

Groupe de travail Réseau
Request for Comments : 5322
 RFC rendue obsolète : 2822
 RFC mise à jour : 4021
 Catégorie : En cours de normalisation

P. Resnick, éditeur,
 Qualcomm Incorporated
 octobre 2008

Traduction Claude Brière de L'Isle

Format de message Internet

Statut du présent mémoire

Le présent document spécifie un protocole Internet en cours de normalisation pour la communauté de l'Internet. Il appelle à la discussion et à des suggestions pour son amélioration. Prière de se référer à l'édition actuelle des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Le présent document spécifie le format de message Internet (IMF, *Internet Message Format*) une syntaxe pour les messages de texte qui sont envoyés entre les utilisateurs d'ordinateurs, dans le cadre de messages de "courrier électronique". La présente spécification est une révision de la demande de commentaires (RFC, *Request For Comments*) 2822, qui remplace elle-même la RFC 822, "Norme pour le format des messages de texte de l'ARPA Internet", et la met à jour pour refléter les pratiques actuelles et incorporer les changements qui ont été spécifiés dans d'autres RFC.

Table des Matières

1.	Introduction.....
1.1	Domaine d'application.....
1.2	Conventions rédactionnelles.....
2.	Analyse lexicale des messages.....
2.1	Description générale.....
2.2	Champs d'en-tête.....
2.3	Corps.....
3.	Syntaxe.....
3.1	Introduction.....
3.2	Jetons lexicaux.....
3.3	Spécification de la date et de l'heure.....
3.4	Spécification d'adresse.....
3.5	Syntaxe globale de message.....
3.6	Définitions des champs.....
4.	Syntaxe obsolète.....
4.1	Divers jetons obsolètes.....
4.2	Espace de pliage obsolète.....
4.3	Date et heure obsolète.....
4.4	Adressage obsolète.....
4.5	Champs d'en-tête obsolètes.....
5.	Considérations pour la sécurité.....
6.	Considérations relatives à l'IANA.....
Appendice A.	Exemples de messages.....
A.1	Exemples d'adressage.....
A.2	Messages de réponse.....
A.3	Messages de renvoi.....
A.4	Messages avec champs de traçage.....
A.5	Espaces blanches, commentaires, et autres particularités.....
Appendice B	Différences avec les spécifications antérieures.....
Appendice C.	Remerciements.....
7.	Références.....
7.1	Références normatives.....
7.2	Références pour information.....
	Adresse de l'auteur.....
	Déclaration complète de droits de reproduction.....

1. Introduction

1.1 Domaine d'application

Le présent document spécifie le format de message de l'Internet (IMF, *Internet Message Format*), une syntaxe pour les messages de texte qui sont envoyés entre utilisateurs d'ordinateurs, dans le cadre de messages de "courrier électronique". La présente spécification est une mise à jour de la [RFC2822], qui elle-même remplaçait la [RFC0822], la mettant à jour pour refléter les pratiques actuelles et incorporant les changements spécifiés dans d'autres RFC telles que la [RFC1123].

Le présent document spécifie une syntaxe pour les seuls messages de texte. En particulier, elle ne prend aucune disposition pour la transmission des images, de l'audio, ou autres sortes de données structurées dans les messages de courrier électronique. Plusieurs extensions ont été publiées, comme la série des documents MIME ([RFC2045], [RFC2046], [RFC2049]), qui décrivent les mécanismes pour la transmission de telles données par la messagerie électronique, soit en étendant la syntaxe fournie ici, soit en structurant de tels messages pour se conformer à la présente syntaxe. Ces mécanismes sortent du domaine d'application de la présente spécification.

Dans le contexte de la messagerie électronique, les messages sont vus comme ayant une enveloppe et un contenu. L'enveloppe contient les informations qui sont nécessaires pour accomplir la transmission et la livraison. (Voir dans la [RFC5321] un exposé sur l'enveloppe.) Le contenu comporte l'objet à livrer au receveur. La présente spécification ne s'applique qu'au format et à une partie de la sémantique du contenu du message. Elle ne contient aucune spécification des informations de l'enveloppe.

Cependant, certains systèmes de message peuvent utiliser des informations provenant du contenu pour créer l'enveloppe. La présente spécification est destinée à faciliter l'acquisition de telles informations par les programmes.

Cette spécification est destinée à définir le format du contenu de message qui va passer entre les systèmes. Bien que certains systèmes de messagerie mémorisent localement les messages dans ce format (ce qui élimine le besoin de traduction entre formats) et que d'autres utilisent des formats qui diffèrent de celui spécifié ici, la mémorisation locale sort du domaine d'application de la présente spécification.

Note : La présente spécification n'est pas destinée à dicter les formats internes utilisés par les sites, les caractéristiques de système spécifiques du message qu'on s'attend qu'ils prennent en charge, ou aucune des caractéristiques des programmes d'interface d'utilisateur qui créent ou lisent les messages. De plus, le présent document ne spécifie pas de codage des caractères pour le transport ou la mémorisation, c'est-à-dire qu'il ne spécifie pas le nombre de bits utilisés ni comment ces bits sont spécifiquement transférés sur le réseau ou mémorisés sur les disques.

1.2 Conventions rédactionnelles

1.2.1 Exigences pour la notation

Le présent document utilise occasionnellement des termes qui apparaissent en lettres majuscules. Lorsque les termes "DOIT", "DEVRAIT", "RECOMMANDE", "NE DOIT PAS", "NE DEVRAIT PAS" et "PEUT" apparaissent en majuscules, ils sont utilisés pour indiquer des exigences particulières de la présente spécification. Un exposé de la signification de ces termes figure dans la [RFC2119].

1.2.2 Notation syntaxique

La présente spécification utilise la notation en forme Backus-Naur augmentée (ABNF, *Augmented Backus-Naur Form*) [RFC5234] pour la définition formelle de la syntaxe des messages. Les caractères seront spécifiés par une valeur décimale (par exemple, la valeur %d65 pour le A majuscule et %d97 pour le a minuscule) ou par une valeur littérale insensible à la casse incluse entre des guillemets (par exemple, "A" pour le A aussi bien majuscule que minuscule).

1.2.3 Structure du document

Le présent document est divisé en plusieurs sections.

Cette section, la section 1, est une brève introduction au document.

La section 2 pose la description générale d'un message et de ses parties constitutives. C'est un survol du problème pour aider le lecteur à comprendre les principes généraux utilisés dans les portions ultérieures du document. Aucun exemple de

cette section NE DOIT être considéré comme spécification de la syntaxe formelle d'une partie quelconque d'un message.

La section 3 spécifie les règles formelles de l'ABNF pour la structure de chaque partie d'un message (la syntaxe) et décrit les relations entre ces parties et leur signification dans le contexte d'un message (la sémantique). C'est-à-dire qu'elle pose les règles réelles de la structure de chaque partie d'un message (la syntaxe) ainsi qu'une description des parties et des instructions pour leur interprétation (la sémantique). Cela inclut l'analyse de la syntaxe et de la sémantique des sous parties des messages qui ont une structure spécifique. La syntaxe incluse dans la section 3 représente les messages comme ils DOIVENT être créés. Il y a aussi des notes dans la section 3 pour indiquer si une des options spécifiées dans la syntaxe DEVRAIT être utilisée plutôt qu'une des autres.

Les sections 2 et 3 décrivent toutes deux les messages qu'il est légal de générer pour les besoins de cette spécification.

La section 4 de ce document spécifie une syntaxe "obsolète". Il y a dans la section 3 des références à ces éléments syntaxiques obsolètes. Les règles de la syntaxe obsolète sont des éléments qui sont apparus dans des versions antérieures de la présente spécification ou ont été précédemment largement utilisés dans les messages Internet. Comme tels, ces éléments DOIVENT être interprétés par les analyseurs de messages afin d'être conformes à la présente spécification. Cependant, comme des éléments de cette syntaxe ont été déterminés non interopérables ou causer des problèmes significatifs aux receveurs des messages, ils NE DOIVENT PAS être générés par les créateurs de messages conformes.

La section 5 précise les considérations pour la sécurité à prendre en compte lors de la mise en œuvre de la présente spécification.

L'Appendice A donne des exemples de différentes sortes de messages. Ces exemples ne donnent pas une description exhaustive des types de messages qui apparaissent sur l'Internet, mais donnent une large vue d'ensemble de certaines formes syntaxiques.

L'Appendice B donne la liste des différences entre cette spécification et les spécifications antérieures des messages Internet.

L'Appendice C contient les remerciements.

2. Analyse lexicale des messages

2.1 Description générale

Au niveau le plus basique, un message est une série de caractères. Un message conforme à la présente spécification est composé de caractères qui ont des valeurs dans la gamme de 1 à 127 et sont interprétés comme des caractères US-ASCII [ANSI.X3-4.1986]. Pour faire court, le présent document se réfère parfois à cette gamme de caractères simplement comme "caractères US-ASCII".

Note : Le présent document spécifie que les messages sont constitués de caractères dans la gamme US-ASCII de 1 à 127. Il y a d'autres documents, en particulier les documents de la série MIME ([RFC2045], [RFC2046], [RFC2047], [RFC2049], [RFC4288], [RFC4289]) qui étendent cette spécification pour permettre des valeurs en dehors de cette gamme. L'exposé de ces mécanismes sort du domaine d'application de la présente spécification.

Les messages sont divisés en lignes de caractères. Une ligne est une série de caractères qui est délimitée par les deux caractères retour-chariot et saut-à-la-ligne ; c'est-à-dire le caractère retour-chariot (CR) (valeur ASCII 13) suivie immédiatement par le caractère saut-à-la-ligne (LF) (valeur ASCII 10). (La paire retour-chariot/saut-à-la-ligne est habituellement présentée comme "CRLF" dans le présent document.)

Un message consiste en champs d'en-tête (appelés collectivement "la section en-tête du message") suivis facultativement par un corps. La section d'en-tête est une séquence de lignes de caractères avec une syntaxe spéciale définie dans la présente spécification. Le corps est simplement une séquence de caractères qui suivent la section d'en-tête et en est séparée par une ligne vide (c'est-à-dire, une ligne sans rien avant le CRLF).

Note : Le langage courant et les versions antérieures de la présente spécification utilisent le terme "en-tête" soit pour se référer à la section d'en-tête entière, soit pour se référer à un champ d'en-tête individuel. Pour éviter l'ambiguïté, le présent document n'utilise pas les termes "en-tête" ou "en-têtes" isolément, mais utilise plutôt toujours "champ d'en-tête" pour se référer au champ individuel et "section d'en-tête" pour se référer à leur collection entière.

2.1.1 Limites de longueur de ligne

La présente spécification établit deux limites au nombre de caractères d'une ligne. Chaque ligne de caractères DOIT être au plus de 998 caractères, et DEVRAIT être au plus de 78 caractères, à l'exclusion du CRLF.

La limite des 998 caractères est due aux limitations de nombreuses mises en œuvre qui envoient, reçoivent ou mémorisent des messages IMF qui tout simplement ne peuvent pas traiter plus de 998 caractères sur une ligne. Les mises en œuvre receveuses feraient bien de traiter un nombre arbitraire de caractères sur une ligne pour les besoins de la robustesse. Cependant, il y a de si nombreuses mises en œuvre qui (conformément aux exigences de transport de la [RFC5321]) n'acceptent pas de messages contenant plus de 1000 caractères par ligne, y compris le CR et le LF, qu'il est important que les mises en œuvre ne créent pas de tels messages.

La recommandation plus conservatrice de 78 caractère est destinée à s'accommoder des nombreuses mises en œuvre d'interfaces d'utilisateur qui affichent ces messages en les tronquant, ou (désastre) réécrivent sur le début de ligne, lorsque l'affichage donnerait plus de 78 caractères par ligne, en dépit du fait que de telles mises en œuvre ne sont pas conformes aux intentions de la présente spécification (et de celles de la [RFC5321] si elles causent en fait la perte d'informations). Même si cette limitation est mise sur les messages, c'est aux mises en œuvre qu'il revient d'afficher les messages qui contiennent un nombre arbitrairement grand de caractères sur une ligne (certainement au moins jusqu'à la limite de 998 caractères) au nom de la robustesse.

2.2 Champs d'en-tête

Les champs d'en-tête sont des lignes qui commencent par un nom de champ, suivi par deux-points (":"), suivi par un corps de champ, et terminées par CRLF. Un nom de champ DOIT être composé de caractères US-ASCII imprimables (c'est-à-dire, de caractères qui ont des valeurs entre 33 et 126, inclus), sauf les deux points. Un corps de champ peut être composé de caractères US-ASCII imprimables aussi bien que de caractères espace (SP, valeur ASCII 32) et tabulation horizontale (HTAB, valeur ASCII 9) (appelés collectivement caractères d'espace blanche, WSP). Un corps de champ NE DOIT PAS inclure de CR et de LF excepté quand utilisés dans le "pliage" (*folding*) et le "dépliage" (*unfolding*), qui sont décrits au paragraphe 2.2.3. Tous les corps de champ DOIVENT se conformer à la syntaxe décrite aux sections 3 et 4 de la présente spécification.

2.2.1 Corps de champ d'en-tête non structurés

Certains corps de champ de la présente spécification sont définis simplement comme "non structurés" (ce qui est spécifié au paragraphe 3.2.5 comme tout caractère US-ASCII imprimable plus les caractères d'espace) sans autre restriction. On les appelle les corps de champ non structurés. Sémantiquement, les corps de champ non structurés sont simplement à traiter comme une seule ligne de caractères sans autre traitement (sauf pour ce qui concerne le "pliage" et le "dépliage" comme expliqué au paragraphe 2.2.3).

2.2.2 Corps de champ d'en-tête structurés

Certains corps de champ de la présente spécification ont une syntaxe qui est plus restrictive que celle des corps de champ non structurés décrits ci-dessus. On les appelle des corps de champ "structurés". Les corps de champs structurés sont des séquences de jetons lexicaux spécifiques qui sont décrits aux sections 3 et 4 de la présente spécification. Beaucoup de ces jetons peuvent (selon leur syntaxe) être introduits ou se terminer par des commentaires (comme décrit au paragraphe 3.2.2) ainsi que par des caractères espace, et ces caractères espace sont sujets au "pliage" et "dépliage" décrit au paragraphe 2.2.3. L'analyse sémantique des corps de champ structurés est donnée avec leur syntaxe.

2.2.3 Champs d'en-tête longs

Chaque champ d'en-tête est logiquement une seule ligne de caractères comportant le nom du champ, les deux points, et le corps du champ. Cependant, par commodité, et pour s'accommoder des limitations à 998/78 caractères par ligne, la portion corps de champ d'un champ d'en-tête peut être partagée sur une représentation multi lignes ; ce qu'on appelle "pliage". La règle générale est que partout où la présente spécification permet l'espace blanche de pliage (pas simplement les caractères WSP) un CRLF peut être inséré avant toute WSP.

Par exemple, le champ d'en-tête :

Subject: C'est un essai

peut être représenté par :

Subject: C'est
un essai

Note : Bien que les corps de champ structurés soient définis de telle façon que le pliage puisse avoir lieu entre beaucoup des jetons lexicaux (et même au sein de certains des jetons lexicaux), le pliage DEVRAIT être limité à placer le CRLF à des coupures syntaxiques de niveau supérieur. Par exemple, si un corps de champ est défini comme des valeurs séparées par des points, il est recommandé que le pliage survienne après le point qui sépare les éléments structurés plutôt qu'à d'autres endroits où le champ pourrait être plié, même si c'est permis ailleurs.

Le processus de passer de cette représentation de pliage sur plusieurs lignes d'un champ d'en-tête à sa représentation sur une seule ligne est appelé "dépliage". Le dépliage est réalisé par la simple suppression de tout CRLF qui est immédiatement suivi d'une WSP. Chaque champ d'en-tête devrait être traité dans sa forme dépliée pour la suite de l'évaluation syntaxique et sémantique. Un champ d'en-tête déplié n'a pas de restriction de longueur et peut donc être d'une longueur indéterminée.

2.3 Corps

Le corps d'un message est simplement constitué de lignes de caractères US-ASCII. Les deux seules limitations portant sur le corps sont les suivantes :

- o CR et LF DOIVENT survenir uniquement ensemble comme CRLF ; ils NE DOIVENT PAS apparaître séparément dans le corps.
- o Les lignes de caractères dans le corps DOIVENT être limitées à 998 caractères, et DEVRAIENT être limitées à 78 caractères, CRLF exclu.

Note : Comme mentionné plus haut, d'autres documents, en particulier MIME ([RFC2045], [RFC2046], [RFC2049], [RFC4288], [RFC4289]) étendent (et limitent) la présente spécification en permettant différentes sortes de corps de message. Ces mécanismes sortent du domaine d'application du présent document.

3. Syntaxe

3.1 Introduction

Telle qu'elle est donnée dans la présente section, la syntaxe définit la syntaxe légale pour les messages sur l'Internet. Les messages qui se conforment à la présente spécification DOIVENT se conformer à la syntaxe donnée dans cette section. Lorsque il y a des options et qu'une option DEVRAIT être générée, c'est indiqué soit dans le texte, soit dans un commentaire à côté de la syntaxe.

Pour les expressions définies, on donne une brève description de la syntaxe et de l'utilisation, suivie par la syntaxe en ABNF, suivie par une analyse sémantique. Les jetons de primitives suivants qui sont utilisés mais non spécifiés par ailleurs sont tirés du "Cœur des règles" de la [RFC5234], Appendice B.1 : CR, LF, CRLF, HTAB, SP, WSP, DQUOTE, DIGIT, ALPHA et VCHAR.

Dans certaines des définitions, il y aura des non-terminaux dont le nom commence par "obs-". Ces éléments "obs-" se réfèrent aux jetons définis dans la syntaxe obsolète de la section 4. Dans tous les cas, ces produits sont à ignorer lorsque on veut générer des messages Internet légaux et ils NE DOIVENT PAS être utilisés au titre d'un tel message. Cependant, pour interpréter les messages, ces jetons DOIVENT être honorés au titre de la syntaxe légale. En ce sens, la section 3 définit une grammaire pour la génération des messages, avec des éléments "obs-" qui sont à ignorer, alors que la section 4 ajoute de la grammaire pour l'interprétation des messages.

3.2. Jetons lexicaux

Les règles suivantes sont utilisées pour définir un analyseur lexical sous-jacent, qui présente les jetons aux analyseurs de niveau supérieur. Ce paragraphe définit les jetons utilisés dans les corps de champ d'en-tête structurés.

Note : Les lecteurs de la présente spécification devraient porter une attention particulière à la façon dont ces jetons lexicaux sont utilisés plus loin dans ce document à la fois dans la syntaxe de niveau inférieur et de niveau supérieur. En particulier, les jetons d'espace blanche et de commentaire définis au paragraphe 3.2.2 sont utilisés dans les jetons de niveau inférieur définis ici, et ces jetons de niveau inférieur sont à leur tour utilisés au titre des jetons du niveau supérieur définis plus loin.

Donc, les espaces blanches et les commentaires peuvent être admis dans les jetons de niveau supérieur même si ils n'apparaissent pas explicitement dans une définition particulière.

3.2.1 Caractères entre guillemets

Certains caractères sont réservés pour une interprétation particulière, comme les jetons de délimitation lexicale. Pour permettre l'utilisation de ces caractères comme données non interprétées, un mécanisme de guillemets est fourni.

paire-entre-guillemets = ("\" (VCHAR / WSP)) / obs-qp

Partout où apparaît une paire entre guillemets, elle doit être interprétée comme le caractère seul. C'est à dire que le caractère "\" qui apparaît au titre d'une paire entre guillemets est sémantiquement "invisible".

Note : Le caractère "\" peut apparaître dans un message où il ne fait pas partie d'une paire entre guillemets. Un caractère "\" qui n'apparaît pas dans une paire entre guillemets n'est pas sémantiquement invisible. Les seuls endroits de la présente spécification où apparaissent actuellement des paires entre guillemets sont ccontent, qcontent, et dans obs-dtext à la section 4.

3.2.2 Espace de pliage et commentaires

Les caractères d'espace, y compris les espaces blanches utilisées dans le pliage (décrit au paragraphe 2.2.3) peuvent apparaître entre de nombreux éléments dans les corps de champ d'en-tête. Et aussi, des chaînes de caractères qui sont traités comme des commentaires peuvent être incluses dans des corps de champ structurés comme des caractères inclus entre parenthèses. Ce qui suit définit l'espace blanche de pliage (FWS, *folding white space*) et les constructions de commentaires.

Les chaînes de caractères incluses entre parenthèses sont considérées comme commentaires pour autant qu'elles n'apparaissent pas au sein d'une "chaîne entre guillemets", comme défini au paragraphe 3.2.4. Les commentaires peuvent être incorporés.

Il y a plusieurs endroits de cette spécification où commentaires et FWS peuvent être insérés librement. Pour s'accommoder de cette syntaxe, un jeton supplémentaire pour "CFWS" est défini pour les endroits où commentaires et/ou FWS peuvent survenir. Cependant, lorsque un CFWS survient dans la présente spécification, il NE DOIT PAS être inséré d'une façon telle qu'une ligne d'un champ d'en-tête replié soit entièrement constituée de caractères WSP et rien d'autre.

FWS = ([*WSP CRLF] 1*WSP) / obs-FWS ; Espace blanche de pliage

ctext = %d33-39 / ; Caractères US-ASCII non imprimables n'incluant pas "(", ")", ou "\"
%d42-91 /
%d93-126 /
obs-ctext

ccontent = ctext / paire-entre-guillemets / commentaire

commentaire = "(" *([FWS] ccontent) [FWS] ")"

CFWS = (1*([FWS] commentaire) [FWS]) / FWS

Tout au long de la présente spécification, lorsque FWS (le jeton d'espace blanche de pliage) apparaît, il indique un endroit où le pliage, tel qu'exposé au paragraphe 2.2.3, peut avoir lieu. Chaque fois que le pliage apparaît dans un message (c'est-à-dire, un corps de champ d'en-tête contenant un CRLF suivi par une ou des WSP), le dépliage (suppression du CRLF) est effectué avant toute analyse sémantique sur ce champ d'en-tête selon la présente spécification. Cela revient à dire que tout CRLF qui apparaît dans les FWS est sémantiquement "invisible".

Un commentaire est normalement utilisé dans un corps de champ structuré pour apporter du texte d'information lisible par l'homme. Comme il est permis à un commentaire de contenir des FWS, le pliage est permis au sein du commentaire. Noter aussi que comme une paire entre guillemets est permise dans un commentaire, les parenthèses et barres obliques inverses peuvent apparaître dans un commentaire, pour autant qu'elles apparaissent comme paire entre guillemets. Sémantiquement, les parenthèses englobantes ne font pas partie du commentaire ; le commentaire est ce qui est contenu entre les deux parenthèses. Comme il est dit plus haut, le caractère "\" dans toute paire entre guillemets et le CRLF dans toute FWS apparaissant au sein du commentaire sont sémantiquement "invisibles" et ne font donc pas non plus partie du commentaire.

Les séries de FWS, commentaire, ou CFWS qui surviennent entre des jetons lexicaux dans un champ d'en-tête structuré

sont sémantiquement interprétées comme un seul caractère d'espace.

3.2.3 Atome

Dans les corps de champ d'en-tête structurés, plusieurs produits sont simplement des chaînes de certains caractères de base. On appelle ces produits des atomes.

Certains corps de champ d'en-tête structuré permettent aussi le caractère point (".", valeur ASCII 46) au sein de séries de atext. Un jeton "point-atome" supplémentaire est défini à cette fin.

Note : Le jeton "specials" n'apparaît nulle part ailleurs dans la présente spécification. C'est simplement le caractère visible (c'est-à-dire, ni caractère de contrôle, ni d'espace) qui n'apparaît pas en atext. Il n'est fourni que parce que il est utile pour les mises en œuvre qui utilisent des outils qui analysent lexicalement les messages. Chacun des caractères en specials peut être utilisé pour indiquer un jeton dans l'analyse lexicale.

```
atext      = ALPHA / DIGIT /
           "!" / "#" /
           "$" / "%" /
           "&" / "'" /
           "*" / "+" /
           "-" / "/" /
           "=" / "?" /
           "^" / "_" /
           "`" / "{" /
           "|" / "}" /
           "~"
           ; Caractères US-ASCII imprimables, non inclus les spécials.
           ; Utilisé pour les atomes.
```

```
atome      = [CFWS] 1*atext [CFWS]
```

```
point-atome-texte = 1*atext *("." 1*atext)
```

```
point-atome      = [CFWS] point-atome-texte [CFWS]
```

```
specials    = "(" / ")" /
           "<" / ">" /
           "[" / "]" /
           ":" / ";" /
           "@" / "\" /
           "," / "." /
           DQUOTE
           ; Caractères spéciaux qui n'apparaissent pas en atext
```

atom et dot-atom sont tous deux interprétés comme une seule unité, comprenant la chaîne de caractères qui la constitue. Sémantiquement, les commentaires et FWS facultatifs entourant le reste des caractères ne font pas partie de l'atome ; l'atome est seulement la série de caractères atext dans un atome, ou les caractères atext et "." dans un point-atome.

3.2.4 Chaînes entre guillemets

Les chaînes de caractère qui comportent des caractères autres que ceux permis dans les atomes peuvent être représentées dans un format de chaîne entre guillemets, où les caractères sont entourés de guillemets (DQUOTE, valeur ASCII 34).

```
qtext      = %d33 /
           %d35-91 /
           %d93-126 /
           obs-qtext
           ; Caractères US-ASCII imprimables,
           ; non inclus "\" ou le caractère guillemet.
```

```
qcontent   = qtext / quoted-pair
```

```
chaîne-entre-guillemets = [CFWS] DQUOTE *([FWS] qcontent) [FWS] DQUOTE [CFWS]
```

Une chaîne entre guillemets est traitée comme une unité. C'est-à-dire que sémantiquement, chaîne-entre-guillemets est identique à atome. Comme il est permis à une chaîne entre guillemets de contenir des FWS, le pliage est permis. Noter aussi que comme la paire entre guillemets est permise dans une chaîne entre guillemets, les caractères guillemet et barre

oblique inverse peuvent apparaître dans une chaîne entre guillemets pour autant qu'ils apparaissent comme paire entre guillemets.

Sémantiquement, ni le CFWS facultatif en dehors des caractères guillemets ni les caractères guillemets eux-mêmes ne font partie de la chaîne entre guillemets ; la chaîne entre guillemets est ce qui est contenu entre les deux caractères de guillemet. Comme on l'a dit plus haut, le caractère "\" dans toute paire entre guillemets et le CRLF dans tout FWS/CFWS qui apparaît au sein de la paire entre guillemets sont sémantiquement "invisibles" et ne font donc pas partie non plus de la chaîne entre guillemets.

3.2.5 Jetons divers

Trois jetons supplémentaires sont définis : mot et phrase pour des combinaisons d'atomes et/ou de chaînes entre guillemets et non-structuré à utiliser dans les champs d'en-tête non structurés et dans certains endroits au sein de champs d'en-tête structurés.

mot = atome / chaîne-entre-guillemets
 phrase = 1*mot / obs-phrase
 non-structuré = (*([FWS] VCHAR) *WSP) / obs-unstruct

3.3 Spécification de la date et de l'heure

Des valeurs de date et d'heure surviennent dans plusieurs champs d'en-tête. Ce paragraphe spécifie la syntaxe pour une spécification complète de la date et de l'heure. Bien que l'espace blanche de pliage soit permise au sein de la spécification de date et d'heure, il est RECOMMANDÉ qu'une seule espace soit utilisée à chaque endroit où FWS apparaît (qu'il soit obligé ou facultatif) ; certaines mises en œuvre anciennes ne vont pas interpréter correctement de plus longues séquences d'espaces blanches de pliage.

date-heure = [jour-de-la semaine ", "] date heure [CFWS]
 jour-de-la semaine = ([FWS] nom-du-jour) / obs-day-of-week
 nom-du-jour = "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun"
 date = jour mois année
 jour = ([FWS] 1*2DIGIT FWS) / obs-day
 mois = "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun" / "Jul" / "Aug" / "Sep" / "Oct" / "Nov" / "Dec"
 année = (FWS 4*DIGIT FWS) / obs-year
 heure = heure-du-jour zone
 heure-du-jour = heure ":" minute [":" seconde]
 heure = 2DIGIT / obs-hour
 minute = 2DIGIT / obs-minute
 seconde = 2DIGIT / obs-second
 zone = (FWS ("+" / "-") 4DIGIT) / obs-zone

Le jour est le numéro du jour du mois. L'année est un numéro d'année postérieur à 1900.

L'heure-du-jour spécifie le nombre d'heures, minutes, et facultativement de secondes depuis minuit de la date indiquée.

La date et l'heure-du-jour DEVRAIENT exprimer l'heure locale.

La zone spécifie le décalage par rapport au Temps Universel Coordonné (UTC, anciennement appelé "Heure Moyenne de Greenwich") que représentent la date et l'heure-du-jour. Le "+" ou "-" indiquent si l'heure-du-jour est avant (c'est-à-dire, à l'est) ou derrière (c'est-à-dire, à l'ouest) le temps universel. Les deux premiers chiffres indiquent le nombre d'heures de différence avec le Temps Universel, et les deux derniers chiffres indiquent le nombre de minutes additionnelles de différence avec le Temps Universel. (Et donc, +hhmm signifie +(hh * 60 + mm) minutes, et -hhmm signifie -(hh * 60 + mm) minutes). La forme "+0000" DEVRAIT être utilisée pour indiquer un fuseau horaire du Temps Universel. Bien que "-0000" indique aussi le Temps Universel, c'est utilisé pour indiquer que l'heure a été générée sur un système qui peut être dans un fuseau horaire autre que du Temps Universel et que la date-heure ne contient aucune information sur le fuseau horaire.

Une spécification de date-heure DOIT être sémantiquement valide. C'est à dire que le jour-de-la-semaine (s'il est inclus) DOIT être le jour impliqué par la date, le numéro du jour-de-la-semaine DOIT être entre 1 et le nombre de jours permis pour le mois spécifié (dans l'année spécifiée) l'heure-du-jour DOIT être dans la gamme 00:00:00 à 23:59:60 (le nombre de secondes permettant un saut de seconde voir la [RFC1305]) ; et les deux derniers chiffres de la zone DOIVENT être dans la gamme de 00 à 59.

3.4 Spécification d'adresse

Les adresses surviennent dans plusieurs champs d'en-tête de message pour indiquer les envoyeurs et receveurs des messages. Une adresse peut être une boîte aux lettres individuelle, ou un groupe de boîtes aux lettres.

```

adresse      = boîte-aux-lettres / groupe
boîte-aux-lettres = nom-d'adresse / adresse-spec
nom-d'adresse  = [nom-affiché] angle-addr
angle-addr    = [CFWS] "<" adresse-spec ">" [CFWS] / obs-angle-addr
groupe       = nom-affiché ":" [liste-du-groupe] ";" [CFWS]
nom-affiché  = phrase
liste-de-boîte-aux-lettres = (boîte-aux-lettres *("," boîte-aux-lettres)) / obs-mbox-list
liste-d'adresses = (adresse *("," adresse)) / obs-addr-list
liste-du-groupe = liste-de-boîte-aux-lettres / CFWS / obs-group-list

```

Une boîte à lettre reçoit du courrier. C'est une entité conceptuelle qui ne doit pas nécessairement relever de la mémorisation de fichiers. Par exemple, des sites peuvent choisir d'imprimer leur courrier sur une imprimante et de le livrer au bureau spécifié par l'adresse.

Normalement, une boîte aux lettres se compose de deux parties : (1) un nom d'affichage facultatif qui indique le nom du receveur (qui peut être un système ou une personne) qui peut être affiché à l'utilisateur de l'application de courrier, et (2) une adresse-spec comprise entre des crochets angulaires "<" et ">". Il y a aussi une forme simple de remplacement d'une boîte aux lettres où l'adresse-spec apparaît seule, sans le nom du receveur ou les crochets angulaires. L'adresse-spec de l'Internet est définie au paragraphe 3.4.1.

Note : Certaines mises en œuvre utilisent la forme simple où l'adresse-spec apparaît sans les crochets angulaires, mais inclut le nom du receveur entre parenthèses comme s'il s'agissait d'un commentaire suivant l'adresse-spec. Comme la signification des informations dans un commentaire est non spécifiée, les mises en œuvre DEVRAIENT utiliser la forme nom-d'adresse complète de la boîte aux lettres, au lieu de la forme traditionnelle ci-dessus, pour spécifier le nom d'affichage associé à une boîte aux lettres. Et aussi, comme certaines mises en œuvre traditionnelles interprètent le commentaire, les commentaires NE DEVRAIENT généralement PAS être utilisés dans les champs d'adresse pour éviter de plonger de telles mises en œuvre dans la confusion.

Lorsque il est souhaitable de traiter plusieurs boîtes aux lettres comme une seule unité (c'est-à-dire, dans une liste de distribution) la construction groupe peut être utilisée. La construction groupe permet à l'envoyeur d'indiquer un groupe désigné de receveurs. Cela se fait en donnant un nom d'affichage pour le groupe, suivi par deux points, suivi par une liste séparée par des virgules de tout nombre de boîtes aux lettres (y compris zéro et une) et se terminant par un point virgule.

Parce que la liste des boîtes aux lettres peut être vide, l'utilisation de la construction groupe est aussi une façon simple de communiquer aux receveurs que le message a été envoyé à un ou plusieurs ensembles de receveurs désignés, sans réellement fournir les adresses individuelles de boîte aux lettres d'aucun de ces receveurs.

3.4.1 Spécification de adresse-spec

Un adresse-spec est un identifiant spécifique de l'Internet qui contient une chaîne interprétée localement suivie par le caractère arobase ("@", valeur ASCII 64) suivi par un domaine Internet. La chaîne interprétée localement est une chaîne entre guillemets ou un point-atome. Si la chaîne peut être représentée comme un point-atome (c'est-à-dire, si elle ne contient que des caractères atext ou "." entouré par des caractères atext) la forme point-atome DEVRAIT être utilisée et la forme de chaîne entre guillemets NE DEVRAIT PAS être utilisée. Commentaires et espace de pliage NE DEVRAIENT PAS être utilisés autour du "@" dans une adresse-spec.

Note : On donne ici une syntaxe libérale pour la portion domaine de adresse-spec. Cependant, la portion domaine contient des informations d'adressage spécifiées par, et utilisées par, d'autres protocoles (par exemple, [RFC1034], [RFC1035], [RFC1123], [RFC5321]). Il appartient donc aux mises en œuvre de se conformer à la syntaxe des adresses selon le contexte dans lequel elles sont utilisées.

```

adresse-spec    = partie-locale "@" domaine

partie-locale   = point-atome / chaîne-entre-guillemets / obs-local-part

domaine         = point-atome / domaine-littéral / obs-domain

domaine-littéral = [CFWS] "[" *([FWS] dtext) [FWS] "]" [CFWS]

dtext           = %d33-90 /                ; caractères US-ASCII imprimables,
                  %d94-126 /              ; non inclus "[", "]", ou "\"
                  obs-dtext

```

La portion domaine identifie le point auquel le message est livré. Dans la forme point-atome, c'est interprété comme un nom de domaine Internet (un nom d'hôte ou un nom d'échangeur de messagerie) comme décrit dans la [RFC1034], la [RFC1035] et la [RFC1123]. Sous la forme domaine-littéral, le domaine est interprété comme l'adresse Internet littérale de l'hôte particulier. Dans les deux cas, la façon dont est utilisé l'adressage et dont les messages sont transportés jusqu'à un hôte particulier est traitée dans des documents distincts, tels que la [RFC5321]. Ces mécanismes sortent du domaine d'application du présent document.

La portion partie-locale est une chaîne qui dépend du domaine. Dans les adresses, elle est simplement interprétée chez l'hôte particulier comme un nom d'une boîte aux lettres particulière.

3.5 Syntaxe globale de message

Un message consiste en champs d'en-tête, facultativement suivis par un corps de message. Les lignes d'un message DOIVENT être au maximum de 998 caractères, CRLF exclu, mais il est RECOMMANDÉ que les lignes se limitent à 78 caractères, CRLF exclu. (Voir les explications au paragraphe 2.1.1.) Dans un corps de message, bien que tous les caractères dont la liste figure dans la règle texte PUISSENT être utilisés, l'utilisation des caractères de contrôle US-ASCII (valeurs de 1 à 8, 11, 12, et 14 à 31) est déconseillés car leur interprétation par les receveurs n'est pas garantie pour l'affichage.

```

message         = (champs / obs-fields) [CRLF corps]

corps           =>(*998texte CRLF) *998texte / obs-body

texte           = %d1-9 /                  ; Caractères à l'exclusion de CR et LF
                  %d11 /
                  %d12 /
                  %d14-127

```

Les champs d'en-tête portent la plupart des informations sémantiques et sont définis au paragraphe 3.6. Le corps est simplement une série de lignes de texte qui ne sont pas interprétées pour les besoins de cette spécification.

3.6 Définitions des champs

Les champs d'en tête d'un message sont définis ici. Tous les champs d'en tête ont la même structure syntaxique générale : un champ de nom, suivi par deux points, suivi par le corps de champ. La syntaxe spécifique pour chaque champ d'en tête est définie dans les paragraphes suivants.

Note : Dans la syntaxe ABNF pour chaque champ dans les paragraphes suivants, chaque nom de champ est suivi par deux points obligatoires. Cependant, pour simplifier, il n'est pas toujours fait usage de ces deux points dans la description textuelle de la syntaxe. Ils sont cependant obligatoires.

Il est important de noter qu'il n'est pas garanti que les champs d'en tête soient classés dans un ordre particulier. Ils peuvent apparaître dans n'importe quel ordre, et il est connu qu'ils sont réordonnés occasionnellement lors du transport sur l'Internet. Cependant, pour les besoins de la présente norme, les champs d'en tête NE DEVRAIENT PAS être réordonnés quand un message est transporté ou transformé. Le plus important, les champs d'en tête "trace" et "resent" NE DOIVENT PAS être réordonnés et DEVRAIENT être gardés en blocs ajoutés au message. Voir les paragraphes 3.6.6 et 3.6.7 pour plus d'informations.

Les seuls champs obligatoires sont le champ de date de création et le ou les champs d'adresse d'origine. Tous les autres champs sont facultatifs du point de vue de la syntaxe. Des informations complémentaires sont fournies dans le tableau qui suit cette définition.

```
champs      = *(trace
               *optional-field /
               *(resent-date /
                 resent-from /
                 resent-sender /
                 resent-to /
                 resent-cc /
                 resent-bcc /
                 resent-msg-id))
               *(orig-date /
                 from /
                 sender /
                 reply-to /
                 to /
                 cc /
                 bcc /
                 message-id /
                 in-reply-to /
                 references /
                 subject /
                 comments /
                 keywords /
                 optional-field)
```

Le tableau suivant indique les limites sur le nombre de fois qu'un champ peut intervenir dans la section d'en-tête ainsi que toute limitation spécifique sur l'utilisation de ces champs. Un astérisque ("*") près d'une valeur dans la colonne minimum ou maximum indique qu'une restriction particulière apparaît dans la colonne Notes.

Champ	Nombre mini.	Nombre maximum	Notes
trace	0	illimité	Bloc ajouté – voir au paragraphe 3.6.7
resent-date	0*	illimité	Un par bloc, obligatoire si d'autres champs resent sont présents – voir en 3.6.6
resent-from	0	illimité	Un par bloc – voir au paragraphe 3.6.6
resent-sender	0*	illimité	Un par bloc, DOIT survenir avec multi-address resent-from - voir en 3.6.6
resent-to	0	illimité	Un par bloc – voir au paragraphe 3.6.6
resent-cc	0	illimité	Un par bloc – voir au paragraphe 3.6.6
resent-bcc	0	illimité	Un par bloc – voir au paragraphe 3.6.6
resent-msg-id	0	illimité	Un par bloc – voir au paragraphe 3.6.6
orig-date	1	1	
from	1	1	Voir à sender et au paragraphe 3.6.2
sender	0*	1	DOIT survenir avec multi-address from – voir en 3.6.2

reply-to	0	1	
to	0	1	
cc	0	1	
bcc	0	1	
message-id	0*	1	DEVRAIT être présent – voir en 3.6.4
in-reply-to	0*	1	DEVRAIT survenir dans certaines réponses – voir en 3.6.4
references	0*	1	DEVRAIT survenir dans certaines réponses – voir en 3.6.4
subject	0	1	
comments	0	illimité	
keywords	0	illimité	
optional-field	0	illimité	

L'interprétation exacte de chaque champ est décrite dans les paragraphes suivants.

3.6.1 Champ Date de création (*Origination Date*)

Le champ Date de création comporte le nom du champ "Date" suivi par une spécification de date et heure.

orig-date = "Date:" date-heure CRLF

La date de création spécifie la date et l'heure à laquelle le créateur du message indique que le message a été terminé et est prêt à passer dans le système de livraison des messages. Par exemple, cela peut être l'heure à laquelle l'utilisateur a appuyé sur le bouton "envoi" ou "soumettre" dans un programme d'application. Dans tous les cas, il n'est précisément pas envisagé de lui faire porter l'heure à laquelle le message a réellement été transporté, mais plutôt l'heure à laquelle l'homme ou tout autre créateur du message l'a mis en forme finale, prêt pour le transport. (Par exemple, un utilisateur d'ordinateur portable non connecté à un réseau peut mettre le message en file d'attente en vue de sa livraison. La date de création est destinée à contenir la date et l'heure à laquelle le message a été mis en file d'attente, pas l'heure à laquelle l'utilisateur s'est connecté au réseau pour l'envoi du message.)

3.6.2 Champs d'origine

Les champs d'origine d'un message comportent les champs from (*provenance*), sender (*envoyeur*) (lorsque applicable), et facultativement le champ reply-to (*en réponse à*). Le champ from comporte le nom de champ "From: " et une liste séparée par des virgules d'une ou plusieurs spécifications de boîtes aux lettres. Si le champ from spécifie plusieurs spécifications de boîtes aux lettres dans la liste des boîtes aux lettres, le champ d'envoyeur, qui contient le nom de champ "Sender" et une seule spécification de boîte aux lettres) DOIT alors apparaître dans le message. Dans tous les cas, un champ facultatif reply-to PEUT aussi être inclus, contenant le nom de champ "Reply-To" suivi d'une liste de une ou plusieurs adresses séparées par des virgules.

from = "From:" liste-des-boîtes-aux-lettres CRLF

sender = "Sender:" boîte-aux-lettres CRLF

reply-to = "Reply-To:" liste-d'adresses CRLF

Les champs d'origine indiquent la ou les boîtes aux lettres de la source du message. Le champ "From:" spécifie le ou les auteurs du message, c'est-à-dire la ou les boîtes aux lettres de la ou des personnes ou systèmes responsables de la rédaction du message. Le champ "Sender:" spécifie la boîte aux lettres de l'agent responsable de la transmission effective du message. Par exemple si une secrétaire doit envoyer un message pour une autre personne, la boîte à lettre de la secrétaire apparaîtra dans le "Sender:" et le nom de l'auteur réel apparaîtra dans le champ "From:". Si le créateur du message peut être indiqué par une seule boîte aux lettres et que l'auteur et l'émetteur sont identiques, le champ "Sender:" NE DEVRAIT PAS être utilisé. Sinon, les deux champs DEVRAIENT apparaître.

Note : Les informations sur l'émetteur sont toujours présentes. L'absence du champ "Sender:" est parfois comprise à tort comme signifiant que l'agent responsable de l'émission du message n'a pas été spécifié. Cette absence signifie simplement que l'émetteur est identique à l'auteur et il n'y a donc pas redondance dans le champ "Sender:".

Les champs d'origine fournissent aussi les informations nécessaires pour répondre à un message. Quand le champ "Reply-To:" est présent, il indique la ou les adresses auxquelles l'auteur du message suggère d'envoyer les réponses. En l'absence du champ "Reply-To:", les réponses DEVRAIENT par défaut être envoyées à la ou aux boîtes aux lettres spécifiées dans le champ "From:" sauf spécification contraire par la personne qui compose la réponse.

Dans tout les cas, le champ "From:" NE DEVRAIT PAS contenir de boîte aux lettres n'appartenant pas à l'auteur du message. Voir aussi au paragraphe 3.6.3 plus d'informations sur la formation des adresses de destination pour une réponse.

3.6.3 Champs d'adresse de destination

Les champs de destination d'un message consistent en trois champs possibles, chacun de la même forme : le nom du champ, qui est "To", "Cc", ou "Bcc" suivi d'une liste d'une ou plusieurs adresses (de syntaxe boîte aux lettres ou groupe) séparées par des virgules.

```
to      = "To:" liste-d'adresses CRLF
cc      = "Cc:" liste-d'adresses CRLF
bcc     = "Bcc:" [liste-d'adresses / CFWS] CRLF
```

Les champs de destination spécifient les receveurs du message. Chaque champ de destination peut avoir une ou plusieurs adresses, et chacune des adresses indique le receveur attendu du message. La seule différence entre les trois champs est la façon dont ils sont utilisés.

Le champ "To:" contient la ou les adresses du ou des receveurs principaux du message.

Le champ "Cc:" (où "Cc" veut dire copie carbone dans le sens où l'on fait une copie à la machine à écrire en utilisant du papier carbone) contient les adresses des autres personnes qui doivent recevoir le message, bien que le contenu du message puisse ne pas s'adresser à elles.

Le champ "Bcc:" (ou "Bcc" veut dire copie carbone aveugle) contient les adresses des receveurs du message dont l'adresse n'est pas à révéler aux autres receveurs du message. Il y a trois manières d'utiliser le champ "Bcc". La première, quand un message contenant un champ "Bcc:" est préparé pour l'envoi, la ligne "Bcc:" est enlevée même si tous les receveurs (y compris ceux spécifiés dans le champ "Bcc") se voient envoyer une copie de message. Dans le second cas, les receveurs qui sont spécifiés dans les lignes "To" et "Cc" reçoivent chacun une copie du message avec la ligne "Bcc" enlevée comme ci-dessus, mais les receveurs figurant sur la ligne "Bcc" obtiennent une copie séparée du document contenant la ligne "Bcc". (Quand il y a plusieurs adresses de receveurs dans le champ "Bcc", certaines mises en œuvre envoient en fait une copie séparée du message à chaque receveur avec un "Bcc" contenant seulement l'adresse de ce receveur particulier.) Enfin, comme un champ "Bcc" peut ne contenir aucune adresse, un champ "Bcc" peut être envoyé sans aucune adresse, indiquant aux receveurs que des copies en aveugle ont été envoyées à quelqu'un. La méthode à utiliser avec les champs "Bcc" dépend de la mise en œuvre, mais on peut se reporter à la section Considérations sur la sécurité du présent document pour un exposé sur chaque cas.

Quand un message est une réponse à un autre message, les boîtes aux lettres des auteurs du message d'origine (celles du champ "From:") ou les boîtes aux lettres spécifiées dans le champ "Reply-To:" (s'il existe) PEUVENT apparaître dans le champ "To:" de la réponse car ils seront normalement les receveurs principaux de la réponse. Si une réponse est envoyée à un message qui contient des champs de destination, il est souvent souhaitable d'envoyer une copie de la réponse à tous les receveurs du message, en plus de l'auteur. Quand une telle réponse est formée, les adresses dans les champs "To:" et "Cc:" du message d'origine PEUVENT apparaître dans le champ "Cc:" de la réponse, puisqu'ils sont normalement les receveurs secondaires de la réponse. Si un champ "Bcc:" est présent dans le message d'origine, les adresses de ce champ PEUVENT apparaître dans le champ "Bcc:" de la réponse mais NE DEVRAIENT PAS apparaître dans les champs "To" ou "Cc".

Note : Certaines applications de messagerie ont des commandes de réponse automatique qui incluent les adresses de destination du message d'origine dans les adresses de destination de la réponse. La manière dont ces commandes de réponse agissent dépend de la mise en œuvre et sort du domaine d'application du présent document. En particulier, qu'il faille ou non à inclure les adresses de destination originales lorsque le message d'origine a un champ "Reply-To:" n'est pas traité ici.

3.6.4 Champs d'identification

Bien que ce soit noté comme facultatif dans le tableau du paragraphe 3.6, chaque message DEVRAIT avoir un champ "Identifiant de message:". De plus, les messages de réponse DEVRAIENT avoir des champs "En-réponse-à:" et "Références:" appropriés et comme décrit ci-dessous.

Le champ "Message-ID:" contient un identifiant de message univoque. Les champs "References:" et "In-Reply-To:" contiennent chacun un ou plusieurs identifiants de message univoques, facultativement séparés par un CFWS.

La syntaxe d'identifiant de message (msg-id) est une version limitée de la construction adresse-spec enclose entre les caractères de crochet angulaire, "<" et ">". À la différence de adresse-spec, cette syntaxe permet seulement la forme point-atome-texte sur la partie gauche du "@" et n'a nulle part de CFWS interne dans l'identifiant de message.

Note : Comme avec adresse-spec, une syntaxe libérale est donnée pour le côté droit du "@" dans un msg-id. Cependant, plus loin dans ce paragraphe, l'utilisation d'un domaine pour le côté droit du "@" est RECOMMANDÉE. Ici encore, la syntaxe des constructions de domaine est spécifiée et utilisée par d'autres protocoles (par exemple, [RFC1034], [RFC1035], [RFC1123], [RFC5321]). Il appartient donc aux mises en œuvre de se conformer à la syntaxe des adresses dans le contexte de leur utilisation.

```

message-id    = "Message-ID:" msg-id CRLF
in-reply-to   = "In-Reply-To:" 1*msg-id CRLF
references    = "References:"  1*msg-id CRLF

msg-id        = [CFWS] "<" id-left "@" id-right ">" [CFWS]
id-left       = point-atome-texte / obs-id-left
id-right      = point-atome-texte / no-fold-literal / obs-id-right
no-fold-literal = "[" *dtext "]"

```

Le champ "Message-ID:" fournit un identifiant de message unique qui se réfère à une version particulière d'un message particulier. L'unicité de cet identifiant de message est garantie par l'hôte qui le génère (voir ci-dessous). Cet identifiant de message est destiné à être lisible par une machine et pas forcément compréhensible par l'humain. Un identifiant de message s'applique à exactement une version d'un message particulier ; les révisions ultérieures du message reçoivent chacune un nouvel identifiant de message.

Note: Il y a de nombreuses instances où le message est "changé", mais ces changements ne constituent pas une nouvelle instantiation de ce message, et donc le message ne recevra pas un nouvel identifiant. Par exemple, quand des messages sont introduits dans le système de transport, il leur est souvent ajouté des champs d'en-tête supplémentaires comme des champs de traçage (décrits au paragraphe 3.6.7) et des champs de renvoi (décrits au paragraphe 3.6.6). L'ajout de tels champs d'en-tête ne change pas l'identité du message et donc le champ "Message-ID:" d'origine est conservé. Dans tout les cas, c'est la signification que l'expéditeur du message souhaite véhiculer (c'est-à-dire s'il s'agit du même message ou d'un message différent) qui détermine si le champ "Message-ID:" change ou non, et pas une différence syntaxique particulière qui apparaît (ou n'apparaît pas) dans le message.

Les champs "In-Reply-To:" et "Reference:" sont utilisés lors de la création d'une réponse à un message. Ils contiennent l'identifiant du message d'origine et les identifiants de message des autres messages (par exemple, dans le cas d'une réponse à un message qui était lui-même une réponse). Le champ "In-Reply-To:" peut être utilisé pour identifier le message (ou les messages) auquel répond le nouveau message, alors que le champ "References:" peut être utilisé pour identifier un "fil" de conversation.

Quand on crée une réponse à un message, les champs "In-Reply-To:" et "References:" du message résultant sont construits comme suit :

Le champ "In-Reply-To:" contiendra le contenu du champ "Message-ID:" du message auquel il répond (le message parent). Si il y a plus d'un message parent, alors le champ "In-Reply-To:" devra contenir le contenu de tous les champs "Message-ID:" parents. Si il n'y a de champ "Message-ID:" dans aucun des messages parents, alors le nouveau message n'aura pas de champ "In-Reply-To:".

Le champ "References:" contiendra les contenus des champs "References:" parents (s'il en est) suivis par le contenu des champs "Message-ID:" parents (s'il en est). Si le message parent ne contient pas de champ "References:" mais a un champ "In-Reply-To:" contenant un seul identifiant de message, alors le champ "References:" contiendra le contenu du champ "In-Reply-To:" du parent suivi par le contenu du champ "Message-ID:" du parent (si présent). Si le parent n'a pas de champ "References:", "In-Reply-To:" ou "Message-ID:" alors le nouveau message n'aura pas de champ "References:".

Note: Certaines mises en œuvre analysent le champ "References:" pour afficher le "sujet de discussion". Ces mises en œuvre supposent que chaque nouveau message est une réponse à un parent unique et donc qu'on peut remonter à travers le champ "References:" pour trouver le parent de chaque message qui y est listé. En fait, essayer de former un champ "References:" pour une réponse qui a plusieurs parents est déconseillé et la manière de le faire n'est pas décrite dans ce document.

L'identifiant de message (msg-id) lui-même DOIT être un identifiant unique au monde pour un message. Le générateur de l'identifiant du message DOIT garantir que le msg-id est unique. Il y a plusieurs algorithmes qui peuvent être utilisés pour le faire. Comme le msg-id a une syntaxe similaire à celle de adresse-spec (identique sauf que les chaînes entre guillemets, les commentaires et les espaces de pliage ne sont pas autorisés), une bonne méthode consiste à mettre le nom de domaine (ou une adresse IP littérale de domaine) de l'hôte sur lequel l'identifiant du message a été créé à droite du "@" dans la mesure où les noms de domaines et les adresses IP sont normalement uniques) et de mettre une combinaison de la date et de l'heure absolues courantes avec quelque autre identifiant unique (peut-être séquentiel) disponible sur le système, à gauche. Bien que d'autres algorithmes fonctionnent, il est RECOMMANDÉ que le côté droit contienne un identifiant de domaine (de l'hôte lui même ou d'un autre) de manière à ce que le générateur de l'identifiant du message puisse garantir l'unicité de la partie gauche dans la portée de ce domaine.

Sémantiquement, les caractères de crochet angulaire ">" et "<" ne font pas partie du msg-id; le msg-id est ce qui est contenu entre ces deux caractères.

3.6.5 Champs d'information

Les champs d'information sont tous facultatifs. Les champs "Subject:" et "Comments:" sont des champs non structurés comme défini au paragraphe 2.2.1, et peuvent donc contenir du texte ou des espaces de pliage. Le champ "Keywords:" contient une liste séparée par des virgules d'un ou plusieurs lors ou chaînes entre guillemets.

sujet = "Subject:" non-structuré CRLF
commentaires = "Comments:" non-structuré CRLF
mots-clés = "Keywords:" phrase *(", phrase) CRLF

Ces trois champs sont destinés à avoir uniquement des contenus lisibles par l'homme avec des informations sur le message. Le champ "Subject:" est le plus courant et contient un courte chaîne identifiant le sujet du message. Quand on l'utilise dans une réponse, le corps du champ PEUT commencer par la chaîne "Re:" (abréviation du latin "in re", au sujet de) suivi par le contenu du corps du champ "Subject:" du message original. Si cela est fait, une seule instance de la chaîne littérale "Re:" doit être utilisée car l'utilisation d'autres chaînes ou de plusieurs instances de cette chaîne pourrait conduire à des conséquences indésirables. Le champ "Comments:" contient n'importe quel commentaire supplémentaire sur le texte du corps du message. Le champ "Keywords:" (*mots-clés*) contient une liste séparée par des virgules de mots et phrases importants qui peuvent être utiles au receveur.

3.6.6 Champs de renvoi

Des champs de renvoi DEVRAIENT être ajoutés à tout message réintroduit par un utilisateur dans le système de transport. Un jeu séparé de champs de renvoi DEVRAIT être ajouté chaque fois que cela est fait. Tous les champs de renvoi correspondant à un renvoi particulier du message DEVRAIENT être rassemblés. Chaque nouveau jeu des champs de renvoi est ajouté au message ; c'est-à-dire que le jeu le plus récent apparaît le plus en haut du message. Aucun autre champ du message n'est changé quand les champs de renvoi sont ajoutés.

Chacun des champs de renvoi correspond à un champ particulier ailleurs dans la syntaxe. Par exemple, le champ "Resent-Date:" correspond au champ "Date:" et le champ "Resent-To:" correspond au champ "To:" Dans chaque cas, la syntaxe pour le corps du message est identique à la syntaxe donnée précédemment pour le champ correspondant.

Lorsque les champs de renvoi sont utilisés, les champs "Resent-From:" et "Resent-Date:" DOIVENT être envoyés. Le champ "Resent-Message-ID:" DEVRAIT être envoyé. "Resent-Sender:" NE DEVRAIT PAS être utilisé si "Resent-Sender:" serait identique à "Resent-From:".

resent-date = "Resent-Date:" date-heure CRLF
resent-from = "Resent-From:" liste-de-boîte-aux-lettres CRLF
resent-sender = "Resent-Sender:" boîte-aux-lettres CRLF
resent-to = "Resent-To:" liste-d'adresses CRLF
resent-cc = "Resent-Cc:" liste-d'adresses CRLF

resent-bcc = "Resent-Bcc:" [liste-d'adresses / CFWS] CRLF

resent-msg-id = "Resent-Message-ID:" msg-id CRLF

Les champs de renvoi sont utilisés pour identifier qu'un message a été réintroduit dans le système de transport par un utilisateur. L'objet de l'utilisation de ces champs de renvoi est de faire apparaître le message au receveur final comme s'il était envoyé directement par l'expéditeur d'origine, avec tous les champs d'origine restant inchangés. Chaque ensemble de champs de renvoi correspond à un événement de renvoi particulier. En fait, si un message est envoyé plusieurs fois, chaque jeu de champs de renvoi donne des informations d'identification de chaque phase du message. Les champs de renvoi sont purement informatifs. Ils NE DOIVENT PAS être utilisés dans le traitement normal des réponses ou autre action automatique sur les messages.

Note : Réintroduire un message dans le système de transport et utiliser les champs de renvoi est une opération différente de "Transmettre". Cette dernière a deux significations : la première dans le sens où un utilisateur peut donner pour instruction à un programme de lecture de la messagerie de transmettre une copie d'un message à une autre personne, faisant du message transmis le corps du nouveau message. Un message transmis dans ce sens ne paraît pas venir de l'émetteur d'origine, mais est un message entièrement nouveau de l'émetteur du message. Transmettre peut aussi signifier qu'un programme de transport de messagerie reçoit un message et le fait suivre sur une destination différente pour sa livraison finale. Les champs de renvoi ne sont destinés à être utilisés avec aucun de ces types de retransmission.

Les champs d'origine de renvoi indiquent la boîte aux lettres de la ou des personnes ou systèmes qui renvoient ce message. Comme avec les champs d'origine réguliers, il existe deux formes : une forme simple "Resent-From:" qui contient la boîte aux lettres de l'individu qui a fait le renvoi, et une forme plus complexe, quand un individu (identifié dans le champ "Resent-Sender:") renvoie un message au nom d'un ou plusieurs individus (identifiés dans le champ "Resent-From:").

Note : Quand on répond à un message de renvoi, les réponses se comportent de la même manière qu'avec un message classique en utilisant les champs "From:", "Reply-To:", "Message-ID:", et autres. Les champs de renvoi sont seulement informatifs et NE DOIVENT PAS être utilisés dans le traitement normal des réponses.

"Resent-Date:" indique la date et l'heure à laquelle le message de renvoi a été expédié par le retransmetteur du message. Tout comme pour le champ "Date:" il ne s'agit pas des date et heure à laquelle le message a été réellement transporté.

Les champs "Resent-To:", "Resent-Cc:" et "Resent-Bcc:" fonctionnent respectivement de la même manière que les champs "To:", "Cc:", et "Bcc:" sauf qu'ils indiquent le receveur du message de renvoi et pas les receveurs du message original.

Le champ "Resent-Message-ID:" fournit un identifiant univoque pour le message renvoyé.

3.6.7 Champs de traçage

Les champs de traçage sont un groupe de champs d'en-tête comprenant un champ "Return-Path:" facultatif, et un ou plusieurs champs "Received:". Le champ d'en-tête "Return-Path:" contient une paire de crochets angulaires qui entourent une adresse-spec facultative. Le champ "Received:" contient une liste (qui peut être vide) de jetons suivie par un point virgule et une spécification de date/heure. Chaque jeton doit être un mot, une angle-addr, une adresse-spec, ou un domaine. Des restrictions supplémentaires sont appliquées à la syntaxe des champs de traçage par les spécifications qui traitent de leur utilisation (comme la RFC 5321).

trace = [return]
l*received

return = "Return-Path:" chemin CRLF

chemin = angle-addr / ([CFWS] "<" [CFWS] ">" [CFWS])

received = "Received:" *jeton-reçu ";" date-heure CRLF

jeton-reçu = mot / angle-addr / adresse-spec / domaine

Un exposé complet sur l'utilisation des champs de traçage dans la messagerie Internet figure dans la [RFC5321]. Pour les besoins de la présente spécification, les champs de traçage sont strictement pour information, et leur interprétation formelle sort du domaine d'application du présent document.

3.6.8 Champs facultatifs

Des champs peuvent apparaître dans des messages qui ne sont pas autrement spécifiés dans ce document. Ils DOIVENT se conformer à la syntaxe de champ facultatif. C'est un nom de champ, constitué de caractères US-ASCII imprimables sauf SP et deux points, suivi par deux points, suivis par tout texte qui se conforme à la syntaxe de non structuré.

Le nom de champ de tout champ facultatif NE DOIT être identique à aucun nom de champ spécifié ailleurs dans ce document.

champ-facultatif = nom-de-champ ":" non-structuré CRLF

nom-de-champ = 1*ftext

ftext = %d33-57 / ; Caractères US-ASCII nom imprimables, ":" non inclus.
%d59-126

Pour les besoins de la présente spécification, tout champ facultatif est non interprété.

4. Syntaxe obsolète

Les versions antérieures de la présente spécification permettaient des syntaxes différentes (habituellement plus libérales) que celle admise dans cette version. De plus, il y a eu des éléments syntaxiques utilisés dans les messages sur Internet dont l'interprétation n'avait jamais été documentée. Bien que certaines de ces formes NE DOIVENT PAS être générées conformément à la grammaire de la Section 3, elles DOIVENT être acceptées et analysées par un receveur conforme. Cette section spécifie beaucoup de ces éléments syntaxiques. En prenant la grammaire de la Section 3 et en y ajoutant les définitions de cette section, on aura toute la grammaire à utiliser pour l'interprétation des messages.

Note : Cette section identifie les formes syntaxiques que toute mise en œuvre DOIT raisonnablement interpréter. Cependant, il y a certainement des messages Internet qui ne se conforment même pas à la syntaxe supplémentaire de cette section. Le fait qu'une forme particulière n'apparaisse dans aucune section de ce document n'est pas une raison pour que les programmes connaissent des défaillances ou que des données mal formées soient irrémédiablement perdues par une mise en œuvre quelconque. Il appartient aux mises en œuvre de traiter les messages de façon robuste.

Une différence importante entre la syntaxe obsolète (l'interprétation) et la syntaxe actuelle (générée) est que dans les corps de champs d'en tête structurés (c'est-à-dire entre les deux points et le CRLF de tout champ d'en-tête structuré) des caractères d'espace blanche, y compris d'espace de pliage, et des commentaires peuvent être librement insérés entre n'importe quels jetons lexicaux. Ce qui permet de nombreuses formes complexes qui ont donné quelques difficultés d'analyse à certaines mises en œuvre.

Une autre différence clé entre la syntaxe obsolète et l'actuelle est que la règle du paragraphe 3.2.2 concernant les lignes entièrement composées d'espaces blanches dans des commentaires et d'espaces de pliage ne s'applique pas. Voir la discussion sur l'espace de pliage au paragraphe 4.2 ci-dessous.

Enfin, certains caractères qui étaient précédemment permis dans les messages apparaissent dans cette section. Le caractère NUL (valeur ASCII 0) était permis mais ne l'est plus pour des raisons de compatibilité. De même, les caractères de contrôle US-ASCII autres que CR, LF, SP, et HTAB (valeurs ASCII de 1 à 8, 11, 12, 14 à 31, et 127) pouvaient apparaître dans les corps de champ d'en-tête. CR et LF étaient autorisés dans les messages, autrement que sous la forme CRLF, cet usage sera aussi expliqué ici.

D'autres différences dans la syntaxe et la sémantique sont notées dans les paragraphes suivants.

4.1 Divers jetons obsolètes

Ces éléments syntaxiques sont utilisés ailleurs dans la syntaxe obsolète ou dans la syntaxe principale. Les caractères CR seul, LF seul, et NUL sont ajoutés à obs-qp, obs-body, et obs-unstruct. Les caractères de contrôle US-ASCII sont ajoutés à obs-qp, obs-unstruct, obs-ctext, et obs-qttext. Le caractère point est ajouté à obs-phrase. La liste obs-phrase-list permet une liste séparée par des virgules (éventuellement vide) de phrases qui peuvent comporter des éléments "nuls". C'est à dire qu'il peut y avoir deux virgules ou plus dans une telle liste avec rien entre elles, ou des virgules au début ou à la fin de la liste.

Note : Le caractère "point" (ou "point final") (".") dans une obs-phrase est une forme qui n'était pas permise dans les

versions précédentes de la présente spécification ou de toute autre. Le point (pas plus qu'aucun autre caractère de la série spéciale) n'était pas permis dans la phrase car il introduisait une difficulté d'analyse pour faire la distinction entre les phrases et les portions d'adresse-spec (voir au paragraphe 4.4). Il apparaît ici parce que le caractère point est couramment utilisé dans de nombreux messages dans la portion affichage-du-nom des adresses, surtout pour les initiales dans les noms, et donc il doit être interprété correctement.

```
obs-NO-WS-CTL = %d1-8 /      ; caractères de contrôle US-ASCII qui n'incluent pas les caractères
                %d11 /      ; retour-chariot, saut à la ligne, et espace blanche
                %d12 /
                %d14-31 /
                %d127
```

```
obs-ctext = obs-NO-WS-CTL
```

```
obs-qttext = obs-NO-WS-CTL
```

```
obs-utext = %d0 / obs-NO-WS-CTL / VCHAR
```

```
obs-qp = "\" (%d0 / obs-NO-WS-CTL / LF / CR)
```

```
obs-body = *((*LF *CR *((%d0 / text) *LF *CR)) / CRLF)
```

```
obs-unstruct = *((*LF *CR *(obs-utext *LF *CR)) / FWS)
```

```
obs-phrase = mot *(mot / "." / CFWS)
```

```
obs-phrase-list = [phrase / CFWS] *(", " [phrase / CFWS])
```

CR seul et LF seul apparaissent dans les messages avec deux significations différentes. Dans de nombreux cas, CR seul ou LF seul sont utilisés de façon impropre à la place de CRLF pour indiquer les séparateurs de ligne. Dans d'autres cas, le CR seul et le LF seul sont utilisés simplement comme caractères de contrôle US-ASCII avec leur signification ASCII traditionnelle.

4.2. Espace de pliage obsolète

Dans la syntaxe obsolète, n'importe quelle quantité d'espaces de pliage PEUT être insérée lorsque la règle obs-FWS est permise. Cela offre la possibilité d'avoir deux "pliages" consécutifs dans une ligne, et donc la possibilité qu'une ligne qui constitue un champs d'en-tête replié soit entièrement composée d'espaces blanches.

```
obs-FWS = 1*WSP *(CRLF 1*WSP)
```

4.3. Date et heure obsolète

La syntaxe de format de date obsolète permet d'utiliser dans les versions antérieures de cette spécification une année sur deux chiffres dans le champ de date et une liste de spécificateurs alphabétiques de fuseaux horaires. Elle permet aussi des commentaires et des espaces de pliage entre beaucoup des jetons.

```
obs-jour-de-semaine = [CFWS] nom-de-jour [CFWS]
```

```
obs-jour = [CFWS] 1*2DIGIT [CFWS]
```

```
obs-an = [CFWS] 2*2DIGIT [CFWS]
```

```
obs-heure = [CFWS] 2DIGIT [CFWS]
```

```
obs-minute = [CFWS] 2DIGIT [CFWS]
```

```
obs-seconde = [CFWS] 2DIGIT [CFWS]
```

```
obs-zone = "UT" / "GMT" /      ; Décalages Temps Universel UT Nord américain
```

"EST" / "EDT" /	; Est : - 5/ - 4
"CST" / "CDT" /	; Central : - 6/ - 5
"MST" / "MDT" /	; Montagne : - 7/ - 6
"PST" / "PDT" /	; Pacifique : - 8/ - 7
%d65-73 /	; Zones militaires – de "A" à "I" et de "K" à "Z", en majuscules
%d75-90 /	; et minuscules
%d97-105 /	
%d107-122	

Lorsque une année à deux ou trois chiffres survient dans une date, l'année doit être interprétée comme suit : si on rencontre une année à deux chiffres dont la valeur est entre 00 et 49, l'année est interprétée en ajoutant 2000, ce qui donne une valeur entre 2000 et 2049. Si on rencontre une année à deux chiffres avec une valeur entre 50 et 99, ou si on rencontre une année à trois chiffres, l'année est interprétée en ajoutant 1900.

Dans la zone horaire obsolète, "UT" et "GMT" sont des indications du "temps universel" et de "l'heure moyenne de Greenwich" qui sont sémantiquement égales à "+0000".

Les trois caractères restants de zone sont pour le temps US. La première lettre "E", "C", "M", ou "P" vaut pour "Est", "Central", "Montagnes", et "Pacifique". La deuxième lettre est soit "S" pour l'heure "Standard", soit "D" pour "Daylight" (l'heure d'été). Leur interprétation se fait comme suit :

EDT est sémantiquement équivalent à -0400
 EST est sémantiquement équivalent à -0500
 CDT est sémantiquement équivalent à -0500
 CST est sémantiquement équivalent à -0600
 MDT est sémantiquement équivalent à -0600
 MST est sémantiquement équivalent à -0700
 PDT est sémantiquement équivalent à -0700
 PST est sémantiquement équivalent à -0800

Les zones horaires militaires à un caractère étaient définies de manière non standard dans la [RFC0822] et sont donc imprévisibles quant à leur signification. Les définitions d'origine des zones militaires de "A" à "I" sont équivalentes à "+0100" à "+0900" respectivement ; "K", "L", et "M" sont équivalentes à "+1000", "+1100", et "+1200" respectivement ; de "N" à "Y" est équivalent à de "-0100" à "-1200" respectivement ; et "Z" est équivalent à "+0000". Cependant, à cause d'erreurs dans la [RFC0822], elles DEVRAIENT toutes être considérées comme équivalentes à "-0000" sauf s'il y a des informations hors bande qui confirment leur signification.

D'autres zones horaires alphabétiques multi-caractères (habituellement entre 3 et 5) ont été utilisées dans les messages Internet. Toute zone horaire dont la signification n'est pas connue DEVRAIT être considérée comme équivalente à "-0000" sauf si des informations hors bande confirment leur signification.

4.4 Adressage obsolète

Il y a quatre différences principales dans l'adressage. Tout d'abord, il était permis aux adresses de boîte aux lettres d'avoir une portion de chemin avant l'adresse-spec lorsque incluse entre "<" et ">". Le chemin est simplement une liste séparée par des virgules de noms de domaines, précédé chacun par "@", et la liste est terminée par deux points. Ensuite, les CFWS étaient permis entre les éléments séparés par des points de partie locale et de domaine (c'est-à-dire que point-atome n'était pas utilisé). De plus, il est permis à la partie locale de contenir une chaîne entre guillemets en plus d'un simple atome. Troisièmement, liste de boîte aux lettres et liste d'adresses pouvaient avoir des membres "nuls". C'est à dire qu'il pouvait y avoir deux virgules ou plus dans une telle liste avec rien entre elles, ou des virgules au début ou à la fin de la liste. Enfin, les caractères de contrôle US-ASCII et les paires entre guillemets étaient permises dans les littéraux de domaines et ils sont ajoutés ici.

obs-angle-addr = [CFWS] "<" obs-route addr-spec ">" [CFWS]

obs-route = obs-domain-list ":"

obs-domain-list = *(CFWS / ",") "@" domaine
 *(", " [CFWS] ["@" domaine])

obs-mbox-list = *([CFWS] ",") boîte-aux-lettres *(", " [boîte-aux-lettres / CFWS])

obs-addr-list = *([CFWS] ",") adresse *(",") [adresse / CFWS])

obs-group-list = 1*([CFWS] ",") [CFWS]

obs-local-part = mot *("." mot)

obs-domain = atome *("." atome)

obs-dtext = obs-NO-WS-CTL / paire-entre-guillemets

Pour interpréter les adresses, la portion route DEVRAIT être ignorée.

4.5 Champs d'en-tête obsolètes

Syntaxiquement, la différence principale dans la syntaxe des champs d'en-tête obsolètes est qu'elle permet plusieurs occurrences de n'importe quel champ et qu'ils peuvent survenir dans n'importe quel ordre. De plus n'importe quel nombre d'espaces est permis avant le ":" à la fin du champ de nom.

obs-fields = *(obs-return /
 obs-received /
 obs-orig-date /
 obs-from /
 obs-sender /
 obs-reply-to /
 obs-to /
 obs-cc /
 obs-bcc /
 obs-message-id /
 obs-in-reply-to /
 obs-references /
 obs-subject /
 obs-comments /
 obs-keywords /
 obs-resent-date /
 obs-resent-from /
 obs-resent-send /
 obs-resent-rply /
 obs-resent-to /
 obs-resent-cc /
 obs-resent-bcc /
 obs-resent-mid /
 obs-optional)

Sauf pour les champs d'adresse de destination (décrits au paragraphe 4.5.3), l'interprétation d'occurrences multiples des champs est non spécifiée. De même l'interprétation des champs de traçage et des champs de renvoi qui ne surviennent pas dans les blocs ajoutés en tête du message sont aussi non spécifiés. Sauf indication contraire dans les paragraphes suivants, l'interprétation des autres champs est identique à l'interprétation de leur contrepartie non obsolète de la section 3.

4.5.1 Champs de date d'origine obsolètes

obs-orig-date = "Date" *WSP ":" date-heure CRLF

4.5.2 Champs d'origine obsolètes

obs-from = "From" *WSP ":" liste-de-boîte-aux-lettres CRLF

obs-sender = "Sender" *WSP ":" boîte-aux-lettres CRLF

obs-reply-to = "Reply-To" *WSP ":" liste-d'adresses CRLF

4.5.3 Champs d'adresse de destination obsolètes

obs-to = "To" *WSP ":" liste-d'adresses CRLF

obs-cc = "Cc" *WSP ":" liste-d'adresses CRLF

obs-bcc = "Bcc" *WSP ":" (liste-d'adresses / (*(CFWS] ",") [CFWS])) CRLF

Quand plusieurs occurrences de champs d'adresse de destination apparaissent dans un message, elles DEVRAIENT être traitées comme si la liste d'adresses dans la première occurrence du champ était combinée avec la liste d'adresses de l'occurrence suivante en ajoutant une virgule et en l'enchaînant.

4.5.4 Champs d'identification obsolètes

Les champs obsolètes "In-Reply-To:" et "References:" diffèrent de la syntaxe actuelle en ce qu'ils permettent l'apparition d'une phrase (mots ou chaînes entre guillemets). Les formes obsolètes des côtés gauche et droit du msg-id permettent des CFWS épars, les rendant respectivement syntaxiquement identiques à la partie locale et au domaine.

obs-message-id = "Message-ID" *WSP ":" msg-id CRLF
 obs-in-reply-to = "In-Reply-To" *WSP ":" *(phrase / msg-id) CRLF
 obs-references = "References" *WSP ":" *(phrase / msg-id) CRLF
 obs-id-gauche = partie-locale
 obs-id-droit = domaine

Pour les besoins de l'interprétation, les phrases dans les champs "In-Reply-To:" et "References:" sont ignorées.

Sémantiquement, aucun des CFWS facultatifs dans la partie locale et dans le domaine ne fait partie, respectivement, de obs-id-gauche ni de obs-id-droit.

4.5.5 Champs d'information obsolètes

obs-subject = "Subject" *WSP ":" non-structuré CRLF
 obs-comments = "Comments" *WSP ":" non-structuré CRLF
 obs-keywords = "Keywords" *WSP ":" obs-phrase-list CRLF

4.5.6 Champs de renvoi obsolètes

La syntaxe obsolète ajoute un champ "Resent-Reply-To:", qui consiste en un champ de nom, un commentaire facultatif et une espace de pliage, deux points, et une liste d'adresses séparées par une virgule.

obs-resent-from = "Resent-From" *WSP ":" liste-de-boîte-aux-lettres CRLF
 obs-resent-send = "Resent-Sender" *WSP ":" boîte-aux-lettres CRLF
 obs-resent-date = "Resent-Date" *WSP ":" date-heure CRLF
 obs-resent-to = "Resent-To" *WSP ":" liste-d'adresses CRLF
 obs-resent-cc = "Resent-Cc" *WSP ":" liste-d'adresses CRLF
 obs-resent-bcc = "Resent-Bcc" *WSP ":" (liste-d'adresses / (*(CFWS] ",") [CFWS])) CRLF
 obs-resent-mid = "Resent-Message-ID" *WSP ":" msg-id CRLF
 obs-resent-rply = "Resent-Reply-To" *WSP ":" liste-d'adresses CRLF

Comme les autres champs de renvoi, le champ "Resent-Reply-To:" est à traiter seulement comme informations de traçage.

4.5.7 Champs de traçage obsolètes

Les champs obs-return et obs-received ne sont encore ici donnés que comme gabarit de définition, tout comme return et received à la section 3. Leur syntaxe complète est donnée dans la [RFC5321].

obs-return = "Chemin-de-retour" *WSP ":" chemin CRLF
 obs-received = "Reçu" *WSP ":" *jeton-reçu CRLF

4.5.8 Champs facultatifs obsolètes

obs-optional = nom-de-champ *WSP ":" non-structuré CRLF

5. Considérations pour la sécurité

Il faut faire attention quand on affiche des messages sur un terminal ou son émulateur. Des terminaux puissants peuvent

agir sur les séquences d'échappement ou toute autre combinaison de caractères de contrôle ASCII avec des conséquences variées. Ils peuvent réallouer les touches du clavier ou permettre d'autres modifications au terminal qui peuvent conduire à un déni de service ou même à endommager les données. Ils peuvent déclencher des messages de réponse (parfois programmables) qui peuvent permettre à un message de produire des commandes au nom du receveur. Ils peuvent aussi affecter le fonctionnement des appareils connectés au terminal tel qu'une imprimante. Ceux qui visualisent les messages peuvent souhaiter neutraliser dans le message les séquences d'échappement terminales potentiellement dangereuses avant l'affichage. Cependant, d'autres séquences d'échappement apparaissent dans les messages dans un but utile (cf. [ISO.2022.1994], [RFC2045], [RFC2046], [RFC2047], [RFC2049], [RFC4288], [RFC4289]) et ne devraient donc pas être éliminées sans discrimination.

La transmission d'objets non textuels dans un message soulève des problèmes de sécurité supplémentaires. Ces problèmes sont exposés dans les [RFC2045], [RFC2046], [RFC2047], [RFC2049], [RFC4288] et [RFC4289].

De nombreuses mises en œuvre utilisent le champ "Bcc:" (copie carbone aveugle) décrit au paragraphe 3.6.3 pour faciliter l'envoi de messages au receveur sans montrer l'adresse d'un ou plusieurs autres receveurs. Un mauvais usage de "Bcc:" peut révéler des informations confidentielles qui pourraient finalement conduire à des problèmes de sécurité par la simple connaissance de l'existence d'une adresse de messagerie particulière. Par exemple, si en utilisant la première méthode décrite au paragraphe 3.6.6, où la ligne "Bcc:" est enlevée du message, les receveurs en aveugle n'ont pas d'indication explicite qu'une copie en aveugle leur a été envoyée, sauf par le fait que leurs adresses n'apparaissent pas dans l'en tête du message. À cause de cela, un des receveurs en aveugle pourrait envoyer une réponse à tous les receveurs apparents et révéler accidentellement que le message a été envoyé à ce receveur en aveugle. Quand on utilise la seconde méthode du paragraphe 3.6.3, l'adresse du receveur en aveugle apparaît dans le champ "Bcc:" d'une copie séparé du message. Si le champ "Bcc:" envoyé contient tous les destinataires en aveugle, tous les receveurs du "Bcc:" seront vus par chaque receveur du "Bcc:". Même si un message séparé est envoyé à chaque "Bcc:" avec seulement une adresse individuelle, il faut encore que les mises en œuvre veillent à traiter les réponses au message conformément au paragraphe 3.6.3 de manière à ne pas révéler accidentellement le receveur en aveugle aux autres receveurs.

6. Considérations relatives à l'IANA

Le présent document met à jour les enregistrements qui apparaissaient dans la [RFC4021] qui se référaient aux définitions de la [RFC2822]. L'IANA a mis à jour le répertoire permanent des champs d'en-tête de message avec les champs d'en-tête suivants, conformément aux procédures établies dans la [RFC3864].

Nom de champ d'en-tête : Date
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.1)

Nom de champ d'en-tête : From
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.2)

Nom de champ d'en-tête : Sender
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.2)

Nom de champ d'en-tête : Reply-To
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.2)

Nom de champ d'en-tête : To
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF

Document(s) de spécification : ce document (paragraphe 3.6.3)

Nom de champ d'en-tête : Cc
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.3)

Nom de champ d'en-tête : Bcc
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.3)

Nom de champ d'en-tête : Message-ID
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.4)

Nom de champ d'en-tête : In-Reply-To
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.4)

Nom de champ d'en-tête : References
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.4)

Nom de champ d'en-tête : Subject
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.5)

Nom de champ d'en-tête : Comments
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.5)

Nom de champ d'en-tête : Keywords
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.5)

Nom de champ d'en-tête : Resent-Date
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)

Nom de champ d'en-tête : Resent-From
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)
Nom de champ d'en-tête : Resent-Sender
Protocole applicable : Mail

Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)

Nom de champ d'en-tête : Resent-To
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)

Nom de champ d'en-tête : Resent-Cc
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)

Nom de champ d'en-tête : Resent-Bcc
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)

Nom de champ d'en-tête : Resent-Reply-To
Protocole applicable : Mail
Statut : obsolète
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 4.5.6)

Nom de champ d'en-tête : Resent-Message-ID
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.6)

Nom de champ d'en-tête : Return-Path
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.7)

Nom de champ d'en-tête : Received
Protocole applicable : Mail
Statut : standard
Auteur/contrôleur des changements : IETF
Document(s) de spécification : ce document (paragraphe 3.6.7)
Informations en rapport : [RFC5321]

Appendice A. Exemples de messages

Cette section présente un choix de messages. Ils sont destinés à aider à la mise en œuvre de cette spécification, mais ne devraient pas être considérés comme normatifs ; cela signifie que bien que les exemples de cette section aient été revus avec soin, s'il se trouvait un conflit entre ces exemples et la syntaxe décrite aux sections 3 et 4 de ce document, c'est la syntaxe de ces sections qui doit être considérée comme correcte.

Dans la version texte de ce document, les messages de cette section sont délimités entre les lignes de "----". Les lignes de "----" ne font pas partie du message lui-même.

A.1 Exemples d'adressage

Les exemples qui suivent sont des messages qui pourraient être envoyés entre deux individus.

A.1.1. Message d'une personne à une autre avec simple adressage

Cela pourrait s'appeler un message canonique. Il y a un seul auteur, John Doe, un seul receveur, Mary Smith, un sujet, une date, un identifiant de message, et un message textuel dans le corps.

From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Dire bonjour
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

C'est un message pour dire simplement bonjour.
Alors, "Bonjour".

Si le secrétaire de John, Michael a en fait envoyé le message, bien que John soit l'auteur et que la réponse au message doivent lui revenir, le champ "Sender:" serait utilisé :

From: John Doe <jdoe@machine.example>
Sender: Michael Jones <mjones@machine.example>
To: Mary Smith <mary@example.net>
Subject: Dire bonjour
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

C'est un message juste pour dire bonjour
alors, "Bonjour".

A.1.2 Différents types de boîtes aux lettres

Ce message inclut plusieurs adresses dans les champs de destination et utilise aussi plusieurs formes d'adresse différentes.

From: "Joe Q. Public" <john.q.public@example.com>
To: Mary Smith <mary@x.test>, jdoe@example.org, Who? <one@y.test>
Cc: <boss@nil.test>, "Giant; \"Big\" Box" <syssservices@example.net>
Date: Tue, 1 Jul 2003 10:52:37 +0200
Message-ID: <5678.21-Nov-1997@example.com>

Salut tout le monde.

Noter que les noms d'affichage pour Joe Q. Public et Giant; "Big" Box ont dus être mis entre guillemets parce que le premier contient un point et que l'autre contient à la fois les caractères point virgule et guillemets (les guillemets apparaissant comme une construction de paire entre guillemets). À l'inverse, le nom d'affichage pour Who? n'a pas besoin de guillemets car le point d'interrogation est légal dans un atome. Noter également que jdoe@example.org et boss@nil.test n'ont pas du tout de nom d'affichage associé, et jdoe@example.org utilise la forme d'adresse plus simple sans les crochets angulaires.

A.1.3 Adresses de groupe

From: Pete <pete@silly.example>
To: A Group:Ed Jones <c@a.test>,joe@where.test,John <jdoe@one.test>;
Cc: Undisclosed recipients;;
Date: Thu, 13 Feb 1969 23:32:54 -0330
Message-ID: <testabcd.1234@silly.example>

Essai.

Dans ce message, le champ "To:" a un unique groupe de receveurs nommé "Groupe A" qui contient trois adresses, et un champ "Cc:" qui contient un groupe vide nommé "Undisclosed recipients" (*receveurs non révélés*).

A.2 Messages de réponse

Ce qui suit est une série de trois messages qui suivent le fil d'une conversation entre John et Mary. John envoie un premier message à Mary, qui lui répond puis John répond au message de réponse de Mary.

Noter surtout les champs "Message-Id:", "References:", et "In-Reply-To:" dans chaque message.

From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Dire bonjour
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

C'est un message juste pour dire bonjour
alors, "Bonjour".

Quand on envoie une réponse, le champ Subject est souvent conservé, bien qu'on y rajoute "Re:" devant, comme décrit au paragraphe 3.6.5.

From: Mary Smith <mary@example.net>
To: John Doe <jdoe@machine.example>
Reply-To: "Mary Smith: Compte personnel" <smith@home.example>
Subject: Re: Dire bonjour
Date: Fri, 21 Nov 1997 10:01:10 -0600
Message-ID: <3456@example.net>
In-Reply-To: <1234@local.machine.example>
References: <1234@local.machine.example>

C'est une réponse à ton bonjour

Noter le champ "Reply-To:" dans le message ci-dessus. Quand John répond au message de Mary, la réponse devrait aller à l'adresse spécifiée dans le champ "Reply-To:" au lieu de celle du champ "From:".

To: "Mary Smith: Compte personnel" <smith@home.example>
From: John Doe <jdoe@machine.example>
Subject: Re: Dire bonjour
Date: Fri, 21 Nov 1997 11:00:00 -0600
Message-ID: <abcd.1234@local.machine.test>
In-Reply-To: <3456@example.net>
References: <1234@local.machine.example> <3456@example.net>

C'est une réponse à ta réponse.

A.3 Messages de renvoi

Commençons avec le message qui a déjà été utilisé plusieurs fois comme exemple :

From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>

Subject: Dire bonjour
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

C'est un message juste pour dire bonjour.
Alors, "Bonjour".

Disons que Mary, à réception de ce message, souhaite en envoyer une copie à Jane de manière à ce que, (a) le message semble provenir directement de John et que (b) si Jane répond à ce message, la réponse devrait aller à John, et que (c) toutes les informations d'origine telles que la date d'envoi originale du message à Mary, l'identifiant du message, et le destinataire d'origine soient préservées. Dans ce cas, les champs de renvoi sont ajoutés devant le message :

Resent-From: Mary Smith <mary@example.net>
Resent-To: Jane Brown <j-brown@other.example>
Resent-Date: Mon, 24 Nov 1997 14:22:01 -0800
Resent-Message-ID: <78910@example.net>
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Dire bonjour
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>

C'est un message juste pour dire bonjour.
Alors, "Bonjour".

Si Jane souhaite à son tour envoyer ce message à une autre personne, elle devra ajouter son propre jeu de champs de renvoi par dessus et envoyer le tout. (Noter que pour faire court, les champs de traçage ne sont pas montrés.)

A.4 Messages avec champs de traçage

Lorsque les messages sont envoyés à travers le système de transport comme décrit dans la [RFC5321], les champs de traçage sont ajoutés en tête du message. Ce qui suit est un exemple de ce à quoi peuvent ressembler ces champs de traçage. Noter qu'il y a des espaces de pliage dans le premier car ces lignes peuvent être longues.

Received: from x.y.test
by example.net
via TCP
with ESMTP
id ABC12345
for <mary@example.net>; 21 Nov 1997 10:05:43 -0600
Received: from node.example by x.y.test; 21 Nov 1997 10:01:22 -0600
From: John Doe <jdoe@node.example>
To: Mary Smith <mary@example.net>
Subject: Dire bonjour
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.node.example>

C'est un message juste pour dire bonjour.
Alors, "Bonjour".

A.5 Espaces blanches, commentaires, et autres particularités

Les espaces, y compris celles de pliage, et les commentaires peuvent être insérés entre beaucoup des jetons des champs. En reprenant l'exemple du A.1.3, espaces et commentaires peuvent être insérés dans tous les champs.

From: Pete(Un type \) sympa) <pete(son compte)@silly.test(son hôte)>
 To:Un groupe(de plusieurs personnes)
 :Chris Jones <c@(l'hôte de Chris.)public.example>,
 joe@example.org,
 John <jdoe@one.test> (mon cher ami); (la fin du groupe)Cc:(Liste vide)(début)Undisclosed recipients :(personne (à ma connaissance)) ;
 Date: Thu,
 13
 Feb
 1969
 23:32
 -0330 (Heure de Terre-Neuve)
 Message-ID: <testabcd.1234@silly.test>

Essai.

L'exemple ci-dessus est déplaisant du point de vue esthétique, mais parfaitement légal. Noter en particulier (1) les commentaires dans le champ "From:" (y compris celui qui a un caractère ")") qui apparaît dans une paire entre guillemets ; (2) l'espace absente après le ":" dans le champ "To:" ainsi que le commentaire et l'espace de pliage après le nom de groupe, le caractère spécial (".") dans le commentaire dans l'adresse de Chris Jones, et l'espace de pliage avant et après "joe@example.org," ; (3) les commentaires multiples et incorporés dans le champ "Cc:" ainsi que le commentaire qui suit immédiatement le ":" après "Cc" ; (4) l'espace de pliage (mais pas de commentaire sauf à la fin) et les secondes qui manquent dans l'heure du champ de date ; et (5) l'espace avant (mais pas dans) l'identifiant dans le champ "Message-ID:".

Les exemples qui suivent sont ceux d'éléments de syntaxe obsolète (c'est à dire, à "NE PAS générer") décrits à la section 4 de ce document.

A.6.1 Adressage obsolète

Noter dans l'exemple ci-dessous l'absence des guillemets autour de Joe Q. Public, le chemin qui apparaît dans l'adresse de Mary Smith, les deux virgules qui apparaissent dans le champ "To:", et les espaces qui apparaissent autour du "." dans l'adresse de jdoe.

From: Joe Q. Public <john.q.public@example.com>
 To: Mary Smith <@node.test:mary@example.net>, , jdoe@test . example
 Date: Tue, 1 Jul 2003 10:52:37 +0200
 Message-ID: <5678.21-Nov-1997@example.com>

Salut à tous.

A.6.2 Dates obsolètes

Le message suivant utilise un format de date obsolète, incluant une zone d'heure non numérique et une année à deux chiffres. Notez que bien que le jour de la semaine (*day-of-week*) soit manquant, cela n'est pas spécifique de la syntaxe obsolète ; il est facultatif dans la syntaxe courante.

From: John Doe <jdoe@machine.example>
 To: Mary Smith <mary@example.net>
 Subject: Saying Hello
 Date: 21 Nov 97 09:55:06 GMT
 Message-ID: <1234@local.machine.example>

C'est un message pour simplement dire bonjour.
 Alors, "Bonjour".

A.6.3 Espaces et commentaires obsolètes

Les espaces et les commentaires peuvent apparaître entre beaucoup plus d'éléments que dans la syntaxe actuelle. Aussi, les retours à la ligne qui sont composés uniquement d'espaces sont légaux.

```

-----
From : John Doe <jdoe@machine(commentaire). example>
To : Mary Smith

      <mary@example.net>
Subject : Saying Hello
Date : Fri, 21 Nov 1997 09(commentaire): 55 : 06 -0600
Message-ID : <1234 @ local(blah) .machine .example>

```

C'est un message pour simplement dire bonjour.
Alors, "Bonjour".

Noter particulièrement la second ligne du champ "To:". Elle commence par deux caractères espace. (Noter que " _" représente des espaces.) Donc, elle est considérée comme faisant partie du repli à la ligne, comme décrit au paragraphe 4.2. Les commentaires et espaces dans les adresses, dates, et identifiants de message font tous partie de la syntaxe obsolète.

Appendice B Différences avec les spécifications antérieures

Cet appendice contient une liste des changements qui ont été apportés au format de message Internet par rapport aux spécifications antérieures, en particulier les [RFC0822], [RFC1123] et [RFC2822]. Les éléments marqués d'un astérisque (*) ci-dessous sont des éléments qui apparaissent dans la section 4 de ce document et ne doivent donc plus être générés.

Les changements qui suivent aux [RFC0822] et [RFC1123] par la [RFC2822] sont ceux qui subsistent dans ce document :

1. Le point est permis dans la forme de phrase obsolète.
2. L'ABNF a été sorti du document. Il est maintenant dans la [RFC5234].
3. Quatre chiffres ou plus sont permis pour l'année.
4. L'ordre (et son absence) des champs d'en-tête est rendu explicite.
5. Le champ d'en-tête chiffré est supprimé.
6. Permission spécifique de la zone horaire "-0000" avec sa signification.
7. L'espace de pliage n'est pas permis entre tous les jetons.
8. Les exigences pour les destinations ont été supprimées.
9. L'émission et le renvoi sont redéfinis.
10. Les champs d'en-tête d'extension ne sont plus invoqués de façon spécifique.
11. Le caractère ASCII 0 (nul) est retiré.*
12. Les lignes de continuation de pliage ne peuvent plus contenir que des espaces.*
13. L'insertion libre de commentaires n'est plus permise dans la date.*
14. Les zones horaires non numériques ne sont plus permises.*
15. L'année à deux chiffres n'est plus permise.*
16. Les années à trois chiffres sont interprétées, mais il n'est plus permis d'en générer.*
17. Les chemins ne sont plus permis dans les adresses.*
18. Les CFWS au sein des parties locale et des domaines ne sont plus permises.*
19. Les membres vides de listes d'adresses ne sont plus permises.*
20. Les espaces de pliage entre le nom de champ et les deux points ne sont plus permises.*
21. Les commentaires entre le nom de champ et les deux points ne sont plus permis.
22. Resserrement de la syntaxe de in-reply-to et de references.*
23. Les CFWS au sein de msg-id ne sont plus admises.*
24. La sémantique des champs de renvoi a été restreinte à seulement informative.
25. Resent-Reply-To n'est plus admis.*
26. Les occurrences multiples de champs (excepté resent et received) ne sont plus admises.*
27. Les CR et LF libres ne sont plus admis.*
28. Les limites de longueur de ligne sont spécifiées.
29. Bcc est spécifié plus clairement.

Les changements suivants ont été apportés à la [RFC2822].

1. Les erreurs typographiques/grammaticales sont corrigées et les éclaircissements apportés.
2. Remplacement de "standard" par "document" ou "spécification" partout.

3. La distinction est faite entre "champ d'en-tête" et "section d'en-tête".
4. Retrait de NO-WS-CTL de ctext, qtext, dtext, et de non-structuré.*
5. La discussion de specials passe au paragraphe "Atome". Du texte est déplacé au paragraphe "Syntaxe globale du message".
6. Syntaxe simplifiée de CFWS.
7. Correction de la syntaxe de non-structuré.
8. Changement de la syntaxe de la date et heure pour traiter l'espace dans la syntaxe de date obsolète.
9. La paire entre guillemets est retirée des littéraux de domaine et des identifiants de message.*
10. IL est précisé que d'autres spécifications limitent la syntaxe de domaine.
11. Simplification de la syntaxe de "Bcc:" et "Resent-Bcc:".
12. IL est permise à des champs facultatifs d'apparaître dans les informations de traçage.
13. Retrait de no-fold-quote de msg-id. Précision des limitations syntaxiques.
14. La syntaxe de "Received:" est généralisée pour corriger des erreurs et sortir la définition de ce document.
15. Simplification de obs-qp. Correction et simplification de obs-utext (qui n'apparaît plus maintenant que dans la syntaxe obsolète). Retrait de obs-text et de obs-char, ajout de obs-body.
16. Correction de la syntaxe de date obsolète pour permettre plus (ou moins) de commentaires et d'espaces.
17. Correction de toute la syntaxe de liste obsolète (obs-domain-list, obs-mbox-list, obs-addr-list, obs-phrase-list, et de la obs-group-list nouvellement ajoutée).
18. Correction de la syntaxe de obs-reply-to.
19. Correction de obs-bcc et de obs-resent-bcc pour permettre des listes vides.
20. Retrait de obs-path.

Appendice C. Remerciements

De nombreuses personnes ont contribué à ce document. Parmi elles des gens qui ont participé au groupe de travail Révision détaillée et mise à jour des normes de messagerie (DRUMS) de l'Internet Engineering Task Force (IETF), le président de DRUMS, les directeurs de zones de l'IETF, et des gens qui ont simplement envoyé leurs commentaires via email. L'éditeur leur est largement redevable et il les remercie très sincèrement. La liste ci-dessous comporte tous ceux qui ont envoyé des messages concernant à la fois ce document et la [RFC2822]. J'espère que tous ceux qui ont contribué sont bien là :

Matti Aarnio	Tanaka Akira	Russ Allbery	Eric Allman	Harald Alvestrand	Ran Atkinson
Jos Backus	Bruce Balden	Dave Barr	Alan Barrett	John Beck	J Robert von Behren
Jos den Bekker	D J Bernstein	James Berriman	Oliver Block	Norbert Bollow	Raj Bose
Antony Bowesman	Scott Bradner	Randy Bush	Tom Byrer	Bruce Campbell	Larry Campbell
W J Carpenter	Michael Chapman	Richard Clayton	Maurizio Codogno	Jim Conklin	R Kelley Cook
Nathan Coulter	Steve Coxa	Mark Crispin	Dave Crocker	Matt Curtin	Michael D'Errico
Cyrus Daboo	Michael D Dean	Jutta Degener	Mark Delany	Steve Dorner	Harold A Driscoll
Michael Elkins	Frank Ellerman	Robert Elz	Johnny Eriksson	Erik E Fair	Roger Fajman
Patrik Falstrom	Claus Andre Faerber	Barry Finkel	Erik Forsberg	Chuck Foster	Paul Fox
Klaus M Frank	Ned Freed	Jochen Friedrich	Randall C Gellens	Sukvinder Singh Gill	Tim Goodwin
Philip Guenther	Arnt Gulbrandsen	Eric A Hall	Tony Hansen	John Hawkinson	Philip Hazel
Kai Henningsen	Robert Herriot	Paul Hethmon	Jim Hill	Alfred Hoenes	Paul E Hoffman
Steve Hole	Kari Hurtta	Marco S Hyman	Ofer Inbar	Olle Jarnefors	Kevin Johnson
Sudish Joseph	Maynard Kang	Prabhat Keni	John C Klensin	Graham Klyne	Brad Knowles
Shuhei Kobayashi	Peter Koch	Dan Kohn	Christian Kutzt	Anand Kumria	Steen Larsen
Eliot Lear	Barry Leiba	Jay Levitt	Bruce Lilly	Lars-Johan Liman	Charles Lindsey
Pete Loshin	Simon Lyall	Bill Manning	John Martin	Mark Martinec	Larry Masinter
Denis McKeon	William P McQuillan	Alexey Melnikov	Perry E Metzger	Steven Mille	S Moonesamy
Keith Moore	John Gardiner Myers	Chris Newman	John W Noerenberg	Eric Norman	Mike O'Dell
Larry Osterman	Paul Overell	Jacob Palme	Michael A Patton	Uzi Paz	Michael A Quinlan
Robert Rapplean	Eric S Raymond	Sam Roberts	Hugh Sasse	Bart Schaefer	Tom Scola
Wolfgang Segmuller	Nick Shelness	John Stanley	Einar Stefferud	Jeff Stephenson	Bernard Stern
Peter Sylvester	Mark Symons	Eric Thomas	Lee Thompson	Karel De Vriendt	Matthew Wall
Rolf Weber	Brent B Welch	Dan Wing	Jack De Winter	Gregory J Woodhouse	Greg A Woods
Kazu Yamamoto	Alain Zahm	Jamie Zawinski	Timothy S Zurcher		

7. Références

(Les liens hypertexte en tête et en queue des références pointent sur l'original anglais ; ceux du corps du titre sur l'éventuelle traduction française.)

7.1 Références normatives

- [ANSI.X3-4.1986] American National Standards Institute, "Coded Character Set - 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.
- [[RFC1034](#)] P. Mockapetris, "[Noms de domaines](#) - Concepts et facilités", STD 13, novembre 1987.
- [[RFC1035](#)] P. Mockapetris, "[Noms de domaines](#) - Mise en œuvre et spécification", STD 13, novembre 1987.
- [[RFC1123](#)] R. Braden, éditeur, "Exigences pour les [hôtes Internet](#) - Application et prise en charge", STD 3, octobre 1989.
- [[RFC2119](#)] S. Bradner, "[Mots clés](#) à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.
- [[RFC5234](#)] D. Crocker, éd., P. Overell, "BNF augmenté pour les spécifications de syntaxe : [ABNF](#)", janvier 2008. ([STD0068](#))

7.2 Références pour information

- [[RFC0822](#)] D. Crocker, "Norme pour le [format des messages](#) de texte de l'ARPA-Internet", STD 11, août 1982.
- [[RFC1305](#)] D. Mills, "[Protocole de l'heure](#) du réseau, version 3, spécification, mise en œuvre et analyse", STD 12, mars 1992.
- [ISO.2022.1994] International Organization for Standardization, "Information technology - Character code structure and extension techniques", ISO Standard 2022, 1994.
- [[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.*)
- [[RFC2047](#)] K. Moore, "MIME (Extensions de [messagerie Internet](#) multi-objets) Partie trois : extensions d'en-tête de message pour texte non ASCII", novembre 1996. (*MàJ par [RFC2184](#), [RFC2231](#)*) (*D.S.*)
- [[RFC2049](#)] N. Freed, N. Borenstein, "Extensions multi-objets de la messagerie Internet ([MIME](#)) Partie cinq : critères de conformité et exemples", novembre 1996. (*Remplace [RFC1521](#), [RFC1522](#), [RFC1590](#)*) (*D.S.*)
- [[RFC2822](#)] P. Resnick, "Format de [message](#) Internet", avril 2001. (*Remplace la [RFC0822](#), [STD 11](#), Remplacée par [RFC5322](#)*)
- [[RFC3864](#)] G. Klyne, M. Nottingham, J. Mogul, "Procédures d'enregistrement pour les champs d'en-tête de message", septembre 2004. ([BCP0090](#))
- [[RFC4021](#)] G. Klyne, J. Palme, "Enregistrement des champs d'[en-tête Mail](#) et MIME", mars 2005. (*MàJ par [RFC5322](#)*) (*P.S.*)
- [[RFC4288](#)] N. Freed et J. Klensin, "Spécifications du [type de support](#) et procédures d'enregistrement", BCP 13, décembre 2005.
- [[RFC4289](#)] N. Freed, J. Klensin, "Extensions multi-objet de messagerie Internet ([MIME](#)) Partie quatre : Procédures d'enregistrement", décembre 2005. (Remplace [RFC2048](#)) ([BCP0013](#))
- [[RFC5321](#)] J. Klensin, "Protocole [simple de transfert](#) de messagerie", octobre 2008. (Remplace [RFC2821](#)) (MàJ [RFC1123](#)) (*D.S.*)

Adresse de l'auteur

Peter W. Resnick (éditeur)
Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
USA
téléphone : +1 858 651 4478
mél : presnick@qualcomm.com
URI: <http://www.qualcomm.com/~presnick/>

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2008).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations y contenues sont fournies 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, le IETF TRUST 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.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourrait être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur répertoire en ligne des IPR de l'IETF à <http://www.ietf.org/ipr>.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.