

Groupe de travail Réseau  
**Request for Comments : 3862**  
 Catégorie : En cours de normalisation  
 Traduction Claude Brière de L'Isle

G. Klyne, Nine by Nine  
 D. Atkins, IHTFP Consulting  
 août 2004

## Présence commune et messagerie instantanée (CPIM) : Format de message

### Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

### Notice de Copyright

Copyright (C) The Internet Society (2004).

### Résumé

Le présent mémoire définit le type de contenu MIME 'Message/CPIM', un format de message pour les protocoles qui se conforment à la spécification du profil commun pour la messagerie instantanée (CPIM, *Common Profil for Instant Messaging*).

### Table des Matières

1. Introduction.....	2
1.1 Motivation.....	2
1.2 Fondements.....	2
1.3 Objectifs.....	3
1.4 Terminologie et conventions.....	3
2. Structure globale de message.....	3
2.1 En-têtes MIME de message/CPIM.....	4
2.2 En-têtes de message.....	4
2.3 Mécanisme d'échappement de caractère.....	4
2.4 Contenu du message.....	5
3. Syntaxe de l'en-tête de message.....	6
3.1 Noms d'en-tête.....	6
3.2 Valeur d'en-tête.....	6
3.3 Étiquetage de langage.....	6
3.4 Espaces de noms pour l'extensibilité des noms d'en-tête.....	7
3.5 Caractéristiques qu'il est obligatoire de reconnaître.....	8
3.6 Syntaxe collectée d'en-tête de message.....	8
4. Définitions d'en-tête.....	9
4.1 En-tête 'From'.....	10
4.2 En-tête 'To'.....	10
4.3 En-tête 'cc'.....	10
4.4 En-tête 'DateTime'.....	11
4.5 En-tête 'Subject'.....	11
4.6 En-tête 'NS'.....	11
4.7 En-tête 'Require'.....	12
5. Exemples.....	12
5.1 Exemple de message Message/CPIM.....	12
5.2 Exemple utilisant MIME multipart/signed.....	13
6. Considérations de conception des applications.....	13
7. Considérations relatives à l'IANA.....	14
7.1 Enregistrement du type de contenu Message/CPIM.....	14
7.2 Enregistrement de urn:ietf:params:cpim-headers.....	15
8. Considérations d'internationalisation.....	15
9. Considérations pour la sécurité.....	15
10. Remerciements.....	15
11. Références.....	15

11.1 Références normatives.....	15
11.2 Références pour information.....	16
12. Adresse des auteurs.....	17
13. Déclaration complète de droits de reproduction.....	17

## 1. Introduction

Le présent mémoire définit le type de contenu MIME 'Message/CPIM', un format de message pour les protocoles qui se conforment à la spécification du profil commun pour la messagerie instantanée (CPIM, *Common Profile for Instant Messaging*). C'est un format de message commun pour les protocoles de messagerie conformes à CPIM [RFC3860].

Bien que préparé pour CPIM, ce format est assez général et peut être réutilisé par d'autres applications qui ont des exigences similaires. Les spécifications d'application qui adoptent ceci comme format de base devraient satisfaire aux questions soulevées dans la section 6 du présent document.

### 1.1 Motivation

La spécification du profil commun pour la messagerie instantanée (CPIM, *Common Profile for Instant Messaging*) [RFC3860] définit un certain nombre d'opérations à prendre en charge et de critères à satisfaire pour l'interfonctionnement entre divers protocoles de messagerie instantanée. L'intention est de permettre une diversité de protocoles interfonctionnant entre des passerelles pour prendre en charge la messagerie inter protocoles et satisfaisant aux exigences de la [RFC2779].

Pour satisfaire de façon adéquate aux exigences de sécurité de la RFC 2779, un format de message commun est nécessaire afin que puissent être appliquées des signatures et un chiffrement de bout en bout. Le présent document décrit un format canonique commun de message qui doit être utilisé par tout protocole de transfert de message conforme à CPIM, par lequel les signatures sont calculées pour la sécurité de bout en bout.

Le concept de ce format de message est destiné à permettre d'appliquer la sécurité, tout en restant lui-même extérieur aux mécanismes de sécurité spécifiques qui peuvent être appropriés pour une certaine application. Pour la messagerie instantanée CPIM et la présence, les protocoles de sécurité sont spécifiés par les spécifications de messagerie instantanée CPIM [RFC3860] et la présence CPIM [RFC3859].

Noter aussi que le format de message décrit ici n'est pas lui-même un format de données MIME, bien qu'il puisse être contenu dans un objet MIME, et puisse contenir des objets MIME. Voir les détails à la Section 2.

### 1.2 Fondements

La RFC 2779 exige qu'un message instantané puisse porter une charge utile MIME [RFC2045], [RFC2046] ; donc un certain niveau de prise en charge de MIME sera un élément commun de tout protocole conforme à CPIM. Il semble donc raisonnable qu'un format de message commun doive utiliser une syntaxe du style de celle de la RFC2822/MIME, car les mises en œuvre de protocole doivent déjà contenir le code pour l'analyser.

Malheureusement, l'utilisation de la pure syntaxe RFC2822/MIME peut poser des problèmes :

- o Structure lexicale irrégulière – la RFC2822/MIME permet un certain nombre de codages facultatifs et plusieurs façons de coder une valeur particulière. Par exemple, les commentaires RFC2822/MIME peuvent être codés de plusieurs façons. Pour des raisons de sécurité, une seule méthode de codage doit être définie comme base de calcul des valeurs de résumé de message. Les protocoles qui transmettent des données dans un format différent perdraient les informations nécessaires pour vérifier une signature.
- o Faible internationalisation – la RFC2822/MIME exige que les valeurs d'en-tête utilisent de l'ASCII à 7 bits, ce qui pose problème pour le codage des jeux de caractères internationaux. Des mécanismes pour l'étiquetage de langue dans les en-têtes RFC2822/MIME [RFC2231] sont difficiles à utiliser et ont une applicabilité limitée.
- o Variabilité – l'ajout, la modification ou la suppression des informations d'en-tête. Parce que ce n'est pas explicitement interdit, de nombreuses applications qui traitent le contenu MIME (par exemple, les passerelles MIME) reconstruisent ou restructurent les messages en transit. Cela oblitère la plupart des tentatives de réaliser la sécurité (par exemple, des signatures) rendant les applications receveuses incapables de vérifier les données reçues.
- o Séparation message charge utile – il n'y a pas de claire distinction syntaxique entre les métadonnées de message et le contenu de message.
- o Extensibilité limitée. (Les en-têtes X- sont problématiques parce qu'ils ne peuvent pas être normalisés ; cela conduit à

des situations où un en-tête commence comme expérimental mais trouve ensuite une large application, résultant en un usage courant qui ne peut être normalisé.)

- o Pas de prise en charge d'informations structurées (seulement des valeurs de chaîne de texte).
- o Certains processeurs imposent des limitations de longueur de ligne.

Le format de message défini par le présent mémoire surmonte certaines de ces difficultés en ayant une syntaxe simplifiée qui est généralement compatible avec le format accepté par les analyseurs RFC2822/MIME et en ayant une syntaxe plus stricte. Il définit aussi des mécanismes pour prendre en charge certaines caractéristiques désirées non couvertes par les spécifications de format RFC2822/MIME.

### 1.3 Objectifs

La présente spécification vise à satisfaire les objectifs suivants :

- o un format sécurisable de bout en bout pour un message (un format de message canonique pour servir de base au calcul de signature, plutôt que pour spécifier des mécanismes de sécurité) ;
- o indépendance de toute application spécifique ;
- o capacité de porter une gamme de différents types d'adresses ;
- o hypothèse d'un protocole de transfert de message de 8 bits stricts ;
- o évolutif : extensible par plusieurs parties ;
- o une claire séparation des métadonnées de message et du contenu de message ;
- o une syntaxe simple, régulière, facile à analyser ;
- o un format compact, à faible redondance pour les messages simples.

### 1.4 Terminologie et conventions

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDÉ", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC 2119].

Note : Les commentaires comme celui-ci apportent des informations supplémentaires non essentielles sur les raisons qui sous-tendent le présent document. De telles informations ne sont pas nécessaires pour construire une mise en œuvre conforme, mais peuvent aider ceux qui souhaitent comprendre plus en profondeur sa conception.

## 2. Structure globale de message

Le format de message CPIM encapsule un contenu de message MIME arbitraire, ainsi que des métadonnées de message et relatives au contenu. Cela peut facultativement être signé ou chiffré en utilisant les multiparties MIME de sécurité en conjonction avec un schéma de sécurité approprié.

Un objet Message/CPIM est une entité en deux parties, où la première partie contient les métadonnées de message et la seconde partie est le contenu de message. Les deux parties sont séparées des champs d'en-tête MIME englobant et aussi de chacune d'entre elles par des lignes blanches. Les informations d'en-tête de métadonnées de message obéissent à des règles de syntaxe plus strictes que les en-têtes de contenu de message MIME qui peuvent être portés au sein du message.

Un message complet ressemble à quelque chose comme :

```
m: Content-type: Message/CPIM
s:
h: (en-têtes de métadonnées de message)
s:
e: (corps de message MIME encapsulé)
```

La fin d'un corps de message est définie par le mécanisme de tramage du protocole utilisé. Les étiquettes 'm:', 's:', 'h:', 'e:', et 'x:' ne font pas partie du format de message et sont utilisées ici pour indiquer les différentes parties du message, donc :

```
m: en-têtes MIME pour le message global
s: ligne blanche de séparation
h: en-têtes de message
e: objet MIME encapsulé contenant le contenu de message
```

x: enveloppe de message multiparties de sécurité MIME

## 2.1 En-têtes MIME de message/CPIM

Les en-têtes de message MIME identifient le message comme un message de format CPIM.

Le seul en-tête MIME exigé est :

Content-type: Message/CPIM

D'autres en-têtes MIME peuvent être utilisés comme approprié pour l'environnement de transfert de message.

## 2.2 En-têtes de message

Les en-têtes de message portent des informations pertinentes pour le transfert de bout en bout du message de l'expéditeur au destinataire. Les en-têtes de message NE DOIVENT PAS être modifiés, reformattés ou réarrangés dans le transit, mais dans certaines circonstances ils PEUVENT être examinés par un protocole de transfert de message CPIM.

Les en-têtes de message servent un but similaire à celui des en-têtes de message de la messagerie de la [RFC2822], et ont une syntaxe similaire mais restreinte.

La syntaxe de base d'en-tête est :

Clé: Valeur

où "Clé" est un nom d'en-tête et "Valeur" est la valeur d'en-tête correspondante.

Les considérations suivantes s'appliquent :

- o L'en-tête entier DOIT être contenu sur une seule ligne. La terminaison de ligne n'est pas considérée comme faisant partie de la valeur d'en-tête.
- o Un seul en-tête par ligne. Plusieurs en-têtes NE DOIVENT PAS être inclus sur une seule ligne.
- o Les processeurs NE DEVRAIENT PAS imposer de limitation de longueur de ligne.
- o Il NE DOIT PAS y avoir d'espace au début ou à la fin d'une ligne.
- o Le codage de caractères UTF-8 [RFC3629] DOIT être utilisé tout du long.
- o La séquence de caractères CR,LF (13,10) DOIT être utilisée pour terminer chaque ligne.
- o Le nom d'en-tête contient seulement des caractères US-ASCII (voir la syntaxe spécifique aux paragraphes 3.1 et 3.6).
- o L'en-tête NE DOIT PAS contenir de caractère de contrôle (0-31). Si une valeur d'en-tête doit représenter un caractère de contrôle, on DOIT utiliser le mécanisme d'échappement décrit ci-dessous.
- o Il DOIT y avoir un seul caractère espace (32) à la suite d'un nom d'en-tête et de deux points.
- o Plusieurs en-têtes qui utilisent la même clé (nom d'en-tête) sont permis. (La sémantique spécifique de l'en-tête peut imposer qu'il n'y ait qu'une seule occurrence d'un certain en-tête.)
- o Les noms d'en-tête DOIVENT correspondre exactement (c'est-à-dire, "From:" et "from:" sont des en-têtes différents).
- o Si un nom d'en-tête n'est pas reconnu ou compris, l'en-tête devrait être ignoré. Mais voir aussi l'en-tête "Require:" (paragraphe 4.7).
- o L'interprétation (par exemple, l'équivalence) des valeurs d'en-tête dépend de la définition de l'en-tête particulier. Les processeurs de message DOIVENT préserver exactement tous les octets de tous les en-têtes (nom et valeur).
- o Les processeurs de message NE DOIVENT PAS changer l'ordre des en-têtes de message.

Exemples :

```
To: Pooh Bear <im:pooh@100akerwood.com>  
From: <im:piglet@100akerwood.com>  
DateTime: 2001-02-02T10:48:54-05:00
```

## 2.3 Mécanisme d'échappement de caractère

Ce mécanisme DOIT être utilisé pour coder les caractères de contrôle dans un en-tête, avec les codets Unicode dans la gamme U+0000 à U+001f ou U+007f. (Plutôt que d'inventer quelque chose de complètement nouveau, le mécanisme d'échappement a été adopté à partir de celui utilisé par le langage de programmation Java.)

Noter que le mécanisme d'échappement est appliqué aux caractères UCS-2, NON aux octets de son codage UTF-8. La transposition du/en codage UTF-8 est effectuée sans considération pour les séquences d'échappement ou le codage de caractères. (La syntaxe d'en-tête est définie de telle sorte que les octets qui correspondent aux caractères de contrôle autres que CR et LF n'apparaissent pas dans le résultat.)

Un caractère UCS-2 arbitraire est échappé en utilisant la forme :

`\uxxxx`

où :

`\` est U+005c (barre oblique inverse)

`u` est U+0075 (u minuscule)

`xxxx` est une séquence d'exactly quatre chiffres hexadécimaux (0-9, a-f ou A-F) ou (U+0030-U+0039, U+0041-U+0046, ou U+0061-0066)

Le nombre hexadécimal 'xxxx' est la valeur du codet UCS du caractère échappé.

De plus, les séquences spéciales suivantes introduites par "\" sont utilisées :

`\\` pour `\` (barre oblique inverse, U+005c)

`\"` pour `"` (guillemets, U+0022)

`\'` pour `'` (apostrophe, U+0027)

`\b` pour espace arrière (U+0008)

`\t` pour tabulation (U+0009)

`\n` pour saut à la ligne (U+000a)

`\r` pour retour chariot (U+000d)

### 2.3.1 Utilisation de mécanisme d'échappement

Lorsque on génère des messages se conformant à la présente spécification :

- o Les séquences spéciales mentionnées ci-dessus DOIVENT être utilisées pour coder toute occurrence des caractères suivants qui apparaissent n'importe où dans un en-tête : barre oblique inverse (U+005c), espace arrière (U+0008), tabulation (U+0009), saut à la ligne (U+000a) ou retour chariot (U+000d).
- o La séquence spéciale `\"` DOIT être utilisée pour toute occurrence de guillemets (U+0022) qui apparaît au sein d'une chaîne délimitée par des guillemets.
- o La séquence spéciale `\'` DOIT être utilisée pour toute occurrence d'une apostrophe (U+0027) qui apparaît au sein d'une chaîne délimitée par des apostrophes.
- o Les caractères guillemets ou apostrophe qui délimitent une valeur de chaîne NE DOIVENT PAS être échappés.
- o La séquence d'échappement générale `\uxxxx` DOIT être utilisée pour tout autre caractère de contrôle (U+0000 à U+0007, U+000b à U+000c, U+000e à U+000f ou u+007f) qui apparaît n'importe où dans un en-tête.
- o Aucun autre caractère NE DOIT être représenté en utilisant une séquence d'échappement.

Lorsque on traite un message fondé sur la présente spécification, l'usage de la séquence d'échappement décrite ci-dessus DOIT être reconnu.

De plus, toute autre occurrence d'une séquence échappée décrite ci-dessus DEVRAIT être reconnue et traitée comme une occurrence du caractère Unicode correspondant.

Tout caractère barre oblique inverse (`\`) DEVRAIT être interprété comme introduisant une séquence d'échappement. Toute séquence d'échappement non reconnue DEVRAIT être traitée comme une instance du caractère suivant le caractère barre oblique inverse. Une barre oblique inverse isolée qui est le dernier caractère d'un en-tête DEVRAIT être ignorée.

## 2.4 Contenu du message

La section finale d'un Message/CPIM est le contenu de message encapsulé dans MIME, qui suit les règles standard de formatage MIME [RFC2045], [RFC2046].

Les en-têtes de contenu MIME DOIVENT inclure au moins un en-tête Content-Type. Le contenu peut être tout type MIME.

Exemple :

```
e: Content-Type: text/plain; charset=utf-8
e: Content-ID: <1234567890@foo.com>
e:
e: C'est mon contenu de message de texte encapsulé
```

### 3. Syntaxe de l'en-tête de message

Un en-tête contient deux parties, un nom et une valeur, séparées par un caractère deux points (':') et une seule espace (32). Il est terminé par la séquence CR,LF (13,10).

Les en-têtes utilisent le codage de caractère UTF-8 tout du long, selon la [RFC3629].

Note : dans les descriptions qui suivent, les noms de champ d'en-tête et les autres valeurs de texte spécifiées DOIVENT être utilisées exactement comme ils sont donnés, en utilisant exactement les casses de lettre indiquées en majuscules et minuscules. À cet égard, l'usage de l'ABNF diffère de celui de la [RFC2234].

#### 3.1 Noms d'en-tête

Le nom d'en-tête est une séquence de caractères US-ASCII, excluant les caractères de contrôle, d'espace ou les séparateurs. L'utilisation du caractère "." dans un nom d'en-tête est réservée pour un séparateur de préfixe d'espace de nom.

Les caractères séparateurs sont :

```
SEPARATORS = "(" / ")" / "<" / ">" / "@" / "," / ";" / ":" / "\" / DQUOTE / "/" / "[" / "]" / "?" / "=" / "{" / "}" / SP
```

Note : la gamme de caractères permis a été déterminée par l'examen des formats de nom d'en-tête de HTTP et de la RFC2822 en choisissant le plus restreint. L'intention est de permettre que les en-têtes CPIM suivent une syntaxe compatible avec la syntaxe permise par les deux [RFC2822] et HTTP [RFC2616] (incluant les protocoles dérivés de HTTP tels que SIP [RFC3261]).

#### 3.2 Valeur d'en-tête

Une valeur d'en-tête a une structure définie par la spécification d'en-tête correspondante. Les mises en œuvre qui utilisent un en-tête particulier doivent adhérer aux règles de format et d'usage ainsi définies lorsque elles créent ou traitent un message contenant cet en-tête.

Les autres contraintes générales des formats d'en-tête DOIVENT aussi être suivies (une ligne, codage de caractères UTF-8, pas de caractères de contrôle, etc.)

#### 3.3 Étiquetage de langage

Une pleine internationalisation d'un protocole exige qu'un langage puisse être indiqué pour tout texte lisible par l'homme [RFC2130], [RFC2277].

Un en-tête de message peut indiquer un langage pour sa valeur en incluant ';lang=tag' après le nom d'en-tête et les deux points, où 'tag' est un jeton d'identification de langage selon la [RFC3066].

Exemple :

```
Subject::lang=fr Objet de message
```

Si le paramètre langage n'est pas appliqué à un en-tête, tout texte lisible par l'homme est supposé utiliser le langage identifié comme 'i-default' [RFC2277].

### 3.4 Espaces de noms pour l'extensibilité des noms d'en-tête

Note : ce paragraphe définit un cadre pour l'extensibilité d'en-tête dont l'utilisation est facultative. Si aucune extension d'en-tête n'est permise par une application, ces structures peuvent alors n'être jamais utilisées.

Une application qui utilise ce format de message est supposée définir l'ensemble des en-têtes qui sont requis et permis pour cette application. Ce paragraphe définit un cadre d'extensibilité d'en-tête qui peut être utilisé avec toute application.

Le cadre d'extensibilité se fonde sur celui fourni pour XML [XML1.0] par les espaces de noms XML [XMLNAMES]. Tous les en-têtes sont associés à un "namespace" (*espace de noms*), qui est à son tour associé à un URI unique au monde.

Dans une instance de message particulière, les noms d'en-tête sont associés à un espace de noms particulier par la présence ou l'absence d'un préfixe d'espace de noms, qui est la première partie du nom d'en-tête suivi par un point (".") ; par exemple, préfixe.nom d'en-tête: valeur d'en-tête

Ici, 'préfixe' est le nom du préfixe d'en-tête, 'nom d'en-tête' est le nom de l'en-tête au sein de l'espace de noms associé à 'préfixe', et 'valeur d'en-tête' est la valeur pour cet en-tête.

nom d'en-tête: valeur d'en-tête

Dans ce cas, le préfixe du nom d'en-tête est absent, et le 'nom d'en-tête' donné est associé à un espace de noms par défaut.

L'enregistrement de type de support Message/CPIM désigne un espace de noms par défaut pour tous les en-têtes qui ne sont pas plus explicitement associés à un espace de noms. Dans la plupart des cas, cet espace de noms par défaut est tout ce qui est nécessaire.

Un espace de noms est identifié par un URI. Dans cet usage, l'URI est utilisé simplement comme un identifiant unique au monde, et il n'est pas exigé qu'il puisse être utilisé pour autre chose. Tout URI légal unique au monde PEUT être utilisé pour identifier un espace de noms. (Par "unique au monde", on veut dire construit selon un ensemble de règles telles qu'il soit raisonnable de s'attendre à ce que personne d'autre n'utilise le même URI pour un objet différent.) Un URI utilisé comme identifiant DOIT être un plein URI absolu, selon la [RFC2396]. (Les URI relatifs et les URI références qui contiennent des fragments d'identifiant NE DOIVENT PAS être utilisés à cette fin.)

Au sein d'un message spécifique, un en-tête 'NS' est utilisé pour déclarer un préfixe d'espace de noms et l'associer avec un URI qui identifie un espace de noms. Suivant cette déclaration, dans la portée de ce message, la combinaison du préfixe d'espace de noms et du nom d'en-tête indique un identifiant unique au monde pour l'en-tête (consistant en l'URI d'espace de noms et du nom d'en-tête).

Par exemple :

```
NS: MyFeatures <mid:MessageFeatures@id.foo.com>
MyFeatures.WackyMessageOption: Use-silly-font
```

Ceci définit un préfixe d'espace de noms 'MyFeatures' associé à l'identifiant d'espace de noms 'mid:MessageFeatures@id.foo.com'. Par conséquent, le préfixe indique que le nom d'en-tête WackyMessageOption référencé est associé à l'espace de noms identifié.

Une déclaration de préfixe d'espace de noms DOIT précéder toute utilisation de ce préfixe.

À l'exception de tous préfixes d'espace de noms prédéfinis spécifiques de l'application (voir la Section 6) un préfixe d'espace de noms est strictement local pour le message dans lequel il survient. Le préfixe réel utilisé n'a pas de signification mondiale. Cela signifie que les en-têtes :

```
xxx.nom : valeur
yyy.nom: valeur
```

dans deux messages différents peuvent avoir exactement le même effet si les préfixes d'espace de noms 'xxx' et 'yyy' sont associés au même URI d'espace de noms. Donc, ce qui suit a exactement la même signification :

```
NS: acme <http://id.acme.widgets/wily-headers/>
acme.runner-trap: set
```

et

```
NS: widget <http://id.acme.widgets/wily-headers/>
widget.runner-trap: set
```

Un en-tête 'NS' sans un nom de préfixe d'en-tête spécifie un espace de noms par défaut pour les en-têtes suivants ; c'est-à-dire un espace de noms qui est associé à des noms d'en-tête qui n'ont pas de préfixe. Par exemple :

```
NS: <http://id.acme.widgets/wily-headers/>
runner-trap: set
```

a la même signification que les exemples précédents.

Ce cadre permet à des mises en œuvre différentes de créer des en-têtes d'extension sans se soucier des duplications de nom d'en-tête ; chacune définit des en-têtes au sein de son propre espace de noms.

### 3.5 Caractéristiques qu'il est obligatoire de reconnaître

Il est parfois nécessaire que l'expéditeur d'un message insiste pour qu'une fonctionnalité soit comprise par le receveur. En utilisant l'indicateur "reconnaissance obligatoire", un expéditeur notifie au receveur qu'il DOIT comprendre l'en-tête ou la caractéristique désignée afin de comprendre correctement le message.

Un en-tête ou une caractéristique est indiquée comme de reconnaissance obligatoire par un en-tête 'Require:'. Par exemple :

```
Require: MyFeatures.VitalMessageOption
MyFeatures.VitalMessageOption: Confirmation-requested
```

Plusieurs noms d'en-tête exigés peuvent être cités dans un seul en-tête 'Require', séparés par des virgules.

Note : une utilisation abusive des en-têtes 'Require:' pourrait nuire à l'interopérabilité. On suggère que toute mise en œuvre qui définit des en-têtes exigés publie aussi la spécification d'en-têtes afin que les autres mises en œuvre puissent interopérer avec succès.

L'en-tête 'Require:' PEUT aussi être utilisé pour indiquer qu'une sémantique autre que d'en-tête doit être mise en œuvre par le receveur, même lorsque il n'apparaît pas comme un en-tête. Par exemple :

```
Require: Locale.MustRenderKanji
```

peut être utilisé pour indiquer que le contenu de message comporte des caractères du répertoire Kanji, qui doivent être rendus pour une bonne compréhension du message. Dans ce cas, le nom d'en-tête est juste un jeton (utilisant la syntaxe de nom d'en-tête et l'association d'un espace de noms) qui indique un comportement désiré.

### 3.6 Syntaxe collectée d'en-tête de message

La description suivante de la syntaxe d'en-tête de message utilise l'ABNF, selon la [RFC2234]. La plupart de cette syntaxe peut être interprétée comme définissant des séquences de caractères UCS ou des séquences d'octets UTF-8. D'autres productions à la fin permettent l'une ou l'autre interprétation.

Note : les valeurs de texte spécifiées DOIVENT être utilisées comme elles sont données, en utilisant exactement les lettres en majuscule et minuscules indiquées. À cet égard, l'usage de l'ABNF diffère ici de celui de la [RFC2234].

Récapitulation de la syntaxe :

En-tête = Nom d'en-tête ":" \*( ";" Paramètre ) SP valeur d'en-tête CRLF

Nom d'en-tête = [ Préfixe-de-nom "." ] Nom

Préfixe-de-nom = Nom

Paramètre = Paramètre-de-langue / Ext-param

Paramètre-de-langue = "langue=" Étiquette-de-langue

Ext-param = Nom-de-paramètre "=" Valeur-de-paramètre

Nom-de-paramètre = Nom



Valeur-de-paramètre = Jeton / Nombre / Chaîne

Valeur-d'en-tête = \*HEADERCHAR

Nom = 1\*NAMECHAR

Jeton = 1\*TOKENCHAR

Nombre = 1\*CHIFFRE

Chaîne = DQUOTE \*( Chaîne-de-caractères / Echappement ) DQUOTE

Chaîne-de-caractères = %x20-21 / %x23-5B / %x5D-7E / UCS-high

Echappement = "\" ( "u" 4(HEXDIG) ; codet UCS  
 / "b" ; espace arrière  
 / "t" ; tabulation  
 / "n" ; saut à la ligne  
 / "r" ; retour  
 / DQUOTE ; guillemets  
 / "'" ; apostrophe  
 / "\" ) ; barre oblique inverse

Nom-formel = 1\*( Jeton SP ) / Chaîne

URI = <défini comme URI absolu par la RFC 2396>

Etiquette-de-langage = <définie par la RFC 3066>

; tout caractère UCS sauf les CTL, ou HEADERCHAR d'échappement = UCS-no-CTL / Echappement

; tout caractère US-ASCII sauf ".", CTL ou SEPARATEURS: NAMECHAR = %x21 / %x23-27 / %x2a-2b / %x2d / %x5e-60 / %x7c / %x7e / ALPHA / CHIFFRE

; tout caractère UCS sauf les CTL ou SEPARATEURS : TOKENCHAR = NAMECHAR / "." / UCS-high

SEPARATEURS = "(" / ")" / "<" / ">" / "@" ; 28/29/3c/3e/40  
 / ";" / ":" / "." / "\" / DQUOTE ; 2c/3b/3a/5c/22  
 / "/" / "[" / "]" / "?" / "=" ; 2f/5b/5d/3f/3d  
 / "{" / "}" / SP ; 7b/7d/20

CTL = <Défini par la RFC 2234 -- %x0-%x1f, %x7f>

CRLF = <Défini par la RFC 2234 -- CR, LF>

SP = <défini par la RFC 2234 -- %x20>

CHIFFRE = <défini par la RFC 2234 -- '0'-'9'>

HEXDIG = <défini par la RFC 2234 -- '0'-'9', 'A'-'F', 'a'-'f'>

ALPHA = <défini par la RFC 2234 -- 'A'-'Z', 'a'-'z'>

DQUOTE = <défini par la RFC 2234 -- %x22>

Pour interpréter la syntaxe dans un environnement général de caractères UCS, on utilise les productions suivantes :

UCS-no-CTL = %x20-7e / UCS-high

UCS-high = %x80-7fffffff

Pour interpréter la syntaxe comme définissant des séquences d'octets codées en UTF-8, utiliser les productions suivantes :

UCS-no-CTL = UTF8-no-CTL

UCS-high = UTF8-multi

UTF8-no-CTL = %x20-7e / UTF8-multi

UTF8-multi = %xC0-DF %x80-BF / %xE0-EF %x80-BF %x80-BF / %xF0-F7 %x80-BF %x80-BF %x80-BF / %xF8-FB %x80-BF %x80-BF %x80-BF %x80-BF / %xFC-FD %x80-BF %x80-BF %x80-BF %x80-BF

Note : la syntaxe ci-dessus vient d'une plus ancienne version de l'UTF-8, et est incluse pour la compatibilité avec le logiciel UTF-8 fondé sur les spécifications antérieures. Les applications qui génèrent ce format de message DEVRAIENT générer un UTF-8 qui correspond à la spécification plus restreinte de la [RFC3629].

#### 4. Définitions d'en-tête

La présente spécification définit un ensemble cœur d'en-têtes qui sont disponibles pour les applications : une spécification d'application doit indiquer les en-têtes qui peuvent être utilisés, ceux qui doivent être reconnus, et ceux qui doivent apparaître dans tout message (voir la section 6).

Les définitions d'en-tête qui suivent entrent dans deux catégories :

- a) celles qui font partie du cadre d'extensibilité de format CPIM, et
- b) celles qui se fondent sur des en-têtes similaires de la [RFC2822], spécifiées ici avec la sémantique correspondante.

Les noms et la syntaxe d'en-tête sont décrits sans qualification d'espace de noms, et l'URI d'espace de noms associé est cité au titre de la spécification d'en-tête. Toutes les associations d'espace de noms déjà mentionnées (espace de nom par défaut implicite, espace de nom par défaut explicite ou préfixe d'espace de nom implicite ou déclaration de préfixe d'espace de nom explicite) peuvent être utilisées pour identifier l'espace de noms.

Tous les en-têtes définis ici sont associés à l'URI d'espace de noms <urn:ietf:params:cpim-headers:>, qui est défini selon la [RFC3353].

Note : les noms d'en-tête et autres textes DOIVENT être utilisés comme ils sont donnés, en utilisant exactement les lettres en majuscules et minuscules indiquées. À cet égard, l'usage de l'ABNF est ici différent de celui de la [RFC2234].

#### 4.1 En-tête 'From'

Indique l'expéditeur d'un message.

Nom d'en-tête : From

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi le paragraphe 3.6)

en-tête From = "From" ":" [ Nom-formel ] "<" URI ">" ; "From" est sensible à la casse

Description : indique l'expéditeur ou l'origine d'un message.

Si il est présent, le 'Nom-formel' identifie la personne ou le nom "réel" de l'origine.

L'URI indique une adresse pour l'origine.

Exemples :

From: Winnie l'ourson <im:pooh@100akerwood.com>

From: <im:tigger@100akerwood.com>

#### 4.2 En-tête 'To'

Spécifie un receveur prévu pour un message.

Nom d'en-tête : To

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi le paragraphe 3.6)

En-tête To = "To" ":" [ Nom-formel ] "<" URI ">" ; "To" est sensible à la casse

Description : Indique le receveur d'un message.

Si il est présent, le 'Nom-formel' identifie la personne ou le nom "réel" du receveur.

L'URI indique une adresse pour le receveur.

Plusieurs receveurs peuvent être indiqués en incluant plusieurs en-têtes 'To'.

Exemples :

To: Winnie l'ourson <im:pooh@100akerwood.com>

To: <im:tigger@100akerwood.com>

#### 4.3 En-tête 'cc'

Spécifie un receveur autre que le principal ("copie de courtoisie") pour un message.

Nom d'en-tête : cc

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi au paragraphe 3.6)

En-tête Cc = "cc" ":" [ Nom-formel ] "<" URI ">" ; "cc" est sensible à la casse

Description : Indique un receveur d'une copie de courtoisie d'un message.

Si il est présent, le 'Nom-formel' identifie la personne ou le nom "réel" du receveur.

L'URI indique une adresse pour le receveur.

Plusieurs receveurs de copie de courtoisie peuvent être indiqués en incluant plusieurs en-têtes 'cc'.

Exemples :

cc: Winnie l'ourson <im:pooh@100akerwood.com>

cc: <im:tigger@100akerwood.com>

#### 4.4 En-tête 'DateTime'

Spécifie la date et l'heure d'envoi d'un message.

Nom d'en-tête : DateTime

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi au paragraphe 3.6)

En-tête DateTime = "DateTime" ":" " date-time" ; "DateTime" est sensible à la casse

(la syntaxe de 'date-time' est un profil de [ISO8601] défini dans "Date et heure sur l'Internet" [RFC3339])

Description : l'en-tête 'DateTime' fournit la date et l'heure à laquelle l'expéditeur a expédié le message.

Un des objets de cet en-tête est de fournir une protection contre une attaque en répétition, en permettant au receveur de savoir quand il était prévu que le message soit envoyé. La valeur de l'en-tête date est l'heure en cours chez l'expéditeur lorsque le message a été transmis, en utilisant le format de date et heure de [ISO8601] selon le profil décrit dans la [RFC3339].

Exemple :

DateTime: 2001-02-01T12:16:49-05:00

#### 4.5 En-tête 'Subject'

Contient une description du sujet du message.

Nom d'en-tête : Subject

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi au paragraphe 3.6)

En-tête Subject = "Subject" ":" [ ";" paramètre de langage ] SP \*HEADERCHAR ; "Subject" est sensible à la casse

Description : l'en-tête 'Subject' fournit la description par l'expéditeur du sujet ou du contenu du message.

L'agent expéditeur devrait spécifier le paramètre de langage si il a une connaissance raisonnable de la langue utilisée par l'expéditeur pour indiquer le sujet du message.

Exemple :

Subject:;lang=fr Marie se sent très mal aujourd'hui

#### 4.6 En-tête 'NS'

Déclare un préfixe d'espace de noms local.

Nom d'en-tête : NS

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi au paragraphe 3.6)

En-tête NS = "NS" ":" [ Préfixe-de-nom ] "<" URI ">" ; "NS" est sensible à la casse

Description : Déclare un préfixe d'espace de noms qui peut être utilisé dans les noms d'en-tête suivants. Voir les détails au paragraphe 3.4.

Exemple :

NS: MyAlias <mid:MessageFeatures@id.foo.com>

MyAlias.MyHeader: private-extension-data

#### 4.7 En-tête 'Require'

Spécifie un en-tête ou une caractéristique qui doit être mise en œuvre par le receveur pour un traitement correct du message.

Nom d'en-tête : Require

URI d'espace de noms : <urn:ietf:params:cpim-headers:>

Syntaxe : (voir aussi au paragraphe 3.6)

En-tête Require = "Require" ":" Nom-d'en-tête \*( "," Nom-d'en-tête ) ; "Require" est sensible à la casse

Description : Indique un en-tête ou une caractéristique qui doit être mise en œuvre ou comprise par le receveur pour un traitement correct du message. Voir les détails au paragraphe 3.5.

Noter que l'en-tête ou la caractéristique exigée n'a pas à être utilisée dans le message, mais dans un souci de concision, il est recommandé qu'une mise en œuvre ne produise pas l'en-tête 'Required' pour des caractéristiques non utilisées.

Exemple :

Require: MyAlias.VitalHeader

## 5. Exemples

Les exemples de la présente section utilisent les étiquettes par ligne ci-dessous pour indiquer les différentes parties du format de message global

m : en-têtes MIME pour le message global

s : une ligne blanche de séparation

h : en-tête de messages

e : objet MIME encapsulé qui contient le contenu du message

x : enveloppe MIME de message multiparties de sécurité

Les exemples suivants supposent aussi que <urn:ietf:params:cpim-headers:> est l'espace de noms implicite par défaut pour l'application.

### 5.1 Exemple de message Message/CPIM

L'exemple suivant montre un message Message/CPIM :

m: Content-type: Message/CPIM

s:

h: From: MR SANDERS <im:piglet@100akerwood.com>

h: To: Ane déprimé <im:eevore@100akerwood.com>

h: DateTime: 2000-12-13T13:40:00-08:00

h: Subject: il fera beau aujourd'hui

h: Subject:;lang=fr beau temps prévu pour aujourd'hui

h: NS: MyFeatures <mid:MessageFeatures@id.foo.com>

h: Require: MyFeatures.VitalMessageOption

h: MyFeatures.VitalMessageOption: Confirmation-requested

h: MyFeatures.WackyMessageOption: Use-silly-font

```
s:
e: Content-type: text/xml; charset=utf-8
e: Content-ID: <1234567890@foo.com>
e:
e: <body>
e: Voici le texte de mon message.
e: </corps du message>
```

## 5.2 Exemple utilisant MIME multipart/signed

Afin de sécuriser un message/CPIM, une application ou une mise en œuvre peut utiliser la [RFC1847], et des protocoles de sécurité appropriés (par exemple, S/MIME [RFC2633] ou openPGP [RFC2440]), et un schéma de chiffrement.

En utilisant S/MIME [RFC2633] et pkcs7, le message ci-dessus ressemblerait à ceci :

```
x: Content-Type: multipart/signed; boundary=next; micalg=sha1; protocol=application/pkcs7-signature
x:
x: --next
m: Content-Type: Message/CPIM
s:
h: From: MR SANDERS <im:piglet@100akerwood.com>
h: To: Dopey Donkey <im:eevore@100akerwood.com>
h: DateTime: 2000-12-13T13:40:00-08:00
h: Subject: il fera beau aujourd'hui
h: Subject::lang=fr beau temps prévu pour aujourd'hui
h: NS: MyFeatures <mid:MessageFeatures@id.foo.com>
h: Require: MyFeatures.VitalMessageOption
h: MyFeatures.VitalMessageOption: Confirmation-requested
h: MyFeatures.WackyMessageOption: Use-silly-font
s:
e: Content-type: text/xml; charset=utf-8
e: Content-ID: <1234567890@foo.com>
e:
e: <body>
e: Voici le texte de mon message.
e: </corps du message>
x: --next
x: Content-Type: application/pkcs7-signature
x:
x: (matériel de signature)
:
x: --next--
```

## 6. Considérations de conception des applications

Comme défini, le type de contenu 'Message/CPIM' utilise un URI d'espace de noms par défaut de 'urn:ietf:params-cpim-headers:', et ne définit aucun autre préfixe implicite d'espace de noms. Les applications qui ont des exigences différentes devraient définir et enregistrer un type différent de support MIME, spécifier l'URI d'espace de nom par défaut exigé et définir tous les préfixes d'espace de nom impliqués au titre de la spécification de type de support.

Les applications qui utilisent cette spécification doivent aussi spécifier :

- o tous les en-têtes qui doivent être reconnus par les mises en œuvre de l'application,
- o tous les en-têtes qui doivent être présents dans tous les messages créés par cette application,
- o tous les en-têtes qui peuvent apparaître plus d'une fois dans un message, et comment ils doivent être interprétés (par exemple, comment interpréter plusieurs en-têtes 'Subject:' avec des valeurs de paramètre de langage différentes),
- o les mécanismes de sécurité et les schémas de chiffrement à utiliser avec l'application, incluant toutes les dispositions de sécurité de mise en œuvre obligatoire.

L'objectif de fourniture d'un format de message définitif auquel puissent être appliqués les mécanismes de sécurité impose des contraintes à la conception des applications qui utilisent ce format de message :

- o Dans un réseau d'agents de transfert de messages, une passerelle intermédiaire NE DOIT changer en aucune façon le contenu du Message/CPIM. Cela implique que ces en-têtes ne peuvent pas être changés ou réordonnés, que le codage de transfert ne peut pas être changé, que les langages ne peuvent pas être changés, etc.
- o Parce que les messages Message/CPIM sont immuables, tout agent de transfert qui veut modifier le message devrait créer un nouveau message Message/CPIM avec l'en-tête modifié et avec le message d'origine comme contenu. (Cette approche est similaire au traitement des connaissances dans le monde du commerce, où chaque personne dans la chaîne attache une nouvelle feuille au message. Tout participant peut valider le message d'origine et voir ce qui a changé et qui l'a changé en suivant le cheminement des amendements. Une autre métaphore est d'inclure le vieux message dans une nouvelle enveloppe.)

En choisissant les mécanismes de sécurité pour une application, les documents de l'IAB suivants peuvent être utiles :

- o Mécanismes de sécurité pour l'Internet [RFC3631]
- o Survol des mécanismes d'authentification [29].

## 7. Considérations relatives à l'IANA

Le présent mémoire appelle deux nouveaux enregistrements de l'IANA :

- o une nouvelle valeur de type de contenu MIME, Message/CPIM, selon la [RFC2048]. Le gabarit d'enregistrement se trouve au paragraphe 7.1.
- o un nouveau sous espace de noms d'URN de l'IANA, urn:ietf:params:cpim-headers:, conformément à la [RFC3353]. Le gabarit d'enregistrement se trouve au paragraphe 7.2.

### 7.1 Enregistrement du type de contenu Message/CPIM

To: ietf-types@iana.org

Sujet : Enregistrement du type de support MIME Message/CPIM

Nom du type de support MIME : message

Nom du sous-type MIME : CPIM

Paramètres exigés : (aucun)

Paramètres facultatifs : (aucun)

Considérations de codage : destiné à être utilisé dans des environnements 8 bits stricts, avec un codage non transformant (8 bits ou binaire, selon le contenu du message ; les en-têtes de message CPIM peuvent être traités dans un environnement de texte de 8 bits). Ce type de contenu pourrait être utilisé avec un environnement de transfert à 7 bits si le codage de transfert approprié est utilisé. Noter qu'à cette fin, le contenu MIME inclus DOIT être traité comme des données opaques et codé en conséquence. Tout codage doit être inversé avant d'accéder à un contenu MIME inclus.

Considérations de sécurité : le contenu peut inclure des données signées, de sorte que tout codage de transfert DOIT être exactement inversé avant le traitement du contenu. Voir aussi les considérations de sécurité pour les messages de la messagerie électronique ([RFC2822]).

Considérations d'interopérabilité : ce format de contenu est destiné à être utilisé pour échanger des messages éventuellement sécurisés entre différents protocoles de messagerie instantanée. Une adhésion très stricte au format de message (incluant l'usage des espaces) peut être nécessaire pour réaliser l'interopérabilité.

Spécification publiée : RFC 3862

Applications qui utilisent ce type de support : messagerie instantanée

Informations supplémentaires : l'URI d'espace de noms par défaut associé à ce type de contenu est 'urn:ietf:params:cpim-headers:'. (Voir les détails dans la [RFC3862].) Voir aussi le profil commun pour la messagerie instantanée (CPIM) [RFC3860].

Adresse personnelle & de messagerie à contacter pour plus d'informations : G. Klyne, <GK-IETF@ninebynine.org>

Utilisation prévue : USAGE LIMITÉ

Auteur/contrôleur des changements : IETF

## 7.2 Enregistrement de urn:ietf:params:cpim-headers

Nom du registre : cpim-headers

Spécification : RFC 3862. Des valeurs supplémentaires peuvent être définies par des RFC en cours de normalisation qui mettent à jour ou rendent obsolète la RFC3862.

Répertoire : <http://www.iana.org/assignments/cpim-headers>

Valeur d'indice : la valeur d'indice est un nom d'en-tête de message CPIM, qui peut consister en une séquence tirée d'un ensemble restreint de caractères US-ASCII, comme défini ci-dessus.

Formation d'URN : l'URI pour un en-tête est formé à partir de son nom par :

- a) le remplacement de tout caractère non URN (comme défini par la [RFC2141]) par la séquence d'échappement correspondante '%hh' (selon la [RFC2396]) ; et
- b) l'ajout devant la chaîne résultante de 'urn:ietf:params:cpim-headers:'.

Ainsi, l'URI correspondant à l'en-tête de message CPIM 'From:' serait 'urn:ietf:params:cpim-headers:From'. L'URI correspondant à l'en-tête de message CPIM (imaginaire) 'Top&Tail' serait 'urn:ietf:params:cpim-headers:Top%26Tail'.

## 8. Considérations d'internationalisation

Les en-têtes de message utilisent toujours le codage de caractère UTF-8 ; donc, ils peuvent porter la totalité du répertoire de caractères UCS-4 (Unicode [UNICODE], ISO/CEI 10646 [UCS]).

L'étiquetage des langues est fourni pour les en-têtes de message qui utilisent le paramètre "Lang" (paragraphe 3.3).

Le contenu de message est tout contenu encapsulé dans MIME, et les considérations d'internationalisation de contenu MIME normales s'appliquent.

## 9. Considérations pour la sécurité

Le format Message/CPIM est conçu en pensant à la sécurité. En particulier, il est conçu pour être utilisé avec les multi-parties de sécurité MIME pour la signature et le chiffrement. À cette fin, les messages Message/CPIM doivent être considérés comme immuables une fois créés.

Parce que les messages Message/CPIM sont des messages binaires (du fait du codage UTF-8) si ils sont transmis à travers des transports non strictement à 8 bits, l'agent de transfert doit alors tunneler le message entier. Changer le codage des données du message n'est pas une option. Cela implique que le Message/CPIM doit être encapsulé par le système de transfert de message et désencapsulé à l'extrémité receveuse du tunnel.

Le message résultant ne doit pas avoir de pertes de données à cause du codage/décodage du message. Par exemple, une application peut choisir d'appliquer le codage de transfert de contenu MIME base64 à l'objet Message/CPIM pour satisfaire cette exigence.

## 10. Remerciements

Les auteurs tiennent à remercier Harald Alvestrand, Walter Houser, Leslie Daigle, Mark Day, et Brian Raymor de leurs utiles commentaires.

## 11. Références

### 11.1 Références normatives

[RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet multi-objets \(MIME\) Partie 1 : Format des corps de message Internet](#)", novembre 1996. (*D. S., MàJ par 2184, 2231, 5335.*)

- [RFC2046] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 2 : Types de support", novembre 1996. (*D. S., MàJ par 2646, 3798, 5147, 6657.*)
- [RFC2048] N. Freed, J. Klensin et J. Postel, "Extensions multi-objets de la messagerie Internet (MIME) Partie 4 : Procédures d'enregistrement", BCP 13, novembre 1996. (*Rendue obsolète par les RFC 4288-4289*)
- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [RFC2141] R. Moats, "[Syntaxe des URN](#)", mai 1997.
- [RFC2234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", novembre 1997. (*Obsolète, voir RFC5234*)
- [RFC2277] H. Alvestrand, "Politique de l'IETF en matière de [jeux de caractères et de langages](#)", BCP 18, janvier 1998.
- [RFC2396] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiants de ressource uniformes (URI) : Syntaxe générique", août 1998. (*Obsolète, voir RFC3986*)
- [RFC2822] P. Resnick, "[Format de message Internet](#)", avril 2001. (*Remplace la RFC0822, STD 11, Remplacée par RFC5322*)
- [RFC3036] L. Andersson et autres, "Spécification de LDP", janvier 2001. (*Rendue obsolète par la RFC5036*)
- [RFC3339] G. Klyne, C. Newman, "[La date et l'heure sur l'Internet](#) : horodatages", juillet 2002. (*P.S.*)
- [RFC3353] D. Ooms et autres, "Vue d'ensemble de la diffusion groupée IP dans un environnement de commutation d'étiquettes multi-protocoles (MPLS)", août 2002. (*Information*)
- [RFC3629] F. Yergeau, "[UTF-8, un format de transformation](#) de la norme ISO 10646", STD 63, novembre 2003.

## 11.2 Références pour information

- [ISO8601] International Organization for Standardization, "Data elements and interchange formats - Information interchange - Representation of dates and times", norme ISO 8601, juin 1988.
- [RFC1847] J. Galvin, S. Murphy, S. Crocker, N. Freed, "[Sécurité multiparties pour MIME](#) : multipartie/signée et multipartie/chiffrée", octobre 1995. (*P.S.*)
- [RFC2130] C. Weider, C. Preston, K. Simonsen, H. Alvestrand, R. Atkinson, M. Crispin, P. Svanberg, "Rapport de l'atelier Jeux de caractères de l'IAB tenu du 29 février au 1<sup>er</sup> mars 1996", avril 1997. (*Information*)
- [RFC2231] N. Freed, K. Moore, "Extensions MIME [Valeur de paramètre et Mot codé](#) : jeux de caractères, langages, et continuations", novembre 1997. (*P.S.*)
- [RFC2440] J. Callas, L. Donnerhackle, H. Finney et R. Thayer, "[Format de message OpenPGP](#)", novembre 1998. (*Obs. voir 4880*)
- [RFC2616] R. Fielding et autres, "[Protocole de transfert hypertexte -- HTTP/1.1](#)", juin 1999. (*D.S., MàJ par 2817, 6585*)
- [RFC2633] B. Rmasdell, "Spécification de message S/MIME version 3", juin 1999. (*Obsolète, voir RFC3851*) (*P.S.*)
- [RFC2779] M. Day et autres, "[Exigences des protocoles Messagerie instantanée / Presence](#)", février 2000. (*Information*)
- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (*Mise à jour par RFC3265, RFC3853, RFC4320, RFC4916, RFC5393, RFC6665*)
- [RFC3631] S. Bellovin, J. Schiller et C. Kaufman, éd., "[Mécanismes de sécurité pour l'Internet](#)", décembre 2003. (*Information*)
- [RFC3859] J. Peterson, "[Profil commun pour les services de présence](#) (CPP)", août 2004. (*P.S.*)



- [RFC3860] J. Peterson, "[Profil commun pour la messagerie instantanée](#) (CPIM)", août 2004. (P.S.)
- [29] Rescorla, E., "A Survey of Authentication Mechanisms", Non publiée, mars 2004.
- [UCS] International Organization for Standardization, "Information Technology - Universal Multiple-octet coded Character Set (UCS) - Part 1: Architecture and Basic Multilingual Plane", norme ISO 10646-1, mai 1993.
- [UNICODE] The Unicode Consortium, "The Unicode Standard, Version 4.0", Addison-Wesley, Boston, MA. ISBN 0-321-18578-1, avril 2003,  
<[http://www.unicode.org/unicode/standard/versions/enumeratedversions.html#Unicode\\_4\\_0\\_0](http://www.unicode.org/unicode/standard/versions/enumeratedversions.html#Unicode_4_0_0)>.
- [XML1.0] Bray, T., Paoli, J., Sperberg-McQueen, C., and E. Maler, "Extensible Markup Language (XML) 1.0 (2nd ed)", W3C Recommendation xml, octobre 2000, <<http://www.w3.org/TR/2000/REC-xml-20001006>>.
- [XMLNAMES] Bray, T., Hollander, D., and A. Layman, "Namespaces in XML", W3C Recommendation xml-names, janvier 1999, <<http://www.w3.org/TR/REC-xml-names>>.

## 12. Adresse des auteurs

Derek Atkins  
IHTFP Consulting  
6 Farragut Ave  
Somerville, MA 02144  
USA

téléphone : +1 617 623 3745  
mél : [derek@ihtfp.com](mailto:derek@ihtfp.com), [warlord@alum.mit.edu](mailto:warlord@alum.mit.edu)

Graham Klyne  
Nine by Nine  
mél : [GK-IETF@ninebynine.org](mailto:GK-IETF@ninebynine.org)  
URI : <http://www.ninebynine.net/>

## 13. Déclaration complète de droits de reproduction

Copyright (c) The Internet Society (2001). Tous droits réservés.

Le présent document et ses traductions peuvent être copiés et fournis aux tiers, et les travaux dérivés qui les commentent ou les expliquent ou aident à leur mise en œuvre peuvent être préparés, copiés, publiés et distribués, en tout ou partie, sans restriction d'aucune sorte, pourvu que la déclaration de droits de reproduction ci-dessus et le présent paragraphe soient inclus dans toutes ces copies et travaux dérivés. Cependant, le présent document lui-même ne peut être modifié d'aucune façon, en particulier en retirant la notice de droits de reproduction ou les références à la Internet Society ou aux autres organisations Internet, excepté autant qu'il est nécessaire pour le besoin du développement des normes Internet, auquel cas les procédures de droits de reproduction définies dans les procédures des normes Internet doivent être suivies, ou pour les besoins de la traduction dans d'autres langues que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Internet Society, ses successeurs ou ayant droits.

Le présent document et les informations contenues sont fournis sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations ci encloses ne violent aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

### Remerciement

Le financement de la fonction d'édition des RFC est actuellement fourni par l'Internet Society.