# RFC 6873 : Format for the Session Initiation Protocol (SIP) Common Log Format (CLF)

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 février 2013

Date de publication du RFC : Février 2013

https://www.bortzmeyer.org/6873.html

---

La question d'un format standard et commun pour la journalisation des événements SIP est exposée et discutée dans le RFC 6872 [1]. Ce RFC 6873, lui, spécifie un format standard conforme aux exigences et au modèle de données du RFC 6872. Techniquement, sa principale originalité est d'être un format texte... indexé. Il a donc les avantages des formats textes (lisible sans outil particulier, utilisable avec des outils génériques comme grep ou awk) tout en gardant de bonnes performances si on développe des outils qui lisent ces index.

Le projet d'un format standard pour les journaux de SIP s'appuie évidemment sur le grand succès du CLF des serveurs HTTP. Ce format CLF a facilité la vie de tous les administrateurs système, et a permis le développement de plein d'outils de statistiques et d'analyse. Il a toutefois quelques défauts, le principal étant de performance. Si on cherche un champ de taille variable, comme le nom de la ressource demandée, il faut parcourir chaque ligne jusqu'à ce champ. Or, en SIP, les lignes peuvent être longues (SIP est plus complexe que HTTP et il y a plein de choses à journaliser), d'où le souhait d'un format où on puisse sauter facilement à un champ donné. Les formats binaires permettent cela, mais au prix de la souplesse et de la facilité : il faut en effet des outils spécialisés pour tout accès au journal. Le nouveau format décrit par ce RFC va essayer de vous donner le beurre (le format texte lisible et facile à traiter) et l'argent du beurre (l'indexation, pour sauter directement aux informations intéressantes). Ce format « ASCII indexé » est une innovation : peu de programmes utilisent cette technique. En raison de sa nouveauté dans le monde IETF, ce choix a fait l'objet de chaudes discussions dans le groupe de travail.

La section 4 du RFC décrit ce format : chaque événement est journalisé sous forme d'un enregistrement qui compte **deux** lignes. La première, qui commence toujours par une lettre, est un ensemble

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6872.txt

de métadonnées et de pointeurs vers les champs principaux. Tous sont de taille fixe et donc d'accès direct rapide. La seconde, qui commence toujours par un chiffre, comprend les champs eux-mêmes, dans l'ordre indiqué par le RFC. Ces champs sont de taille variable, mais indexés par les pointeurs cités plus haut. Il y a les champs obligatoires (définis par le modèle de données du RFC 6872) comme `From:` ou bien l'adresse IP source, et les champs facultatifs. Pour les champs obligatoires, toute l'astuce est dans les pointeurs. Ils sont à un emplacement fixe de la première ligne (pour être trouvés facilement) et ils pointent vers l'octet de la deuxième ligne où se trouve le champ. Pour connaître la fin du champ, on regarde simplement le pointeur suivant, qui indique le début du champ suivant. Ainsi, si on veut l'adresse IP de destination, on lit les octets 20 à 23 de la première ligne (006D dans l'exemple suivant), puis les octets 24 à 27 pour avoir le pointeur suivant (007D). Entre ces deux pointeurs (109 à 125 en décimal), on trouve `192.0.2.10:5060`. (Pour les champs optionnels, il faut faire une analyse syntaxique classique.) Voici un exemple d'un enregistrement (section 5 du RFC) :

```
A000100,0053005C005E006D007D008F009E00A000BA00C700EB00F70100
1328821153.010 RORUU 1 INVITE -sip:192.0.2.10 192.0.2.10:5060 192.0.2.200:56485 sip:192.0.2.10 -sip:1001@exa
DL88360fa5fc DL70dff590c1-1079051554@example.com S1781761-88 C67651-11
```

On note que les champs numériques (les pointeurs) sont encodés en chiffres hexadécimaux, et pas en binaire (pour faciliter l'analyse par des outils orientés texte). Ce petit programme (en ligne sur `https://www.bortzmeyer.org/files/sip-clf-reader.py`) en Python (récupéré sur le Wiki du groupe de travail <http://trac.tools.ietf.org/wg/sipclf/trac/wiki>) permet de décoder l'enregistrement :

```
% python sip-clf-reader.py sip.log
String length 257

str=005C CSeq             message[83:92] = 1 INVITE

str=005E Response         message[92:94] = -

str=006D ReqURI           message[94:109] = sip:192.0.2.10

...
```

Ici, on avait un programme spécifique comprenant ce format. Et si on veut utiliser les outils texte traditionnels ? La section 6 donne quelques détails à ce sujet. La méthode recommandée est de sauter les lignes d'index (elles commencent toujours par une lettre) et de ne traiter que les lignes de données (elles commencent toujours par un chiffre décimal, qui encode l'estampille temporelle). Ensuite, on éclate la ligne de données en champs, ceux-ci étant séparés par une tabulation. C'est moins rapide que les index, mais plus simple pour les utilisateurs de awk.

Merci à Régis Montoya pour sa relecture.