

RFC 6858 : Simplified POP/IMAP Downgrading for Internationalized Email

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 12 mars 2013

Date de publication du RFC : Mars 2013

<https://www.bortzmeyer.org/6858.html>

Lorsqu'un message complètement internationalisé (avec en-têtes en Unicode, adresse d'expéditeur en Unicode, etc) est stocké sur un serveur de messagerie, et qu'un client IMAP ou POP veut le récupérer, que faire si le dit client n'est pas capable de gérer les messages entièrement internationalisés ? Le message a déjà été reçu et stocké. Il est trop tard pour le renvoyer à l'expéditeur (le courrier électronique étant asynchrone, on ne peut pas en général signaler à l'expéditeur qu'on ne sait pas gérer ses messages). Il faut donc se résigner à effectuer un repli stratégique ("*downgrading*") et à transcrire plus ou moins bien le message en ASCII. Il existe deux algorithmes pour cela, un très simple dans ce RFC 6858¹ et un plus riche dans le RFC 6857.

Car notre RFC 6858 ne prétend pas faire un travail satisfaisant. Si M. Li, en Chine, a utilisé son nom comme adresse (From: [Caractère Unicode non montré²]@[Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré].[Caractère Unicode non montré][Caractère Unicode non montré]), il n'existe pas de mécanisme automatique satisfaisant pour transformer cette adresse en une adresse qui marche toujours (par exemple, qui permette d'y répondre). Le repli ("*downgrade*") est donc toujours rudimentaire et le message ne sera donc qu'imparfaitement rendu.

Une des solutions permises par le RFC 6855 est de dissimuler le message aux yeux des clients IMAP archaïques. Notre RFC en propose une autre, afficher un message dégradé (le « message de synthèse »), considérant que cela vaut mieux que rien.

L'algorithme de ce court RFC se veut simple, considérant que, si le programmeur est courageux et prêt à faire des choses compliquées, il a plutôt intérêt à mettre en œuvre le RFC 6855, pour gérer complètement l'Unicode. (L'autre algorithme, celui du RFC 6857, n'est pas d'accord.)

Le gros du RFC est la section 2, qui indique les transformations opérées pour transformer le message Unicode en du pur ASCII :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc6858.txt>

2. Car trop difficile à faire afficher par L^AT_EX

- Les adresses en Unicode sont remplacées par une adresse ASCII bidon (qui ne marchera pas), indiquant ce qui s’est passé. Ainsi, le From: [Caractère Unicode non montré]@[Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré].[Caractère Unicode non montré][Caractère Unicode non montré] ci-dessus pourra devenir, par exemple, From: Internationalized Address =?utf-8?b?5p20?=<international@address.invalid>. Le TLD .invalid est réservé par le RFC 2606. Ici, le nom de M. Li a été encodé avec le RFC 2047 (le truc qui commence avec =?). Cette transformation s’applique à tous les champs contenant des adresses (From:, ici, mais aussi Cc:, Reply-To:, etc).
- Les paramètres MIME en Unicode sont simplement retirés. Ainsi, Content-Disposition: attachment; filename=[Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré]_[Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré][Caractère Unicode non montré] deviendra Content-Disposition: attachment.
- Le Subject: sera encodé en RFC 2047. Ainsi, Subject: Venez découvrir ce café deviendra Subject: =?utf-8?q?Venez_d=C3=A9couvrir_ce_caf=C3=A9?=. Contrairement au cas de l’adresse (où le repli produit un en-tête qui ne marchera plus), ici, le MUA pourra reconstruire le sujet original.
- Tous les autres champs qui contiennent de l’Unicode seront tout simplement retirés (dans l’algorithme du RFC 6857, ils sont transformés).

Voilà, j’avais promis un algorithme simple, il l’est. Il permet à un serveur IMAP ou POP de ne pas trop dérouter ses lecteurs non-Unicode. Il reste quelques petits détails, dans les sections 3 et 5. Ainsi, des commandes IMAP comme FETCH RFC822.SIZE permettent de connaître la taille d’un message avant de le télécharger. Comme le repli en ASCII change la taille, notre RFC permet au serveur de renvoyer la taille du message originel, pas celle du message de synthèse, pour lui simplifier la vie.

Question sécurité, le repli n’est pas idéal (section 5) :

- Il invalide les signatures,
- Des champs importants comme Message-ID: ont pu disparaître purement et simplement dans l’opération,
- Et le plus gênant : si le client POP ou IMAP (comme le permet fetchmail) récupère le message en local, puis le détruit sur le serveur, le message original est irrémédiablement perdu et le client se retrouve avec uniquement le message de synthèse...

À noter que les réponses du serveur IMAP pourront désormais inclure le mot-clé DOWNGRADED, désormais dans le registre IANA des réponses possibles <<https://www.iana.org/assignments/imap-response-codes/imap-response-codes.xml>>.