# RFC 7622 : Extensible Messaging and Presence Protocol (XMPP): Address Format

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 septembre 2015

Date de publication du RFC : Septembre 2015

https://www.bortzmeyer.org/7622.html

---

Ce RFC normalise le format des **adresses** du protocole de messagerie instantanée XMPP, protocole également connu sous son ancien nom de Jabber. Il remplace l'ancien RFC 6122[1] avec notamment un changement important sur l'internationalisation de ces adresses, le passage au nouveau système PRECIS (décrit dans le RFC 7564), qui remplace stringprep. Les adresses XMPP peuvent en effet être entièrement en Unicode.

Le protocole XMPP est le standard actuel de messagerie instantanée de l'IETF (il est normalisé dans le RFC 6120). Les utilisateurs sont identifiés par une **adresse** également connue, pour des raisons historiques, sous le nom de **JID** ("*Jabber IDentifier*"). Ces adresses ont une forme qui ressemble aux adresses de courrier électronique mais elles sont en fait complètement différentes. Ainsi, j'ai personnellement deux adresses XMPP, bortzmeyer@gmail.com (le service de messagerie Google Talk utilise en effet la norme XMPP), qui me sert surtout pour la distraction, et bortzmeyer@dns-oarc.net qui me sert pour le travail (j'utilise aussi le service XMPP de la Quadrature du Net <https://jabber.lqdn.fr/> mais plus rarement). Mais rien ne dit qu'un courrier envoyé à ces adresses fonctionnera : adresses XMPP et de courrier vivent dans des mondes différents.

Le format des adresses XMPP était spécifié à l'origine dans le document XEP-0029 <http://xmpp.org/extensions/xep-0029.html>. Puis il a été normalisé dans la section 3 du RFC 3920. La norme XMPP ayant subi une refonte complète, qui a mené entre autres à un nouveau RFC, le RFC 6120, se posait la question du format d'adresses, dont l'internationalisation, compte tenu de la nouvelle norme IDN <https://www.bortzmeyer.org/idnabis.html>, suscitait des débats. Ces débats n'étant pas terminés, la décision a été prise de sortir le format d'adresses du RFC principal, et d'en faire le RFC 6122,

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6122.txt

jusqu'à ce qu'un consensus apparaisse sur le format « définitif ». Ce format final est celui de notre RFC 7622, qui utilise pour Unicode le cadre PRECIS, normalisé dans le RFC 8264.

Donc, quels sont les points essentiels des adresses XMPP (section 3) ? Une adresse (ou JID) identifie une entité (un humain mais peut-être aussi un programme automatique). Elle comprend trois parties (dont une seule, le domaine, est obligatoire), la **partie locale**, le **domaine** et une **ressource**. Ainsi, dans `bortzmeyer@gmail.com/Home`, `bortzmeyer` est la partie locale, `gmail.com` le domaine et `Home` la ressource. Le @ et le / servent de séparateurs. Chacune des trois parties peut être composée de caractères Unicode, qui doivent être encodés en UTF-8. La partie locale doit être canonicalisée avec PRECIS et son profil `UsernameCaseMapped` (voir RFC 8265), le domaine avec les méthodes IDN des RFC 5890 et RFC 5892. Seul le domaine est obligatoire et `bot.example.org` est donc un JID valide.

Les adresses XMPP (ou JID) sont souvent représentées sous forme d'un IRI, selon le RFC 5122. En gros, un IRI XMPP est un JID préfixé par `xmpp:`. Mais ces IRI ne sont pas utilisés par le protocole XMPP. Dans les champs `to` et `from` des strophes ("*stanzas*") XMPP, on ne trouve que des JID (sans le `xmpp:` devant). Les IRI ne servent que dans les liens dans les pages Web, comme `<xmpp:bortzmeyer@gmail.com>` (attention, ce lien ne marchera pas avec tous les navigateurs).

Le RFC détaille ensuite chaque partie. Le domaine, seule partie obligatoire, est dans la section 3.2. On peut le voir comme une identification du service auquel on se connecte (`gmail.com` = Google Talk), et la création de comptes et l'authentification se font typiquement en fonction de ce service. En théorie, une adresse IP est acceptable pour cette partie mais, en pratique, c'est toujours un FQDN. Ce nom de domaine peut être un IDN donc `instantanée.nœud.example` est un nom acceptable pour former un JID. Auquel cas, ce doit être un IDN légal (ce qui veut dire que toute chaîne de caractères Unicode n'est pas forcément un nom légal pour la partie domaine d'un JID). À noter que XMPP n'utilise **pas** l'encodage en ASCII des IDN (le Punycode).

Et la partie locale, celle avant le @ ? La section 3.3 la couvre. Elle est optionnelle et identifie en général une personne (mais peut aussi indiquer un programme ou bien un salon de conversation à plusieurs). Si elle est en Unicode, elle doit être canonicalisée avec le profil `UsernameCaseMapped` de PRECIS, profil décrit dans le RFC 8265. Cette partie locale est donc insensible à la casse. Certains caractères normalement autorisés par ce profil sont explicitement interdits en XMPP comme les séparateurs / ou @ mais aussi comme " ou ¡. (Voir XEP-0106 `<http://xmpp.org/extensions/xep-0106.html>` pour un mécanisme d'échappement permettant de mettre quand même ces caractères.)

Quant à la ressource, troisième partie du JID (après le /), également optionnelle, elle sert à distinguer plusieurs sessions par le même utilisateur (par exemple `/Home` et `/Office`). La section 3.4 la décrit en détail. Également en Unicode, elle est canonicalisée par le profil `OpaqueString` de la classe `FreeformClass` de PRECIS (RFC 8264, section 4.3). Elle n'a pas de structure officielle. Un / dans une ressource (par exemple `example.com/foo/bar`) n'implique donc pas une hiérarchie. De même, si on y trouve quelque chose ressemblant à une adresse (par exemple `joe@example.net/nic@host`), cette ressource `nic@host` doit être traitée comme un identificateur opaque (et pas être analysée en « nom (at) machine »).

La section 3.5 contient quelques exemples de JID. Si certains sont « ordinaires » (le classique `juliet@example.co` reprenant la tradition XMPP des noms tirés de Roméo et Juliette), d'autres sont plus déroutants à première vue comme `juliet@example.com/foo@bar` (le domaine est `example.com`, la ressource `foo@bar`), `foo\20bar@example.com` (avec un échappement pour l'espace), `[Caractère Unicode non montré [2] ]@example.com` (partie locale en Unicode), `king@example.com/[Caractère Unicode non montré ]` (ressource en Unicode), `example.com` (juste le nom de domaine)...

2. Car trop difficile à faire afficher par LaTeX

Il y a aussi des exemples de JID **illégaux**, qu'il ne faut pas utiliser, comme `"juliet"@example.com` (les guillemets), `foo bar@example.com` (l'espace), `henri[Caractère Unicode non montré ]@example.com` (le caractère Unicode à la fin de la partie locale a un équivalent canonique et n'est donc pas autorisé), `[Caractère Unicode non montré ]@example.com` (la partie locale doit obéir aux règles restrictives du RFC 8265, qui n'autorise pas les symboles dans les identificateurs, seulement les lettres et chiffres), `juliet@` (domaine absent)...

En parlant d'adresses illégales, qui doit vérifier qu'elles sont légales ? Évidemment, l'émetteur devrait le faire. Mais notre RFC va plus loin en recommandant (section 4) que le récepteur jette les messages XMPP (les strophes, "*stanzas*" en anglais) contenant de telles adresses. C'est donc plus strict que le traditionnel principe de robustesse <https://www.bortzmeyer.org/principe-robustesse.html>.

Quelles sont les conséquences de sécurité de ces adresses ? Il n'y en a guère mais, bon, il y a toujours des inquiets donc la section 7 examine en détail tous les risques, même très théoriques. Bien sûr, les adresses XMPP héritent des questions de sécurité de PRECIS mais le RFC mentionne surtout les risques de triche sur les adresses. Il y en a de deux sortes, les usurpations et les imitations. Les usurpations (section 7.3.1) sont les cas où un méchant arrive à envoyer un message XMPP en trichant sur l'origine, par exemple si `jean@example.net`, une fois authentifié auprès de son propre serveur, arrive à transmettre un message prétendant venir de `jeanne@jabber.example`. Normalement, le protocole XMPP empêche cela, par l'authentification mutuelle des serveurs (sauf attaques un peu sophistiquées, par exemple sur le DNS). Cela dit, cette authentification n'empêche pas un serveur d'annoncer une autre partie locale (`jeanne@example.net` au lieu `jean@example.net`).

L'autre risque est celui d'imitation (section 7.3.2). Cette fois, il s'agit d'utiliser des JID légitimes et authentiques mais qui ressemblent à celui de la victime, par exemple `fric@paypa1.com` au lieu de `fric@paypal.com` (si vous ne voyez pas la différence, regardez mieux). Cette technique est souvent connue sous le nom de "*typejacking*". Comme le montre l'exemple du RFC ou bien celui cité plus haut, le problème arrive même en se limitant à ASCII. Si vous voulez un joli exemple avec Unicode (mais le résultat dépend de l'environnement avec lequel vous lisez cet article), regardez `[Caractère Unicode non montré ][Caractère Unicode non montré ][Caractère Unicode non montré ][Caractère Unicode non montré ][Caractère Unicode non montré ][Caractère Unicode non montré ][Caractère Unicode non montré ]` qui ressemble à `STPETER` mais est écrit en Cherokee. Comme un JID peut contenir à peu près n'importe quel caractère Unicode, il n'y a pas vraiment de prévention technique possible contre ce problème. Il est peu probable que cela ait des conséquences en pratique <https://www.bortzmeyer.org/idn-et-phishing.html>.

La liste complète des changements par rapport au RFC 6122 figure en annexe A. Rappelons que le RFC 6122 s'appuyait, pour les parties en Unicode, sur des profils `Stringprep` (RFC 3454) comme `Nameprep`. Ce `Nameprep`, décrit dans le RFC 3491, ayant été supprimé à l'occasion de la réforme IDNA bis <https://www.bortzmeyer.org/idnabis.html>, le grand changement dans ce nouveau RFC est l'abandon complet de stringprep et l'utilisation de PRECIS pour les identificateurs (comme la partie locale d'un JID) et d'IDN pour le nom de domaine.

Vu les changements entre stringprep et PRECIS, on ne peut pas garantir à 100 % que les JID autrefois valides le seront toujours. Mais, dans la plupart des cas, les anciens JID Unicode resteront légaux et utilisables.