

# RFC 2047 : MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 2 Novembre 2008

Date de publication du RFC : Novembre 1996

<http://www.bortzmeyer.org/2047.html>

---

La série de RFC sur MIME, qui a introduit les caractères composés (et les écritures non-latines), ainsi que les objets multimédia, dans le monde du courrier électronique, comprend, entre autres, ce RFC qui normalise une méthode pour mettre des caractères non-ASCII dans les **en-têtes** du courrier, ces méta-données situées avant le corps du message et structurées en `nom: valeur`.

Normalement, le courrier, tel qu'il était normalisé par le RFC 822<sup>1</sup> (aujourd'hui RFC 5322), ne permettait que le jeu de caractères ASCII dans les en-têtes. C'est ainsi qu'il fallait écrire :

```
From: Stephane Bortzmeyer <bortzmeyer@sources.org>  
Subject: Du cafe bien fort !
```

au lieu de la forme correcte en français :

```
From: Stéphane Bortzmeyer <bortzmeyer@sources.org>  
Subject: Du café bien fort !
```

Notre RFC 2047 vise tout simplement à fournir une solution à ce problème, en **encodant** les caractères non-ASCII, en Quoted-Printable ou en Base64. Les en-têtes ci-dessus apparaîtront donc sur le réseau comme :

---

1. Pour voir le RFC de numéro NNN, <http://www.ietf.org/rfc/rfcNNN.txt>, par exemple <http://www.ietf.org/rfc/rfc822.txt>

---

```
From: =?iso-8859-1?q?St=E9phane?= Bortzmeyer <bortzmeyer@sources.org>
Subject: Du =?iso-8859-1?q?caf=E9?= bien fort !
```

Ainsi, le message pourra passer à travers toute l'infrastructure du courrier qui n'accepte que l'ASCII. Ce sera au MUA de les remettre sous une forme lisible (voir par exemple mon article [Décoder les en-têtes du courrier électronique <http://www.bortzmeyer.org/decoder-en-tetes-courrier.html>](http://www.bortzmeyer.org/decoder-en-tetes-courrier.html)).

L'approche de notre RFC est donc conservatrice : on ne demande pas à tous les logiciels sur le trajet de connaître MIME, autrement ce dernier n'aurait jamais été déployé. (MIME n'utilise même pas, par prudence, certaines fonctions du RFC 822 qui auraient pu aider mais qui sont trop souvent boguées, cf. section 1.) Aujourd'hui où l'infrastructure du courrier est très différente, une méthode plus radicale pourrait être réaliste et c'est l'approche du bien plus récent RFC 5335.

La section 2 du RFC donne la grammaire exacte : un terme encodé est précédé de `=?` suivi de l'encodage des caractères. (Pour le corps du message, tel que normalisé dans le RFC 2045, l'encodage est indiqué dans un en-tête, `Content-Type`. Pour les en-têtes eux-mêmes, il a fallu trouver une autre solution.) Puis on trouve le surencodage appliqué par MIME, `Q` pour `Quoted-Printable` et `B` pour `Base64` puis le terme lui-même, ainsi encodé. Le tout est terminé par un `?=`. Voici le résultat, produit par un programme Python avec le module `email`  [<http://docs.python.org/library/email.html>](http://docs.python.org/library/email.html) :

```
% python
>>> from email.header import Header
>>> h = Header('niçoise', 'iso-8859-1')
>>> print h
=?iso-8859-1?q?ni=E7oise?=-
```

Notons que l'encodage utilisé est appelé "*charset*" (jeu de caractères) par MIME, ce qui n'est pas vraiment le terme correct (`<< utf-8 >>` est un encodage, le jeu de caractères est Unicode). La section 3 normalise ce paramètre, pour lequel les valeurs standard de MIME  [<http://www.iana.org/assignments/character-sets>](http://www.iana.org/assignments/character-sets) sont utilisées.

La section 4 décrit en détail les surencodages possibles, `Quoted-Printable` et `Base64`. Le premier (RFC 2045) convient mieux lorsque le texte comprend des caractères latins, avec quelques caractères composés. Le second (RFC 4648) est préférable si le texte est composé de caractères non-latins. La section 4.2 détaille quelques spécificités de "*Quoted-Printable*" et notamment l'utilisation du `_` à la place de l'espace comme dans :

```
% python
>>> from email.header import Header
>>> h = Header("réveillés dans une Citroën niçoise", "ISO-8859-1")
>>> print h
=?iso-8859-1?q?r=E9veill=E9s_dans_une_Citro=E9n_ni=E7oise?=-
```

Pour le même genre de tâches, les programmeurs Perl peuvent utiliser `Encode : :MIME : :Header`  [<http://kobesearch.cpan.org/htdocs/perl/Header.html>](http://kobesearch.cpan.org/htdocs/perl/Header.html).