RFC 8266 : Preparation, Enforcement, and Comparison of Internationalized Strings Representing Nicknames

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 6 octobre 2017

Date de publication du RFC : Octobre 2017

https://www.bortzmeyer.org/8266.html

Bien des protocoles Internet manipulent des noms qui doivent être parlants pour les utilisateurs et donc, de nos jours, doivent pouvoir être en Unicode. Les noms purement ASCII appartiennent à un passé révolu. Le groupe de travail PRECIS <https://tools.ietf.org/wg/precis> de l'IETF établit des règles pour ces noms, de manière à éviter que chaque protocole, chaque application, ne soit obligé de définir ses propres règles. Ce RFC contient les règles pour un sous-ensemble de ces noms : les noms qui visent plutôt à communiquer avec un utilisateur humain (par opposition aux noms qui sont indispensables aux protocoles réseaux, traités dans le RFC 8265 [1]). Il remplace le RFC 7700 (mais il y a peu de changements).

Ces noms « humains » sont typiquement ceux qui sont présentés aux utilisateurs. Ils doivent donc avant tout être « parlants » et il faut donc qu'ils puissent utiliser la plus grande part du jeu de caractères Unicode, sans restrictions arbitraires (contrairement aux identificateurs formels du RFC 8265, pour lesquels on accepte des limites du genre « pas d'espaces » ou « pas d'emojis »).

Le terme utilisé par le RFC pour ces noms « humains » est "*nicknames*", terme qui vient du monde de la messagerie instantanée. Il n'y a pas de terme standard pour les désigner, certains protocoles (comme le courrier) parlent de "*display names*" (par opposition au "*login name*" ou "*account name*"), d'autres utilisent encore d'autres termes (l'article « "*An Introduction to Petname Systems*" <http://www.skyhunter.com/marcs/petnames/IntroPetNames.html> » peut vous intéresser). Par exemple, dans un message électronique, on pourrait voir :

```
From: Valérie Pécresse <vp@les-républicains.fr>
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8265.txt

Et, dans cet exemple, `vp` serait le nom formel (”*mailbox name*” dans le courrier, ”*login name*” pour se connecter), alors que `Valérie Pécresse` est le ”*nickname*”, le nom montrable aux humains. (Le concept de ”*display name*” dans le courrier est normalisé dans la section 3.4.1 du RFC 5322, son exact équivalent XMPP, le ”*nickname*”, est dans XEP-0172 <`http://www.xmpp.org/extensions/xep-0172.html`>.)

Autre exemple, le réseau social Mastodon où mon nom formel est `bortzmeyer@mastodon.gougere.fr` alors que la description, le terme affiché est « S. Bortzmeyer [Caractère Unicode non montré [2]] » (avec un symbole à la fin, le [Caractère Unicode non montré]U+1F5F8).

Comme l'illustre l'exemple ci-dessus, on veut évidemment que le nom puisse être en Unicode, sauf pour la petite minorité des habitants de la planète qui utilisent une langue qui peut s'écrire uniquement en ASCII.

Ces noms « parlants », ces ”*nicknames*”, ne servent pas qu'à désigner des humains, ils peuvent aussi être utilisés pour des machines, des sites Web (dans les signets), etc.

On pourrait penser qu'il n'y a rien à spécifier pour permettre leur internationalisation. On remplace juste ASCII par Unicode comme jeu de caractères autorisé et vas-y, poupoule. Mais Unicode recèle quelques surprises et, pour que les ”*nicknames*” fonctionnent d'une manière qui paraitra raisonnable à la plupart des utilisateurs, il faut limiter légèrement leur syntaxe.

Ces limites sont exposées dans la section 2 de notre RFC, qui définit un **profil** de PRECIS. PRECIS, ”*Preparation, Enforcement, and Comparison of Internationalized Strings*” est le sigle qui désigne le projet « Unicode dans tous les identificateurs » et le groupe de travail IETF qui réalise ce projet. PRECIS définit (RFC 8264) plusieurs classes d'identificateurs et les ”*nicknames*” sont un cas particulier de la classe `FreeformClass` (RFC 8264, section 4.3), la moins restrictive (celle qui permet le plus de caractères).

Outre les restrictions de `FreeformClass` (qui n'est pas complètement laxiste : par exemple, cette classe ne permet pas les caractères de contrôle), le profil `Nickname` :

— Convertit tous les caractères Unicode de type « espace » (la catégorie Unicode Zs) en l'espace ASCII (U+0020),
— Supprime les espaces en début et en fin du ”*nickname*”, ce qui fait que ” `Thérèse`” et ”`Thérèse`” sont le même nom,
— Fusionne toutes les suites d'espaces en un seul espace,
— Met tout en minuscules (donc les ”*nicknames*” sont insensibles à la casse),
— Normalise en NFKC, plus violent que NFC, et réduisant donc les possibilités que deux ”*nicknames*” identiques visuellement soient considérés comme distincts (cf. section 6, qui prétend à tort que ce serait un problème de sécurité ; comme souvent à l'IETF, le groupe de travail a passé beaucoup de temps sur un faux problème de « confusabilité », cf. UTS#39 <`http://unicode.org/reports/tr39`>).

À noter qu'un ”*nickname*” doit avoir une taille non nulle, après l'application des ces règles (autrement, un ”*nickname*” de trois espaces serait réduit à... zéro).

Une fois ce filtrage et cette canonicalisation faite, les ”*nicknames*” peuvent être comparés par une simple égalité bit à bit (s'ils utilisent le même encodage, a priori UTF-8). Un test d'égalité est utile si, par exemple, un système veut empêcher que deux utilisateurs aient le même ”*nickname*”.

La section 3 de notre RFC fournit quelques exemples amusants et instructifs de ”*nicknames*” :

2. Car trop difficile à faire afficher par LaTeX

— "`Foo`" et "`foo`" sont acceptables, mais sont le même nom (en application de la régle d'insensibilité à la casse),
— "`Foo Bar`" est permis (les espaces sont autorisés, avec les quelques restrictions indiquées plus haut),
— "`Échec au roi [Caractère Unicode non montré ]`" est permis, rien n'interdit les symboles comme cette pièce du jeu d'échecs, le caractère Unicode U+265A,
— "`Henri [Caractère Unicode non montré ]`" est permis (ouvrez l'œil : c'est le chiffre romain à la fin, U+2163) mais la normalisation NFKC (précédée du passage en minuscules) va faire que ce nom est équivalent à "`henri iv`" (avec, cette fois, deux caractères à la fin).

Notez que ces règles ne sont pas idempotentes et le RFC demande qu'elles soient appliquées répétitivement jusqu'à la stabilité (ou, au maximum, jusqu'à ce que trois applications aient été faites).

Comme le rappelle la section 4 de notre RFC, les applications doivent maintenant définir l'utilisation de ces noms parlants, où peut-on les utiliser, etc. L'application peut donc avoir des règles supplémentaires, comme une longueur maximale pour ces "*nicknames*", ou des caractères interdits car ils sont spéciaux pour l'application.

L'application ou le protocole applicatif a également toute latitude pour gérer des cas comme les duplicatas : j'ai écrit plus haut que "`Foo Bar`" et "`foo bar`" étaient considérés comme le même "*nickname*" mais cela ne veut pas dire que leur coexistence soit interdite : certaines applications permettent à deux utilisateurs distincts d'avoir le même "*nickname*". Même chose avec d'autres règles « métier » comme la possibilité d'interdire certains noms (par exemple parce qu'ils sont grossiers).

Le profil `Nickname` est désormais ajouté au registre IANA <`https://www.iana.org/assignments/precis-parameters/precis-parameters.xml#profiles`> (section 5 du RFC).

L'annexe A décrit les changements depuis le RFC 7700. Le principal est le remplacement de l'opération Unicode `CaseFold()` par `toLower()` pour assurer l'insensibilité à la casse. La différence est subtile, et ne change rien pour la plupart des écritures. Sinon, l'erreur notée #4570 <`https://www.rfc-editor.org/errata_search.php?eid=4570`> a été corrigée. Le reste n'est que de la maintenance.