La liste des en-têtes n'est pas figée, on la trouve dans un registre IANA <https://www.iana.org/assignments/message-headers/perm-headers.html> (créé à l'origine par le RFC 4021) décrit en section 6.

Notre RFC 5322 remplace le RFC 2822, par rapport auquel il y a peu de changements. Le RFC 2822 remplaçait le RFC 822 qui est resté en service très longtemps, accompagnant l'essor du courrier électronique, et qui a été tellement influent que son numéro sert souvent à désigner le format (comme dans l'ancien module Python `rfc822` <http://docs.python.org/lib/module-rfc822.html>), remplacé depuis par `email` <http://docs.python.org/lib/module-email.html>). L'annexe B donne la liste des changements par rapport au RFC 2822, essentiellement des corrections de bogues.