Groupe de travail Réseau
Request for Comments : 5092
RFC rendue obsolète : 2192
RFC mise à jour : 4467
Catégorie : Sur la voie de la normalisation

A. Melnikov, éditeur, Isode Ltd.
C. Newman, Sun Microsystems
novembre 2007

Traduction Claude Brière de L'Isle

Schéma d'URL IMAP

Résumé
IMAP (RFC 3501) est un protocole intéressant pour accéder aux mémorisations distantes de message. Il fournit un mécanisme idéal pour accéder aux archives publiques et privées de listes de messagerie électronique et aux mémorisations partagées de messages. Ce document définit un schéma d'URL pour référencer des objets sur un serveur IMAP.

Le présent document rend obsolète la RFC 2192. Il met aussi à jour la RFC 4467.

Table des matières

1. Introduction

Le schéma d'URL IMAP est utilisé pour désigner les serveurs IMAP, les boîtes aux lettres de messagerie, les messages, les corps MIME [RFC2045], et les programmes de recherche sur les hôtes Internet accessibles en utilisant le protocole IMAP sur TCP.

L'URL IMAP suit la syntaxe commune de schéma Internet définie dans la [RFC3986]. Si :<port> est omis, l'accès par défaut est 143 (comme défini au paragraphe 2.1 de la [RFC3501]).

Un URL IMAP absolu prend une des formes suivantes :

imap://<iserver>[/]
imap://<iserver>/<enc-mailbox>[<uidvalidity>][?<enc-search>]
imap://<iserver>/<enc-mailbox>[<uidvalidity>]<iuid>[<isection>][<ipartial>][<iurlauth>]

La première forme est utilisée pour se référer à un serveur IMAP (voir la Section 4) ; la seconde forme se réfère au contenu d'une boîte aux lettres ou d'un ensemble de messages résultant d'une recherche (voir la Section 5) ; et la dernière forme se réfère à un message ou partie de message spécifique, et éventuellement une gamme d'octets dans cette partie (voir la Section 6). Si l'extension de la [RFC4467] est prise en charge, alors la dernière forme peut avoir le composant <iurlauth> (voir les détails au paragraphe 6.1).

Le composant <iserver> commun à tous les types d'URL IMAP absolus a la syntaxe suivante, exprimée en ABNF [RFC4234] :

[iuserinfo "@"] hôte [":" port]

Le composant <iserver> est le même que "authority" défini dans la [RFC3986]. La syntaxe et les utilisations de <iuserinfo> ("composant IMAP userinfo") sont décrits en détail à la Section 3. La syntaxe de <hôte> et <port> est décrite dans la [RFC3986].

2. Conventions utilisées dans ce document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Le présent document fait référence à de nombreuses productions de la [RFC3986]. Quand le document a besoin de souligner des différences spécifiques d'URI IMAP par rapport à la [RFC3986] (c'est-à-dire, pour des parties d'URI IMAP qui ont une syntaxe plus restrictive que les URI génériques) il utilise un i<foo> non terminal pour définir une version spécifique de IMAP du <foo> non terminal de la [RFC3986].

Noter que la syntaxe ABNF montrée à la Section 11 est normative. Les Sections 2 à 6 peuvent utiliser une syntaxe moins formelle qui ne correspond pas nécessairement à l'ABNF normatif montré à la Section 11. Si il y a des différences entre la syntaxe montrée dans les Sections 2 à 6 et la Section 11, c'est la syntaxe montrée à la Section 11 qui doit être traitée comme d'autorité. Les exigences non syntaxiques dans les Sections 2 à 6 sont, bien sûr, normatives.

3. Composant IMAP userinfo (iuserinfo)

Le composant <iuserinfo> se conforme à la syntaxe générique de <userinfo> définie dans la [RFC3986]. Il a la syntaxe suivante exprimée en ABNF [RFC4234] :

enc-user [iauth] / [enc-user] iauth

La signification des différentes parties est décrite dans les paragraphes qui suivent.

3.1 Portée de désignation de boîte aux lettres IMAP

La partie "enc-user" du composant "iuserinfo", si il est présent, note la portée de la désignation de boîte aux lettres. Si il est absent, l'URL IMAP peut seulement faire référence à des boîtes aux lettres avec des noms uniques au monde, c'est-à-dire,

des boîtes aux lettres avec des noms qui ne changent pas selon l'utilisateur que le client a authentifié comme serveur IMAP. Noter que toutes les mises en œuvre de IMAP ne prennent pas en charge des noms uniques au monde.

Par exemple, une boîte aux lettres personnelle décrite par l'URL suivant <imap://michael@exemple.org/INBOX> est très probablement différente d'une boîte aux lettres personnelle décrite par <imap://bester@exemple.org/INBOX>, même si les deux URL utilisent le même nom de boîte aux lettres.

3.2 Mécanisme de désignation et d'authentification d'utilisateur IMAP

Le composant userinfo (voir la [RFC3986]) d'un URI IMAP peut contenir un nom d'utilisateur IMAP (autrement dit une identité d'autorisation [RFC4442], "enc-user") et/ou un mécanisme d'authentification. (Noter que le "enc-user" définit aussi une portée de désignation de boîte aux lettres comme décrit au paragraphe 3.1). Le nom d'utilisateur IMAP et le mécanisme d'authentification sont utilisés dans les commandes "LOGIN" ou "AUTHENTICATE" après avoir établi la connexion au serveur IMAP.

Si ni un nom d'utilisateur, ni un mécanisme d'authentification n'est fourni, le client DOIT s'authentifier comme anonyme auprès du serveur. Si le serveur annonce la capacité IMAP AUTH=ANONYMOUS, le client DOIT utiliser la commande AUTHENTICATE avec le mécanisme SASL ANONYMOUS [RFC4505]. Si SASL ANONYMOUS n'est pas disponible, le nom d'utilisateur "anonymous" (insensible à la casse) est utilisé avec la commande "LOGIN" et l'adresse de messagerie Internet de l'utilisateur d'extrémité qui accède à la ressource est fournie comme mot de passe. Cette dernière option est donnée afin d'assurer l'interopérabilité avec les serveurs déployés.

Noter que, comme décrit dans la RFC 3501, la commande "LOGIN" NE DOIT PAS être utilisée quand le serveur IMAP annonce la capacité LOGINDISABLED *(connexion désactivée)*.

Un mécanisme d'authentification (comme utilisé par la commande IMAP AUTHENTICATE) peut être exprimé par l'ajout de ";AUTH=<enc-auth-type>" à la fin du nom d'utilisateur dans un URL IMAP. Quand un tel <enc-auth-type> est indiqué, le client DEVRAIT demander des accréditifs appropriés à ce mécanisme et utiliser la commande "AUTHENTICATE" au lieu de la commande "LOGIN". Si aucun nom d'utilisateur n'est spécifié, il DOIT en être obtenu un du mécanisme, ou demandé à l'utilisateur/configuration, comme approprié.

La chaîne ";AUTH=*" indique que le client DEVRAIT choisir un mécanisme d'authentification approprié. (Bien que le caractère '*' dans cet usage ne soit pas strictement un délimiteur, il est traité comme un sous-délimiteur [RFC3986] dans cette instance. Elle NE DOIT PAS être codée en pourcentage dans cet usage, car ";AUTH=%2A" ne va pas correspondre à cette production.) Il PEUT utiliser tout mécanisme mentionné dans la réponse à la commande CAPABILITY (ou au code de réponse CAPABILITY) ou utiliser un service de sécurité hors bande résultant en une connexion PREAUTH. Si aucun nom d'utilisateur n'est spécifié et si aucun mécanisme d'authentification approprié n'est disponible, le client DEVRAIT revenir à la connexion anonyme comme décrit ci-dessus. Le comportement prescrit dans ce paragraphe permet un URL qui accorde un accès en lecture-écriture aux utilisateurs autorisés et un accès anonyme en lecture seule aux autres utilisateurs.

Si un nom d'utilisateur est inclus sans mécanisme d'authentification, alors";AUTH=*" est supposé.

Les clients doivent faire attention quand ils résolvent un URL qui exige ou demande toute sorte d'authentification, car les URL peuvent facilement venir de sources qui ne sont pas de confiance. Fournir des accréditifs d'authentification au mauvais serveur peut compromettre la sécurité du compte de l'utilisateur ; donc dans ce cas, le programme qui résout l'URL devrait satisfaire au moins un des critères suivants :

1) L'URL vient d'une source de confiance, comme un serveur de référence que le client a validé et auquel il fait confiance en accord avec la politique du site. Noter que l'entrée de l'utilisateur de l'URL peut ou non compter comme source de confiance, selon le niveau d'expérience de l'utilisateur et la politique du site.

2) Une politique explicite de site local permet au client de se connecter au serveur dans l'URL. Par exemple, une compagnie exemple.com peut avoir pour politique de site de faire confiance à tous les noms de serveur IMAP qui se terminent par exemple.com, tandis qu'une telle politique serait peu sage pour exemple.edu où des étudiants aléatoires peuvent établir des serveurs IMAP.

3) L'utilisateur confirme que se connecter à ce nom de domaine avec les accréditifs et/ou le mécanisme spécifiés est permis. Par exemple, quand on utilise "LOGIN" ou SASL PLAIN avec la sécurité de la couche transport (TLS, *Transport Layer Security*) le client d'URL IMAP présente une boîte de dialogue "D'accord pour envoyer votre mot de passe au serveur "exemple.com" ? Soyez conscient que le propriétaire de exemple.com va être capable de réutiliser le mot de passe pour se connecter en votre nom à d'autres serveurs".

4) Un mécanisme est utilisé pour valider le serveur avant de passer des accréditifs de client potentiellement compromettants. Par exemple, un site a un certificat TLS désigné qui est utilisé pour certifier des certificats de serveur IMAP de confiance pour le site, et cela a été configuré explicitement dans le client d'URL IMAP. Un autre exemple est d'utiliser un mécanisme de simple authentification et couche de sécurité (SASL, *Simple Authentication and Security Layer*) comme DIGEST-MD5 [RFC2831], qui prend en charge l'authentification mutuelle.

5) Un mécanisme d'authentification est utilisé qui ne va pas révéler au serveur d'informations qui pourraient être utilisées pour compromettre de futures connexions. Des exemples sont SASL ANONYMOUS [RFC4505] ou GSSAPI [RFC4752].

Les URL qui n'incluent pas de nom d'utilisateur mais incluent un mécanisme d'authentification (";AUTH=<mech>") doivent être traités avec une extrême attention, car pour certains <mech> ils vont très probablement compromettre le compte principal de l'utilisateur. Un URL contenant ";AUTH=*" doit aussi être traité avec une extrême prudence car il pourrait retomber sur un mécanisme de sécurité plus faible. Finalement, il est déconseillé aux clients d'utiliser un mot de passe en clair comme solution de repli avec ";AUTH=*" sauf si la connexion a un chiffrement fort.

Un programme qui interprète des URL IMAP PEUT mettre en antémémoire des connexions ouvertes avec un serveur IMAP pour une réutilisation ultérieure. Si un URL contient un nom d'utilisateur, seules les connexions authentifiées comme étant celles de cet utilisateur peuvent être réutilisées. Si un URL ne contient pas de nom d'utilisateur ou de mécanisme d'authentification, seule une connexion anonyme peut être réutilisée.

Noter que si des caractères non sûrs ou réservés comme " " (espace) ou ";" sont présents dans le nom d'utilisateur ou mécanisme d'authentification, ils DOIVENT être codés en pourcentage comme décrit dans la [RFC3986].

3.3 Limitations de enc-user

Selon les paragraphes 3.1 et 3.2 du présent document, l'URI IMAP enc-user a deux objets :
1) Il fournit le contexte pour les chemins de boîte aux lettres spécifiques de l'utilisateur comme "INBOX" (paragraphe 3.1).
2) Il spécifie que la résolution de l'URL exige une connexion de cet utilisateur et limite l'usage de cet URL à ce seul utilisateur (paragraphe 3.2).

Une limitation évidente de l'utilisation du même champ pour les deux objets est que l'URL ne peut être résolu que par le propriétaire de la boîte aux lettres. Afin d'éviter cette restriction, les mises en œuvre devraient utiliser des noms de boîte aux lettres uniques au monde (voir le paragraphe 3.1) chaque fois que possible.

Note : il n'y a actuellement pas de moyen général dans IMAP d'apprendre un nom unique au monde pour une boîte aux lettres. Cependant, en cherchant le résultat de la commande NAMESPACE [RFC2342], il est possible de déterminer si le nom d'une boîte aux lettres est ou non unique au monde.

Le composant URLAUTH outrepasse le second objet de enc-user dans l'URI IMAP et par défaut permet à l'URI d'être résolu par tout utilisateur permis par l'identifiant <access>. URLAUTH et l'identifiant <access> sont décrits au paragraphe 6.1.

4. Serveur IMAP

Un URL IMAP qui se réfère à un serveur IMAP a la forme suivante :

imap://<iserver>[/]

Ce type d'URL est fréquemment utilisé pour décrire la localisation d'un serveur IMAP, à la fois dans les références et dans la configuration. Il peut facultativement contenir le composant <iuserinfo> (voir les Sections 3 et 11). Un programme qui interprète cet URL va produire l'ensemble standard de commandes qu'il utilise pour présenter une vue du contenu du serveur IMAP, comme il est visible à l'utilisateur décrit dans la partie "enc-user" du composant <iuserinfo>, si la partie "enc-user" est spécifiée.

5. Listes de messages

Un URL IMAP qui se réfère à une liste de messages a la forme suivante :

imap://<iserver>/<enc-maibox>[<uidvalidity>][?<enc-search>]

Le champ <enc-maibox> est utilisé comme argument de la commande IMAP4 "SELECT" ou "EXAMINE". Noter que si des caractères non sûrs ou réservés comme " " (espace), ";", ou "?" sont présents dans <enc-mailbox>, ils DOIVENT être codés en pourcentage comme décrit dans la [RFC3986].

Le champ <uidvalidity> est facultatif. Si il est présent, il DOIT être le même que la valeur du code de réponse IMAP4 UIDVALIDITY au moment où l'URL a été créé. Cela DOIT être utilisé par le programme qui interprète l'URL IMAP pour déterminer si l'URL est périmé. Si l'URL IMAP est périmé, alors le programme devrait se comporter comme si la boîte aux lettres correspondante n'existait pas.

Noter que le champ <uidvalidity> est un modificateur de <enc-mailbox>, c'est-à-dire, il est considéré comme faisant partie du dernier "composant" (comme utilisé dans la [RFC3986]) de la <enc-mailbox>. Ceci est significatif durant la résolution d'URI relatif.

Le champ "?<enc-search>" est facultatif. Si il n'est pas présent, le programme qui interprète l'URL va présenter le contenu entier de la boîte aux lettres.

Si le champ "?<enc-search>" est présent, le programme qui interprète l'URL devrait utiliser le contenu de ce champ comme argument suivant une commande IMAP4 SEARCH. Ces arguments vont probablement contenir des caractères non sûrs comme " " (espace) (qui sont probablement présents dans le <enc-search>). Si des caractères non sûrs sont présents, ils DOIVENT être codés en pourcentage comme décrit dans la [RFC3986].

Noter que les chaînes entre guillemets et les littéraux non synchronisateurs [RFC2088] sont permis dans le contenu de <enc-search> ; cependant, les littéraux de synchronisation ne sont pas permis, car leur présence signifierait que l'agent qui interprète des URL IMAP a besoin d'analyser un contenu de <enc-search>, de trouver tous les littéraux de synchronisation, et d'effectuer un traitement approprié de continuation de commande (voir le paragraphe 4.3 et la section 7 de la [RFC3501]).

6. Message ou partie de message spécifique

Un URL IMAP qui se réfère à un message ou partie de message spécifique a la forme suivante :

imap://<iserver>/<enc-maibox>[<uidvalidity>]<iuid>[<isection>][<ipartial>][<iurlauth>]

<enc-maibox> et [uidvalidity] sont comme défini dans la Section 5 ci-dessus.

Si <uidvalidity> est présent sous cette forme, il DEVRAIT être utilisé par le programme qui interprète l'URL pour déterminer si l'URL est périmé.

Le champ <iuid> se réfère à un identifiant univoque (UID, *Unique Identifier*) de message IMAP4, et il DEVRAIT être utilisé comme argument <set> à la commande IMAP4 "UID FETCH".

Le champ <isection> est facultatif. Si il n'est pas présent, l'URL se réfère au message Internet entier comme retourné par la commande IMAP "UID FETCH <uid> BODY.PEEK[]". Si il est présent, l'URL se réfère à l'objet retourné par une commande "UID FETCH <uid> BODY.PEEK[<section>]". Le type de l'objet peut être déterminé en utilisant une commande "UID FETCH <uid> BODYSTRUCTURE" et en localisant la partie appropriée dans la BODYSTRUCTURE *(structure de corps)* résultante. Noter que des caractères non sûrs dans [isection] DOIVENT être codés en pourcentage comme décrit dans la [RFC3986].

Le champ <ipartial> est facultatif. Si il est présent, il ajoute effectivement "<<partial-range>>" à la fin de la commande "UID FETCH BODY.PEEK[<section>]" construite comme décrit au paragraphe précédent. En d'autres termes, il permet au client de demander une gamme d'octets du message/partie de message.

Le champ <iurlauth> est décrit en détails au paragraphe 6.1.

6.1 URL URLAUTH autorisé

Les URL URLAUTH autorisés ne sont pris en charge que par un serveur IMAP qui annonce la capacité IMAP URLAUTH [RFC4467].

6.1.1 Concepts

6.1.1.1 URLAUTH

URLAUTH est un composant, ajouté à la fin d'un URL, qui porte l'autorisation d'accéder aux données visées par cet URL. Il contient un identifiant d'accès autorisé, un nom de mécanisme d'autorisation, et un jeton d'autorisation. Le jeton d'autorisation est généré à partir de l'URL, de l'identifiant d'accès autorisé, du nom du mécanisme d'autorisation, et d'une clé d'accès de boîte aux lettres.

Note : la présente spécification permet seulement le composant URLAUTH dans des URL IMAP décrivant un message ou partie de message.

6.1.1.2 Clé d'accès de boîte aux lettres

La clé d'accès de boîte aux lettres est une chaîne aléatoire imprévisible. Pour assurer l'imprévisibilité, la chaîne aléatoire avec au moins 128 bits d'entropie est générée par le logiciel ou matériel (pas par l'utilisateur humain).

Chaque utilisateur a un tableau des boîtes aux lettres et une clé d'accès de boîte aux lettres associée pour chaque boîte aux lettres. Par conséquent, la clé d'accès de boîte aux lettres est par utilisateur et par boîte aux lettres. En d'autres termes, deux utilisateurs qui partagent la même boîte aux lettres ont chacun une clé d'accès de boîte aux lettres différente pour cette boîte aux lettres, et chaque boîte aux lettres accédée par un seul utilisateur a aussi une clé d'accès de boîte aux lettres différente.

6.1.1.3 Identifiant d'accès autorisé

L'identifiant <access> autorisé restreint l'utilisation de l'URL URLAUTH autorisé à certains utilisateurs autorisés sur le serveur, comme décrit au paragraphe 6.1.2.

6.1.1.4 Mécanisme d'autorisation

Le mécanisme d'autorisation est l'algorithme par lequel le URLAUTH est généré et ensuite vérifié, en utilisant la clé d'accès de boîte aux lettres.

6.1.1.5 Jeton d'autorisation

Le jeton d'autorisation est une chaîne déterministe d'au moins 128 bits qu'une entité qui a connaissance de la clé d'accès secrète de boîte aux lettres et du mécanisme d'autorisation de l'URL peut utiliser pour vérifier l'URL.

6.1.2 Extensions URLAUTH d'URL IMAP

Un message ou partie de message URL IMAP spécifique peut facultativement contenir ";EXPIRE=<datetime>" et/ou ";URLAUTH=<access>:<mech>:<token>".

Quand ";EXPIRE=<datetime>" est utilisé, cela indique la date et heure au plus tard de validité de l'URL. Après cette date et heure, l'URL a expiré et les mises en œuvre de serveur DOIVENT rejeter l'URL. Si ";EXPIRE=<datetime>" n'est pas utilisé, l'URL n'a pas de date d'expiration, mais peut toujours être révoqué en utilisant la commande RESETKEY [RFC4467].

URLAUTH prend la forme ";URLAUTH=<access>:<mech>:<token>", et il DOIT être à la fin de l'URL. Il est composé de trois parties. La portion <access> donne les identifiants d'accès autorisé qui peuvent contraindre les opérations et les utilisateurs à qui l'utilisation de cet URL est permise. La portion <mech> donne le mécanisme d'autorisation utilisé par le serveur IMAP pour générer le jeton d'autorisation qui suit. La portion <token> donne le jeton d'autorisation, qui peut être généré en utilisant la commande GENURLAUTH [RFC4467].

Le préfixe d'identifiant d'accès "submit+", suivi par un identifiant d'utilisateur *(userid)* indique que seul un userid autorisé comme entité de soumission de message au nom de l'identifiant d'utilisateur spécifié a la permission d'utiliser cet URL. Le serveur IMAP ne valide pas l'identifiant d'utilisateur spécifié mais valide que la session IMAP a une identité d'autorisation qui est autorisée comme entité de soumission de message. L'entité de soumission de message autorisée DOIT valider l'identifiant d'utilisateur avant de contacter le serveur IMAP.

Le préfixe d'identifiant d'accès "user+", suivi par un identifiant d'utilisateur, indique que l'utilisation de cet URL est limitée aux sessions IMAP qui sont enregistrées comme dans l'identifiant d'utilisateur spécifié (c'est-à-dire, ont l'identité d'autorisation identique à cet identifiant d'utilisateur).

Note : Si un mécanisme SASL qui fournit des identifiants à la fois d'autorisation et d'authentication est utilisé pour s'authentifier au serveur IMAP, l'identifiant d'accès "user+" DOIT correspondre à l'identifiant d'autorisation. Si le mécanisme SASL ne peut pas transporter l'identifiant d'autorisation, l'identifiant d'accès "user+" DOIT correspondre à l'identifiant d'autorisation déduit de l'identifiant d'authentification (voir la [RFC4442]).

L'identifiant d'accès "authuser" indique que l'utilisation de cet URL est limitée aux sessions IMAP authentifiées qui sont enregistrées comme tout utilisateur non anonyme (c'est-à-dire, ont une identité d'autorisation comme utilisateur non anonyme) de ce serveur IMAP. Pour déclarer cela autrement : l'utilisation de ce type d'URL est interdite aux sessions IMAP anonymes, c'est-à-dire, toute commande URLFETCH qui contient ce type d'URL produit dans une session anonyme DOIT retourner NIL dans la réponse URLFETCH.

L'identifiant d'accès "anonymous" indique que l'utilisation de cet URL n'est pas restreinte par l'identitié d'autorisation de session ; c'est-à-dire, toute session IMAP dans l'état authentifié ou choisi (comme défini dans la [RFC3501]) y compris des sessions anonymes, peut produire une commande URLFETCH [RFC4467] en utilisant cet URL.

Le jeton d'autorisation est représenté comme une chaîne hexadécimale codée en ASCII, qui est utilisée pour autoriser l'URL. La longueur et le calcul du jeton d'autorisation dépendent du mécanisme utilisé, mais dans tous les cas, le jeton d'autorisation est d'au moins 128 bits (et donc d'au moins 32 chiffres héxadécimaux).

Exemple :

<imap://joe@exemple.com/INBOX/;uid=20/;section=1.2;urlauth=submit+fred:internal:91354a473744909de610943775f92038>

7. URL IMAP relatifs

Des URL IMAP relatifs sont permis et sont résolus en accord avec les règles définies dans la [RFC3986]. En particulier, dans des paramètres d'URL IMAP (comme ";uid=" ou ";section=") ils sont traités au titre du chemin normal par rapport à la résolution des URL relatifs.

La [RFC3986] définit quatres formes d'URL relatifs : <inetwork-path>, <iabsolute-path>, <irelative-path>, et <ipath-empty>. Leur syntaxe est définie à la Section 11.

Une référence relative qui commence par deux caractères barre oblique est appelée une référence de chemin réseau (<inetwork-path>) ; de telles références sont rarement utilisées, parce que dans la plupart des cas, elles peuvent être remplacées par un URL absolu équivalent. Une référence relative qui commence par un seul caractère barre oblique est appelée une référence de chemin absolu (<iabsolute-path> ; voir aussi le paragraphe 7.1). Une référence relative qui ne commence pas par un caractère barre oblique est appelée une référence de chemin relatif (<irelative-path> ; voir aussi le paragraphe 7.2). La quatrième forme est <ipath-empty>, qui est "la référence du même document" (voir le paragraphe 4.4 de la [RFC3986]).

Les observations suivantes sur les URL relatifs sont importantes :

L'élément de grammaire <iauth> (qui fait partie de <iuserinfo>, qui est à son tour, une partie de <iserver> ; voir la Section 3) est considéré faire partie du nom d'utilisateur pour les besoins de la résolution des URL IMAP relatifs. Cela signifie que sauf si une nouvelle spécification de nom d'utilisateur/serveur est incluse dans l'URL relatif, le mécanisme d'authentification est hérité de l'URL IMAP de base.

Les URL utilisent toujours "/" comme délimiteur de hiérarchie pour les besoins de la résolution des chemins dans les URL relatifs. IMAP4 permet l'utilisation de tout délimiteur de hiérarchie dans les noms de boîte aux lettres. Pour cette raison, les chemins relatifs de boîte aux lettres ne vont fonctionner que si la boîte aux lettres utilise "/" comme délimiteur de hiérarchie. Les URL relatifs peuvent être utilisés sur des boîtes aux lettres qui utilisent d'autres délimiteurs, mais dans ce cas, le nom entier de boîte aux lettres DOIT être spécifié dans l'URL relatif ou hérité comme un tout de l'URL de base.

Si un serveur IMAP permet que les noms de boîte aux lettres commencent par "./" ou "../", se terminent par "/." ou "/..", ou contiennent des séquences "/../" ou "/./", alors ces noms de boîte aux lettres DOIVENT être codés en pourcentage comme décrit dans la [RFC3986]. Autrement, ils seraient mal interprétés comme des segments séparés par des points (voir le paragraphe 3.3 de la [RFC3986]) qui subissent un traitement spécial durant le processus de résolution de chemin relatif.

7.1 Références de chemin absolu

Une référence relative qui commence par un seul caractère barre oblique est appelée une référence de chemin absolu (voir le paragraphe 4.2 de la [RFC3986]). Si un serveur IMAP permet des noms de boîte aux lettres avec une "/" en-tête, alors la "/" en tête DOIT être codée en pourcentage comme décrit dans la [RFC3986]. Autrement, l'URI de référence de chemin absolu produit va être mal interprété comme une référence de chemin réseau [RFC3986] décrite par le <inetwork-path> non terminal.

7.2 Références de chemin relatif

Une référence relative qui ne commence pas par un caractère barre oblique est apelée une référence de chemin relatif [RFC3986]. Les mises en œuvre NE DOIVENT PAS générer ou accepter de références de chemin relatif IMAP.

Voir aussi au paragraphe 4.2 de la [RFC3986] les restrictions sur les références de chemin relatif.

8. Considérations d'internationalisation

Le paragraphe 5.1.3 de la [RFC3501] IMAP4, inclut une convention pour le codage des caractères non US-ASCII dans les noms de boîte aux lettres IMAP. Parce que cette convention est réservée à IMAP, il est nécessaire de convertir le codage de IMAP en un qui puisse être plus facilement interprété par un programme d'affichage d'URL. Pour cette raison, le codage UTF-7 modifié d'IMAP pour les boîtes aux lettres DOIT être converti en UTF-8 [RFC3629]. Comme les octets de 8 bits ne sont pas permis dans les URL, les octets UTF-8 sont codés en pourcentage comme exigé par le paragraphe 2.1 de la spécification d'URL [RFC3986]. Un échantillon de code est inclus dans l'Appendice A pour montrer cette conversion.

Les noms d'utilisateur IMAP sont des chaînes UTF-8 et DOIVENT être codées en pourcentage comme exigé par le paragraphe 2.1 de la spécification d'URL [RFC3986].

Noter aussi que le critère IMAP SEARCH peut contenir des caractères non US-ASCII. Les octets de 8 bits de ces chaînes DOIVENT être codés en pourcentage comme exigé par le paragraphe 2.1 de la spécification d'URL [RFC3986].

9. Exemples

Les exemples suivants montrent comment un programme de client IMAP4 pourrait traduire divers URL IMAP4 en une série de commandes IMAP4. Les commandes envoyées du client au serveur sont précédées de "C:", et les réponses envoyées du serveur au client sont précédées de "S:".

L'URL <imap://minbari.exemple.org/gray-council;UIDVALIDITY=385759045/;UID=20/;PARTIAL=0.1024> peut résulter en les commandes de client et réponses de serveur suivantes :

<se connecter à minbari.exemple.org, port 143>
S: * OK [CAPABILITY IMAP4rev1 STARTTLS AUTH=ANONYMOUS] Bienvenue
C: A001 AUTHENTICATE ANONYMOUS
S: +
C: c2hlcmlkYW5AYmFieWxvbjUuZXhhbXBsZS5vcmc=
S: A001 OK Bienvenue à sheridan@babylon5.exemple.org
C: A002 SELECT gray-council
<le client vérifie que UIDVALIDITY correspond>
C: A003 UID FETCH 20 BODY.PEEK[]<0.1024>

L'URL : <imap://psicorp.exemple.org/~peter/%E6%97%A5%E6%9C%AC%E8%AA%9E/%E5%8F%B0%E5%8C%97> peut résulter en les commandes de client suivantes :

<se connecter à psicorp.exemple.org, port 143>

S: * OK [CAPABILITY IMAP4rev1 STARTTLS AUTH=CRAM-MD5] Bienvenue
C: A001 LOGIN ANONYMOUS bester@psycop.psicorp.exemple.org
C: A002 SELECT ~peter/&ZeVnLIqe-/&U,BTFw-
<les commandes que le client utilise pour voir le contenu de la boîte aux lettres>

L'URL : <imap://;AUTH=GSSAPI@minbari.exemple.org/gray-council/;uid=20/;section=1.2> peut résulter en les commandes de client suivantes :

<se connecter à minbari.exemple.org, port 143>
S: * OK Greetings
C: A000 CAPABILITY
S: * CAPABILITY IMAP4rev1 STARTTLS AUTH=GSSAPI
S: A000 OK
C: A001 AUTHENTICATE GSSAPI
<échange d'authentification>
C: A002 SELECT gray-council
C: A003 UID FETCH 20 BODY.PEEK[1.2]

Si l'URL relatif suivant est situé dans cette partie de corps : <;section=1.4> il pourrait en résulter les commandes de client suivantes :

C: A004 UID FETCH 20 (BODY.PEEK[1.2.MIME]
  BODY.PEEK[1.MIME]
  BODY.PEEK[HEADER.FIELDS (Content-Location)])
<le client cherche les en-têtes Content-Location dans le résultat. Si il n'y a aucun de ces en-têtes, il fait alors ce qui suit>
C: A005 UID FETCH 20 BODY.PEEK[1.4]

L'URL : <imap://;AUTH=*@minbari.exemple.org/gray%20council?SUBJECT%20shadows> pourrait résulter en ce qui suit :

<se connecter à minbari.exemple.org, port 143>
S: * OK Bienvenue
C: A001 CAPABILITY
S: * CAPABILITY IMAP4rev1 AUTH=DIGEST-MD5
S: A001 OK
C: A002 AUTHENTICATE DIGEST-MD5
<échange d'authentification>
S: A002 OK utilisateur lennier authentifié
C: A003 SELECT "gray council"
...
C: A004 SEARCH SUBJECT shadows
S: * SEARCH 8 10 13 14 15 16
S: A004 OK SEARCH completed
C: A005 FETCH 8,10,13:16 ALL
...

Dans l'exemple ci-dessus, le client a des choix qui dépendent de la mise en œuvre. Le mécanisme d'authentification pourrait être n'importe quoi, y compris PREAUTH. La commande finale FETCH pourrait aller chercher plus ou moins d'informations sur les messages, selon ce qu'on souhaite afficher à l'utilisateur.

L'URL : <imap://john;AUTH=*@minbari.exemple.org/babylon5/personel?charset%20UTF-8%20SUBJECT%20%7B14+%7D%0D%0A%D0%98%D0%B2%D0%B0%D0%BD%D0%BE%D0%B2%D0%B0> montre que des données en 8 bits peuvent être envoyées en utilisant des littéraux non synchronisants [RFC2088]. Il pourrait en résulter ce qui suit :

<se connecter à minbari.exemple.org, port 143>
S: * OK Coucou
C: A001 CAPABILITY
S: * CAPABILITY IMAP4rev1 LITERAL+ AUTH=DIGEST-MD5
S: A001 OK
C: A002 AUTHENTICATE DIGEST-MD5
<échange d'authentification>
S: A002 OK utilisateur john authentifié

C: A003 SELECT babylon5/personel
...
C: A004 SEARCH CHARSET UTF-8 SUBJECT {14+}
C: XXXXXXXXXXXXXX
S: * SEARCH 7 10 12
S: A004 OK SEARCH completed
C: A005 FETCH 7,10,12 ALL
...

où XXXXXXXXXXXXXX sont 14 octets de données codées en UTF-8 comme spécifié dans l'URL ci-dessus.

9.1 Exemples d'URL relatifs

La référence de chemin absolu suivante </foo/;UID=20/..> est la même que </foo>. C'est-à-dire que toutes deux se réfèrent à la boîte aux lettres "foo" située sur le serveur IMAP décrit par l'URI de base correspondant.

La référence de chemin relatif suivante <;UID=20> fait référence à un message avec l'UID dans la boîte aux lettres spécifiée par l'URI de base.

L'exemple de cas limite suivant montre que le ";UIDVALIDITY= modificateur" fait partie du nom de la boîte aux lettres pour autant que la résolution d'URI relatif est concernée :

<..;UIDVALIDITY=385759045/;UID=20>

Dans cet exemple, ".." n'est pas un segment séparé par des points de la [RFC3986].

10. Considérations sur la sécurité

Les considérations sur la sécurité discutées dans la spécification IMAP [RFC3501] et la spécification d'URI [RFC3986] sont pertinentes. Les considérations sur la sécurité relatives aux URL authentifiés sont discutées au paragraphe 3.2 du présent document.

De nombreux clients de messagerie électronique mémorisent le mot de passe en clair pour une utilisation ultérieure après la connexion à un serveur IMAP. Ces clients NE DOIVENT PAS utiliser un mot de passe mémorisé en réponse à un URL IMAP sans la permission explicite de cet utilisateur de fournir ce mot de passe au nom d'hôte spécifié.

Les clients qui résolvent des URL IMAP et souhaitent réaliser la confidentialité et/ou l'intégrité des données DEVRAIENT utiliser la commande STARTTLS (si elle est prise en charge par le serveur) avant de commencer l'authentification, ou utiliser un mécanisme SASL, comme GSSAPI, qui fournit une couche de sécurité de confidentialité.

10.1 Considération sur la sécurité spécifique de l'URL autorisé URLAUTH

L'identifiant d'accès "user+<userid>" limite la résolution de cet URL à un identifiant d'utilisateur particulier, tandis que l'identifiant d'accès "submit+<userid>" est plus général et exige simplement que la session soit autorisée par un utilisateur à qui le rôle "submit" a été accordé au sein du système d'authentification. L'utilisation de l'un de ces mécanismes limite la portée de l'URL. Un attaquant qui ne peut pas s'authentifier en utilisant les accréditifs appropriés ne peut pas utiliser l'URL.

Les identifiants d'accès "authuser" et "anonymous" n'ont pas ce niveau de protection. Ces identifiants d'accès sont principalement utiles pour l'exportation publique de données provenant d'un serveur IMAP, sans exiger qu'elles soient copiées d'un serveur de la Toile uu d'un serveur FTP anonyme.

La décision d'utiliser l'identifiant d'accès "authuser" devrait être prise avec précaution. Un identifiant d'accès "authuser"peut être utilisé par tout utilisateur autorisé du serveur IMAP ; donc, l'utilisation de cet identifiant d'accès devrait être limitée au contenu qui peut être divulgué à tout utilisateur autorisé du serveur IMAP.

La décision d'utiliser l'identifiant d'accès "anonymous" devrait être prise avec une extrême prudence. Un identifiant d'accès "anonymous" peut être utilisé par n'importe qui ; donc, l'utilisation de cet identifiant d'accès devrait être limitée au contenu qui peut être divulgué à tous.

11. ABNF pour le schéma d'URL IMAP

La syntaxe formelle est définie en utilisant l'ABNF [RFC4234], étendant les règles d'ABNF de la Section 9 de la [RFC3501]. Les éléments non définis ici se trouvent dans les [RFC3501], [RFC3986], [RFC4234], ou [RFC4466]. Les chaînes ne sont pas sensibles à la casse, et la libre insertion d'espaces blanches linéaires n'est pas permise.

```
sub-delims-sh = "!" / "$" / "'" / "(" / ")" / "*" / "+" / ","
;; les mêmes que les sous délimiteurs de la [RFC3986], mais sans ";", "&" et "=".

uchar = unreserved / sub-delims-sh / pct-encoded

achar = uchar / "&" / "="
;; les mêmes que 'unreserved / sub-delims / pct-encoded' dans la [RFC3986], mais ";" n'est pas permis.

bchar = achar / ":" / "@" / "/"

enc-auth-type = 1*achar                    ; version codée en pourcentage du "auth-type" de la [RFC3501]

enc-mailbox = 1*bchar                      ; version codée en pourcentage du "mailbox" de la [RFC3501]

enc-search = 1*bchar
;; version codée en pourcentage de "search-program" de la [RFC4466]. Noter que les littéraux IMAP4 ne peuvent pas être utilisés dans un "search-program", c'est-à-dire, seuls des littéraux entre guillemets ou non synchronisants (si le serveur prend en charge LITERAL+ [RFC2088]) sont permis.

enc-section = 1*bchar              ; version codée en pourcentage du "section-spec" de la [RFC3501].

enc-user = 1*achar                 ; version codée en pourcentage de l'identité d'autorisation ou "userid" de la [RFC3501].

imapurl = "imap://" iserver ipath-query          ; définit un URL IMAP absolu.

ipath-query = ["/" [ icommand ]]                 ; correspond au "path-abempty [ "?" query ]" dans la [RFC3986].
```

La syntaxe générique pour les URL relatifs est définie au paragraphe 4.2 de la [RFC3986]. Pour faciliter la mise en œuvre, la syntaxe d'URL IMAP relatif est définie ci-dessous :

```
imapurl-rel = inetwork-path
              / iabsolute-path
              / irelative-path
              / ipath-empty

inetwork-path   = "//" iserver ipath-query     ; correspond à '"//" authority path-abempty [ "?" query ]' dans la [RFC3986]

iabsolute-path  = "/" [ icommand ]             ; icommand, si il est présent, NE DOIT PAS commencer par '/'.
                                               ; correspond à "path-absolute [ "?" query ]" dans la [RFC3986]

irelative-path  = imessagelist / imsg-or-part  ; correspond à "path-noscheme [ "?" query ]" dans la[RFC3986]

imsg-ou-part = ( imailbox-ref "/" iuid-only ["/" isection-only] ["/" ipartial-only] ) / ( iuid-only ["/" isection-only]
               ["/" ipartial-only] ) / ( isection-only ["/" ipartial-only] ) / ipartial-only

ipath-empty = 0<pchar>                         ; caractères zéro. la référence au même document.
```

Les trois règles suivantes ne sont utilisées qu'en présence de l'extension IMAP [RFC4467] :

```
authimapurl = "imap://" iserver "/" imessagepart  ; la même que "imapurl" quand "[icommand]" est "imessagepart".

authimapurlfull = authimapurl iurlauth            ; la même que "imapurl" quand "[icommand]" est "imessagepart iurlauth"

authimapurlrump = authimapurl iurlauth-rump
```

```
enc-urlauth = 32*HEXDIG

iurlauth = iurlauth-rump iua-verifier

iua-verifier = ":" uauth-mechanism ":" enc-urlauth

iurlauth-rump = [expire] ";URLAUTH=" access

access = ("submit+" enc-user) / ("user+" enc-user) / "authuser" / "anonymous"

expire = ";EXPIRE=" date-time                    ; date-time est défini dans la [RFC3339]

uauth-mechanism = "INTERNAL" / 1*(ALPHA / DIGIT / "-" / ".")
                                  ; insensible à la casse. Les nouveaux mécanismes DOIVENT être enregistrés par l'IANA.

iauth  = ";AUTH=" ( "*" / enc-auth-type )

icommand = imessagelist / imessagepart [iurlauth]

imailbox-ref = enc-mailbox [uidvalidity]

imessagelist = imailbox-ref [ "?" enc-search ]          ; "enc-search" est le "query" de la [RFC3986].

imessagepart = imailbox-ref iuid [isection] [ipartial]

ipartial = "/" ipartial-only

ipartial-only = ";PARTIAL=" partial-range

isection = "/" isection-only

isection-only = ";SECTION=" enc-section

iserver = [iuserinfo "@"] host [ ":" port ]
        ; c'est le même que "authority" défini dans la [RFC3986]. Voir dans la [RFC3986] les définitions de "host" et "port".

iuid = "/" iuid-only

iuid-only = ";UID=" nz-number                    ; voir dans la [RFC3501] la définition de "nz-number".

iuserinfo = enc-user [iauth] / [enc-user] iauth     ; conforme à la syntaxe générique de "userinfo" définie dans la [RFC3986].

partial-range = number ["." nz-number]
          ; FETCH partiel. Le premier nombre est le décalage du premier octet, le second est la longueur du fragment.

uidvalidity = ";UIDVALIDITY=" nz-number      ; voir dans la [RFC3501] la définition de "nz-number".
```

12. Considérations relatives à l'IANA

L'IANA a mis à jour la définition de "imap" dans le registre "Schéma d'identifiant de ressource universel" pour pointer sur le présent document.

Le gabarit d'enregistrement (conformémeent à la [RFC4395]) est spécifié au paragraphe 12.1 du présent document.

12.1 Enregistrement par l'IANA du schéma d'URI imap:

Ce paragraphe donne les informations requises pour enregistrer le schéma d'URI imap:.

Nom de schéma d'URI : imap
Statut : permanent

Syntaxe de schéma d'URI : voir la Section 11 de la [RFC5092].
Sémantique du schéma d'URI : le schéma d'URI imap: est utilisé pour désigner les serveurs IMAP, les boîtes aux lettres, les messages, les corps MIME [RFC2045] et leurs parties, et les programmes de recherche sur les hôtes Internet accessibles en utilisant le protocole IMAP. Il n'y a pas de type MIME associé à cet URI.
Considérations de codage : voir la Section 8 de la [RFC5092].
Applications/protocoles qui utilisent ce nom de schéma d'URI : l'URI imap: est destiné à être utilisé par les applications qui pourraient avoir besoin d'accéder à une mémorisation de messages IMAP. De telles applications peuvent inclure (mais ne s'y limitent pas) des navigateurs de la Toile à capacité IMAP, des clients IMAP qui souhaitent accéder à une boîte aux lettres, à un message, ou éditer un message sur le serveur en utilisant la [RFC4469], des clients et serveurs de la [RFC4409] à qui il est demandé d'assembler un message complet à la soumission en utilisant la [RFC4468].
Considérations d'interopérabilité : un client IMAP de messagerie Netscape largement déployé (et éventuellement Mozilla, Thunderbird, Seamonkey) utilise un schéma imap: différent en interne.
Considérations de sécurité : voir la Section Considérations sur la sécurité (Section 10) de la [RFC5092].
Contact: Alexey Melnikov <alexey.melnikov@isode.com>
Auteur/contrôleur des changements : IESG
Références : [RFC5092] et [RFC3501].

13. Références

13.1 Références normatives

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

[RFC2088] J. Myers, "Littéraux IMAP4 sans synchronisation", janvier 1997. (*MàJ par* RFC4466) *(P.S. Remplacée par* RFC7888)

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC2342] M. Gahrns, C. Newman, "Espace de noms IMAP4", mai 1998. (*MàJ par* RFC4466) *(P.S.)*

[RFC3339] G. Klyne, C. Newman, "La date et l'heure sur l'Internet : horodatages", juillet 2002. *(P.S.)*

[RFC3501] M. Crispin, "Protocole d'accès au message Internet - version 4rev1", mars 2003. *(P.S. ; MàJ par* RFC4466, 4469, 4551, 5032, 5182, 7817, 8314, 8437, 8474 ; *remplacée par la* RFC9051

[RFC3629] F. Yergeau, "UTF-8, un format de transformation de la norme ISO 10646", STD 63, novembre 2003.

[RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "Identifiant de ressource uniforme (URI) : Syntaxe générique", STD 66, janvier 2005. *(P.S. ; MàJ par* RFC8820)

[RFC4234] D. Crocker et P. Overell, "BNF augmenté pour les spécifications de syntaxe : ABNF", octobre 2005. *(Remplace* RFC2234, *remplacée par* RFC5234)

[RFC4466] A. Melnikov, C. Daboo, "Récapitulation des extensions à l'ABNF pour IMAP4", avril 2006. *(P.S.)*

[RFC4467] M. Crispin, "Protocole d'accès au message Internet (IMAP) - Extension URLAUTH", mai 2006. (*P.S. ; MàJ par* RFC5092)

[RFC4505] K. Zeilenga, "Mécanisme anonyme d'authentification simple et de couche de sécurité (SASL)", juin 2006. *(P.S.)*

13.2 Références pour information

[RFC2831] P. Leach et C. Newman, "Utilisation de l'authentification par résumé comme mécanisme SASL", mai 2000. (*Historique, voir* RFC6331)

[RFC4395] T. Hansen et autres, "Lignes directrices et procédures d'enregistrement pour les nouveaux schémas d'URI", février 2006. (*Remplace* RFC2717, RFC2718) (BCP0115)

[RFC4409] R. Gellens, J. Klensin, "Soumission du message de messagerie électronique", avril 2006. (*Remplacé par la* RFC6409 STD072*)*

[RFC4442] S. Fries, H. Tschofenig, "Amorçage de l'authentification tolérante aux pertes de flux à synchronisation efficace (TESLA)", mars 2006. *(P.S.)*

[RFC4468] C. Newman, "Extension BURL de soumission de message", mai 2006. (*MàJ* RFC3463) (*MàJ par* RFC5248) *(P.S.)*

[RFC4469] P. Resnick, "Extension CATENATE au protocole d'accès au message Internet (IMAP)", avril 2006. *(P.S.)*

[RFC4752] A. Melnikov, éd., "Méthode d'utilisation de l'interface de programme d'application de service générique de sécurité (GSS-API) Kerberos v5 dans le mécanisme d'authentification simple et couche de sécurité (SASL)", novembre 2006.

Appendice A. Échantillon de code

Voici un échantillon de code de source C pour convertir les chemins d'URL en noms de boîte aux lettres IMAP, en tenant compte de la transposition entre l'UTF-7 modifié de IMAP [RFC3501] et l'UTF-8 codé en hexadécimal, qui est plus approprié pour les URL. Ce code n'a pas fait l'objet d'essais rigoureux et ne se comporte pas nécessairement de façon raisonnable avec une entrée invalide, mais il devrait servir d'exemple utile. Ce code convertit juste la portion boîte aux lettres de l'URL et ne traite pas les paramètres, interrogations, ou composants de serveur de l'URL.


```
#include <stdio.h>
#include <string.h>

/* tableau de recherche hexadécimal */
static const char hex[] = "0123456789ABCDEF";

#définir XX 127
/*
 * Tableau pour décoder l'hexadécimal en codage en pourcentage
 */
static const char index_hex[256] = {
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
0, 1, 2, 3,  4, 5, 6, 7,  8, 9,XX,XX, XX,XX,XX,XX,
XX,10,11,12, 13,14,15,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,10,11,12, 13,14,15,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
```

```
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX, XX,XX,XX,XX,
};
#définit HEXCHAR(c)  (index_hex[(unsigned char)(c)])

/* "gen-delims" excluant "/" mais incluant "%" */
#définir GENERAL_DELIMS_NO_SLASH     ":?#[]@" "%"

/* "gen-delims" (excluant "/", mais incluant "%") plus le sous ensemble de "sub-delims" */
#définir GENERAL_UNSAFE_NO_SLASH     GENERAL_DELIMS_NO_SLASH ";&=+"
#définir OTHER_UNSAFE  " \"<>\\^`{|}"

/* Caractères imprimables d'URL non sûrs */
static const char mailbox_url_unsafe[] = GENERAL_UNSAFE_NO_SLASH OTHER_UNSAFE;

/*alphabet UTF7 modifié en base64 */
static const char base64chars[] =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+,";

#définir UNDEFINED 64

/* définitions UTF16 */
#définir UTF16MASK  0x03FFUL
#définir UTF16SHIFT  10
#définir UTF16BASE  0x10000UL
#définir UTF16HIGHSTART  0xD800UL
#définir UTF16HIGHEND  0xDBFFUL
#définir UTF16LOSTART  0xDC00UL
#définir UTF16LOEND  0xDFFFUL

/* Convertir une boîte aux lettres IMAP en un chemin d'UmailboxRL de destination doit avoir en gros 4 fois l'espace de
mémorisation du codage Hex de source et peut tripler la taille de l'entrée UTF-7 qui peut être légèrement plus dense que
UTF-8 (dans le pire des cas : 8 octets UTF-7 deviennent 9 octets UTF-8)
 */
void MailboxToURL(char *dst, char *src)
{
   unsigned char c, i, bitcount;
   unsigned long ucs4, utf16, bitbuf;
   unsigned char base64[256], utf8[6];

/* initialiser le tableau de décodage base64 modifié */

   memset(base64, UNDEFINED, sizeof (base64));
   for (i = 0; i < sizeof (base64chars); ++i) {
    base64[(int) base64chars[i]] = i;
   }

/* boucle jusqu'à la fin de lla chaîne */
   while (*src != '\0') {
    c = *src++;
/* traiter les caractères littéraux et &- */
    if (c != '&' || *src == '-') {
/* note : Il n'y a pas de caractère "URL sûr" après le '~' */
       if (c < ' ' || c > '~' || strchr(mailbox_url_unsafe, c) != NULL) {
/* coder en hex si nécessaire */
         dst[0] = '%';
         dst[1] = hex[c >> 4];
         dst[2] = hex[c & 0x0f];
         dst += 3;
```

```
        } else {
/* coder littéralement */
         *dst++ = c;
        }
/* sauter le '-' si c'est une séquence de &- */
        if (c == '&') ++src;

      } else {
/* convertir l'UTF-7 modifiéé en  -> UTF-16 -> UCS-4 -> UTF-8 -> HEX */
        bitbuf = 0;
        bitcount = 0;
        ucs4 = 0;
        while ((c = base64[(unsigned char) *src]) != UNDEFINED) {
         ++src;
         bitbuf = (bitbuf << 6) | c;
         bitcount += 6;
/* assez de bits pour un caractère UTF-16 ? */
         if (bitcount >= 16) {
            bitcount -= 16;
            utf16 = (bitcount ? bitbuf >> bitcount : bitbuf) & 0xffff;
/* convertir UTF16 en UCS4 */
            if
                (utf16 >= UTF16HIGHSTART && utf16 <= UTF16HIGHEND) {
             ucs4 = (utf16 - UTF16HIGHSTART) << UTF16SHIFT;
             continue;
            } else if
                (utf16 >= UTF16LOSTART && utf16 <= UTF16LOEND) {
             ucs4 += utf16 - UTF16LOSTART + UTF16BASE;
            } else {
             ucs4 = utf16;
            }
/* convertir la gamme de UCS4 UTF-16 en UTF-8 */
            if (ucs4 <= 0x7fUL) {
             utf8[0] = (unsigned char) ucs4;
             i = 1;
            } else if (ucs4 <= 0x7ffUL) {
             utf8[0] = 0xc0 | (unsigned char) (ucs4 >> 6);
             utf8[1] = 0x80 | (unsigned char) (ucs4 & 0x3f);
             i = 2;
            } else if (ucs4 <= 0xffffUL) {
             utf8[0] = 0xe0 | (unsigned char) (ucs4 >> 12);
             utf8[1] = 0x80 | (unsigned char) ((ucs4 >> 6) & 0x3f);
             utf8[2] = 0x80 | (unsigned char) (ucs4 & 0x3f);
             i = 3;
            } else {
             utf8[0] = 0xf0 | (unsigned char) (ucs4 >> 18);
             utf8[1] = 0x80 | (unsigned char) ((ucs4 >> 12) & 0x3f);
             utf8[2] = 0x80 | (unsigned char) ((ucs4 >> 6) & 0x3f);
             utf8[3] = 0x80 | (unsigned char) (ucs4 & 0x3f);
             i = 4;
            }
/* convertir utf8 en hex */
            for (c = 0; c < i; ++c) {
             dst[0] = '%';
             dst[1] = hex[utf8[c] >> 4];
             dst[2] = hex[utf8[c] & 0x0f];
             dst += 3;
            }
         }
        }
/* sauter le '-' en queue dans le codage UTF-7 modifié */
        if (*src == '-') ++src;
```

```
    }
   }
/* terminer la chaîne de destination */
   *dst = '\0';
}

/* Convertir le chemin d'URL UTF-8 codé en hexadécimal en boîte aux lettres IMAP de destination en UTF-7 modifié
devrait être d'environ deux fois la longueur de la source pour traiter les URL non codés en hexadécimal */

int URLtoMailbox(char *dst, char *src)
{
   unsigned int utf8pos = 0;
   unsigned int utf8total, i, c, utf7mode, bitstogo, utf16flag;
   unsigned long ucs4 = 0, bitbuf = 0;

   utf7mode = 0;              /* le résultat UTF7 est-il actuellement en mode base64 ? */
   utf8total = 0;             /* de combien d'octets est l'entrée actuelle de UTF-8 char 0 == entre caractères */
   bitstogo = 0;              /* bits qui doivent être codés en base64 ; si bitstogo != 0 alors utf7mode == 1 */
   while ((c = (unsigned char)*src) != '\0') {
    ++src;
/* défaire le codage hexadécimal */
    if (c == '%' && src[0] != '\0' && src[1] != '\0') {
       c = HEXCHAR(src[0]);
       i = HEXCHAR(src[1]);
       if (c == XX || i == XX) {
          return 0;
       } else {
          c = (char)((c << 4) | i);
       }
       src += 2;
    }
/* est-ce un caractère normal ? */
    if (c >= ' ' && c <= '~') {
/* passer au mode UTF-7 */
       if (utf7mode) {
        if (bitstogo) {
        *dst++ = base64chars[(bitbuf << (6 - bitstogo)) & 0x3F];
        }
        *dst++ = '-';
        utf7mode = 0;
        bitstogo = bitbuf = 0;
       }
       *dst++ = c;
/* coder '&' comme '&-' */
       if (c == '&') {
        *dst++ = '-';
       }
       continue;
    }
/* passer au mode UTF-7 */
    if (!utf7mode) {
       *dst++ = '&';
       utf7mode = 1;
    }
/* coder les caractères US-ASCII comme eux-mêmes */
    if (c < 0x80) {
       ucs4 = c;
       utf8total = 1;
    } else if (utf8total) {
/* c'est un octet suivant d'un caractère multi octets */
/* sauvegarder les bits UTF8 en UCS4 */
```

```
        ucs4 = (ucs4 << 6) | (c & 0x3FUL);
        if (++utf8pos < utf8total) {
         continue;
        }
      } else {
/* c'est le premier octet d'un caractère multi octets */
        utf8pos = 1;
        if (c < 0xE0) {
         utf8total = 2;
         ucs4 = c & 0x1F;
        } else if (c < 0xF0) {
         utf8total = 3;
         ucs4 = c & 0x0F;
        } else {
/* Note : ne peut pas convertir des séquences UTF8 plus longue que 4 */
         utf8total = 4;
         ucs4 = c & 0x03;
        }
        continue;
      }
/* Terminé avec le caractère UTF-8. S'assurer qu'il n'a pas une séquence trop longue. Si elle l'est, retourner un échec. */
      if ((ucs4 < 0x80 && utf8total > 1) ||
        (ucs4 < 0x0800 && utf8total > 2) ||
        (ucs4 < 0x00010000 && utf8total > 3) ||
        (ucs4 < 0x00200000 && utf8total > 4) ||
        (ucs4 < 0x04000000 && utf8total > 5) ||
        (ucs4 < 0x80000000 && utf8total > 6)) {
        return 0;
      }
/* boucle pour partager ucs4 en deux caractères utf16 si nécessaire */
      utf8total = 0;
      do {
        if (ucs4 >= UTF16BASE) {
            ucs4 -= UTF16BASE;
         bitbuf = (bitbuf << 16) | ((ucs4 >> UTF16SHIFT)
                    + UTF16HIGHSTART);
         ucs4 = (ucs4 & UTF16MASK) + UTF16LOSTART;
         utf16flag = 1;
        } else {
         bitbuf = (bitbuf << 16) | ucs4;
         utf16flag = 0;
        }
        bitstogo += 16;
/* rejeter le base64 */
        while (bitstogo >= 6) {
         bitstogo -= 6;
         *dst++ = base64chars[(bitstogo ? (bitbuf >> bitstogo) : bitbuf) & 0x3F];
        }
      } while (utf16flag);
    }
/* si on est en mode UTF-7, finir en ASCII */
    if (utf7mode) {
      if (bitstogo) {
        *dst++ = base64chars[(bitbuf << (6 - bitstogo)) & 0x3F];
      }
      *dst++ = '-';
    }
/* attacher la chaîne */
    *dst = '\0';
    return 1;
}
```

Appendice B. Liste des changements par rapport à la RFC 2192

Mise à jour des usuels, liste des éditeurs, etc.
Mise à jour des références.
Mise à jour de l'ABNF pour ne plus utiliser "_" et utiliser SP à la place de SPACE, etc.
Mise à jour des domaines d'exemple pour utiliser exemple.org.
Correction d'une erreur de l'ABNF dans le non terminal "imessagelist".
Mise à jour de l'ABNF, du fait de changements dans les RFC 3501, RFC 3986, et RFC 4466,.
Remplacement du non terminal "iuserauth" par <iuserinfo>.
Précisé que le composant userinfo décrit à la fois l'identité d'autorisation et la portée du nom de boîte aux lettres.
Permet des littéraux non synchronisants dans "enc-search".
Ajout du spécificateur "ipartial" qui note un FETCH partiel.
Emprunt du texte sur URLAUTH de la RFC 4467 pour ce document.
Mise à jour de l'ABNF pour le serveur entier pour permettre des "/" en queue manquantes (par exemple, "imap://imap.exemple.com" est maintenant valide et est le même que "imap://imap.exemple.com/").
Précisé comment les références de chemin relatif sont construites.
Ajout de plus d'exemples montrant des références de chemin relatif.
Ajout de règles pour les URL relatifs et restructuration de l'ABNF qui en découle.
Suppression du texte sur l'utilisation des URL relatifs dans MHTML.
Ajout d'exemples montrant les considérations de sécurité lors de la résolution des URL.
Recommande l'usage de la couche de sécurié STARTTLS/SASL pour protéeger les données confidentielles.
Suppression des avis sur la réutilisation de connexion qui étaient incorrects.
Suppression des URL de référence à une liste de boîte aux lettres, car cette caractéristique n'a connu aucun déploiement.
Précisé que le nom d'utilisateur "anonymous" est insensible à la casse.

Appendice C. Liste des changements par rapport à la RFC 4467

<mechanism> a été renommé <uauth-mechanism>. L'ABNF a été restructuré.

Appendice D. Remerciements

Le texte qui décrit URLAUTH a été pris à la [RFC4467] de Mark Crispin.

Stephane H. Maes a contribué à certaines idées de ce document ; il a aussi co-édité les versions antérieures de ce document.

Les éditeurs tiennent à remercier Mark Crispin, Ken Murchison, Ted Hardie, Zoltan Ordogh, Dave Cridland, Kjetil Torgrim Homme, Lisa Dusseault, Spencer Dawkins, Filip Navara, Shawn M. Emery, Sam Hartman, Russ Housley, et Lars Eggert pour le temps qu'ils on consacré à la relecture de ce document et/ou pour les commentaires fournis.

Adresse des auteurs

Chris Newman
Sun Microsystems
3401 Centrelake Dr., Suite 410
Ontario, CA 91761
USA
mél : chris.newman@sun.com

Alexey Melnikov
Isode Limited
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX, UK
mél : Alexey.Melnikov@isode.com
URI : http://www.melnikov.ca/

Déclaration complète de droits de reproduction

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

Propriété intellectuelle

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