

- *Format des messages internet* -

Statut de ce mémo

Ce document spécifie un protocole Internet normalisé pour la communauté de l'Internet. Merci de se référer à l'édition actuelle du "Protocole officiel des normes de l'Internet" (STD1) pour l'état et le statut de normalisation de ce protocole. La distribution de ce mémo n'est pas limitée.

Mention légale

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

Attention

Ce standard spécifie une syntaxe pour les messages textuels qui sont envoyés entre les utilisateurs d'ordinateur, dans le cadre des messages par "courrier électronique". Ce standard remplace celui spécifié par la RFC 822 "Standard pour les formats des messages textuels d'ARPA sur Internet" l'actualisant pour mieux correspondre aux pratiques actuelles et incorporer des changements qui furent fait dans d'autres RFC.

SOMMAIRE

1	Introduction	2
1.1	Domaine d'application	2
1.2	Conventions de notation	3
1.2.1	Exigences pour la notation	3
1.2.2	Notation syntaxique	3
1.3	Structure de ce document	3
2	Analyse lexicale des messages	4
2.1	Description générale	4
2.1.1	Limite de taille des lignes	5
2.2	Champs d'en tête	5
2.2.1	Corps de champ d'en-tête non structurés	5
2.2.2	Corps de champ d'en tête structurés	5
2.2.3	Champs d'en tête longs	6
2.3	Le corps	6
3	Syntaxe	6
3.1	Introduction	6
3.2	Marques lexicales	7
3.2.1	Marques primitives	7
3.2.2	Caractères marqués	8
3.2.3	Espace blancs et commentaires	8
3.2.4	Atome	9
3.2.5	Chaîne marquée	9
3.2.6	Marquage divers	10
3.3	Spécification pour la date et l'heure	10

3.4	Spécification de l'adresse	11
3.4.1	Spécification d'addr-spec	12
3.5	Syntaxe globale des messages	12
3.6	Définition des champs	12
3.6.1	Champ de date de création	14
3.6.2	Champs d'origine	14
3.6.3	Champs d'adresse de destination	15
3.6.4	Champs d'identification	16
3.6.5	Champs d'information	17
3.6.6	Champs de renvoi	17
3.6.7	Champs de traçage	18
3.6.8	Champs facultatifs	19
4	Syntaxe obsolète	19
4.1	Diverses marques obsolètes	20
4.2	FWS Obsolète	20
4.3	Date et heure obsolète	20
4.4	Adressage obsolète	21
4.5	Champs d'en tête obsolète	22
4.5.1	Champ de date d'origine obsolète	22
4.5.2	Champ d'origine obsolète	23
4.5.3	Champs adresse de destination obsolètes	23
4.5.4	Champs d'identification obsolètes	23
4.5.5	Champs d'information obsolètes	23
4.5.6	Champs de renvoi obsolètes	23
4.5.7	Champs de traçage obsolètes	24
4.5.8	Champs facultatifs obsolètes	24
5	Considérations sur la sécurité	24
6	Bibliographie	24
7	Adresse de l'éditeur	25
8	Remerciements	25
Appendice A	Exemples de messages	26
A.1	Exemples d'adressage	27
A.1.1	Message d'une personne à une autre	27
A.1.2	Différents types de boîte à lettre	27
A.1.3	Adresses de groupe	28
A.2	Messages de réponses	28
A.3	Messages de renvoi	29
A.4	Messages avec des champs de traçage	29
A.5	Espaces blancs, commentaires et autres raretés	30
A.6	Formes obsolètes	30
A.6.1	Adressage obsolète	31
A.6.2	Dates obsolètes	31
A.6.3	Commentaire et espaces blancs obsolètes	31
Appendice B	Différences avec les précédents standards	32
Appendice C	Déclarations de propriété intellectuelle	33

1 Introduction

1.1 Domaine d'application

Ce standard spécifie une syntaxe pour les messages textuels qui sont envoyés entre les utilisateurs d'ordinateur, dans le cadre des messages par "courrier électronique". Ce standard remplace celui spécifié par la RFC 822 "Standard pour les formats des messages textuels d'ARPA sur Internet" l'actualisant pour mieux correspondre aux pratiques actuelles et incorporer des changements qui furent fait dans d'autres RFC [STD3].

Ce standard spécifie une syntaxe seulement pour les messages textuels. En particulier, il ne fournit aucune disposition pour le transfert d'images, de son et autres données structurées dans un courrier électronique. Il existe plusieurs extensions publiées, telles que la série des documents sur MIME, [RFC2045, RFC2046, RFC2049], qui décrivent les mécanismes pour la transmission de telles données par messagerie électronique, soit par une extension de la syntaxe fournie ici, soit en structurant de tels messages pour les conformer à cette syntaxe. Ces mécanismes sortent du domaine d'application de la présente norme.

Dans le contexte du courrier électronique, les messages sont vus comme ayant une enveloppe et un contenu. L'enveloppe contient toute l'information nécessaire pour accomplir la transmission et la livraison. (Voir la [RFC2821] pour un exposé sur l'enveloppe). Le contenu comprend l'objet qui doit être livré au destinataire. Ce standard s'applique seulement au format et à quelques aspects sémantiques des contenus de messages. Il ne contient aucune spécification sur les informations de l'enveloppe.

Quoi qu'il en soit, quelques systèmes de messages doivent utiliser l'information du contenu pour créer l'enveloppe. Ce standard est destiné à faciliter l'acquisition de telles informations par les programmes.

Cette spécification est entendue comme une définition du format du contenu du message qui doit être passé entre les systèmes. Bien que quelques systèmes de messagerie mémorisent ces messages localement dans ce format (ce qui évite tout besoin de traduction entre formats) et que d'autres utilisent des formats qui diffèrent de celui spécifié dans ce standard, la mémorisation locale est en dehors du domaine d'application de la présente norme.

Note : Ce standard n'est pas destiné à dicter les formats internes utilisés par les sites, les composants spécifiques de système de messages qu'ils sont censés prendre en charge, ou n'importe laquelle des caractéristiques d'une interface d'utilisateur qui crée, ou lit des messages. De plus ce standard ne spécifie pas le nombre de bits utilisés ou comment ces bits sont transférés sur le réseau ou mémorisés sur le disque.

1.2 Conventions de notation

1.2.1 Exigences pour la notation

Ce document utilise occasionnellement des termes qui apparaissent en lettres capitales. Quand les termes "DOIT", "DEVRAIT", "RECOMMANDE", "NE DOIT PAS", "NE DEVRAIT PAS", et "PEUT" apparaissent en capitales, ils sont utilisés pour indiquer ce qui est requis pour cette spécification. Une discussion sur la signification de ces termes se trouve dans la [RFC2119].

1.2.2 Notation syntaxique

Ce standard utilise la notation de forme Backus-Naur augmentée (ABNF, *Augmented Backus-Naur Form*) spécifiée dans la [RFC2234] pour les définitions formelles de la syntaxe des messages. Les caractères seront spécifiés soit en valeur décimale (par exemple, la valeur %d65 pour A et %d97 pour a) ou par une valeur littérale non sensible à la casse entre guillemets (par exemple, "A" pour soit la majuscule ou la minuscule). Voir la [RFC2234] pour une description complète de cette notation.

1.3 Structure de ce document

Ce document est divisé en plusieurs sections.

La section 1 est une courte introduction au document.

La section 2 établit la description générale d'un message et de ces parties constitutives. C'est une vue générale pour aider le lecteur à comprendre quelques principes utilisés plus tard dans ce document. Les exemples de cette section NE DOIVENT PAS être pris pour des spécifications 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 la relation entre ces parties et leur signification dans le contexte d'un message (la sémantique). C'est-à-dire qu'elle décrit les règles réelles de la structure de chaque partie d'un message (la syntaxe) ainsi qu'une description des parties et instructions sur la façon dont elles doivent être interprétées (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 en section 3 représente les messages tels qu'il DOIVENT être créés. Il y a aussi des notes au paragraphe 3 qui indiquent si une des options spécifiées dans la syntaxe POURRAIT être utilisé par dessus une autre.

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

La section 4 de ce document spécifie une syntaxe "obsolète". Il y a des références au paragraphe 3 à ces éléments syntaxiques obsolètes. Les règles de la syntaxe obsolète sont des éléments qui sont apparus dans des révisions précédentes sur ce standard ou qui ont été utilisés largement dans les messages Internet. Comme tels, ces éléments DOIVENT être interprété par les analyseurs de messages de manière à se conformer à la présente norme. Cependant, comme les éléments de cette syntaxe ont été déterminés comme non-interopérables ou causant des problèmes significatif au receveur du message, ils NE DOIVENT PAS être générés par des créateurs de messages conformes.

La section 5 détaille les considérations sur la sécurité à prendre en compte lors de la mise en œuvre de la présente norme.

La section 6 est une bibliographie des références de ce document.

La section 7 contient l'adresse de l'éditeur.

La section 8 contient les remerciements.

L'appendice A, donne une liste d'exemples des différentes sortes de messages. Ces exemples ne sont pas exhaustifs de tous les types de messages que l'on rencontre sur l'internet, mais donnent une large vue sur certaines formes syntaxiques.

L'appendice B, liste les différences entre ce standard et les précédents.

L'appendice C comporte les déclarations de copyright et de propriété intellectuelle.

2 Analyse lexicale des messages

2.1 Description générale

Au niveau le plus basique, un message est une série de caractères. Un message qui se conforme à ce standard comprend des caractères avec des valeur dans la gamme de 1 à 127 et sont interprétés comme caractères US-ASCII [ASCII]. Pour faire court, ce document se réfère parfois à cette gamme de caractères simplement comme "caractères US-ASCII".

Note : ce standard spécifie que les messages sont composés de caractères US-ASCII de 1 à 127. Il y a d'autres documents surtout la série des documents sur MIME [RFC2045, RFC2046, RFC2047, RFC2048, RFC2049] qui étendent ce standard pour permettre des valeurs en dehors de cette gamme. Les discussions sur ces mécanismes sortent du domaine d'application de la présente norme.

Les messages sont divisés en lignes de caractères. Une ligne est une série de caractères qui est délimitée par deux caractères : retour chariot et saut à la ligne ; c'est-à-dire le retour chariot (CR) (ASCII 13) suivit immédiatement par un saut à la ligne (LF) (ASCII 10). (La paire CR/LF est généralement écrite dans ce document comme "CRLF".)

Un message est composé de champs d'en tête (collectivement appelés l'en tête du message) suivis, facultativement, par un corps. L'en-tête est une séquence de lignes de caractères avec une syntaxe spéciale telle que définie dans ce standard. Le corps est simplement une séquence de caractères qui suit l'en-tête et en est séparé par une ligne vide (c'est-à-dire, une ligne avec rien devant le CRLF).

2.1.1 Limite de taille des lignes

Il y a deux limites que ce standard émet sur le nombre de caractères dans une ligne. Chaque ligne de caractères NE DOIT PAS dépasser 998 caractères, et ne DEVRAIT pas avoir plus de 78 caractères, CRLF exclu.

La limite des 998 caractères est le fait de limitations dans beaucoup de mises en œuvre qui envoient, reçoivent, ou mémorisent des messages au format de message Internet qui ne peuvent tout simplement pas traiter plus de 998 caractères sur une ligne. Recevoir les mises en œuvre serait parfait pour traiter un grand nombre arbitraire de caractères dans une ligne pour des raisons de robustesse. Cependant, il y a tant de mises en œuvre qui (conformément aux exigences sur les transports de la RFC 2821) n'acceptent pas les messages contenant plus de 1000 caractères CR et LF compris, par ligne, qu'il est important pour les mises en œuvre de ne pas créer de tels messages.

La recommandation la plus conservatrice des 78 caractères est destinée à tenir compte des nombreuses mises en œuvre d'interfaces utilisateurs qui en affichant ces messages peuvent les tronquer, ou envelopper d'une façon désastreuse, l'affichage de plus de 78 caractères par ligne. En dépit du fait que de telles mises en œuvre ne sont pas conforme à l'objectif de cette spécification (et à celui de la [RFC2821] si les informations sont en fait perdues). A nouveau, même si cette limitation est mise dans les messages, elle incombe à la mise en œuvre qui affiche les messages.

2.2 Champs d'en tête

Les champs d'en tête sont des lignes composées d'un champ de nom, suivi par deux point (":"), suivi par un corps de champ, et terminé par un CRLF. Un nom de champ DOIT être composé de caractères US-ASCII imprimables (c'est-à-dire, les caractères qui ont des valeurs comprises entre 33 et 126 inclus), sauf les deux points. Un corps de champ peut être composé de tout caractère US-ASCII, sauf CR et LF. Cependant, un champ d'en tête peut contenir un CRLF quand il est utilisé dans l'en tête "folding" ou "unfolding" comme décrit au paragraphe 2.2.3. Tous les corps de champ DOIVENT se conformer à la syntaxe décrite dans les sections 3 et 4 de ce standard.

2.2.1 Corps de champ d'en-tête non structurés

Certains corps de champ de ce standard sont définis simplement comme "unstructured" (qui est spécifié ci-dessous comme tous caractères US-ASCII, sauf CR et LF) sans aucune autre restriction. Sémantiquement parlant, les corps de champ non structurés doivent être traités comme une seule ligne de caractères sans aucun autre traitement (sauf pour l'en tête "folding" et "unfolding" comme décrit au paragraphe 2.2.3).

2.2.2 Corps de champ d'en tête structurés

Certains champs de ce standard ont des structures syntaxiques spécifiques plus restrictives que le champ non structuré décrit ci-dessus. On s'y réfère en tant que corps de champ "structuré". Les corps de champ structurés sont les séquences de marques lexicales spécifiques, comme décrit dans les sections 3 et 4 de ce standard. Nombre de ces marques peuvent (si l'on se réfère à leur syntaxe) être introduites ou se conclure par des commentaires (comme décrit au paragraphe 3.2.3) aussi bien que des caractères espaces (SP, valeur ASCII 32) et tabulations horizontales (HTAB, valeur ASCII 9) (connus conjointement comme caractère espace blanc, WSP), et ces caractères WSP sont sujet à l'en tête "folding" et "unfolding" comme décrit au paragraphe 2.2.3. L'analyse sémantique des corps de champ structurés est donnée plus loin avec leur syntaxe.

2.2.3 Champs d'en tête longs

Chaque champ d'en tête est logiquement une seule ligne de caractères qui comprend le nom du champ, les deux points et le corps du champ. Pour plus de facilité, et pour se plier à la limite des 998/78 caractères par ligne, la portion de corps de champ d'un champ d'en tête peut être divisée en une représentation sur plusieurs lignes ; ce qui est appelé "folding" (*aller à la ligne*). La règle générale est que partout où la présente norme permet les espaces blancs "folding" (pas juste un WSP), un CRLF peut être inséré avant n'importe quel WSP. Par exemple, un 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 sorte que le saut à la ligne puisse prendre place entre beaucoup des marques lexicales (et même à l'intérieur des marques lexicales), le retour à la ligne DEVRAIT être limité au placement des CRLF au haut niveau des ruptures syntaxiques. Si un corps de champ est défini avec une virgule comme séparateur, il est recommandé de placer le retour à la ligne après la virgule qui sépare les items structurés plutôt qu'en d'autres endroits où le champ pourrait être coupé, même s'il est permis de le faire ailleurs.

Le processus de passage de cette représentation multi-ligne d'un champ d'en tête à sa représentation sur une seule ligne est appelée "unfolding" (*déploiement*). Le déploiement est accompli en enlevant simplement tout CRLF qui est immédiatement suivi d'un WSP. Chaque champ d'en tête devrait être traité dans sa forme déployée pour les évaluations syntaxiques et sémantiques.

2.3 Le corps

Le corps d'un message est simplement des lignes de caractères US-ASCII. Les deux seules limitations relatives au corps du message sont les suivantes :

- CR et LF DOIVENT toujours apparaître ensemble en tant que CRLF ; ils NE DOIVENT PAS apparaître indépendamment dans le corps.
- Les lignes de caractères dans le corps DOIVENT être limitées à 998 caractères, et DEVRAIENT être limitées à 78 caractères, CRLF exclu.

Note : Comme nous l'avons fait remarqué plus haut, il y a d'autres documents de normalisation, spécialement MIME [RFC2045, RFC2046, RFC2048, RFC2049] qui étendent la présente norme pour permettre différents types de corps de message. A nouveau ces mécanismes sont au delà du domaine d'application de ce document.

3 Syntaxe

3.1 Introduction

La syntaxe telle que donnée dans cette section définit la syntaxe légale des messages Internet. Les messages qui se conforment à ce standard DOIVENT se conformer à la syntaxe de cette section. Si il y a des options dans cette section où une option DEVRAIT être générée, cela est indiqué soit dans le texte soit dans un commentaire accolé à la syntaxe.

Pour les expressions définies, une courte description de la syntaxe et de son utilisation est donnée, suivie par la syntaxe en ABNF, suivie par une analyse sémantique. Les marques de primitives qui sont utilisées mais non spécifiées autrement proviennent de la [RFC2234].

Dans certaines des définitions, il y aura des "non-terminaux" dont les noms commencent par "obs-". Ces éléments "obs-" se réfèrent aux marques définies dans la syntaxe obsolète de la section 4. Dans tout les cas, ces productions doivent être ignorées pour la génération de messages légaux sur Internet et NE DOIVENT PAS être utilisées en tant que partie d'un tel message. Cependant, quand on interprète les messages, ces marques DOIVENT être honorées au titre de la syntaxe légale. Dans ce sens, la section 3 définit une grammaire pour la génération des messages, avec les éléments "obs-" qui sont à ignorer, tandis que la section 4 ajoute de la grammaire pour l'interprétation des messages.

3.2 Marques lexicales

Les règles suivantes sont utilisées pour définir un analyseur lexical sous-jacent, qui fournit les marques à un analyseur de niveau supérieur. Cette section définit les marques utilisées dans les corps de champ d'en tête structurés.

Note : Les lecteurs de la présente norme porteront une attention particulière à la façon dont les marques lexicales sont utilisées à la fois dans la syntaxe de niveau inférieur et supérieur dans la suite de ce document. En particulier, les marques d'espace blanc et de commentaire définies au paragraphe 3.2.3 sont utilisées dans les marques de niveau inférieur définies ici, et les marques de niveau inférieur sont aussi utilisées au titre des marques de niveau supérieur définies plus loin. Donc, les espaces blancs et les commentaires peuvent être autorisés dans les marques de niveau supérieur même si ils n'apparaissent pas explicitement dans une définition.

3.2.1 Marques primitives

Ce qui suit correspond aux marques primitives auxquelles on se réfère ailleurs dans le présent document, mais qui ne sont pas autrement définies dans la [RFC2234]. Quelques unes d'entre elles n'apparaîtront plus ailleurs dans la syntaxe, mais elles sont utiles en tant que référence pour d'autres parties de ce document.

Note :Les "spéciales" ci dessous sont données en tant qu'exemple. Bien que les marques spéciales n'apparaissent pas ailleurs dans la présente norme, elles sont utiles pour les développeurs qui utilisent des outils qui font l'analyse lexicographique des messages. Chacun des caractères des spéciales peut être utilisé pour indiquer un marquage dans l'analyse lexicale.

NO-WS-CTL	=	%d1-8 / ; caractères de contrôle US-ASCII %d11 / ; qui n'incluent pas les caractères %d12 / ; retour chariot, le saut à la ligne %d14-31 / ; et les espaces blancs %d127
text	=	%d1-9 / ; caractères excluant CR et LF %d11 / %d12 / %d14-127 / obs-text
specials	=	"(" / ")" / ; caractères spéciaux utilisés dans d'autres parties de la syntaxe "<" / ">" / ; "[" / "]" / ":" / ";" / "@"/ "\" / "," / "." / DQUOTE

Aucune sémantique spéciale n'est attachée à ces marques. Ce sont simplement des caractères seuls.

3.2.2 Caractères marqués

Certains caractères sont réservés pour une interprétation particulière, comme la délimitation des marques lexicales. Pour permettre l'utilisation de ces caractères comme des données non interprétées un mécanisme de marquage est fourni.

quoted-pair = ("\" texte) / obs-qp

Quand la quoted-pair (*paire marquée*) apparaît, elle doit être interprétée comme caractère de texte seul. C'est à dire que le caractère "\" qui apparaît en tant que membre d'une quoted-pair est sémantiquement invisible.

Note : Le caractère "\" peut apparaître dans un message où il ne fait pas partie d'une quoted-pair. Un "\" qui n'apparaît pas dans une quoted-pair n'est pas sémantiquement invisible. La seule place dans la présente norme où les quoted-pair apparaissent sont ccontent, qcontent, dcontent, no-fold-quote, et no-fold-literal.

3.2.3 Espace blancs et commentaires

Les caractères espace blanc, y compris les espaces blancs utilisé dans un retour à la ligne (comme décrit au paragraphe 2.2.3), peuvent apparaître entre de nombreux éléments dans les corps de champ d'en tête. Aussi, les chaînes de caractères qui sont traitées comme commentaires peuvent être incluses dans les corps de champ structuré comme des caractères entre parenthèses. Ce qui suit définit les constructions d'espace blanc de retour à la ligne (FWS) et de commentaire.

Les chaînes de caractères entre parenthèses sont considérées comme commentaires tant qu'elles n'apparaissent pas à l'intérieur d'une "chaîne marquée", comme défini au paragraphe 3.2.5.

Il y a plusieurs endroits de la présente norme où les commentaires et le FWS peuvent être insérés librement. Pour arranger cette syntaxe, une marque supplémentaire pour "CFWS" est définie pour les endroits où le commentaire et/ou le FWS peut intervenir. Cependant, là où le CFWS apparaît dans ce standard, il NE DOIT PAS être inséré d'une façon telle qu'une ligne d'un champ d'en-tête sur plusieurs lignes soit composé uniquement de WSP et de rien d'autre.

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

ctex = NO-WS-CTL / %d33-39 / %d42-91 / %d93-126 ; contrôles de non espace blanc ; Le reste des caractères US-ASCII ; n'incluant pas "(", ")", ou "\" ;

ccontent = ctext / quoted-pair / comment

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

CFWS = *([FWS] comment) (([FWS] comment) / FWS)

Tout au long de la présente norme, où le FWS (la marque d'espace blanc de déploiement) apparaît, il indique un emplacement où le déploiement d'en-tête, comme exposé paragraphe 2.2.3, peut prendre place. N'importe où qu'il apparaisse dans le message (c'est à dire, un champ d'en tête contenant un CRLF suivi d'un WSP), le déploiement d'en-tête (sans le CRLF) est exécuté avant toute analyse lexicale sur le champ d'en-tête conformément à la présente norme. C'est-à-dire que si un CRLF apparaît dans un FWS, il est sémantiquement "invisible".

Un commentaire est normalement utilisé dans un corps de champ structuré pour fournir une information textuelle lisible par l'homme. Comme un commentaire est autorisé à contenir un FWS, le passage à la ligne est autorisé dans un commentaire. Noter également que comme les guillemets sont autorisés dans les commentaires, les caractères parenthèses et les barres obliques inversées peuvent apparaître dans un commentaire tant qu'ils sont entre guillemets. Sémantiquement, les parenthèses d'enveloppe ne font pas partie du commentaire ; un commentaire est ce qui est contenu entre les deux parenthèses. Comme dit précédemment, le "\" dans des guillemets et le CRLF dans tout FWS qui apparaissent à l'intérieur d'un commentaire sont sémantiquement invisibles et donc ne font pas non plus partie du commentaire.

Les séries de FWS, les commentaires ou CFWS qui surviennent entre des marques lexicales dans un en-tête de champ structuré sont sémantiquement interprétés comme un seul caractère d'espace.

3.2.4 Atome

Plusieurs productions dans des corps de champ d'en-tête structuré sont simplement des chaînes de certains caractères de base. De telles productions sont appelées atomes.

Quelques corps de champ d'en-tête structurés autorisent le caractère point (".", ASCII 46) à l'intérieur du texte. Une marque supplémentaire "dot-atom" est définie dans ce but.

Atext = ALPHA / DIGIT / ; Tout caractère sauf contrôle,
 "!" / "#" / ; SP, et spéciaux.
 "\$" / "%" / ; Utilisé pour atomes
 "&" / "" /
 "*" / "+" /
 "-" / "/" /
 "=" / "?" /
 "^" / "_" /
 "`" / "{" /
 "|" / "}" /
 "~"

atom = [CFWS] 1*atext [CFWS]

dot-atom = [CFWS] dot-atom-text [CFWS]

dot-atom-text = 1*atext *("." 1*atext)

atom et dot-atom sont tous deux interprétés de manière unitaire, comprenant une chaîne de caractères qui la compose. Sémantiquement, le commentaire facultatif et le FWS entourant le reste des caractères qui ne font pas partie de l'atome ; l'atome est seulement une suite de caractères atext dans un atome, ou atext et "." dans un dot-atom.

3.2.5 Chaîne marquée

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

qtext = NO-WS-CTL / ; contrôle d'espace non blanc
 %d33 / ; le reste de l'US-ASCII excluant "/" ou le guillemet
 %d35-91 / ;
 %d93-126 ;

qcontent = qtext / quoted-pair

quoted-string = [CFWS]
 DQUOTE *([FWS] qcontent) [FWS] DQUOTE
 [CFWS]

Une chaîne marquée est traitée en tant qu'unité. Une chaîne marquée est sémantiquement identique à un atome. Comme il est permis à une chaîne marquée de contenir un FWS, le passage à la ligne est permis. Noter aussi que comme les guillemets sont autorisés dans les chaînes marquées, les caractères "" et "/" peuvent apparaître dans une chaîne marquée tant qu'ils apparaissent comme paire marquée.

Sémantiquement, ni le CFWS facultatif en dehors des guillemets, ni les guillemets eux-mêmes ne font partie de la chaîne marquée ; la chaîne marquée est ce qui est contenu entre les guillemets. Comme dit précédemment, le "\" dans toute paire de guillemets et le CRLF dans tout FWS/CFWS apparaissant à l'intérieur d'une chaîne marquée sont sémantiquement "invisibles" et donc ne font pas non plus partie de la chaîne marquée.

3.2.6 Marquage divers

Trois marquages additionnels ont été définis, mot et phrase pour la combinaison avec les atomes et/ou les chaînes marquées, et les non-structuré à utiliser dans des champs d'en tête non structurés et dans quelques endroits à l'intérieur des champs d'en-tête structurés.

word = atom / quoted-string

phrase = 1*word / obs-phrase

utext = NO-WS-CTL / ; commandes d'espace non blanc
 %d33-126 / ; le reste de l'US-ASCII
 obs-utext

unstructured = *([FWS] utext) [FWS]

3.3 Spécification pour la date et l'heure

La date et l'heure interviennent dans plusieurs champs d'en-tête. Ce paragraphe spécifie la syntaxe pour la date et l'heure. Bien que l'espace blanc déployé soit permis dans la spécification de la date et l'heure, il est RECOMMANDÉ qu'un seul espace soit utilisé partout où un FWS apparaît (qu'il soit exigé ou facultatif) ; les mises en œuvre plus anciennes pourraient ne pas interpréter d'autres occurrences d'espace blanc déployé correctement.

date-time = [jour de la semaine " ,"] date FWS heure [CFWS]

day-of-week = ([FWS] nom du jour) / obs-day-of-week

day-name = "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun"

date = jour mois an

year = 4*DIGIT / obs-year

month = (FWS nom du mois FWS) / obs-month

month-name = "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun" / "Jul" / "Aug" / "Sep" / "Oct" / "Nov" / "Dec"

day = ([FWS] 1*2DIGIT) / obs-day

time = heure FWS zone

time-of-day = heure ":" minute [":" seconde]

hour = 2DIGIT / obs-hour

minute = 2DIGIT / obs-minute

second = 2DIGIT / obs-second

zone = (("+" / "-") 4DIGIT) / obs-zone

Le jour est le jour du mois. L'année est n'importe quel numérique à partir de 1900.

"heure du jour" spécifie le nombre d'heures, de 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, Coordinated Universal Time, connu aussi sous le nom d'"heure moyenne de Greenwich", GMT) que représentent la date et l'heure. Le "+" ou le "-" indique si l'heure est en avance (à l'est de Greenwich) ou retarde (à l'ouest de Greenwich) sur 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 de différence avec le temps universel. (Et donc, +hhmm veut dire $+(hh*60+mm)$ minutes, et -hhmm veut dire $-(hh*60+mm)$ minutes). La forme "+0000" DEVRAIT être utilisée pour indiquer une zone sur le fuseau du temps universel. Bien que "-0000" indique aussi le temps universel, elle est utilisée pour indiquer que l'heure a été générée sur un système qui peut être dans un fuseau horaire autre que celui du temps universel et donc indiquer que le champ de date et heure ne contient pas d'information sur le fuseau horaire.

Une spécification de date-heure DOIT être sémantiquement valide. C'est à dire que le jour de la semaine (si inclus) DOIT être le jour impliqué par la date, le numéro du jour du mois DOIT être compris entre 1 et le nombre de jour permis par le mois spécifié (dans l'année spécifiée), l'heure du jour DOIT être comprise entre 00:00:00 et 23:59:60 (le nombre de secondes permettant une seconde sautée, voir le STD12) et la zone doit être comprise entre -9959 et +9959.

3.4 Spécification de l'adresse

Les adresses interviennent dans plusieurs champs d'en-tête de message pour indiquer les envoyeurs et receveurs des messages. Une adresse peut être soit une boîte à lettre individuelle soit un groupe de boîtes à lettre.

Address	=	mailbox / group
Mailbox	=	name-addr / addr-spec
name-addr	=	[display-name] angle-addr
angle-addr	=	[CFWS] "<" addr-spec ">" [CFWS] / obs-angle-addr
group	=	display-name ":" [mailbox-list / CFWS] ";" [CFWS]
display-name	=	phrase
mailbox-list	=	(mailbox *(", " mailbox)) / obs-mbox-list
address-list	=	(address *(", " address)) / obs-addr-list

Une boîte à lettre reçoit du courrier. C'est une entité conceptuelle qui ne doit pas nécessairement avoir de rapport avec le fait de le mémoriser. 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 à lettre 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 addr-spec comprise entre des crochets angulaires "<" et ">". Il y a aussi une forme simple de remplacement où l'adresse addr-spec apparaît seule, sans le nom du receveur ou les crochets angulaires. L'adresse addr-spec de l'Internet est définie au paragraphe 3.4.1.

Note : Certaines mises en œuvre utilisent la forme simple où le addr-spec apparaît sans crochets angulaires "<" et ">", mais inclut le nom du receveur entre parenthèses comme s'il s'agissait d'un commentaire suivant l'addr-spec. Comme la signification de l'information dans un commentaire est non spécifiée, les mises en œuvre DEVRAIENT utiliser la forme name-addr complète de la boîte à lettre, au lieu de la forme ordinaire ci dessus, pour spécifier l'affichage du nom associé à la boîte à lettre. Aussi, parce que quelques mises en œuvre ordinaires interprètent le commentaire, les commentaires NE DEVRAIENT PAS généralement être utilisés dans les champs d'adresse pour éviter de telles confusions.

Quand on veut traiter plusieurs boîtes à lettre comme un ensemble (c'est-à-dire une liste de diffusion), le groupe construct peut être utilisé. Il permet à l'envoyeur d'indiquer un groupe désigné de receveurs. Ceci se fait en donnant un nom au groupe suivi de deux points, suivi par une liste séparée par des virgules d'un nombre quelconque de boîtes à lettre (incluant 0 et 1), et terminé par un point virgule. Parce que la liste des boîtes à lettre peut être vide, utiliser un groupe construct est aussi un moyen simple pour communiquer aux receveurs

que le message a été envoyé à un ou plusieurs ensembles désignés de receveurs, sans en fait fournir l'adresse individuelle de chacun des receveurs.

3.4.1 Spécification d'addr-spec

L'addr-spec est un identifiant spécifique d'Internet qui contient une chaîne interprétée localement suivie par un caractère à ("@", valeur ASCII 64) suivi par un domaine Internet. La chaîne interprétée localement est soit une chaîne marquée soit un dot-atom. Si la chaîne peut être représentée comme dot-atom (c'est à dire qu'elle ne contient pas de caractères autres que des atext ou "." entourés par des caractères atext), alors le dot-atom DEVRAIT être utilisé et la chaîne marquée NE DEVRAIT PAS être utilisée. Les commentaires et le FWS NE DEVRONT PAS être utilisés autour du "@" dans l'addr-spec.

addr-spec	=	local-part "@" domain
local-par	=	dot-atom / quoted-string / obs-local-part
domain	=	dot-atom / domain-literal / obs-domain
domain-literal	=	[CFWS] "[" *(FWS) dcontent [FWS] "]" [CFWS]
dcontent	=	dtext / quoted-pair
dtext	=	NO-WS-CTL ; contrôles d'espace non blancs %d33-90 / ; Le reste des caractères US-ASCII %d94-126 ; non inclus "[", ";", "]", ou "\"

La portion domaine identifie le point auquel le message est envoyé. Dans la forme dot-atom, ceci est interprété comme un nom de domaine Internet (un nom d'hôte ou un nom d'échangeur de messagerie électronique) comme décrit dans les [STD3,STD13, STD14]. Dans la forme domain-literal, le domaine est interprété comme adresse Internet littérale d'un hôte particulier. Dans les deux cas, la manière dont l'adressage est utilisé et la manière dont les messages sont acheminés vers un hôte est couverte par la RFC 2821 sur le transport de la messagerie électronique. Ces mécanismes sont en dehors du domaine d'application du présent document.

La partie local-part est une chaîne qui dépend du domaine. Dans les adresses, il est simplement interprété sur l'hôte commm d'une boîte à lettre particulière.

3.5 Syntaxe globale des messages

Un message consiste en champs d'en-tête, facultativement suivis par un corps de message. Les lignes d'un message NE DOIVENT PAS dépasser 998 caractères, CRLF exclu, mais il est RECOMMANDÉ de ne pas dépasser la limite de 78, 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 sur le texte PUISSENT être utilisés, l'utilisation de caractères non US-ASCII (valeurs 1 à 8, 11, 12 et 14 à 31) est déconseillée car l'interprétation par le receveur pour l'affichage n'est pas garantie.

Message	=	(fields / obs-fields) [CRLF body]
body	=	*(998text CRLF) 998text

Le champ d'en tête porte la plupart des informations sémantiques qui sont définies au paragraphe 3.6. Le corps est simplement une série de lignes de texte qui ne sont pas interprétées par la présente norme.

3.6 Définition 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 que les champs d'en tête ne sont pas classés dans un ordre particulier. Ils peuvent apparaître dans n'importe quelle ordre, et 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.

```

Champ      =      *(trace
                    *(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 un en tête ainsi que des limitations spécifiques sur l'utilisation de ces champs. Un astérisque près d'une valeur dans la colonne minimum ou maximum indique qu'une restriction apparaît dans la colonne notes.

Champ	Nombre minimum	Nombre maximum	Notes
Trace	0	sans limite	Bloc ajouté – voir au paragraphe 3.6.7
resent-date	0*	sans limite*	Un par bloc, obligatoire si d'autres champs resent sont présents – voir au paragraphe 3.6.6
resent-from	0	sans limite*	Un par bloc – voir au paragraphe 3.6.6
resent-sender	0*	sans limite*	Un par bloc, DOIT survenir avec resent-from multi-adresses – voir au paragraphe 3.6.6
resent-to	0	sans limite*	Un par bloc – voir au paragraphe 3.6.6
resent-cc	0	sans limite*	Un par bloc – voir au paragraphe 3.6.6
resent-bcc	0	sans limite*	Un par bloc – voir au paragraphe 3.6.6
resent-msg-id	0	sans limite*	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 from multi-adresses – 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	sans limite	
keywords	0	sans limite	
optional-field	0	sans limite	

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

3.6.1 Champ de date de création

La champ date de création consiste en un nom de champ "Date" suivi par la spécification de la date et heure.

orig-date = "Date:" date-time 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 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" dans un programme d'application. Dans tous les cas, il n'est pas envisagé de lui faire porter l'heure à laquelle le message a été 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 a un réseau peu faire attendre la livraison d'un message. La date de création est destinée à contenir la date et l'heure à laquelle le message a été mis en 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*) (si applicable), et facultativement le champ reply-to (*répondre à*). Le champ from comporte le nom de champ "From: " et une liste séparée par des virgules de la ou des spécifications de boîte à lettre. Si le champ from spécifie plusieurs boîtes à lettre dans la liste, le champ sender, qui contient le nom de champ "Sender" et une seule spécification de boîte à lettre) DOIT 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:" mailbox-list CRLF

sender = "Sender:" mailbox CRLF

reply-to = "Reply-To:" address-list CRLF

Les champs d'origine indiquent la ou les boîtes à lettres de la source du message. Le champ "From:" spécifie le ou les auteurs du message, c'est-à-dire la ou les boîtes à lettre de la ou les personnes ou systèmes responsables de l'écriture du message. Le champs "Sender:" spécifie la boîte à lettre 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 simple boîte au lettre et que l'auteur et le transmetteur sont les mêmes, le champ "Sender:" NE DEVRAIT PAS être utilisé. Sinon, les deux champs DEVRAIENT apparaître.

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 boîtes à lettre 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 à lettre 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 à lettre n'appartenant pas à l'auteur du message. Voir aussi au paragraphe 3.6.3 pour plus d'information 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 soit "To", "Cc", ou "Bcc" suivi d'une liste d'une ou plusieurs adresses (de syntaxe de boîte à lettre ou de groupe) séparées par des virgules.

To = "To:" address-list CRLF
Cc = "Cc:" address-list CRLF
Bcc = "Bcc:" (address-list / [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 leur utilisation.

Le champ "To:" contient la ou les adresses du receveur principal 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 ne s'adresse pas à eux.

Le champ "Bcc:" (ou "Bcc" veut dire copie carbone aveugle) contient les adresses des receveurs des messages dont l'adresse n'est pas à révéler aux autres receveurs du message. Il y a trois manières d'utiliser le champ "Bcc". Le premier, quand un message contenant un "Bcc:" est préparé pour l'envoi, la ligne "Bcc:" est enlevée même si tous les receveurs (incluant 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 à lettre des auteurs du message d'origine (celles du champ "From:") ou les boîtes à lettre 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 contient des champs de destination, il est souvent souhaitable d'envoyer une copie de la réponse à tous les receveurs du message, en plus des auteurs. 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 du message. 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 facultatif, chaque message DEVRAIT avoir un champ "Message-ID". De plus, les messages de réponse DEVRAIENT avoir des champs "In-Reply-To:" et "References:", en tant que de besoin, comme décrit ci-dessous.

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

L'identifiant de message (msg-id) est similaire en syntaxe à une construction de angle-addr sans le CFWS interne.

```
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       =    dot-atom-text / no-fold-quote / obs-id-left
id-right      =    dot-atom-text / no-fold-literal / obs-id-right
no-fold-quote =    DQUOTE *(qtext / quoted-pair) DQUOTE
no-fold-literal =  "[" *(dtext / quoted-pair) "]"
```

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 instanciation 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 instanciation 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 (c'est-à-dire qu'il s'agit du même message ou d'un message différent) qui détermine si oui ou non le champ "Message-ID:" change, et non 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 angle-addr (identique sauf que les commentaires et les FWS ne sont pas autorisés), une bonne méthode consiste à mettre le nom de domaine (ou une adresse IP de domaine littéral) de l'hôte sur lequel l'identifiant du message a été créé à droite du "@", 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. Utiliser la date à gauche et un nom de domaine ou une adresse IP à droite rend possible de garantir l'unicité puisqu'il n'y a pas deux hôtes utilisant le même nom de domaine ou d'adresse IP en même temps. Bien que d'autres algorithmes marcheraient, il est RECOMMANDÉ que le côté droit contienne un identifiant de domaine (soit de l'hôte lui-même ou soit autres) de manière à ce que le générateur de l'identifiant du message puisse garantir l'unicité de la partie gauche à l'intérieur de ce domaine.

Sémantiquement, les caractères ">" et "<" ne font pas partie du msg-id; le msg-id est ce qui est contenu à l'intérieur de ces caractères.

3.6.5 Champs d'information

Les champs d'information sont tous facultatifs. Le champ "Keyword:" contient une liste (séparée par des virgules) de un ou plusieurs mots ou de chaînes marquées. 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 FWS.

```
subject      = "Subject:" unstructured CRLF
comments     = "Comments:" unstructured CRLF
keywords     = "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 une courte chaîne identifiant le sujet du message. Quand on l'utilise dans une réponse, le corps du champ PEUT commencer par la chaîne "Re:" (du latin "res", 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:" 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. 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.

Quand ces champs 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 envoyé si "resent-Sender:" est identique à "Resent-From:".

resent-date	=	"Resent-Date:" date-time CRLF
resent-from	=	"Resent-From:" mailbox-list CRLF
resent-sender	=	"Resent-Sender:" mailbox CRLF
resent-to	=	"Resent-To:" address-list CRLF
resent-cc	=	"Resent-Cc:" address-list CRLF
resent-bcc	=	"Resent-Bcc:" (address-list / [CFWS]) CRLF
resent-msg-id	=	"Resent-Message-ID:" msg-id CRLF

Les champs de renvoi sont utilisés pour identifier un message qui a été réintroduit dans le système de transport par un utilisateur. L'intérêt d'utiliser ces champs 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 particulier de renvoi. 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 "Forwarding". Ce dernier a deux significations : la première dans le sens où un utilisateur peut donner pour instruction à un programme de lecture de la messagerie de retransmettre une copie d'un message à une autre personne, faisant du message retransmis le corps du nouveau message. Un message retransmis de cette manière ne paraît pas venir de l'émetteur d'origine, mais est un message entièrement nouveau du transmetteur du message. D'un autre côté, retransmettre est aussi utilisé dans le sens où 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 à lettre de ou des personnes ou systèmes qui retransmettent ce message. Comme avec les champs d'origine réguliers, il existe deux formes : une forme simple "Resent-From:" qui contient la boîte à lettre de l'individu qui a fait le renvoi, et une forme plus complexe, quand un individu (identifié dans le champ "Resent-Sender:") retransmet 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 diffusé par le retransmetteur du message. Tout comme pour le champ "Date:" il ne s'agit ni de la date et l'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:", and "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 un addr-spec facultatif. Le champ "Received:" contient une liste (qui peut être vide) de paires nom/valeur suivie par un point virgule et une spécification de date/heure. Le premier élément de paire nom/valeur est défini par un nom d'élément, et le second élément est soit une addr-spec, un atome, un domaine, ou un msg-id. Des restrictions supplémentaires peuvent être appliquées à la syntaxe des champs de traçage par des normes qui traitent de leur utilisation (comme la RFC 2821).

trace	=	[return] 1*received
return	=	"Return-Path:" path CRLF
path	=	([CFWS] "<" ([CFWS] / addr-spec) ">" [CFWS]) / obs-path
received	=	"Received:" name-val-list ";" date-time CRLF
name-val-list	=	[CFWS] [name-val-pair *(CFWS name-val-pair)]
name-val-pair	=	item-name CFWS item-value
item-name	=	ALPHA *(["-"]) (ALPHA / DIGIT)
item-value	=	1*angle-addr / addr-spec / atom / domain / msg-id

Une discussion complète sur l'usage des champs de traçage dans la maessaging Internet est fournie dans la RFC 2821. Pour les besoins de la présente norme, les champs de traçage sont purement informatifs, et toute interprétation formelle sort du domaine d'application du présent document.

3.6.8 Champs facultatifs

Des champs peuvent apparaître dans les messages qui ne soient pas autrement spécifié dans la présente norme. Ils **DOIVENT** se conformer à la syntaxe des champs facultatifs. C'est un champ de nom, fait de caractères US-ASCII imprimables sauf SP et deux points, suivi par deux points, suivis par un texte qui se conforme à la syntaxe de non structuré.

Les noms de champ de tout champ facultatif **NE DOIVENT PAS** être identiques à tout champ de nom spécifiés ailleurs dans ce standard.

optional-field	=	field-name ":" unstructured CRLF
field-name	=	1*ftext
ftext	=	%d33-57 / ; n'importe quel caractères sauf %d59-126 ; SP et ":"

Pour les besoins de la présente norme, aucun champ facultatif n'est interprété.

4 Syntaxe obsolète

Les versions précédentes de la présente norme permettaient des syntaxes différentes (habituellement plus libérales) de celles permises 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. Pour reprendre un exemple, bien que ce document suggère d'utiliser seulement 998 caractères par ligne, détruire le 999^e caractère dans une ligne sans avertissement serait un mauvais comportement pour une mise en œuvre. Il appartient aux mises en œuvre de traiter les messages de façon cohérente.

Une différence importante entre la syntaxe obsolète (interprétation) et la syntaxe actuelle (génération) 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) les caractères d'espace blanc, même les FWS, et les commentaires peuvent être librement insérés entre n'importe quelle marque lexicale. Ce qui permet de nombreuses formes complexes qui ont donné quelques difficultés d'analyse à certaines mises en œuvre.

Une autre différence fondamentale entre la syntaxe obsolète et la courante est que la règle du paragraphe 3.2.3 concernant les lignes entièrement composées d'espaces blancs dans des commentaires et les FWS ne s'appliquent pas. Voir la discussion sur le FWS au paragraphe 4.2.

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. CR et LF étaient autorisés dans les messages, autrement que sous la forme CRLF, cet usage sera expliqué ici.

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

4.1 Diverses marques obsolètes

Ces éléments syntaxiques sont utilisés ailleurs dans la syntaxe obsolète ou dans la syntaxe principale. Les éléments obs-char et obs-qp ajoutent chacun une valeur ASCII 0. Le CR et le LS nus sont ajoutés à obs-text et obs-utext. Le caractère de "période" est ajouté à obs-phrase. Le obs-phrase-list fournit des éléments vides dans une liste de phrases séparées par des virgules.

Note : Le caractère "point" (ou "point final") (".") dans une obs-phrase n'est pas une forme qui était permise dans les versions précédentes de la présente norme ou de toute autre. Le point (pas plus qu'aucun autre caractères de la série spéciale) n'était pas permis dans les phrases car il introduisait des difficultés d'analyse entre les phrases et les portions d'addr-spec (voir au paragraphe 4.4). Il est apparu ici parce que le caractère point est couramment utilisé dans de nombreux messages dans la partie d'affichage du nom des adresse, surtout pour les initiales, et donc il doit être interprété correctement. Dans le futur, le point pourrait apparaître dans la syntaxe courante des phrases.

```
obs-qp      =  "\" (%d0-127)
obs-text    =  *LF *CR *(obs-char *LF *CR)
obs-char    =  %d0-9 / %d11 / ; %d0-127 excepté CR et %d12 / %d14-127 ; LF
obs-utext   =  obs-text
obs-phrase  =  word *(word / "." / CFWS)
obs-phrase-list =  phrase / 1*([phrase] [CFWS] ", " [CFWS]) [phrase]
```

CR et LF nus apparaissent dans les messages avec deux significations différentes. Dans de nombreux cas, CR et LF nus sont utilisés à tort au lieu du CRLF pour indiquer la séparation de ligne. Dans d'autres cas ils sont utilisés simplement en tant que caractères de contrôle ASCII avec leurs significations ASCII traditionnelles.

4.2 FWS Obsolète

Dans la syntaxe obsolète, un nombre illimité de FWS PEUT être inséré où le obs-FWS est permis. Ce qui crée la possibilité d'avoir deux "sauts à la ligne" consécutifs dans une ligne, et donc la possibilité qu'une ligne qui est composée d'un champ d'en tête de saut à la ligne soit entièrement composée d'espaces blancs.

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

4.3 Date et heure obsolète

La syntaxe pour le format obsolète de date et d'heure permet deux chiffres pour les années dans le champ de date et permet une liste de spécifications de zone horaire alphabétique qui était utilisé dans des précédentes versions de ce standard. Elle permet aussi des commentaires et des FWS entre de nombreuses marques.

obs-day-of-week =	[CFWS] day-name [CFWS]
obs-year =	[CFWS] 2*DIGIT [CFWS]
obs-month =	CFWS month-name CFWS
obs-day =	[CFWS] 1*2DIGIT [CFWS]
obs-hour =	[CFWS] 2DIGIT [CFWS]
obs-minute =	[CFWS] 2DIGIT [CFWS]
obs-second =	[CFWS] 2DIGIT [CFWS]
obs-zone =	"UT" / "GMT" / ; Temps Universel ; Décalages Nord Américain sur l'UT ; "EST" / "EDT" / ; Est: - 5/ - 4 "CST" / "CDT" / ; Central: - 6/ - 5 "MST" / "MDT" / ; Montagnes: - 7/ - 6 "PST" / "PDT" / ; Pacifique: - 8/ - 7 %d65-73 / ; Zones militaires – de "A" %d75-90 / ; à "I" et de "K" %d97-105 / ; à "Z", en majuscules %d107-122 ; et minuscules

Là où deux ou trois chiffres d'année apparaissent dans une date, l'année est interprétée comme suit : si il y a deux chiffres avec une valeur comprise entre 00 et 49, l'année est interprétée comme comprise entre 2000 et 2049. Si les deux chiffres sont compris entre 50 et 99 ou s'il y a un troisième chiffre, il faut lui ajouter 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", ou "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 RFC 822 et sont donc imprévisibles quandt 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 [RFC822], elles DEVRAIENT toutes être considérée 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 trois différences principales dans l'adressage. Tout d'abord, il était permis aux adresses de messagerie électronique d'avoir une portion de chemin avant le addr-spec lorsqu'il était compris entre "<" et ">". Le chemin est simplement une liste de noms de domaine séparés par des virgules, chacun d'eux étant précédé par "@", et la liste étant terminée par deux points. Ensuite, les CFWS étaient admis entre les éléments, séparés par des points, de partie locale et de domaine (c'est-à-dire que dot-atom n'était pas utilisé). Enfin, il est permis à la partie locale de contenir une chaîne marquée en plus d'un simple atome. Finalement, mailbox-list et address-list ont pu avoir

des membres "null". C'est-à-dire qu'il pouvait y avoir deux virgules ou plus dans une telle liste, avec rien entre elles.

```
obs-angle-addr = [CFWS] "<" [obs-route] addr-spec ">" [CFWS]
obs-route      = [CFWS] obs-domain-list ":" [CFWS]
obs-domain-list = "@" domain *(*(CFWS / "," ) [CFWS] "@" domain)
obs-local-part = word *("." word)
obs-domain     = atom *("." atom)
obs-mbox-list  = 1*([mailbox] [CFWS] "," [CFWS]) [mailbox]
obs-addr-list  = 1*([address] [CFWS] "," [CFWS]) [address]
```

Quand on interprète ces adresses, la portion route DEVRAIT être ignorée.

4.5 Champs d'en tête obsolète

Syntaxiquement, la différence principale dans la syntaxe des champs d'en tête obsolètes est qu'elle permet l'occurrence multiple de n'importe quel champ et qu'ils peuvent survenir dans n'importe quel ordre. De plus n'importe quel nombre d'espace blanc 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 au 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 Champ de date d'origine obsolète

```
obs-orig-date = "Date" *WSP ":" date-time CRLF
```

4.5.2 Champ d'origine obsolète

obs-from = "From" *WSP ":" mailbox-list CRLF
obs-sender = "Sender" *WSP ":" mailbox CRLF
obs-reply-to = "Reply-To" *WSP ":" mailbox-list CRLF

4.5.3 Champs adresse de destination obsolètes

obs-to = "To" *WSP ":" address-list CRLF
obs-cc = "Cc" *WSP ":" address-list CRLF
obs-bcc = "Bcc" *WSP ":" (address-list / [CFWS]) CRLF

Quand des occurrences multiples de champs adresse de destination apparaissent dans un message, ils DEVRAIENT être traités comme si l'address-list dans la première occurrence du champ était combinée avec les address-list de l'occurrence suivante en ajoutant une virgule et en l'enchaînant.

4.5.4 Champs d'identification obsolètes

Les champs "In-Reply-To:" et "References:" obsolètes diffèrent de la syntaxe actuelle par le fait qu'ils permettent l'apparition de phrases (des mots ou des chaînes marquées). Les formes obsolètes de la partie gauche / droite du msg-id permettent d'y inclure des CFWS, les rendant sémantiquement identiques, respectivement, 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-left = local-part
obs-id-right = domain

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

Sémantiquement, aucun des CFWS facultatifs entourant la partie locale et le domaine ne fait partie, respectivement du obs-id gauche ou droit.

4.5.5 Champs d'information obsolètes

obs-subject = "Subject" *WSP ":" CRLF non structuré
obs-comments = "Comments" *WSP ":" CRLF non structuré
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, le commentaire facultatif et un FWS, deux points, et une liste d'adresses séparées par une virgule.

obs-resent-from = "Resent-From" *WSP ":" mailbox-list CRLF
obs-resent-send = "Resent-Sender" *WSP ":" mailbox CRLF
obs-resent-date = "Resent-Date" *WSP ":" date-time CRLF
obs-resent-to = "Resent-To" *WSP ":" address-list CRLF
obs-resent-cc = "Resent-Cc" *WSP ":" address-list CRLF
obs-resent-bcc = "Resent-Bcc" *WSP ":" (address-list / [CFWS]) CRLF
obs-resent-mid = "Resent-Message-ID" *WSP ":" msg-id CRLF
obs-resent-rply = "Resent-Reply-To" *WSP ":" address-list CRLF

Comme avec tout les autres champs de ce type, le champ "Resent-Reply-To:" doit être uniquement traité en tant qu'information de traçage.

4.5.7 Champs de traçage obsolètes

Les "obs-return" et "obs-received" sont donnés ici à titre d'information, comme au paragraphe 3 pour return et received. Leur syntaxe complète est donnée dans la RFC 2821.

```
obs-return      = "Return-Path" *WSP ":" path CRLF
obs-received    = "Received" *WSP ":" name-val-list CRLF
obs-path        = obs-angle-addr
```

4.5.8 Champs facultatifs obsolètes

```
obs-optional    = field-name *WSP ":" unstructured CRLF
```

5 Considérations sur la sécurité

Il faut faire attention quand on affiche des messages sur un terminal ou son émulateur. Les terminaux puissants peuvent agir sur les séquences d'échappement ou tout 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 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 de terminal potentiellement dangereuses avant l'affichage. Cependant, d'autres séquences d'échappement apparaissent dans les messages dans un but utile (cf. [RFC2045, RFC2046, RFC2047, RFC2048, RFC2049, ISO2022]) ne doivent pas être neutralisées sans discrimination.

La transmission d'objets non textuels dans un message soulève des problèmes supplémentaires de sécurité. Ceci est exposé dans [RFC2045, RFC2046, RFC2047, RFC2048, RFC2049].

De nombreuses mises en oeuvre utilisent le champ "Bcc:" 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:" a des implications sur la confidentialité des informations qui pourraient être révélées, qui pourrait également mener à des problèmes de sécurité par la connaissance de l'existence d'une adresse de messagerie électronique 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 ait été envoyée, sauf par le fait que leurs adresses n'apparaissent pas dans l'en tête du message. A 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 toutes les adresses aveugles, alors tous les receveurs du "Bcc:" seront vus par chaque receveur d'un "Bcc:". Même si un message séparé est envoyé à chaque "Bcc:" avec seulement une adresse individuelle, il faut encore que les mises en oeuvre veillent à traiter les réponses au message conformément au paragraphe 3.6.3 de manière à ne pas révéler le receveur en aveugle aux autres receveurs.

6 Bibliographie

[ASCII] American National Standards Institute (ANSI), Coded Character Set - 7-Bit American National Standard Code for Information Interchange, ANSI X3.4, 1986.

[ISO2022] International Organization for Standardization (ISO), Information processing - ISO 7-bit and 8-bit coded character sets - Code extension techniques, Third edition - 1986-05-01, ISO 2022, 1986.

[RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", RFC 822, August 1982.

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

[RFC2047] Moore, K., "Multipurpose Internet Mail Extensions (MIME) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2048] Freed, N., Klensin, J. and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Format of Internet Message Bodies", RFC 2048, November 1996.

[RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2234] Crocker, D., Editor, and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[RFC2821] Klensin, J., Editor, "Simple Mail Transfer Protocol", RFC 2821, March 2001.

[STD3] Braden, R., "Host Requirements", STD 3, RFC 1122 and RFC 1123, October 1989.

[STD12] Mills, D., "Network Time Protocol", STD 12, RFC 1119, September 1989.

[STD13] Mockapetris, P., "Domain Name System", STD 13, RFC 1034 and RFC 1035, November 1987.

[STD14] Partridge, C., "Mail Routing and the Domain System", STD 14, RFC 974, January 1986.

7 Adresse de l'éditeur

Peter W. Resnick
QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
USA
Phone: +1 858 651 4478
Fax: +1 858 651 1102
EMail: presnick@qualcomm.com

8 Remerciements:

Beaucoup de personnes ont contribué à ce document. Cela inclut les membres du groupe de travail Detailed Revision and Update of Messaging Standards (DRUMS) de l'Internet Engineering Task Force (IETF), le président de DRUMS, les directeurs de zone de l'IETF, et les personnes qui m'ont simplement envoyés leurs commentaires par messagerie électronique. L'éditeur a une dette importante envers elles et remercie sincèrement toutes ces personnes. La liste ci dessous inclut toutes les personnes qui m'ont envoyé un message relatif à ce document. J'espère que tous y sont présents :

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

Appendice A Exemples de messages

Cette section présente une sélection de messages. Ils sont destinés à aider à la mise en œuvre de ce standard, mais ne doivent pas être pris comme éléments normatifs ; en fait bien que les exemples aient été attentivement étudiés, s'il y a conflit entre la syntaxe utilisée dans ces exemples et la syntaxe utilisée aux sections 3 et 4 de ce document, c'est la syntaxe de ces sections qui doit être considérée comme correcte.

Les messages sont délimités dans cette section entre des lignes de "----". Ces lignes ne font pas partie du message lui-même.

A.1 Exemples d'adressage

Exemple d'envoi de messages entre deux individus.

A.1.1 Message d'une personne à une autre

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 juste pour dire 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 du 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 boîte à lettre.

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" <syservices@example.net>  
Date: Tue, 1 Jul 2003 10:52:37 +0200  
Message-ID: <5678.21-Nov-1997@example.com>
```

Salut tout le monde.

Notez que les noms d'affichage pour Joe Q. Public et Giant; "Big" 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 marquée). A l'inverse, le nom d'affichage pour Who? n'a pas besoin de guillemets car le point d'interrogation est légal dans un atome. Notez également que jdoe@example.org et boss@nil.test n'ont pas du tout de noms d'affichage associés, et jdoe@example.org utilise la forme d'adresse la plus simple sans les crochets angulaires.

A.1.3 Adresses de groupe

From: Pete <pete@silly.example>
To: A Group:Chris 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 A Group qui contient trois adresses, et un champ "Cc:" qui contient un groupe vide nommé Undisclosed recipients (*receveurs non révélés*).

A.2 Messages de réponses

Ce qui suit est une série de trois messages qui font 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.

Notez 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 de sujet 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: Personal Account" <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

Notez 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: Personal Account" <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.tld>
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 préservés. 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.

A.4 Messages avec des champs de traçage

Lorsque les messages sont envoyés à travers le système de transport comme décrit dans la [RFC2821], les champs de traçage sont rajoutés devant le message. Ce qui suit montre un exemple de ce que ce à quoi ces champs pourraient ressembler. Notez qu'il y a un FWS 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 machine.example by x.y.test; 21 Nov 1997 10:01:22 -0600
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".

A.5 *Espaces blancs, commentaires et autres raretés*

Les espaces blancs, y compris les FWS, et les commentaires peuvent être insérés entre de nombreux marqueurs de champs. Prenons l'exemple du A.1.3, des espaces blancs et des commentaires peuvent être insérés dans tous les champs.

From: Pete(Un type \) super) <pete(son compte)@silly.test(son hôte)>
To:A Group(Certaines gens)
:Chris Jones <c@(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, n'est pas satisfaisant d'un point de vu esthétique mais parfaitement légal. Notez en particulier (1) les commentaires dans le champ "From:" (y compris celui qui a un caractère ")" apparaissant au titre d'une paire marquée) ; (2) l'espace blanc absent après les ":" dans le champ "To:" de même que le commentaire et le FWS après le nom de groupe, et le caractère spécial "." dans le commentaire de l'adresse de Chris Jones, et le FWS avant et après joe@example.org ; (3) les commentaires incrustés multiples dans le champ "Cc:" ainsi que le commentaire qui suit immédiatement les ":" après le "Cc:" ; (4) le FWS (mais pas de commentaire sauf à la fin) et les secondes manquantes dans l'heure du champ de date ; et (5) l'espace blanc avant (mais pas dedans) l'identifiant dans le champ "Message-ID:".

A.6 *Formes obsolètes*

Les exemples suivants sont des éléments syntaxiques obsolètes (correspondant au "NE DOIT PAS être généré") décrits à la Section 4 du présent document.

A.6.1 Adressage obsolète

Notez dans l'exemple suivant l'absence de guillemet autour de Joe Q. Public, le chemin qui apparaît dans l'adresse pour Mary Smith, les 2 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 <@machine.tld: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 day-of-week est manquant, cela n'est pas spécifique à la syntaxe obsolète; il est optionnel dans la syntaxe courante.

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

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

```
----
```

A.6.3 Commentaire et espaces blancs obsolètes

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

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

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

```
----
```

Notez surtout la seconde ligne du champ "To:". Elle commence par deux caractères espaces. (Noter que "_" représente un espace blanc). Donc, il est considéré comme faisant partie du renvoi à la ligne décrit au paragraphe 4.2. Les commentaires et espaces blancs dans toutes les adresses, dates, et les identifiants de message font tous aussi partie de la syntaxe obsolète.

Appendice B Différences avec les précédents standards

Cet appendice contient une liste des changements qui ont été apportés au format de message Internet par rapport aux normes antérieures, à savoir la RFC822 et le STD3. Ceux marqués d'un astérisque ("*") sont ceux qui apparaissent à la Section 4 de ce document et qui ne peuvent donc plus être produits.

1. La période permise dans le forme obsolète de la phrase.
2. ABNF révisé [RFC2234].
3. Quatre chiffres ou plus permis pour l'année.
4. Ordre des champs d'en tête (et leur absence) rendu explicite.
5. Champs d'en tête crypté supprimé.
6. La syntaxe de Received assouplie pour permettre toute paire marque/valeur.
7. La zone horaire "-0000" est précisément permise et sa signification est établie.
8. L'espace blanc de retour à la ligne (FWS) n'est pas permis entre chaque marquage.
9. Les exigences sur les destinations sont supprimées.
10. Retransmission et renvoi sont redéfinis.
11. Les champs d'en tête d'extension ne sont plus appelés à chaque fois.
12. ASCII 0 (null) enlevé.*
13. Les lignes de continuation de retour à la ligne ne peuvent plus contenir seulement des espaces blancs.*
14. Insertion libre de commentaire non permise dans la date.*
15. Zones d'heures non numérique non permises.*
16. Année à deux chiffres non permise.*
17. Les années à 3 chiffres sont interprétées, mais non permises à la création.
18. Les acheminements ne sont pas permis dans les adresses.*
19. Les CFWS ne sont pas permis dans parties locales et les domaines.*
20. Listes d'adresses vides non permises.*
21. FWS entre nom de champ et deux points non permis.*
22. Commentaires entre nom de champ et deux points non permis.
23. Syntaxe plus stricte de in-reply-to et references.*
24. CFWS dans un msg-id non permis.*
25. Sémantique plus stricte des champs de renvoi comme informatifs seulement.
26. Resent-Reply-To non permis.*
27. Pas d'occurrences multiples de champs (sauf resent et received).*
28. CR et LF seuls non autorisés.*
29. Les acheminements dans le return-path ne sont plus autorisés.*
30. Les limites de taille des lignes sont spécifiées.
31. Bcc précisé.

Appendice C Déclarations de propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient 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.

Propriété intellectuelle

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

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

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet ni ses successeurs ou ayant-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 ET LA INTERNET ENGINEERING TASK FORCE DÉCLINENT TOUTES GARANTIES, EXPRIMÉES OU IMPLICITES, Y COMPRIS MAIS NON LIMITÉES À TOUTE GARANTIE QUE L'UTILISATION DES INFORMATIONS CI-ENCLOSES NE VIOLENT AUCUN DROIT OU AUCUNE GARANTIE IMPLICITE DE COMMERCIALISATION OU D'APTITUDE À UN OBJET PARTICULIER.

Remerciement

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