

Groupe de travail Réseau  
**Request for Comments : 2049**  
 RFC rendues obsolètes : 1521, 1522, 1590  
 Catégorie : Sur la voie de la normalisation

N. Freed, Innosoft  
 N. Borenstein, First Virtual  
 novembre 1996  
 Traduction Claude Brière de L'Isle

## **Extensions multi-objets de la messagerie Internet (MIME)**

### **Partie cinq : critères de conformité et exemples**

#### **Statut du présent mémoire**

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

#### **Résumé**

Le STD 11, RFC 822, définit un protocole de représentation de message qui spécifie des détails considérables sur les en-têtes de message en US-ASCII, et laisse le contenu du message, un corps de message, en texte US-ASCII plat. Cet ensemble de documents, collectivement appelés "Extensions de messagerie Internet multi-objets", ou MIME, redéfinit le format des messages pour permettre :

- (1) des corps de message textuels dans des jeux de caractères autres que l'US-ASCII,
- (2) un ensemble extensible de formats différents pour les corps de message non textuels,
- (3) des corps de message multi-parties, et
- (4) des informations d'en-tête textuelles dans des jeux de caractères autres que l'US-ASCII.

Ces documents se fondent sur des travaux antérieurs documentés dans la RFC 934, STD 11, et dans la RFC 1049, mais les étendent et les révisent. Parce que la RFC 822 en disait si peu sur les corps de message, ces documents prennent largement le contre-pied de la RFC 822 (plutôt que d'en être une révision).

Le document initial de cet ensemble, la RFC 2045, spécifie les divers en-têtes utilisés pour décrire la structure des messages MIME. Le second document définit la structure générale du système MIME de typologie des supports et définit un ensemble initial de types de supports. Le troisième document, la RFC 2047, décrit les extensions à la RFC 822 qui permettent les données de texte non US-ASCII dans les champs d'en-tête de messagerie Internet. Le quatrième document, la RFC 2048, spécifie diverses procédures d'enregistrement de l'IANA pour les facilités en rapport avec MIME. Ce cinquième et dernier document décrit les critères de conformité à MIME et fournit quelques exemples pour illustrer les formats de message MIME, les remerciements et la bibliographie.

Ces documents sont des révisions des RFC 1521, 1522 et 1590, qui elles-mêmes étaient les révisions des RFC 1341 et 1342. L'Appendice B du présent document décrit les différences et changements par rapport aux versions précédentes.

### **Table des matières**

1. Introduction.....	1
2. Conformité à MIME.....	2
3. Lignes directrices pour l'envoi de données de messagerie électronique.....	3
4. Modèle de codage canonique.....	5
5. Résumé.....	6
6. Considérations pour la sécurité.....	6
7. Adresse des auteurs.....	6
8. Remerciements.....	7
Appendice A – Exemple multipartie complexe.....	7
Appendice B -- Changements par rapport aux RFC 1521, 1522 et 1590.....	8
Appendice C -- Références.....	10

## **1. Introduction**

Le premier et le second document de cet ensemble définissent les champs d'en-tête MIME et l'ensemble initial de types de supports MIME. Le troisième document décrit les extensions aux formats de la RFC0822 pour permettre les jeux de caractères autres que US-ASCII. Le présent document décrit les portions de MIME qui doivent être prises en charge par une mise en œuvre conforme à MIME. Il décrit aussi divers pièges des systèmes de messagerie contemporains ainsi que le

modèle de codage canonique sur lequel est fondé MIME.

## 2. Conformité à MIME

Les mécanismes décrits dans ces documents sont ouverts. Il n'est pas du tout prévu que toutes les mises en œuvre prennent en charge tous les types de supports disponibles, ni qu'elles partagent toutes les mêmes extensions. Afin de promouvoir l'interopérabilité, il est cependant utile de définir le concept de "conformité à MIME" pour arriver à un certain niveau de mise en œuvre permettant un interfonctionnement utile des messages avec des contenus qui diffèrent du texte US-ASCII. Dans cette section, on spécifie les exigences pour une telle conformité.

Un agent d'utilisateur de messagerie qui est conforme à MIME DOIT :

- (1) Toujours générer un champ d'en-tête "MIME-Version: 1.0" dans tout message qu'il crée.
- (2) Reconnaître le champ d'en-tête Codage-de-transfert-de-contenu et décoder toutes les données reçues codées par les mises en œuvre quoted-printable ou base64. Les transformations d'identité 7 bits, 8 bits, et binaires doivent aussi être reconnues.  
Toutes données non en 7 bits qui sont envoyées sans codage doivent être étiquetées de façon appropriée avec un Codage-de-transfert-de-contenu de 8 bits ou binaire, selon le cas approprié. Si le transport sous-jacent ne prend pas en charge le 8 bits ou le binaire (SMTP [RFC0821] ne le fait pas) il est exigé de l'expéditeur à la fois qu'il code et qu'il étiquette les données en utilisant un Codage-de-transfert-de-contenu approprié tel que quoted-printable ou base64.
- (3) Traiter tout Codage-de-transfert-de-contenu non reconnu comme si il avait un champ Type de contenu de "application/flux-d'octet", sans considérer si le Type de contenu réel est reconnu.
- (4) Reconnaître et interpréter le champ d'en-tête Type-de-contenu, et éviter de présenter aux utilisateurs des données brutes avec un champ Type-de-contenu autre que de texte. Les mises en œuvre doivent être capables d'envoyer au moins des messages texte/clair, avec le jeu de caractères spécifié par le paramètre charset si ce n'est pas de l'US-ASCII.
- (5) Ignorer tout paramètre de type de contenu dont le nom n'est pas reconnu.
- (6) Traiter explicitement les valeurs de type de support suivantes, au moins dans la mesure suivante :
  - Texte :
    - Reconnaître et afficher les messages "texte" avec le jeu de caractères "US-ASCII."
    - Reconnaître les autres jeux de caractères au moins au point d'être capable d'informer l'utilisateur sur le jeu de caractères utilisé par le message.
    - Reconnaître les jeux de caractères "ISO-8859-\*" au point d'être capable d'afficher les caractères qui sont communs à ISO-8859-\* et à US-ASCII, à savoir tous les caractères représentés par les valeurs d'octet 1 à 127.
    - Pour les sous-types non reconnus dans un jeu de caractères connu, présenter ou offrir de présenter à l'utilisateur la version "brute" des données après conversion du contenu de la forme canonique à la forme locale.
    - Traiter un matériau qui est dans un jeu de caractères inconnu comme si il était en "application/flux-d'octet".
  - Image, audio et vidéo :
    - Au minimum fournir la facilité de traiter tout sous-type non reconnu comme si il était en "application/flux-d'octet".
  - Application :
    - Offrir la capacité de retirer l'un ou l'autre des codages quoted-printable ou base64 définis dans le présent document si ils étaient utilisés et mettre les informations résultantes dans un fichier d'utilisateur.
  - Multipartie :
    - Reconnaître le sous-type mixte. Afficher toutes les informations pertinentes au niveau du message et au niveau de l'en-tête de partie corps pour afficher ou offrir d'afficher individuellement chacune des parties du corps.
    - Reconnaître le sous-type "alternatif", et éviter de présenter à l'utilisateur les parties redondantes d'un message multipartie/alternatif.
    - Reconnaître le sous-type "multipartie/résumé", spécifiquement celui qui utilise "message/rfc822" au lieu de "texte/clair" comme le type de support par défaut pour les parties de corps à l'intérieur d'entités multipartie/résumé".
    - Traiter tout sous-type non reconnu comme si il était "mixte".
  - Message :
    - Reconnaître et afficher au moins l'encapsulation de message de la RFC 822 (message/rfc822) de façon à préserver toute structure récurrente, c'est-à-dire, en affichant ou en offrant d'afficher les données encapsulées

conformément à leur type de support.

-- Traiter tous les sous-types non reconnus comme si ils étaient en "application/flux-d'octet".

- (7) Lorsque elle rencontre un champ Type-de-contenu inconnu, une mise en œuvre doit le traiter comme si il avait un type de support "application/flux-d'octet" sans aucun sous argument de paramètre. Il dépend de la mise en œuvre de définir la façon de traiter de telles données, mais les options vraisemblables pour le traitement de telles données non reconnues incluent d'offrir à l'utilisateur de les écrire dans un fichier (décodé à partir de son format de transport de messagerie) ou de lui offrir de désigner un programme auquel les données décodées devraient être passées.
- (8) Les agents d'utilisateur conformes sont obligés, si ils fournissent la prise en charge non standard de messages non MIME employant des jeux de caractères autres que l'US-ASCII, de ne le faire que sur les messages reçus. Les agents d'utilisateur conformes ne doivent pas envoyer de messages non MIME contenant autre chose que du texte US-ASCII. En particulier, l'utilisation de texte non US-ASCII dans un message électronique sans un champ Version MIME est fortement déconseillée car elle entrave l'interopérabilité lors de l'envoi de messages entre des régions ayant des conventions de localisation différentes. Les agents d'utilisateur conformes DOIVENT inclure un étiquetage MIME appropriée lors de l'envoi d'autre chose que du texte en clair dans le jeu de caractères US-ASCII.

De plus, les agents d'utilisateur non MIME devraient être mis à niveau si c'est possible pour inclure les informations d'en-tête MIME appropriées dans les messages qu'ils envoient même si rien d'autre de MIME n'est pris en charge. Cette mise à niveau aura peu d'effet, si elle en a, sur les receveurs non MIME et aidera MIME à afficher correctement de tels messages. Elle fournit également un chemin de transition en douceur vers une adoption éventuelle d'autres capacités MIME.

- (9) Les agents d'utilisateur conformes doivent s'assurer que toute chaîne de caractères US-ASCII imprimables autres que des espaces blanches au sein d'un "\*texte" ou "\*ctexte" qui commence par "=?" et se termine par "=?" est un mot-codé valide. ("commence" signifie : au début du corps de champ ou suivant immédiatement une espace blanche linéaire ; "se termine" signifie : à la fin du corps de champ ou précédant immédiatement une espace blanche linéaire.) De plus, tout "mot" au sein d'une "phrase" qui commence par "=?" et se termine par "=?" doit être un mot-codé valide.
- (10) Les agents d'utilisateur conformes doivent être capables de distinguer les mots-codés du "texte", "ctexte", ou "mot", conformément aux règles de la section 4, chaque fois qu'ils apparaissent aux endroits appropriés dans les en-têtes de message. Ils doivent prendre en charge les codages "B" et "Q" pour tout jeu de caractères qu'ils acceptent. Le programme doit être capable d'afficher le texte non codé si le jeu de caractères est "US-ASCII". Pour les jeux de caractères ISO-8859-\*, le programme de lecture de messagerie doit au moins être capable d'afficher les caractères qui sont aussi dans le jeu US-ASCII.

Un agent d'utilisateur qui satisfait aux conditions ci-dessus doit être conforme à MIME. Cette phrase signifie qu'on suppose qu'il est "sûr" d'envoyer virtuellement toutes sortes de données marquées de façon appropriée aux utilisateurs de tels systèmes de messagerie, parce que de tels systèmes seront au moins capables de traiter les données comme du binaire indifférencié, et ne vont pas simplement les étaler sur l'écran d'utilisateurs innocents.

Il y a aussi une autre façon de comprendre qu'il est toujours "sûr" d'envoyer des données sous un format conforme à MIME, qui est que de telles données ne seront pas cassées ou détruites par les systèmes dont il est connu qu'ils sont conformes aux RFC 821 et RFC 822. Les agents d'utilisateur qui sont conformes à MIME ont la garantie supplémentaire que l'utilisateur ne se verra pas présenter des données qu'il n'avait jamais été prévu de présenter comme du texte.

### 3. Lignes directrices pour l'envoi de données de messagerie électronique

La messagerie électronique de l'Internet n'est pas un système homogène parfait. La messagerie peut être corrompue à divers stades de son voyage jusqu'à la destination finale. En particulier, les messages envoyés à travers l'Internet peuvent voyager à travers de nombreuses technologies de réseau. De nombreuses technologies de réseautage et de messagerie ne prennent pas en charge les pleines fonctionnalités possibles dans l'environnement de transport SMTP. La messagerie qui traverse ces systèmes va vraisemblablement être modifiée afin de pouvoir la transporter.

Il existe de nombreux MTA (*agent de transfert de messagerie*) non conformes largement déployés dans l'Internet. Ces MTA, qui parlent le langage du protocole SMTP, altèrent les messages au vol pour tirer parti de la structure interne des données des hôtes sur lesquels ils sont mis en œuvre, ou les coupent simplement en plein milieu.

Les lignes directrices suivantes peuvent être utiles pour quiconque utilise un format de données (type de support) qui est supposé survivre à la plus large gamme de technologies d'inter-réseautage et rester indemnes aux MTA casseurs connus.

Noter que tout ce qui est codé en base64 va satisfaire à ces règles, mais que certains mécanismes bien connus, notamment le dispositif UNIX uuencode, ne le feront pas. Noter aussi que tout ce qui est codé avec Quoted-Printable va survivre intact à la plupart des routeurs, mais peut-être pas à certaines passerelles vers des systèmes qui utilisent le jeu de caractères EBCDIC.

- (1) Dans certaines circonstances, le codage utilisé pour les données peut changer au titre d'une opération normale de passerelle ou d'agent d'utilisateur. En particulier, la conversion de base64 en quoted-printable et vice versa peut être nécessaire. Il peut en résulter une confusion des séquences de CRLF avec les coupures de ligne dans les corps de texte. À ce titre, la persistance du CRLF comme quelque chose d'autre qu'un saut à la ligne ne doit pas être prise pour argent comptant.
- (2) De nombreux systèmes peuvent choisir de représenter et mémoriser les données de texte à l'aide de conventions locales de représentation de nouvelle ligne. Les conventions locales de nouvelle ligne peuvent ne pas respecter la convention du CRLF de la RFC 822 – on connaît des systèmes qui utilisent le CR seul, le LF seul, le CRLF, ou des enregistrements de comptage. Il en résulte que les caractères CR et LF isolés ne sont en général pas bien tolérés ; ils peuvent être perdus ou convertis en délimiteurs sur certains systèmes, et donc on ne peut pas s'y fier.
- (3) La transmission des NUL (valeur US-ASCII de 0) est problématique dans la messagerie Internet. (Ceci résulte largement de l'utilisation des NUL comme caractères de terminaison par de nombreux sous-programmes standard de bibliothèque de démarrage dans le langage de programmation C.) La pratique de l'utilisation des NUL comme caractères de terminaison est si bien établie aujourd'hui que les messages ne devraient pas s'attendre à ce qu'ils soient préservés.
- (4) Les caractères TAB (HT) peuvent être mal interprétés ou peuvent être automatiquement convertis en nombre variable d'espaces. Ceci est inévitable dans certains environnements, notamment ceux qui ne sont pas fondés sur le jeu de caractères US-ASCII. Une telle conversion est **FORTEMENT DÉCONSEILLÉE**, mais elle peut arriver, et les formats de messagerie ne doivent pas compter sur la persistance des caractères TAB (HT).
- (5) Les lignes plus longues que 76 caractères peuvent être renvoyées à la ligne ou tronquées dans certains environnements. Le renvoi à la ligne, ou la coupure de ligne, imposé par les transports de messagerie est **FORTEMENT DÉCONSEILLÉ**, mais inévitable dans certains cas. Les applications qui exigent de longues lignes doivent faire une différence entre la coupure de ligne douce et la brutale. (Une façon simple de le faire est d'utiliser le codage quoted-printable.)
- (6) Les caractères "espace blanche" de queue (SPACE, TAB (HT)) sur la ligne peuvent être éliminés par certains agents de transport, alors que d'autres agents de transport peuvent bourrer les lignes avec ces caractères afin que toutes les lignes d'un fichier de messagerie soient de longueur égale. La persistance des espaces blanches en queue ne doit donc pas être tenue pour acquise.
- (7) De nombreux domaines de messagerie utilisent des variantes du jeu de caractères US-ASCII, ou utilisent des jeux de caractères tels que l'EBCDIC qui contient la plupart des caractères US-ASCII mais pas tous. La traduction correcte des caractères qui ne sont pas dans le jeu "invariant" ne peut pas reposer sur des passerelles de conversion de caractères. Par exemple, cette situation pose problème lors de l'envoi d'informations non codées à travers BITNET, un système EBCDIC. Des problèmes similaires peuvent survenir sans franchir de passerelle, car de nombreux hôtes Internet utilisent en interne des jeux de caractères autres que l'US-ASCII. La définition de chaînes imprimables dans X.400 ajoute de nouvelles restrictions dans certains cas spéciaux. En particulier, les seuls caractères qui soient connus pour être cohérents à travers toutes les passerelles sont les 73 caractères qui correspondent aux lettres A-Z majuscules et a-z minuscules, les 10 chiffres 0-9, et les onze caractères spéciaux suivants :
  - "" (valeur décimale US-ASCII de 39)
  - "(" (valeur décimale US-ASCII de 40)
  - )" (valeur décimale US-ASCII de 41)
  - +" (valeur décimale US-ASCII de 43)
  - ," (valeur décimale US-ASCII de 44)
  - " (valeur décimale US-ASCII de 45)
  - ." (valeur décimale US-ASCII de 46)
  - /" (valeur décimale US-ASCII de 47)
  - :" (valeur décimale US-ASCII de 58)
  - =" (valeur décimale US-ASCII de 61)
  - ?" (valeur décimale US-ASCII de 63)

Une représentation de messagerie à portabilité maximale se limitera elle-même à des lignes relativement courtes de texte dans lequel les seuls caractères significatifs sont tirés de cet ensemble de 73 caractères. Le codage base64 suit cette règle.

- (8) Certains agents de transport de messagerie vont corrompre les données qui incluent certaines chaînes littérales. En particulier, il est connu qu'un point (".") seul sur une ligne est corrompu par certaines mises en œuvre (incorrectes) de SMTP, et qu'une ligne qui commence par les cinq caractères "From " (le cinquième caractère est un SPACE) est aussi couramment corrompue. Un agent de composition soigneux peut empêcher ces corruptions en codant les données (par exemple, avec le codage quoted-printable en utilisant "=46rom " à la place de "From " au début d'une ligne, et "=2E" à la place de "." seul sur une ligne).

Prière de noter que la liste ci-dessus N'EST PAS une liste de pratiques recommandées pour les MTA. Il est interdit aux MTA de la RFC 821 d'altérer les caractères d'espace blanche ou de couper les longues lignes. Ces pratiques MAUVAISES et invalides sont réputées survenir sur des réseaux bien établis, et les mises en œuvre devraient être robustes dans le traitement des mauvais effets qu'ils peuvent causer.

#### 4. Modèle de codage canonique

Il y a eu une certaine confusion, dans des versions antérieures de ces documents, en ce qui concerne le modèle à utiliser lors de la conversion de données de messagerie de forme canonique en forme codée, et en particulier sur la façon dont ce traitement affecterait celui des CRLF, étant donné que la représentation des nouvelles lignes varie considérablement d'un système à l'autre. Pour cette raison, un modèle canonique du codage est présenté ci-dessous.

Le processus de composition d'une entité MIME peut être modélisé comme étant accompli suivant un certain nombre d'étapes. Noter que ces étapes sont en gros similaires à celles utilisées dans PEM [RFC1421] et sont effectuées pour chaque corps "de niveau le plus interne" :

- (1) Création de la forme locale.  
Le corps à transmettre est créé dans le format natif du système. Le jeu de caractère natif est utilisé et, lorsque c'est approprié, les conventions locales de fin de ligne sont utilisées aussi. Le corps peut être un fichier texte de style UNIX, ou une image matricielle Sun, ou un fichier indexé VMS, ou des données audio dans un format dépendant du système stockées seulement en mémoire, ou n'importe quoi d'autre qui corresponde au modèle local pour la représentation d'une forme d'information. Fondamentalement, les données sont créées dans la forme "native" qui correspond au type spécifié par le type de support.
- (2) Conversion en forme canonique.  
Le corps entier, y compris les informations "hors bande" telles que les longueurs d'enregistrements et éventuellement les informations d'attributs de fichier, est converti en une forme canonique universelle. Le type de support spécifique du corps ainsi que ses attributs associés dicte la nature de la forme canonique qui est utilisée. La conversion en la forme canonique appropriée peut impliquer une conversion de jeu de caractères, la transformation de données audio, la compression, ou diverses autres opérations spécifiques des divers types de supports. Si la conversion de jeu de caractères est impliquée, il faut cependant faire attention à comprendre la sémantique du type de support, qui peut avoir de fortes implications pour toute conversion de jeu de caractères, par exemple, à l'égard de caractères syntaxiquement significatifs dans un sous-type de texte autre que "clair".  
  
Par exemple, dans le cas de données texte/clair, le texte doit être converti en un jeu de caractères pris en charge et les lignes doivent être délimitées avec des délimiteurs CRLF conformément à la [RFC0822]. Noter que la restriction sur la longueur de ligne impliquée par la RFC 822 est éliminée si l'étape suivante emploie les codages quoted-printable ou base64.
- (3) Application du codage de transfert.  
On applique un Codage de transfert de contenu approprié à ce corps. Noter qu'il n'y a pas de relation fixée entre le type de support et le codage de transfert. En particulier, il peut être approprié de fonder le choix de base64 ou quoted-printable sur des comptes de fréquence de caractères qui sont spécifiques d'une instance de corps donnée.
- (4) Insertion dans l'entité.  
Le corps codé est inséré dans l'entité MIME avec les en-têtes appropriés. L'entité est alors insérée dans le corps d'une entité de niveau supérieur (message ou multipartie) en tant que de besoin.

La conversion de la forme entité en forme locale est réalisée en inversant ces étapes. Noter que l'inversion de ces étapes peut produire des résultats différents car il n'y a aucune garantie que la forme d'origine et la forme locale finale soient les mêmes.

Il est vital de noter que ces étapes sont seulement un modèle ; elles NE SONT PAS un label spécifique de la façon dont un système réel va fonctionner. En particulier, le modèle ne prend pas en compte deux concepts courants :

- (1) Dans de nombreux cas, la conversion en forme canonique avant le codage sera englobée dans le codeur lui-même, qui comprend directement les formats locaux. Par exemple, la convention locale de nouvelle ligne pour les corps de texte peut être prise en charge par le codeur lui-même avec la connaissance de ce qu'est ce format.
- (2) La sortie des codeurs peut devoir passer par une ou plusieurs étapes additionnelles avant qu'elle ne soit transmise comme message. À ce titre, la sortie du codeur peut n'être pas conforme aux formats spécifiés par la RFC 822. En particulier, une fois encore, il peut être approprié que la sortie du convertisseur soit exprimée en utilisant les conventions locales de nouvelle ligne plutôt que d'utiliser les délimiteurs CRLF standard de la RFC 822.

D'autres variations de mise en œuvre sont aussi concevables. L'aspect essentiel de cette discussion est que, en dépit de toutes les optimisations, fusion des étapes requises, ou insertion de traitements additionnels, les messages résultants doivent être cohérents avec ceux produits par le modèle décrit ci-dessus. Par exemple, un message avec les champs d'en-tête suivants :

```
Content-type: text/foo; charset=bar
Content-Transfer-Encoding: base64
```

doit d'abord être représenté dans la forme `text/foo`, puis (si nécessaire) représenté dans le jeu de caractères "bar", puis finalement transformé via l'algorithme base64 en une forme de messagerie sûre.

Note : Une certaine confusion a été causée par les systèmes qui représentent les messages dans un format qui utilise des conventions locales de passage à la ligne qui diffèrent de la convention CRLF de la RFC 822. Il est important de noter que ces formats ne sont pas canoniques RFC822/MIME. Ces formats sont plutôt des \*codages\* de la RFC 822, où les séquences de CRLF dans la représentation canonique du message sont codées comme convention locale de passage à la ligne. Noter que les formats qui codent les séquences CRLF comme, par exemple, LF ne sont pas capables de représenter les messages MIME qui contiennent des données binaires avec des octets LF qui ne font pas parties des séquences CRLF de séparation de lignes.

## 5. Résumé

Le présent document définit ce que signifie la conformité à MIME. Il précise aussi divers problèmes dont l'existence est connue dans le système de messagerie de l'Internet et la façon d'utiliser MIME pour les surmonter. Finalement, il décrit le modèle de codage canonique de MIME.

## 6. Considérations pour la sécurité

Les questions de sécurité sont exposées dans le second document de cet ensemble, la RFC 2046.

## 7. Adresse des auteurs

Pour des informations complémentaires, les auteurs de ce document pourront être plus facilement contactés via la messagerie Internet :

Ned Freed  
Innosoft International, Inc.  
1050 East Garvey Avenue South  
West Covina, CA 91790  
USA  
téléphone : +1 818 919 3600  
Fax : +1 818 919 3614  
mél : [ned@innosoft.com](mailto:ned@innosoft.com)

Nathaniel S. Borenstein  
First Virtual Holdings  
25 Washington Avenue  
Morristown, NJ 07960  
USA  
téléphone : +1 201 540 8967  
Fax : +1 201 993 3032  
mél : [nsb@nsb.fv.com](mailto:nsb@nsb.fv.com)

MIME est le résultat du travail du groupe de travail sur les extensions à la RFC 822 de l'Équipe d'ingénierie de l'Internet. Le président de ce groupe, Greg Vaudreuil, peut être joint à :

Gregory M. Vaudreuil

Octel Network Services  
 17080 Dallas Parkway  
 Dallas, TX 75248-1905  
 USAmél : [Greg.Vaudreuil@Octel.Com](mailto:Greg.Vaudreuil@Octel.Com)

## 8. Remerciements

Le présent document est le résultat des efforts collectifs d'un grand nombre de personnes, lors de plusieurs réunions de l'IETF, sur les listes de diffusion IETF-SMTP et IETF-822, et ailleurs. Bien que toute énumération semble condamnée à souffrir de criantes omissions, les personnes suivantes ont beaucoup contribué à cet effort :

Harald Tveit Alvestrand	Marc Andreessen	Randall Atkinson	Bob Braden
Philippe Brandon	Brian Capouch	Kevin Carosso	Uhhyung Choi
Peter Clitherow	Dave Collier-Brown	Cristian Constantinof	John Coonrod
Mark Crispin	Dave Crocker	Stephen Crocker	Terry Crowley
Walt Daniels	Jim Davis	Frank Dawson	Axel Deininger
Hitoshi Doi	Kevin Donnelly	Steve Dorner	Keith Edwards
Chris Eich	Dana S. Emery	Johnny Eriksson	Craig Everhart
Patrik Faltstrom	Erik E. Fair	Roger Fajman	Alain Fontaine
Martin Forssen	James M. Galvin	Stephen Gildea	Philip Gladstone
Thomas Gordon	Keld Simonsen	Terry Gray	Phill Gross
James Hamilton	David Herron	Mark Horton	Bruce Howard
Bill Janssen	Olle Jarnefors	Risto Kankkunen	Phil Kam
Alan Katz	Tim Kehres	Neil Katin	Steve Kille
Kyuho Kim	Anders Klemets	John Klensin	Valdis Kletniek
Jim Knowles	Stev Knowles	Bob Kummerfeld	Pekka Kytolaakso
Vincent Lau	Timo Lehtinen	Donald Lindsay	Stellan Lagerstrom
Warner Losh	Carlyn Lowery	Laurence Lundblade	Charles Lynn
John R. MacMillan	Larry Masinter	Rick McGowan	Michael J. McNerny
Leo McLaughlin	Goli Montaser-Kohsari	Tom Moore	John Gardiner Myers
Erik Naggum	Mark Needleman	Chris Newman	John Noerenberg
Mats Ohrman	Julian Onions	Michael Patton	David J. Pepper
Erik van der Poel	Blake C. Ramsdell	Christer Romson	Luc Rooijackers
Marshall T. Rose	Jonathan Rosenberg	Guido van Rossum	Jan Rynning
Harri Salminen	Michael Sanderson	Yutaka Sato	Markku Savela
Richard Alan Schafer	Masahiro Sekiguchi	Mark Sherman	Bob Smart
Peter Speck	Henry Spencer	Einar Stefferud	Michael Stein
Klaus Steinberger	Peter Svanberg	James Thompson	Steve Uhler
Stuart Vance	Peter Vanderbilt	Greg Vaudreuil	Ed Vielmetti
Larry W. Virden	Ryan Waldron	Rhys Weatherly	Jay Weber
Dave Wecker	Wally Wedel	Sven-Ove Westberg	Brian Wideen
John Wobus	Glenn Wright	Rayan Zachariassen	David Zimmerman

Les auteurs présentent leurs excuses pour toute omission de cette liste, qui est certainement involontaire.

## Appendice A – Exemple multipartie complexe

Ci après figurent les grandes lignes d'un message multipartie complexe. Ce message contient cinq parties qui sont à afficher à la suite : deux objets introductifs en clair, un message multipartie incorporé, un objet texte/enrichi, et un message texte encapsulé de clôture dans un jeu de caractères non ASCII. Le message multipartie incorporé lui-même contient deux objets à afficher en parallèle, une image et un fragment audio.

MIME-Version: 1.0  
 From: Nathaniel Borenstein <nsb@nsb.fv.com>  
 To: Ned Freed <ned@innosoft.com>  
 Date: Fri, 07 Oct 1994 16:15:05 -0700 (PDT)  
 Subject: Exemple multi-partie  
 Content-Type: multi-partie/mixte;  
 boundary=unique-boundary-1

C'est la zone de préambule d'un message multipartie. Les lecteurs de messagerie qui comprennent le format multipartie devraient ignorer ce préambule.

Si vous lisez ce texte, vous pourriez vouloir passer à un lecteur de messagerie qui sait comment afficher correctement les messages multipartie.

--unique-boundary-1

... Du texte apparaît ici ...

[Noter que le blanc entre la limite (*boundary*) et le début du texte dans cette partie signifie qu'aucun champ d'en-tête n' a été donné et que c'est du texte dans le jeu de caractères US-ASCII. Cela aurait pu être fait avec des mentions explicites comme dans la partie suivante.]

--unique-boundary-1

Content-type: text/plain; charset=US-ASCII

Cela aurait pu faire partie de la partie précédente, mais illustre la frappe explicite par opposition à la frappe implicite des parties de corps.

--unique-boundary-1

Content-Type: multipart/parallel; boundary=unique-boundary-2

--unique-boundary-2

Content-Type: audio/basic

Content-Transfer-Encoding: base64

... prennent place ici des données audio codées en base64 de format loi  $\mu$  sur un seul canal à 8000 Hz ...

--unique-boundary-2

Content-Type: image/jpeg

Content-Transfer-Encoding: base64

... des données d'image codée en base64 prennent place ici ...

--unique-boundary-2--

--unique-boundary-1

Content-type: text/enriched

C'est ***enriched.***comme défini dans la RFC 1896

Ce n'est pas **cool?**

--unique-boundary-1

Content-Type: message/rfc822

From: (boîte aux lettres en US-ASCII)

To: (adresse en US-ASCII)

Subject: (objet en US-ASCII)

Content-Type: Text/plain; charset=ISO-8859-1

Content-Transfer-Encoding: Quoted-printable

... du texte additionnel en ISO-8859-1 vient ici ...

--unique-boundary-1--

## Appendice B -- Changements par rapport aux RFC 1521, 1522 et 1590

Ces documents sont une révision des RFC 1521, 1522 et 1590. Pour l'agrément de ceux à qui les documents précédents sont familiers, cet appendice résume les changements par rapport à ces documents. Pour approfondir l'histoire, noter que



l'Appendice H de la RFC 1521 spécifiait en quoi ce document différait de son prédécesseur, la RFC 1341.

- (1) Le présent document a été complètement reformaté et partagé en plusieurs documents. Cela a été fait pour améliorer la qualité de la version texte en clair de ce document, qui est nécessaire comme copie de référence.
- (2) Le BNF qui décrit la structure globale des en-têtes d'objet MIME a été ajouté. Ce n'est qu'un changement documentaire – La syntaxe sous-jacente n'a été changée en aucune façon.
- (3) Le BNF spécifique des sept types de support en MIME a été supprimé. Ce BNF était incorrect, incomplet et non cohérent avec le BNF indépendant du type. Et comme le BNF indépendant du type spécifie déjà de façon complète la syntaxe des divers en-têtes MIME, le BNF spécifique du type était, en dernière analyse, totalement inutile et causait plus de problèmes qu'il n'en résolvait.
- (4) Le nom plus spécifique de jeu de caractère "US-ASCII" a remplacé l'utilisation du terme informel ASCII dans de nombreuses parties de ces documents.
- (5) Le concept informel de sous-type principal a été supprimé.
- (6) Le terme "objet" était utilisé de façon incohérente. La définition de ce terme a été précisée, ainsi que celle des termes qui s'y rapportent "corps", "partie de corps" et "entité", et leur utilisation a été corrigée en tant que de besoin.
- (7) Le BNF pour le type de support multipartie a été réarrangé pour rendre clair que le CRLF précédant le marqueur de limite fait en réalité partie du marqueur lui-même plutôt que de la partie de corps précédente.
- (8) La prose et le BNF qui décrivent le type de support multipartie ont été changés pour rendre clair que les parties de corps au sein d'un objet multipartie NE DOIVENT PAS contenir de lignes commençant par la chaîne de paramètres de la limite.
- (9) Dans les règles sur le réassemblage des entités MIME "message/partial", "Subject" est ajouté à la liste des en-têtes à tirer du message interne, et l'exemple est modifié pour préciser ce point.
- (10) Les fragmenteurs "message/partial" sont limités au partage des objets MIME aux seules limites de lignes.
- (11) Dans l'exposé sur le type application/postscript, un paragraphe supplémentaire a été ajouté pour avertir de possibles problèmes d'interopérabilité causés par l'incorporation de données binaires dans une entité MIME PostScript.
- (12) Ajout d'une note de précision aux règles de syntaxe de base pour le champ d'en-tête Content-Type pour préciser que les deux formes suivantes :  
Content-type: text/plain; charset=us-ascii (comment)  
Content-type: text/plain; charset="us-ascii"  
sont complètement équivalentes.
- (13) La phrase suivante a été retirée de l'exposé sur l'en-tête Version MIME : "Cependant, le logiciel conforme est invité à vérifier le numéro de version et au moins avertir l'utilisateur si une version MIME non reconnue est trouvée."
- (14) Correction d'une faute de frappe qui disait "application/corps-externe" au lieu de "message/corps-externe".
- (15) La définition d'un jeu de caractères a été réorganisée pour rendre les exigences plus claires.
- (16) La définition du type de support "image/gif" a été déplacée dans un autre document. Ce changement a été effectué à cause de conflits potentiels avec les règles de IETF sur la normalisation des technologies brevetées.
- (17) La définition de "7bit" et "8bit" a été resserrée afin que l'utilisation de CR et LF seuls ne puisse être acceptée que comme séquence de fin de ligne. Le document n'exige plus que les caractères NUL soient préservés, ce qui met MIME en ligne avec les mises en œuvre existantes.
- (18) La définition du texte canonique dans MIME a été resserrée de façon à ce que les coupures de ligne soient représentées par une séquence CRLF. Les caractères CR et LF ne sont pas admis en dehors de cette utilisation. La définition du codage quoted-printable a été modifiée en conséquence.
- (19) La définition du codage quoted-printable comporte maintenant un certain nombre de suggestions sur la façon dont les codeurs quoted-printable pourraient le mieux traiter le matériel improprement codé.

- (20) De texte a été ajouté pour préciser l'utilisation des codages de transfert "7bit", "8bit" et "binary" sur des entités multi-partie ou de message qui encapsulent des données en "8bit" ou "binary".
- (21) Dans la section sur la conformité à MIME, la prise en charge de "multi-partie/résumé" a été ajoutée à la liste des exigences pour le minimum de conformité à MIME. De plus, l'exigence de la prise en charge de "message/rfc822" a été renforcée pour préciser l'importance de la reconnaissance d'une structure récurrente.
- (22) Les diverses restrictions sur les sous-types de "message" sont maintenant spécifiées entièrement sous-type par sous-type.
- (23) La définition de "message/rfc822" a été changée pour indiquer qu'au moins un des en-têtes "From", "Subject", ou "Date" doit être présent.
- (24) Le traitement requis des sous-types non reconnus comme "application/flux-d'octet" a été rendu plus explicite à la fois dans les paragraphes de définition de type et dans les lignes directrices pour la conformité.
- (25) Les exemples qui utilisent text/richtext ont été changés en text/enriched.
- (26) La définition en BNF du sous-type a été changée pour rendre clair que dans un champ d'en-tête Type de contenu doit être utilisé un sous-type enregistré auprès de l'IANA ou un sous-type "X-" non standard.
- (27) Les types de support MIME qui sont simplement enregistrés pour utilisation et ceux qui sont normalisés par l'IETF sont maintenant distingués dans le BNF MIME.
- (28) Toutes les diverses procédures d'enregistrement MIME ont été largement révisées. Les procédures d'enregistrement de l'IANA pour les jeux de caractères ont été déplacées dans un autre document qui n'est pas inclus dans cet ensemble de documents.
- (29) L'utilisation des mécanismes d'échappement et de substitution que définissent ces documents dans les jeux de caractères US-ASCII et ISO-8859-X ont été précisés: De tels mécanismes ne devraient jamais être utilisés avec ces jeux de caractères et leurs effets en cas d'utilisation sont indéfinis.
- (30) La définition du type d'accès AFS pour message/corps-externe a été supprimée.
- (31) Le traitement de la combinaison de multi-partie/alternative et message/corps-externe est maintenant spécifiquement considéré.
- (32) Les questions de sécurité spécifiques de message/corps-externe sont maintenant exposées en détail.

## Appendice C -- Références

[ATK] Borenstein, Nathaniel S., Multimedia Applications Development with the Andrew Toolkit, Prentice-Hall, 1990.

[ISO-2022] International Standard -- Information Processing -- Character Code Structure and Extension Techniques, ISO/IEC 2022:1994, 4th ed.

[ISO-8859] International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets

- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987, 1st ed.
- Part 2: Latin Alphabet No. 2, ISO 8859-2:1987, 1st ed.
- Part 3: Latin Alphabet No. 3, ISO 8859-3:1988, 1st ed.
- Part 4: Latin Alphabet No. 4, ISO 8859-4:1988, 1st ed.
- Part 5: Latin/Cyrillic Alphabet, ISO 8859-5:1988, 1st ed.
- Part 6: Latin/Arabic Alphabet, ISO 8859-6:1987, 1st ed.
- Part 7: Latin/Greek Alphabet, ISO 8859-7:1987, 1st ed.
- Part 8: Latin/Hebrew Alphabet, ISO 8859-8:1988, 1st ed.
- Part 9: Latin Alphabet No. 5, ISO/IEC 8859-9:1989, 1st ed.
- Part 10: Latin Alphabet No. 6, ISO/IEC 8859-10:1992, 1st ed.

[ISO-646] International Standard -- Information Technology -- ISO 7-bit Coded Character Set for Information

Interchange, ISO 646:1991, 3rd ed..

- [JPEG] JPEG Draft Standard ISO 10918-1 CD.
- [MPEG] Video Coding Draft Standard ISO 11172 CD, ISO IEC/JTC1/SC2/WG11 (Motion Picture Experts Group), May, 1991.
- [PCM] CCITT, Fascicle III.4 - Recommendation G.711, "Pulse Code Modulation (PCM) of Voice Frequencies", Geneva, 1972.
- [POSTSCRIPT] Adobe Systems, Inc., PostScript Language Reference Manual, Addison-Wesley, 1985.
- [POSTSCRIPT2] Adobe Systems, Inc., PostScript Language Reference Manual, Addison-Wesley, Second Ed., 1990.
- [RFC0783] K. Sollins, "Protocole TFTP (révision 2)", juin 1981. (*rendue obsolète par la RFC1350*)
- [RFC0821] J. Postel, "Protocole simple de [transfert de messagerie](#)", STD 10, août 1982.
- [RFC0822] D. Crocker, "Norme pour le [format des messages de texte](#) de l'ARPA-Internet", STD 11, août 1982. (*Obsolète, voir RFC5322*)
- [RFC0934] M. Rose et E. Stefferud, "Proposition de norme pour l'encapsulation de message", janvier 1985.
- [RFC0959] J. Postel et J. Reynolds, "Protocole de [transfert de fichiers](#) (FTP)", STD 9, octobre 1985. (*MàJ par RFC7151*)
- [RFC1049] M. Sirbu, "Champ d'en-tête Type de contenu pour les messages Internet", mars 1988. (*Historique*)
- [RFC1154] D. Robinson et R. Ullmann, "Codage des champs d'en-tête pour les messages Internet", avril 1990. (*obs., voir 1505*)
- [RFC1341] N. Borenstein et N. Freed, "MIME (Extensions multi-objet de messagerie Internet) : mécanismes pour spécifier et décrire le format des corps de message Internet", juin 1992. (*Rendue obsolète par la RFC1521*)
- [RFC1342] K. Moore, "Représentation de texte non ASCII dans les en-têtes de message Internet", juin 1992. (*Obsolète, voir 1522*)
- [RFC1344] N. Borenstein, "Implications de MIME pour les routeurs de messagerie Internet", juin 1992. (*Information*)
- [RFC1345] K. Simonsen, "Mnémoniques de caractères & et jeux de caractères", juin 1992. (*Information*)
- [RFC1421] J. Linn, "Amélioration de la confidentialité pour la messagerie électronique Internet : Partie I : Chiffrement de message et procédures d'authentification", février 1993. (*Historique*)
- [RFC1422] S. Kent, "Amélioration de la confidentialité pour la messagerie électronique Internet : Partie II – Gestion de clés fondée sur le certificat", février 1993. (*Historique*)
- [RFC1423] D. Balenson, D., "Amélioration de la confidentialité pour la messagerie électronique Internet : Partie III -- Algorithmes, modes et identifiants", février 1993. (*Historique*)
- [RFC1424] B. Kaliski, "Amélioration de la confidentialité pour la messagerie électronique Internet : Partie IV – Certification des clés et services qui s'y rapportent", février 1993. (*Historique*)
- [RFC1521] N. Borenstien et N. Freed, "MIME (Extensions multi-usage de messagerie Internet) Partie 1 : Mécanismes pour spécifier et décrire le format des corps de message Internet", septembre 1993. (*Rendue obsolète par les RFC 2045 à 2049*)
- [RFC1522] K. Moore, "MIME (Extensions multi-usage de messagerie Internet) Partie 2 : Extensions d'en-tête de message pour texte non ASCII", septembre 1993. (*Rendue obsolète par les RFC 2045 à 2049*)
- [RFC1524] N. Borenstien, "Mécanisme de configuration d'agent d'utilisateur pour les informations en format de messagerie multimédia", septembre 1993. (*Information*)

- [RFC1543] J. Postel, "Instructions pour les auteurs de RFC", octobre 1993. (*Info., remplacé par 2223*)
- [RFC1556] H. Nussbacher, "Traitement des textes bidirectionnels dans MIME", RFC1556, décembre 1993. (*Info.*)
- [RFC1590] J. Postel, "Procédures d'enregistrement des types de support" mars 1994. (*Info., remplacée par les RFC2045-49*)
- [RFC1602] IAB, IESG, "Le processus des normes de l'Internet – révision 2", mars 1994. (*Info, remplacé par 2026*)
- [RFC1652] J. Klensin et autres, "[Extensions de service SMTP](#) pour transport MIME sur 8 bits", juillet 1994. (*Remplacée par RFC6152*) (*D.S.*)
- [RFC1700] J. Reynolds et J. Postel, "[Numéros alloués](#)", STD 2, octobre 1994. (*Historique, voir [www.iana.org](http://www.iana.org)*)
- [RFC1741] P. Faltstrom, D. Crocker, E. Fair, "Type de contenu MIME pour fichiers codés en BinHex", décembre 1994. (*Info*)
- [RFC1896] P. Resnick, A. Walker, "Le type de contenu texte/MIME enrichi", février 1996. (*Information*)
- [RFC2045] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 1 : Format des corps de message Internet", novembre 1996. (*D. S., MàJ par [2184](#), [2231](#), [5335](#).*)
- [RFC2046] N. Freed et N. Borenstein, "[Extensions de messagerie Internet](#) multi-objets (MIME) Partie 2 : Types de support", novembre 1996. (*D. S., MàJ par [2646](#), [3798](#), [5147](#), [6657](#), [8098](#)*)
- [RFC2047] K. Moore, "MIME ([Extensions de messagerie Internet](#) multi-objets) Partie trois : extensions d'en-tête de message pour texte non ASCII", novembre 1996. (*MàJ par [RFC2184](#), [RFC2231](#)*) (*D.S.*)
- [RFC2048] N. Freed, J. Klensin et J. Postel, "Extensions multi-objets de la messagerie Internet (MIME) Partie 4 : Procédures d'enregistrement", BCP 13, novembre 1996. (*Rendue obsolète par les RFC [4288-4289](#)*)
- [US-ASCII] "Coded Character Set -- 7-Bit American Standard Code for Information Interchange", ANSI X3.4-1986.
- [X400] Schicker, Pietro, "Message Handling Systems, X.400", Message Handling Systems and Distributed Applications, E. Stefferud, O-j. Jacobsen, and P. Schicker, eds., North-Holland, 1989, pp. 3-41.