Universal Acceptance

Introduction à l'acceptation universelle (UA)

Groupe directeur sur l'acceptation universelle (UASG)

23/09/2019

TABLE DES MATIÈRES

À propos du présent document

Les technologies de l'Internet, y compris ses composantes de nommage, changent et évoluent en permanence. Au cours des dernières années, les nouveaux domaines de premier niveau (TLD), certains avec des caractères ASCII traditionnels et d'autre avec des caractères non-ASCII (noms de domaine internationalisés), ont été approuvés par la Société pour l'attribution des noms de domaine et des numéros sur Internet (ICANN). Certains exemples comprennent .nyc, .संगठन, .eco et .католик. Cependant, de nombreuses applications et services n'ont pas été mis à jour pour traiter cette plus grande diversité de TLD. En outre, les normes de messagerie électronique de l'Internet permettent désormais d'utiliser des caractères non-ASCII dans les adresses de courrier électronique, ce qui veut dire que tant que le logiciel ne sera pas mis à niveau, il ne traitera pas correctement ces domaines et adresses. Cela touche l'expérience de l'utilisateur de plusieurs formes :

- Des adresses de courrier électronique valides ne sont pas reconnues ou acceptées.
- Des noms de domaine sont à tort traités comme des termes de recherche dans la barre d'adresses du navigateur.

Jusqu'à ce que le logiciel ne reconnaisse et ne puisse traiter tous les nouveaux domaines et les adresses électroniques - état connu sous le nom d'acceptation universelle (UA) - il sera impossible d'offrir une expérience cohérente et positive aux internautes. Le présent document fournit une introduction générale au concept d'acceptation universelle et des efforts sont réalisés pour aider au développement de logiciels préparés pour l'acceptation universelle.

Public cible

Le présent document vise à présenter l'acceptation universelle à un public technique (développeurs, gestionnaires et opérateurs) qui peuvent connaître certains aspects de la technologie de l'Internet mais pas nécessairement les détails de la manière dont les nouveaux IDN, noms de domaine et adresses de courrier électronique devraient être acceptés, validés, stockés, traités et affichés. Il représente un point de départ pour que les gens de différents milieux techniques commencent leur exploration de l'acceptation universelle.

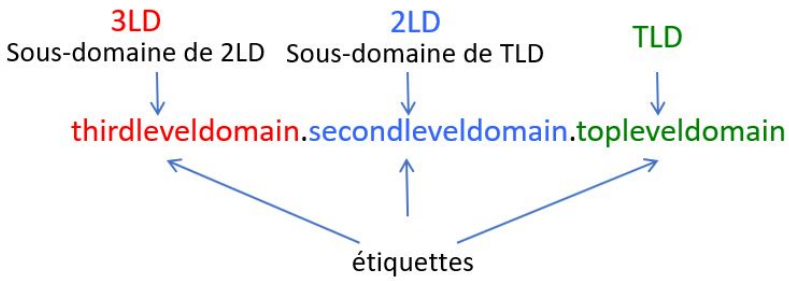
Concepts de base

Noms de domaine

Un nom de domaine est un identificateur convivial pour les ordinateurs et réseaux sur l'Internet. Il est habituellement représenté comme une séquence d'étiquettes de texte séparées par des « points » (point de ponctuation) ; par exemple, www.exemple.tld. Chaque étiquette représente un niveau dans la hiérarchie du système des noms de domaine (DNS).

Au plus haut niveau, ou à la « racine » de la hiérarchie, se trouvent les étiquettes des domaines de premier niveau (TLD) telles que com, jp, et বাংলা, qui apparaissent à la fin d'un nom de domaine. Étant donné qu'elles apparaissent à la fin, les TLD sont parfois appelés des « suffixes ».

Plus bas dans la hiérarchie du DNS à compter de la racine, l'étiquette suivante identifie un sous-domaine du TLD, appelé communément un « domaine de second niveau » ; l'étiquette suivante identifie un sous-domaine du domaine de second niveau, communément appelé un « domaine de troisième niveau », et ainsi de suite, chacune des étiquettes étant séparée de sa voisine par un point. Par exemple, un nom de domaine avec trois niveaux pourrait ressembler à ceci :



Domaines de premier niveau géographique (ccTLD)

Certains TLD sont délégués à des pays ou territoires spécifiques. On les appelle noms de domaine de premier niveau géographique (ccTLD) Dans le passé, tous les ccTLD étaient des codes à deux lettres qui correspondaient aux deux lettres assignées au pays ou au territoire par l'Organisation internationale de normalisation (ISO) dans la norme ISO 3166. Depuis 2010, il existe également des ccTLD internationalisés qui représentent le nom d'un pays ou d'un territoire dans l'écriture propre de ce pays ou de ce territoire.

Domaines génériques de premier niveau (gTLD)

La plupart des TLD qui ne sont pas des ccTLD s'appellent « domaines génériques de premier niveau » (gTLD), et sont soit ouverts à tous ou réservés aux membres d'une communauté définie. Il s'agit notamment du .com, .net et .org, ainsi que d'autres ajouts plus récents.

Grâce au programme des nouveaux gTLD - une initiative coordonnée par l'ICANN - le système des noms de domaine (DNS) s'est élargi de façon exponentielle à travers l'introduction des nouveaux domaines génériques de premier niveau. Ces nouveaux gTLD

peuvent représenter des marques, des communautés d'intérêt, des zones géographiques (villes, régions), et plus.

Internationalisation des noms de domaine

À l'origine, les noms de domaine étaient limités à un sous-ensemble de caractères ASCII (lettres de a à z, chiffres de 0 à 9, et le tiret « - »). Depuis le premier enregistrement .com, à savoir symbolics.com en 1985, le nombre et les caractéristiques des noms de domaine se sont développés pour satisfaire aux besoins liés à l'utilisation croissante de l'Internet en tant que ressource collective au niveau mondial. Aujourd'hui, la majorité des utilisateurs de l'Internet sont non-anglophones ; cependant, la principale langue utilisée sur Internet est l'anglais. En 2003, afin de faciliter l'internationalisation de l'Internet, le Groupe de travail de génie Internet (IETF) a commencé à élaborer des normes fournissant des directives techniques pour le déploiement de noms de domaine internationalisés (IDN) via un mécanisme de traduction pour soutenir les représentations non ASCII de noms de domaine dans tout script supporté par Unicode (par exemple普遍接受-测试.世界, ua-test.كاثوليك,etc.).

Le Conseil d'administration de l'ICANN a approuvé le processus d'introduction de nouveaux ccTLD IDN en octobre 2009 et les premiers ccTLD IDN ont été ajoutés à la zone racine en mai 2010. En juin 2011, le Conseil d'administration a approuvé et autorisé le lancement du programme des nouveaux gTLD, qui comprenait tant les nouveaux TLD ASCII que les TLD IDN. Le premier lot de TLD issu de ce programme a été ajouté à la zone racine en 2013.

L'acceptation universelle (UA), un élément indispensable

Dix ans après la publication des directives de l'IETF relatives aux IDN et grâce au programme des nouveaux TLD, plus de mille nouveaux TLD sont désormais actifs. Cependant, certains logiciels et certaines applications restent obsolètes et sont incapables de traiter ces nouveaux TLD. Cela pose des problèmes pour les utilisateurs de l'Internet, y compris ceux qui utilisent des caractères et des scripts non-ASCII.
L'acceptation universelle assure que tous les noms de domaine et les adresses de courrier électronique valides soient acceptés, validés, stockés, traités et affichés correctement et de façon cohérente par toutes les applications, tous les dispositifs et tous les systèmes de l'Internet. Par exemple, chaque adresse web valide renvoie au site web correct et chaque adresse de courrier électronique valide envoie des messages au destinataire prévu.

Le Groupe directeur sur l'acceptation universelle (UASG) est une initiative de la communauté Internet, soutenue par l'ICANN, et s'occupe d'entreprendre des activités qui auront pour effet de promouvoir l'acceptation universelle et d'aider à assurer une expérience positive et harmonisée pour les utilisateurs de l'Internet à l'échelle mondiale.

Étiquettes U et étiquettes A

Les noms de domaine qui utilisent des caractères non ASCII sont appelés noms de domaine internationalisés (IDN). La partie internationalisée d'un nom de domaine peut correspondre à toute étiquette, pas uniquement au TLD.

Dans la mesure où, au préalable, le DNS lui-même n'utilisait que des caractères ASCII, il est devenu nécessaire de créer un encodage supplémentaire permettant aux points de code Unicode non ASCII d'être représentés comme chaînes ASCII. L'algorithme qui met en œuvre cet encodage Unicode en ASCII est appelé Punycode ; les chaînes résultantes sont

appelées étiquettes A. Les étiquettes A se distinguent d'une étiquette ASCII ordinaire en ce qu'elles commencent toujours par les quatre caractères suivants :

xn--

Ces caractères sont appelés préfixes ACE.[1]

La transformation via Punycode est réversible : on peut transformer Unicode en une étiquette A et inversement transformer une étiquette A en caractères Unicode (connue sous le nom d'étiquette U).

L'algorithme Punycode n'est utilisé que pour exprimer des domaines internationalisés. Bien que l'on puisse hypothétiquement coder d'autres chaînes UTF-8 en utilisant Punycode, ce ne serait pas la norme et ne serait pas compatible avec d'autres systèmes.

Exemples d'IDN (imaginaires)

Version étiquette U	Version étiquette A
exemple.みんな	exemple.xn--q9jyb4c
大坂.info	xn--uesx7b.info
みんな.大坂	xn--q9jyb4c.xn--uesx7b

Internationalisation des adresses de courrier électronique (EAI)

Les adresses de courrier électronique sont constituées de deux parties :

- Une partie locale (avant le caractère « @ »)
- Un partie domaine (après le caractère « @ »).

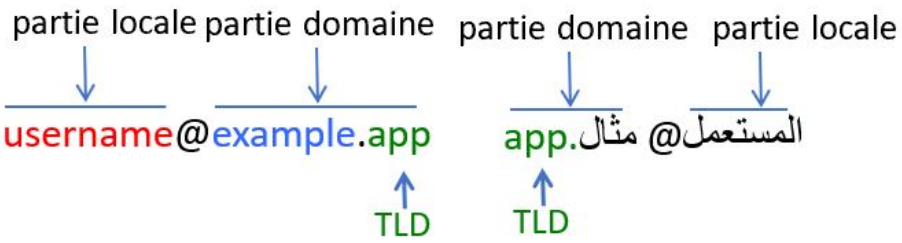
Étant donné que tant les scripts de gauche à droite (LTR) que les scripts de droite à gauche (RTL) peuvent être utilisés dans les adresses de courrier électronique et dans les noms de domaine, « avant » et « après » devraient être compris en fonction de l'orientation du script.

Exemples d'adresses EAI (imaginaires)

utilisateur@exemple.みんな	Utilise un TLD IDN
utilisateur@大坂.info	Utilise un domaine de second niveau IDN
用戶@exemple.avocat	Utilise une partie locale en Unicode et un nouveau gTLD

[1]Le préfixe d'encodage compatible ASCII (ACE) est utilisé afin de distinguer les étiquettes codées par Punycode d'autres étiquettes ASCII.

Exemple de texte de droite à gauche dans une adresse EAI



LTR **RTL**

Dans une adresse de courrier électronique internationalisée, la partie du domaine peut contenir n'importe quel nom de domaine, y compris ceux avec des nouveaux TLD, et peut contenir des étiquettes U en Unicode. La partie locale n'est pas un nom de domaine et peut en principe contenir presque n'importe quel caractère Unicode, bien que dans la pratique les systèmes de messagerie limiteront les caractères utilisés dans le nom de leurs boîte à lettres.

Le terme « internationalisation des adresses de courrier électronique » (EAI) est souvent utilisé pour décrire l'utilisation d'adresses internationalisées dans des courriers électroniques.

Génération de lien dynamique (linkification)

Des logiciels modernes, tels que des applications populaires de traitement de texte ou de tableur, permettent à un utilisateur de créer un hyperlien tout simplement en saisissant une chaîne qui ressemble à une adresse web, une adresse de courrier électronique ou un chemin du réseau. Par exemple, saisir « www.icann.org » dans le corps d'un courrier électronique peut se transformer automatiquement en un lien cliquable http://www.icann.org, si l'application reconnaît « www. » comme un préfixe spécial ou « .org » comme un suffixe spécial.

La génération de lien dynamique devrait marcher pour l'ensemble des adresses web, adresses de courrier électronique ou chemins du réseau correctement saisis et pas seulement pour quelques-uns. La linkification précise est difficile et dépend du contexte du texte (par exemple, dans certaines langues « www » n'indique pas une adresse Web), de sorte qu'elle n'est pas examinée ici.

La nature dynamique du registre de la zone racine

Le DNS est une grande base de données distribuée et divisée en sections appelées « zones ». La section qui contient tous les TLD s'appelle « zone racine » parce qu'elle est conceptuellement à la racine de l'arbre des noms DNS. Toutes les zones DNS, y compris la zone racine, sont mises à jour au besoin. À mesure que de nouveaux TLD sont ajoutés ou de vieux TLD sont retirés, leurs noms sont ajoutés ou supprimés de la zone racine.

Cela signifie que toute liste fixe de TLD, tel qu'une liste stockée dans une application ou dans un fichier, deviendra finalement et inévitablement obsolète. Pour valider de façon fiable les TLD dans un nom de domaine, le logiciel peut les vérifier en ligne à travers une requête

DNS, ou s’il utilise un fichier, actualiser le fichier périodiquement. Ces deux alternatives sont décrites plus en détail plus tard.

Les effets de l'acceptation universelle

Les cinq critères de l'acceptation universelle

L'acceptation universelle est l'état dans lequel tous les noms de domaine et les adresses de courrier électronique valides sont acceptés, validés, stockés, traités et affichés correctement et de façon cohérente par l'ensemble des applications, dispositifs et systèmes Web. Les cinq critères d'acceptation universelle sont décrits ci-dessous.

1. Accepter[2]	**L'*acceptation* est le processus via lequel une adresse de courrier électronique ou un nom de domaine est reçu en tant que chaîne de caractères d'une interface utilisateur, d'un dossier ou d'une API utilisés par une application logicielle ou un service en ligne.** Les applications et services permettent aux noms de domaine et adresses de courrier électronique d'être : ▪ Saisis dans des interfaces utilisateur, ou ▪ Reçus d'autres applications et services via des API.
2. Valider[3]	**La *validation* est un processus pouvant avoir lieu dans de nombreux endroits dès qu'une adresse de courrier électronique ou un nom de domaine est reçu ou émis en tant que chaîne de caractères par une application ou un service en ligne.** La validation a pour but de veiller à ce que les informations saisies soient valides ou au moins manifestement pas invalides. La validation garantit que l'information soit syntaxiquement correcte et peut faire d'autres contrôles. Pour les noms de domaine et les adresses de courrier électronique, de nombreux programmeurs ont toujours compté sur les méthodes de validation ad hoc telles que vérifier qu'un TLD respecte les limites de taille, ou que les caractères utilisés appartiennent à la liste ASCII. Toutefois, ces méthodes sont basées sur des hypothèses qui ne sont plus applicables car l'Internet change en permanence : ▪ Les noms de domaine et les adresses de courrier électronique peuvent désormais inclure des caractères Unicode non ASCII. ▪ La liste de TLD change.

[2] Dans le présent document, on distingue l'acceptation de la validation. Dans la pratique, ces deux actions peuvent se chevaucher.
[3] Dans le présent document, on distingue l'acceptation et le traitement de la validation. Dans la pratique, ces deux actions peuvent se chevaucher.

	▪ Toute étiquette dans un nom de domaine, y compris l'étiquette TLD, peut avoir une longueur maximale de 63 caractères.[4] Il est toujours possible de valider les TLD à l'aide d'autres techniques, tel que décrit ci-dessous.
3. Stockage	**Le *stockage* est le processus via lequel une adresse de courrier électronique ou un nom de domaine sont stockés en tant que chaînes de caractères dans une base de données ou un dossier utilisé par une application logicielle ou un service en ligne et, par la suite, récupérés par la même application logicielle ou d'autres.** Pour les applications et les services, il peut être nécessaire de procéder à un stockage à long terme et/ou temporaire de noms de domaine et adresses de courrier électronique. Indépendamment de la durée de vie des données, ils peuvent être stockés sous : ▪ Des formats définis par un appel à commentaires (RFC) relatif à une norme de l'Internet ou (moins souhaitablement) ▪ D'autres formats pouvant être traduits de ou vers des formats définis par les RFC. Bien que les noms de DNS et d'adresses de courrier électronique en Unicode soient stockés au format UTF-8, d'autres formats sont parfois utilisés dans le code historique. Voir ci-dessous la section consacrée aux « Bonnes pratiques ».

[4] La limite de 63 caractères de longueur s'applique à l'étiquette elle-même si c'est une étiquette ASCII, ou à la forme d'étiquette A de l'étiquette s'il s'agit d'un IDN.

4. Traiter[5]	**Le *traitement* intervient lorsqu'une adresse de courrier électronique ou un nom de domaine sont utilisés par une application ou un service afin de mener une activité (par exemple la recherche ou le tri d'une liste) ou sont transformés en un format distinct (par exemple transformer les étiquettes U en étiquettes A).** Une validation supplémentaire peut être réalisée lors du traitement. Les formes de traiter les adresses de courrier électronique et les noms de domaine ne sont limitées que par l'imagination des développeurs d'applications, mais il est important de ne pas faire des suppositions (p. ex., qu'un courrier électronique adressé à *pākehā@tetaurawhiri.govt.nz* est envoyé à une personne en Nouvelle Zélande) qui dépendent de politiques externes au DNS.
5. Affichage	**L'*affichage* intervient dès qu'une adresse de courrier électronique ou un nom de domaine est délivré au sein d'une interface utilisateur.** L'affichage des noms de domaine et des adresses de courrier électronique ne pose en général aucun problème lorsque les scripts utilisés sont pris en charge dans le système d'exploitation sous-jacent et lorsque les chaînes sont stockées en Unicode[6]. Si ces conditions ne sont pas respectées, des transformations propres aux applications peuvent être requises. En outre, même si le système d'exploitation sous-jacent supporte les chaînes, la visualisation peut être compliquée si, par exemple, elles comprennent une combinaison de scripts RTL et LTR, ou si la directionnalité générale du texte n'est pas claire.

Scénarios utilisateurs

Les exemples et définitions ci-dessus peuvent donner l'impression que l'acceptation universelle ne concerne que les systèmes informatiques et les services en ligne. La réalité est différente. Elle concerne également les individus qui utilisent ces systèmes et services.

Voici des exemples d'activités pour lesquelles l'acceptation universelle est requise :

[5] Dans le présent document, on distingue le traitement de la validation. Dans la pratique, ces deux actions peuvent se chevaucher.

[6] Il est important de reconnaître que l'affichage n'est pas simple, même lorsque ces conditions sont réunies pour certains scripts complexes.

Enregistrement d'un nouveau TLD	Une organisation adopte un TLD « de marque » afin d'offrir à ses clients une expérience client différente en fournissant des adresses de courrier électronique sous le format customername @example.brand. L'acceptation universelle signifie que : ▪ Les sites Web et les applications acceptent ces adresses de courrier électronique « @exemple.marque » tout comme ils accepteraient des TLD plus anciens tel que .com, .net, .org.
Accès à un gTLD	Un utilisateur accède à un site web dont le nom de domaine contient un nouveau TLD en saisissant une adresse dans un navigateur ou en cliquant sur un lien dans un document. L'acceptation universelle signifie que : ▪ Même s'il s'agit d'un nouveau TLD, le navigateur de l'utilisateur doit afficher l'adresse web dans sa forme originelle et accéder au site comme demandé par l'utilisateur. Le navigateur n'affiche pas les noms de domaine comme des étiquettes A à l'utilisateur à moins que ceci bénéficie l'utilisateur d'une quelconque façon.
Utilisation d'une adresse de courrier électronique contenant un nouveau gTLD en tant qu'identité numérique	Un utilisateur acquiert une adresse de courrier électronique avec la partie du domaine utilisant un nouveau gTLD et utilise cette adresse de courrier électronique comme son identité pour accéder à sa banque et au programme de fidélité d'une compagnie aérienne. L'acceptation universelle signifie que : ▪ Même si le domaine utilisé dans l'adresse de courrier électronique est tout neuf, la banque ou le site de la compagnie aérienne accepte l'adresse exactement comme s'il s'agissait d'un TLD établi tel que .biz ou .eu.
Accès à un IDN	Un utilisateur accède à l'URL d'un IDN en saisissant une URL dans un navigateur ou en cliquant sur un lien dans un document. L'acceptation universelle signifie que : ▪ Même si le nom de domaine contient des caractères différents des paramètres de langue de l'ordinateur de l'utilisateur, le navigateur que l'utilisateur souhaite utiliser doit afficher l'adresse web tel que demandé et pouvoir accéder au site.

Utilisation d'une adresse de courrier électronique internationalisée pour les courriers électroniques	Un utilisateur a créé une nouvelle adresses de courrier électronique qui comprend des caractères non-ASCII dans le nom de domaine (p. ex., ĩnfo@普遍接受-测试.世界). L'acceptation universelle signifie que : ▪ L'utilisateur peut envoyer et recevoir des messages de toute adresse de courrier électronique en utilisant un client de messagerie.
Utilisation d'une adresse de courrier électronique internationalisée en tant qu'identité numérique	Un utilisateur acquiert une adresse de courrier électronique non ASCII et l'utilise en tant qu'identité pour accéder à sa banque et au programme de fidélité d'une compagnie aérienne. L'acceptation universelle signifie que : ▪ La banque ou le site de la compagnie aérienne accepte la nouvelle identité exactement comme s'il s'agissait d'une autre identité de courrier électronique.
Création dynamique d'un hyperlien dans une application	Un utilisateur saisit une adresse web dans un document ou dans un message e-mail. L'acceptation universelle signifie que : ▪ Les règles utilisées par l'application afin de générer automatiquement un hyperlien sont les mêmes si l'adresse est non-ASCII ou contient un novueaux TLD.
Développement d'une application	Un développeur développe une application qui accède à des ressources web. L'acceptation universelle signifie que : ▪ Les outils utilisés par les développeurs incluent des bibliothèques qui permettent l'acceptation universelle en prenant en charge Unicode, les IDN et les nouveaux TLD.

Non-respect des pratiques en matière d'acceptation universelle

Les pratiques suivantes sont considérées comme étant de mauvaises pratiques :

✗	Montrer à l'utilisateur les étiquettes A sans que cela implique un avantage pour l'utilisateur, comme pour montrer la correspondance entre une étiquette U et une étiquette A.
✗	Obliger un utilisateur à saisir des étiquettes A lorsqu'il crée une nouvelle adresse de courrier électronique ou obliger un utilisateur à saisir des étiquettes A lorsqu'il enregistre un nouveau domaine hébergé.

✗	Valider la syntaxe d'un nom de domaine ou d'une adresse de courrier électronique en utilisant des critères obsolètes ou des ressources en ligne relatives aux noms de domaine ne faisant pas autorité.
✗	Utiliser une ancienne liste de TLD même si les nouveaux TLD y sont régulièrement ajoutés ou supprimés.
✗	Montrer l'utilisation interne des étiquettes A aux utilisateurs. Par exemple, la conversion des domaines dans les adresses EAI en étiquettes A au moment de répondre à un utilisateur d'EAI.
✗	Traiter certains noms de domaine comme des termes de recherche plutôt que comme des noms de domaine car l'application ne les reconnaît pas en tant que tels.
✗	Définir les filtres anti-spam pour qu'ils bloquent automatiquement des (nouveaux) TLD complets sans preuve d'abus.

Exigences techniques pour la préparation à l'acceptation universelle

Pour qu'une application ou un site Web soit prêt pour l'UA, il doit répondre à une diversité de critères.

Exigences de haut niveau

Une application ou un service qui intègre l'acceptation universelle (UA) :

1. **Prend en charge tous les noms de domaine indépendamment de leur longueur ou des ensembles de caractères.**
Voir le RFC 5892.

2. **Permet d'utiliser plusieurs ensembles de caractères valides pour les noms de domaine et les adresses de courrier électronique.**

Accepter les points de code Unicode ainsi qu'ASCII.

3. **Peut fournir correctement tous les points de code en chaînes Unicode.**

Voir le RFC 3490. Notez que l'Unicode ajoute régulièrement de nouveaux points de code et constitue donc une cible mobile.

4. **Peut fournir correctement les chaînes écrites de droite à gauche (RTL) telles que celles en arabe et en hébreu.**

Pour de plus amples informations sur les scripts RTL, voir le RFC 5893.

5. **Peut communiquer des données entre applications et services en UTF-8 et, le cas échéant, dans les autres encodages pouvant être convertis vers et depuis UTF-8.**

Pour de plus amples informations sur l'UTF-8, voir le RFC 3629..

6. Offre des API publiques et privées qui supportent UTF-8.

Les API privées s'appliquent uniquement aux appels interservices du même fournisseur.

7. Traite les adresses EAI correctement.

En particulier, ne convertit pas les IDN des adresses en étiquettes A.

8. Peut envoyer des courriers électroniques à des destinataires et en recevoir de ces derniers indépendamment du nom de domaine ou de l'ensemble de caractères.

Voir le RFC 6530.

9. Stocke les données des utilisateurs dans des formats qui supportent Unicode et pouvant être convertis en/de l'UTF-8.

De telles conversions ne seraient visibles que par l'opérateur du produit/service.

10. Supporte tous les noms de domaine de premier niveau contenus dans la liste de TLD de l'ICANN faisant autorité indépendamment de la longueur de l'ensemble de caractères.

Consultez la liste officielle à https://data.iana.org/TLD/.

Considérations pour les développeurs

Dans la mesure où de nombreux systèmes logiciels contiennent des hypothèses codées en dur sur des domaines et des adresses de courrier électronique, des changements de code peuvent être requis afin de reconnaître les IDN, les nouveaux TLD et les adresses électroniques EAI. Cette section aborde la façon dont les développeurs peuvent intégrer des changements de code qui permettront l'acceptation universelle.

Conception d'un logiciel qui garantisse la compatibilité et la flexibilité

Le principe de la robustesse, tel que formulé par Jon Postel dans le RFC 793, est une ligne directrice de conception générale applicable au développement de logiciels :

« Soyez conservateur dans ce que vous faites, soyez libéral dans ce que vous acceptez des autres ».

C'est-à-dire, soyez conservateur dans ce que vous envoyez : dans le cas où une spécification applicable était ambigüe ou peu claire, évitez de dire quelque chose qui pourrait surprendre les autres. D'autre part, au moment de la réception, acceptez tout ce qui pourrait être valide. Cela *ne* veut *pas* dire que vous changerez le code pour contourner les erreurs évidentes dans d'autres mises en œuvre car cela générerait des complications non documentées et impossibles à résoudre.

Bonnes pratiques en matière de développement et de mise à jour de logiciels dans la préparation pour l'acceptation universelle

Acceptation	
✔	Dans la mesure du possible, afficher les noms en Unicode. Les utilisateurs devraient être autorisés, mais non obligés, à saisir les noms de domaine comme étiquettes A plutôt que comme étiquettes U. Toutefois, les étiquettes U doivent être affichées par défaut, les étiquettes A étant affichées pour l'utilisateur l'utilisateur uniquement lorsque cela s'avère avantageux.
!	**Ne pas** générer des adresses de courrier électronique EAI avec des étiquettes A mais **être en mesure** d'assurer leur gestion si elles sont présentées par le logiciel de quelqu'un d'autre.
✔	Tout élément d'une interface utilisateur imposant à un utilisateur de saisir un nom de domaine ou une adresse e-mail doit accepter des noms longs. Les noms de domaine ASCII peuvent avoir jusqu'à 63 caractères dans chaque étiquette et peuvent représenter jusqu'à 253 octets au total. Les étiquettes UTF-8 peuvent être beaucoup plus longues que les étiquettes ASCII et leur longueur totale peut représenter jusqu'à 670 octets. N'oubliez pas que le code UTF-8 pour la plupart des points de code Unicode prend plus d'un octet. ▪ Voir le RFC 1035.

Validation	
✔	Valider uniquement si cela est approprié. **Valider uniquement si cela est nécessaire à l'opération de l'application ou du service.** C'est la façon la plus fiable de garantir que tous les noms de domaine valides soient acceptés dans vos systèmes.
✔	Reconnaître que des saisies syntaxiquement correctes peuvent ne pas représenter des noms de domaine ou des adresses de courrier électronique étant actuellement utilisés sur l'Internet. Cela peut être valide ou pas selon l'application.

!	Lorsque vous validez, considérez les éléments suivants : ▪ Vérifiez la partie TLD d'un nom de domaine par rapport à un tableau faisant autorité. L'IANA publie la liste de domaines de premier niveau à : * https://data.iana.org/TLD/tlds-alpha-by-domain.txt * Voir aussi : https://www.icann.org/en/system/files/files/sac-070-en.pdf ▪ Interrogez le nom de domaine par rapport au DNS. * L'API GETDNS (http://getdnsapi.net/) est un moyen hautement portable d'interroger le DNS. * La plupart des systèmes d'exploitation ont également une API de requête DNS native. ▪ Exigez qu'une adresse de courrier électronique soit répétée plusieurs fois afin d'éviter des erreurs de frappe. ▪ Valider les caractères dans les étiquettes en vérifiant que chaque étiquette suit les règles applicables à l'internationalisation des noms de domaine dans les applications (IDNA 2008). * Voir RFC 5892 ▪ Limiter la validation d'étiquettes à un nombre restreint de règles s'appliquant à toutes les étiquettes définies dans les RFC. * Voir RFC 5894 ▪ Veiller à ce que le produit ou la fonctionnalité prenne correctement en charge les chiffres. * Par exemple : Les caractères correspondant à des chiffres arabes-hindi devraient être traités comme des numéros dans les champs de saisie numérique, ainsi que comme des chiffres ASCII. * Noter que les chiffres arabes-hindi sont valides dans les étiquettes U mais ne sont pas considérés comme l'équivalence des chiffres ASCII dans ce contexte.

Stockage	
✔	Les applications et les services devraient supporter les normes Unicode les plus récentes.
✔	Dans la mesure du possible, les informations doivent être stockées sous le format UTF-8. Il est possible que certains systèmes exigent de stocker également sous le format UTF-16, mais en règle générale le format UTF-8 est préféré. Le format UTF-7 est obsolète et l'UTF-32 est trop lourd pour le stockage de fichiers.

	Les chaînes devraient être normalisées si nécessaire (dans certains contextes, la normalisation peut entraîner une perte d'information).
!	Envisager tous les scénarios dans leur globalité avant de convertir des étiquettes A en étiquettes U lors du stockage. Dans les nouvelles applications, il est mieux de ne conserver que les étiquettes U dans un dossier ou une base de données, car cela simplifie la recherche, le tri et la présentation. Toutefois, la conversion peut avoir des conséquences lors de l'interopération avec des applications et services plus anciens ne respectant pas le format Unicode.
✔	Étiqueter les adresses de courrier électronique et les noms de domaine lors du stockage afin d'y accéder plus facilement. Le fait de remplir les adresses de courrier électronique et les noms de domaine dans le champ « auteur » d'un document ou « info contact » d'un journal a entraîné la perte de l'adresse originale.
✔	Quelle que soit la façon dont les adresses et les noms de domaine sont stockés, vous devez être en mesure de faire correspondre les chaînes dans de multiples formats. Par exemple, la recherche de .みんな devrait également trouver exemple.xn--q9jyb4c.

Traitement	
✔	Assurer que toutes les réponses du serveur Web et des courriers MIME aient l'UTF-8 spécifié dans le type de contenu.
✔	Spécifier l'encodage UTF-8 dans l'en-tête http du serveur Web. ▪ Il est important de veiller à ce que l'encodage soit spécifié sur chaque réponse.
!	Envisager le contexte avant de convertir des étiquettes A en étiquettes U et vice versa lors du traitement. Il est souhaitable de ne conserver que les étiquettes U dans un dossier ou une base de données car cela simplifierait la recherche et le tri. Toutefois, la conversion peut avoir des conséquences lors de l'interopération avec des applications et services plus anciens ne respectant pas le format Unicode.
✔	Veillez à ce que le produit ou la fonctionnalité prenne en charge l'ordre de tri, les recherches et le classement en fonction de spécifications locales/linguistiques, et qu'il permette les recherches et le tri multilingue.

✗	Ne pas utiliser l'encodage du pourcentage pour les étiquettes des noms de domaine : ▪ exemple.みんな est correct ▪ exemple.%E3%81%BF%E3%82%93%E3%81%AA n'est pas correct.
✓	Étant donné que la norme Unicode est en développement continu, les points de code non définis lors de la création de l'application ou du service doivent être vérifiés afin de veiller à ce qu'ils ne génèrent pas de résultats erronés ou portant à confusion. Les polices manquantes dans le système d'exploitation sous-jacent pourraient conduire à des caractères non affichables (une petite boîte est souvent utilisée pour les représenter) mais cette situation ne devrait pas entraîner d'interruption du service ou de message d'erreur.
✓	Utilisez les API respectant le format Unicode.
✓	Utilisez les documents relatifs au protocole et les tableaux portant sur les noms de domaine internationalisés dans les applications (IDNA 2008) pour les IDN : ▪ RFC 5891 ▪ RFC 5892
✓	Traiter le texte en format UTF-8, dans la mesure du possible.
✓	Coordonner des mises à niveau des applications et des services desquels elles dépendent. Si le serveur est Unicode et que le client ne l'est pas, ou vice versa, les données devront être converties dans chaque transaction, ce qui est susceptible de générer des erreurs et peut être lent.
✓	Lors de la transformation de caractères, il se peut que des chaînes de texte grandissent ou diminuent considérablement. Chaque point de code UTF-8 peut prendre entre 1 et 4 octets, et dans certains cas un seul caractère dans un autre encodage peut correspondre à plusieurs points de code UTF-8 ou vice versa.

Affichage	
✓	Affichez tous les points de code Unicode pris en charge par le système d'exploitation sous-jacent. Les systèmes d'exploitation modernes ont tous le support de l'Unicode, mais leurs moteurs de rendu ne sont pas toujours corrects pour toutes les langues et écritures. Fournir le rendu des caractères dans les applications seulement lorsque le rendu correct n'est pas disponible à partir du/des système(s) d'exploitation cible(s).
✓	Lors du développement d'une application ou d'un service, passer en revue les langues prises en charge et s'assurer que les systèmes d'exploitation et les applications prennent en charge ces langues.

✔	Convertir les étiquettes A en étiquettes U avant leur affichage. Par exemple, l'utilisateur final devrait voir « `exemple.`みんな » et non pas « exemple.xn--q9jyb4c ». (Cette conversion est un exemple de traitement intégrant l'UA).
✔	Afficher les noms de domaine comme étiquettes U par défaut. Ne présenter les étiquettes A à l'utilisateur que lorsque cela résulte avantageux.
!	Être conscient que les noms de domaine en script mixte sont possibles. ▪ Il se peut que certains caractères Unicode se ressemblent à l'œil humain mais soient différents pour les ordinateurs ; par exemple, le « O » latin, le « O » cyrillique et l'omicron « O » grec. ▪ Les chaînes en script mixte sont courantes dans des scripts proches (*p. ex.*, en japonais entre le kanji, l'hiragana, le katakana et le romaji). Par ailleurs, les scripts mixtes peuvent être mélangés à des fins malveillantes, comme l'hameçonnage. Utilisez la norme technique Unicode nº39, « Mécanismes de sécurité d'Unicode »,[7] pour vérifier que les scripts d'une séquence Unicode suivent les bonnes pratiques. ▪ Si l'interface utilisateur attire l'attention de l'utilisateur sur les chaînes, s'assurer qu'elle le fait sans porter préjudice aux utilisateurs de scripts non latins. Afin d'en savoir davantage sur les impératifs de sécurité Unicode, cliquez ici : http://unicode.org/reports/tr36.
✔	Connaître les caractères non attribués et non autorisés pour les noms de domaine. ▪ Voir le RFC 5892

Unicode	
✔	Utiliser les API respectant le format Unicode.
✘	Utiliser des API normalisées bien déboguées pour : ▪ Conversions du format des chaînes ▪ Déterminer quel script comprend une chaîne ▪ Déterminer si une chaîne contient une combinaison de scripts ▪ Normalisation/décomposition Unicode

[7] Voir https://www.unicode.org/reports/tr39/#Restriction_Level_Detection

✗	Ne pas utiliser l'UTF-7 et limiter l'utilisation de l'UTF-32. ▪ L'UTF-7 est obsolète. ▪ L'UTF-32 utilise quatre octets pour chaque point de code. Vu que chaque point de code prend le même espace et peut être directement indexé dans les tableaux, il s'avère pratique de l'utiliser dans le code du programme, mais il pourrait être trop volumineux pour le stockage dans des fichiers et des bases de données.
✗	Ne pas utiliser l'UTF-16 sauf en cas de disposition explicite contraire (comme pour certaines API de Windows et certaines applications de Javascript). En UTF-16, 16 bits ne peuvent représenter que les caractères de 0x0 à 0xFFFF. Les valeurs au-delà de cette plage (0x10000 à 0x10FFFF) utilisent des paires de pseudo-caractères connus comme substituts. Si la gestion des paires substituts n'est pas minutieusement testée, elle peut entraîner des bogues délicats et éventuellement des trous de sécurité.
✓	Utiliser UTF-8 dans les cookies afin que les applications puissent les lire correctement.
✓	Utiliser le protocole IDNA 2008 et les documents suivants : ▪ RFC 5891 ▪ RFC 5892
✗	Ne pas utiliser l'IDNA 2003 qui a été remplacé par l'IDNA 2008.
!	Tenir à jour les tableaux IDNA et Unicode en fonction des versions prises en charge. Par exemple, à moins que l'application n'applique les règles de classification prévues dans le document des tables pour interpréter les points de code tels que répertoriés (RFC 5892), ses tables IDNA doivent être dérivées de la version d'Unicode prise en charge sur le système. Les tableaux ne doivent pas refléter la dernière version d'Unicode mais doivent être cohérents.
✓	Valider les étiquettes en utilisant les règles IDNA 2008 sur l'ensemble des étiquettes. ▪ Dans certains contextes, il pourrait s'avérer nécessaire de mener une validation plus poussée ; par exemple, si l'application sait quels sont les scripts autorisés dans les noms de domaine qu'elle utilise.

Généralités	
✓	Utiliser des ressources faisant autorité afin de valider les noms de domaine. Ne pas faire des hypothèses ad hoc désuètes, tel que « tous les TLD ont 6 caractères ou moins ».

✔	Veiller à ce que le produit ou la fonctionnalité prennent correctement en charge les numéros. Par exemple, les chiffres ASCII et les représentations asiatiques idéographiques des chiffres doivent tous être traités comme des chiffres dans des contextes numériques.
!	Rechercher des adresses électroniques qui pourraient être des adresses EAI dans des endroits inattendus : ▪ Métadonnées d'artistes, d'auteurs, de photographes ou relatives à des droits d'auteur. ▪ Métadonnées relatives aux polices. ▪ Enregistrements de contacts du DNS. ▪ Informations binaires. ▪ Informations de soutien. ▪ Informations de contact de l'OEM. ▪ Enregistrement, feedback et autres formulaires
!	Limiter les points de code autorisés lors de la génération de nouveaux noms de domaine et d'adresses de courrier électronique : Tous les produits qui utilisent des adresses de courrier électronique doivent accepter les adresses de courrier électronique internationalisées, ce qui impliquerait que la partie locale aurait la plupart de caractères imprimables UTF-8. Toutefois, une application ou un service n'a pas à autoriser tous ces caractères lorsqu'un utilisateur crée un nouvel IDN ou une nouvelle adresse EAI. Le fait d'empêcher dès le départ la création de certains IDN ou de certaines adresses de courrier électronique peut atténuer certains problèmes liés à la sécurité et à l'accessibilité. (REMARQUE : la bonne pratique imposerait tout de même aux logiciels d'accepter de telles chaînes si elles étaient présentées).
!	Garder à l'esprit que l'acceptation universelle ne peut pas toujours être mesurée uniquement via des tests automatisés. Par exemple, il n'est pas toujours possible d'évaluer la mesure dans laquelle une application ou un protocole gère les ressources de réseau et il est parfois plus judicieux de vérifier la conformité via un examen spécifique fonctionnel et un examen de la conception.
!	Ne pas assumer automatiquement qu'une composante, parce qu'elle ne fait pas directement appel à des API de résolution de noms ou n'utilise pas directement des adresses de courrier électronique, n'a aucune incidence sur ces éléments. Comprendre comment les noms de domaine sont obtenus par la composante (pas toujours via l'interaction de l'utilisateur). Voici quelques exemples de la façon dont la composante peut obtenir un nom de domaine : ▪ Politique de groupe. ▪ Requête LDAP. ▪ Fichiers de configuration. ▪ Registre Windows. ▪ Transfert vers/depuis une autre composante/fonctionnalité.

✔	Effectuer des révisions de code afin d'éviter les attaques liées au dépassement de la capacité de la mémoire tampon. ▪ En Unicode, les chaînes peuvent augmenter ou diminuer leur taille lorsqu'elles sont normalisées ou repliées. ▪ Lors de la conversion de caractères, le texte peut grandir ou diminuer considérablement.

Autres défis	
Mécanisme de détection et de conversion d'ensembles de caractères	Certaines anciennes applications de messagerie utilisaient des codages de caractères locaux et ne pouvaient ni détecter et ni convertir un texte en UTF-8 et vice versa au besoin. Cela était spécialement vrai pour les en-têtes de courrier électronique (À, CC, BCC, objet).
Gestion de multiples adresses de courrier électronique en une identité d'utilisateur unique	Lorsqu'un même utilisateur utilise plusieurs adresses de courrier électronique, il peut être compliqué de gérer ces adresses en tant qu'identité d'utilisateur unique. Les programmes de courrier électronique peuvent diriger le trafic destiné à ces faux noms vers la même boîte de réception, mais les applications peuvent toujours percevoir que ces courriers électroniques relèvent de différentes identités.

Sources faisant autorité pour les noms de domaine : Zone racine du DNS et listes de l'IANA

Il existe quelques options pour la liste de TLD faisant autorité. La première option est la zone racine du DNS elle-même. Elle est signée avec les extensions de sécurité du système des noms de domaine (DNSSEC), de sorte que son contenu peut être authentifié par un serveur de nom qui vérifie la signature du DNSSEC, même s'il peut être assez difficile à analyser comme un fichier de texte. Une autre source est le fichier de texte des TLD que publie l'IANA (un TLD par ligne, par ordre alphabétique). Ces fichiers se trouvent sur des serveurs Web https, il est donc recommandé de vérifier que le certificat de sécurité de la couche transport (TLS) soit valide lors du téléchargement pour être sûr que vous accédez au bon fichier.

Vous pouvez obtenir la liste des TLD à partir de l'un des liens suivants :

- https://www.internic.net/domain/root.zone (fichier de la zone racine)
- https://data.iana.org/TLD/tlds-alpha-by-domain.txt (fichier de texte des TLD)

Courrier électronique avec des IDN et en quoi cela diffère de l'EAI

L'internationalisation des adresses de courrier électronique (EAI) préfère les noms de domaine en UTF-8, l'utilisation d'étiquettes A codées en ASCII étant découragée. Certains systèmes de messagerie électronique ont adopté des dispositions partielles pour des adresses de courrier électronique qui incluent des IDN au lieu de supporter pleinement l'EAI. Étant donné que les IDN peuvent être représentés par des étiquettes A en ASCII, certains logiciels existants permettent que la partie d'une adresse de courrier électronique

correspondant à un IDN soit représentée en ASCII ou en Unicode. Par exemple, certains logiciels traiteront de la même façon ces deux adresses de courrier électronique de type IDN à toutes fins (envoi, réception et recherche) :

utilisateur@exemple.みんな = utilisateur@exemple.xn--q9jyb4c

Toutefois, certains logiciels ne traiteront pas ces adresses comme des équivalents, même si elles sont toutes deux valides, car ils ne convertissent pas une étiquette A (« xn--q9jyb4c ») en son étiquette U équivalente (« みんな ») avant de les comparer. Cela peut conduire à une expérience utilisateur imprévisible. L'expérience de l'utilisateur peut devenir particulièrement confuse si certains logiciels convertissent les étiquettes U en étiquettes A pour des questions de « compatibilité ». À mesure que les messages sont répondus ou transférés, les adresses qui sont visiblement différentes pour un utilisateur, ou qui ne parviennent pas à rechercher et classer comme prévu, pourraient augmenter.

Comme dans l'exemple ci-dessous, certains logiciels pourraient tenter de convertir la partie locale de l'adresse de courrier électronique en utilisant Punycode, l'algorithme qui est utilisé pour convertir les étiquettes A en étiquettes U (et vice versa). Cette sorte de conversion n'est pas valide et créera des adresses invalides non distribuables.

Ne jamais essayer de convertir la partie locale d'une adresse de courrier électronique en une forme différente

- ✔ 用戶@exemple.みんな
- ✖ xn--youq53b@exemple.xn--q9jyb4c

Le logiciel et les services robustes prêts à l'UA devraient être capables de gérer et de traiter tous ces formats correctement et devraient être capables de gérer tant les parties locales en UTF-8 que les étiquettes U en UTF-8 dans les adresses, tout en acceptant les étiquettes A des adresses compatibles avec les versions précédentes.

Défis en matière de linkification

Les logiciels modernes permettent parfois à un utilisateur de créer automatiquement un hyperlien simplement en saisissant une chaîne qui ressemble à une adresse web, un nom de courrier électronique ou un chemin d'accès réseau. Par exemple, le fait de saisir « www.icann.org » dans le corps d'un courrier électronique pourrait entraîner la création automatique d'un lien cliquable vers http://www.icann.org si l'application reconnait le « www. » comme une étiquette initiale ou le « .org » comme un TLD.

La linkification est le processus via lequel une application accepte une chaîne et détermine de façon dynamique si elle doit créer un hyperlien sur une page web (URL) ou une adresse de courrier électronique (mailto:) Le cas échéant, la linkification devrait fonctionner de la même façon pour toutes les adresses de courrier électronique, les noms de courrier électronique et les chemins d'accès réseau.

La linkification utilise des algorithmes et des règles créés par des développeurs de logiciels afin de déterminer si une chaîne devrait être considérée ou non comme un lien. Il faut ajouter à cela la façon dont les individus peuvent identifier une chaîne en tant que nom de domaine. Alors que les navigateurs, les clients de messagerie et les systèmes de traitement de texte constituent des lieux évidents, bien d'autres applications prennent ces décisions.

Recommandations de bonnes pratiques

1. Essayer de procéder à la linkification sur la base des préfixes de protocole explicites (par exemple « http:// », « ftp:// », « mailto: ») mais ne compléter l'action que si le reste de la chaîne est bien formée.

Exemple de chaîne	Comportement/résultat escompté
exemple.com	Pas de linkification car le protocole est absent et pas présumé.
http://exemple.com	Créer un hyperlien car le protocole est explicite.
http:exemple.com	Pas de linkification car la syntaxe est incorrecte (il manque //).
http://exemple.a	Pas de linkification parce que « a » n'est pas un TLD.
http://example..ab	Pas de linkification car la syntaxe est incorrecte (points consécutifs).
http://普遍接受-测试.世界	Créer un hyperlien car le protocole est explicite.

2. Essayer de procéder à la linkification sur la base des préfixes de protocole implicites (par exemple « www » implique « http://www »).

Exemple de chaîne	Comportement/résultat escompté
www.exemple.com	Créer un hyperlien car le protocole est implicite[8]
étiquette@exemple.com	Créer un hyperlien vers label@exemple.com car le protocole est implicite.

3. Le HTML entourant les URL qui contiennent le texte bidirectionnel pourrait inclure des codes ayant une incidence sur la direction dans laquelle le texte est affiché. La version linkifiée devrait conserver le même sens d'affichage.

4. Si les TLD sont utilisés en tant que « suffixe spécial » afin de déterminer la possibilité de procéder à une linkification, l'ensemble des TLD doivent alors être inclus. Une liste de TLD valides devrait être mise à jour régulièrement.

Unicode—Contexte et attributs du point de code

[8] Remarque : le site Web pourrait être https exclusivement et exiger l'utilisation de https:// plutôt que http://. Si c'était le cas, l'hyperlien pourrait ne pas pouvoir résoudre.

La norme Unicode a évolué depuis sa première publication en tant qu'Unicode 1.0 en 1991. Chaque version publiée depuis a ajouté plus de caractères et de points de code pour gérer plusieurs langues et scripts. La version actuelle est la version 12.1.

En Unicode, chaque point de code possède un ensemble de propriétés, telles que les lettres majuscules, les numéros décimaux, ou la marque de non-espace. De nombreux caractères appartiennent à un script tel que le Latin, le Han (chinois), ou l'arabe, tandis que d'autres, tels que la ponctuation, non.

Tel que décrit ci-dessous, l'IDNA utilise les attributs du point de code pour déterminer quels sont les caractères autorisés dans les IDN. UAX#44, base de données des caractères Unicode, décrit la base de données des attributs des points de code.

UTF8, UTF16 et autres méthodes d'encodage

Un point de code Unicode peut avoir une valeur numérique allant de 0 à 0x10FFFF. Étant donné qu'un seul octet ne peut contenir que des valeurs de 0 à 0xFF, il est nécessaire d'avoir une sorte d'encodage multi-octets pour stocker les points de code Unicode.

La version originale d'Unicode avait moins de 64K (0xFFFF) points de code, de sorte que chaque point de code pourrait s'inscrire dans un entier de 16 bits. Cela a conduit à l'encodage en deux-octets connu sous le nom d'UCS ou UCS-2. Lorsque l'Unicode s'est élargi au-delà de 64K points de code, l'UCS a été prolongé en UTF-16[9], qui utilise des paires de points de code de 16 bits autrement invalides connues comme *« substituts »* pour représenter des valeurs supérieures à 64K. Bien que cela fonctionne, des problèmes de débogage se sont présentés car les substituts ajoutent de la complexité à tout code qui compte le nombre de points de code dans une chaîne, ou qui ordonne les chaînes par ordre de points de code. Un autre problème est que certains ordinateurs comme ceux fabriqués par IBM gardent d'abord la partie la plus élévée au niveau des octets d'une valeur de 16 bits (« big-endian »), et certains comme ceux d'Intel stockent d'abord la partie la plus réduite (« little-endian »). En conséquence, l'UTF-16 a deux variantes de stockage : UTF-16BE et UTF-16LE. Il existe des techniques pour détecter et corriger des problèmes dûs à cette différence, mais elles peuvent générer des bogues. À ce stade, l'UTF-16 est principalement utilisé dans les applications existantes avec des API de Microsoft Windows, et en langues Java et Javascript.

L'UTF-8 est un encodage alternatif qui encode chaque point de code comme une chaîne de longueur variable d'entre un et quatre octets. L'UTF-8 a plusieurs avantages par rapport à l'UTF-16, y compris que le sous-ensemble ASCII des caractères Unicode est encodé comme un seul octet, c'est-à-dire que toute chaîne ASCII devieint automatiquement une chaîne UTF-8. L'UTF-8 est généralement plus compact que son équivalent UTF-16 ; il est plus facile à trier parce que les chaînes en UTF-8 triées dans l'ordre des octets sont automatiquement en ordre par point de code. L'IDNA et l'EAI exigent l'encodage en UTF-8.

L'UTF-32 est un format simple qui stocke chaque point de code dans un entier de 32 bits. Il est convenable pour le traitement interne dans les programmes car les points de code d'une

[9] Voir la section 3.10 de la norme Unicode pour les détails techniques de l'UTF-8, l'UTF-16 et l'UTF-32, à https://www.unicode.org/versions/Unicode12.0.0/ch03.pdf.

table d'UTF-32 peuvent être indexés directement, mais il est rarement utilisé pour le stockage en raison de son poids.

IDNA - Bref historique IDNA et état actuel

Les noms de domaine internationalisés dans les applications (IDNA) ont été définis pour la première fois par l'IETF en 2003 et dans ce qui est maintenant connu comme IDNA2003[10]. Ils comprenaient un algorithme pour répertorier les points de code Unicode en une forme normalisée des étiquettes des noms de domaine connue comme « Nameprep », et un algorithme pour encoder des étiquettes de point de code Unicode en ASCII appelé « Punycode ». Nameprep comprend des transformations telles que le répertoriage des majuscules et des minuscules.

Suite à ses expérience avec l'IDNA, l'IETF a élaboré et publié une spécification révisée connue comme « IDNA2008 » en 2010[11]. L'IDNA2008 a créé les termes « étiquette U » et « étiquette A » et a supprimé l'étape de Nameprep, conseillant que le répertoriage soit fait par les applications de manière adaptée aux environnements local et de l'application. L'IDNA2008 a été mis à jour pour l'Unicode 6.0 par le RFC 6452 en 2011 et il est constamment révisé par l'IETF.

Dans la pratique, trop de mises en œuvre utilisent toujours l'IDNA2003. Quelques bibliothèques utilisent les tables (comme celles incluses dans l'IDNA2003) créés pour l'IDNA2008. Il n'existe aucun répertoriage pour l'IDNA2008 à l'exception du repliage normal et des règles de normalisation comprises dans la norme Unicode.

La seule exception est qu'il y a quelques correspondances du UTS#46, Traitement de la compatibilité entre l'IDNA et Unicode. Ceci indique si un petit nombre de caractères communs répertoriés dans l'IDNA2003 mais admis comme caractères dans l'IDNA2008 devraient être acceptés ou répertoriés. Il est important que les applications traitent ces caractères suivant l'IDNA2008 et pas l'IDNA2003, et que si elles utilisent l'UTS#46, elles l'utilisent d'une manière qui soit compatible avec l'IDNA2008.

Cas utilisés pour tester

Le logiciel qui est destiné à gérer les IDN et les adresses de courrier électronique EAI devrait être testé avec un large éventail de noms de domaine et d'adresses. Consulter l'UASG 004, « C*as d'utilisation pour évaluer l'état de préparation pour l'UA »*, pour voir un ensemble de cas de test.

Mise à niveau du logiciel pour l'EAI

La conformité avec l'EAI exige la mise à niveau des serveurs de messagerie, du logiciel de soumission et de livraison, des agents de gestion de courrier et de messagerie Web, et de toute application qui gère les adresses de courrier électronique et envoie des courriers.

---

[10] La définition est contenue dans les RFC 3490, 3491, et 3492.
[11] La définition est contenue dans les RFC 5890, 5891, 5892, 5893, 5894 et 5895.

Pour un aperçu détaillé de l'EAI, de ses enjeux et de la façon de l'appliquer, voir l'UASG 012, *« Internationalisation d'adresses de courrier électronique (EAI) : un aperçu technique ».*

Sujets avancés

Scripts complexes

Les détails des scripts complexes ne présentent pas forcément d'intérêt pour les personnes autres que les développeurs qui créent leurs propres bibliothèques d'analyse de chaînes ou d'affichage. Toutefois, un résumé est inclus ici afin de veiller à ce que tous les lecteurs disposent de suffisamment de connaissances leur permettant de reconnaître les bogues de code liés à ces scripts lorsqu'ils se présentent.

Pour le texte HTML formaté dans les pages Web et les courriers électroniques, les normes HTML possèdent des fonctionnalités avancées permettant de gérer et d'afficher du texte complexe et bidirectionnel que les développeurs devraient comprendre et utiliser pour rendre un texte. Voir la section normalisée WHATWG HTML sur le rendu[12] et la section correspondante de la norme W3C HTML[13].

Langues écrites de droite à gauche et conformité avec Unicode

Certains scripts, comme le latin et le dévanagari, présentent les caractères de gauche à droite lorsque le texte est organisé en lignes horizontales. D'autres scripts, comme l'arabe ou l'hébreu, présentent les caractères de droite à gauche. Le texte peut également être bidirectionnel lorsqu'un script s'écrivant de droite à gauche utilise des chiffres qui sont écrits de gauche à droite ou lorsqu'il utilise des termes intégrés issus de l'anglais ou d'autres scripts écrits de gauche à droite.

Des problèmes et des ambiguïtés peuvent se poser lorsque le sens horizontal du texte n'est pas uniforme. Afin d'y remédier, un algorithme permet de déterminer le sens du texte Unicode bidirectionnel.

Il existe un ensemble de règles qui doivent être appliquées par l'application afin de générer le bon ordre au moment de l'affichage ; ces règles sont décrites par l'algorithme bidirectionnel Unicode, souvent appelé « algorithme de Bidi ».

L'algorithme de Bidi

L'algorithme de Bidi décrit la façon dont les logiciels doivent traiter les textes contenant des séquences de caractères écrits de gauche à droite (LTR) et de droite à gauche (RTL). Le sens de base[14] attribué à la phrase déterminera l'ordre dans lequel le texte est affiché. Cela

---

[12] Disponible à https://html.spec.whatwg.org/multipage/rendering.html
[13] Disponible à https://www.w3.org/TR/2018/WD-html53-20181018/rendering.html
[14] En HTML, le sens de base soit découle du sens par défaut du document, c'est-à-dire de gauche à droite, ou bien il est explicitement défini par l'élément parent le plus proche qui utilise l'attribut direction « dir ».

peut être soit de gauche à droite ou de droite à gauche et définit l'ordre dans lequel les séquences de caractères sont affichées. Dans ce document, le sens de base est de gauche à droite, ce qui implique que toutes les séquences de caractères sont affichées avec la première séquence à gauche de la deuxième séquence.

Afin de savoir si une séquence s'écrit de gauche à droite ou de droite à gauche, chaque caractère dans Unicode a une propriété directionnelle associée. La plupart des lettres sont fortement typées (caractères forts) comme LTR (de gauche à droite) ou RTL (de droite à gauche) selon le script dont elles font partie. Une séquence de caractères RTL fortement typées sera affichée de droite à gauche. Cela n'a rien à voir avec le sens de base environnant. Par exemple :

(LTR) exemple - مثال (RTL).

Du texte avec un sens différent peut être mélangé sur les lignes. Dans ce cas, l'algorithme Bidi produit un indicateur directionnel distinct à partir de chaque séquence de caractères contigus avec le même sens.

Les espaces et la ponctuation ne sont pas fortement typées comme LTR ou RTL dans Unicode car ils peuvent être utilisés dans l'un ou l'autre des scripts. Ils sont ainsi qualifiés de caractères neutres ou faibles. Les caractères faibles sont ceux qui sont généralement utilisés dans une direction, mais qui peuvent être utilisés dans l'autre dans certains contextes. Des exemples de ce type de caractères comprennent :

- Les chiffres européens.
- Les chiffres arabes-hindi.
- Les symboles arithmétiques et les symboles monétaires.
- Les signes de ponctuation qui sont communs à bon nombre de scripts tels que les deux-points, la virgule, le point et l'espace no-break.

Sans contexte, le sens des caractères neutres est indéterminé. En voici quelques exemples :

- Onglets.
- Séparateurs de paragraphe.
- La plupart des autres caractères d'espacement.

Lorsqu'un caractère neutre se trouve entre deux caractères fortement typés qui ont le même type directionnel, on retiendra ce type directionnel. Par exemple, un caractère neutre entre deux caractères RTL sera traité en tant que caractère RTL et aura pour effet d'étendre l'indicateur directionnel :

- مثال.نطاق

Même s'il y a plusieurs caractères neutres entre les deux caractères fortement typés, ils seront tous traités de la même façon.

S'il y a un espace ou un signe de ponctuation entre deux caractères fortement typés qui ont un sens distinct, le ou les caractères neutres seront traités comme s'ils avaient le même sens en tant que sens de base principal. Par exemple :

- exemple. مثال

N'oubliez pas que le présent document a une séquence de base de gauche à droite, ce qui veut dire que « exemple » est le domaine de second niveau et مثال , le TLD.

Sauf en cas d'annulation directionnelle, les nombres sont toujours encodés et saisis avec le chiffre le plus élevé dans l'ordre, et les chiffres écrits LTR. La faible directionnalité s'applique uniquement au placement du nombre dans son intégralité.

Tous les détails de l'algorithme Bidi sont décrits dans le rapport technique Unicode n°9.

La règle Bidi pour les noms de domaine

Un nom de domaine Bidi est un nom de domaine qui contient au moins une étiquette RTL. La règle Bidi pour les noms de domaine, définie dans le RFC 5893[15], limite les points de code dans les noms de sorte qu'il n'y ait pas deux noms qui soient différentes séquences de points de code mais qui affichent le même résultat en raison de règles d'affichage bidirectionnel.

Liants

Certaines langues utilisent des scripts alphabétiques dans lesquels des phonèmes uniques sont écrits en utilisant deux caractères, ce que l'on appelle digraphe. Autrement dit, un digraphe est un groupe de deux lettres successives qui représentent un son unique (ou phonème).

Exemples de digraphes en anglais		
ch (comme dans church) ph (comme dans phony)	th (then) th (think)	sh (shoe) gh (rough)

Certains digraphes sont intégralement liés sous forme de ligatures. En écriture et typographie, on parle de ligature lorsqu'au moins deux graphèmes ou lettres sont reliés par un glyphe unique. Comme exemple, on peut citer le symbole (&) qui vient de la jonction des lettres latines « e » et « t » (« et »). Dans les caractères anglais, fi et ffi sont souvent affichées comme des ligatures.

Si les ligatures et les digraphes ont la même interprétation dans toutes les langues qui utilisent un script donné, la normalisation Unicode résout en général les différences et les fait correspondre. En cas d'interprétations différentes, la mise en correspondance doit être effectuée via d'autres méthodes, souvent choisies au niveau du registre, ou alors des utilisateurs doivent avoir été formés de façon à comprendre que la mise en correspondance ne pourra pas être effectuée. Un exemple d'interprétation différente est disponible à la section 4.3 du RFC 5894[16]. Le Consortium Unicode prévoit deux principales stratégies visant à déterminer le comportement liant d'un caractère donné après l'application de l'algorithme Bidi à des caractères antiliants sans chasse connus comme ZWJ et ZWNJ.

---

[15] « *Scripts écrits de droite à gauche pour les noms de domaine internationalisés dans les applications (IDNA)* », RFC 5893, https://www.rfc-editor.org/info/rfc5893

[16] « *Noms de domaine internationalisés dans les applications (IDNA) : contexte, justification et explication* », RFC 5894, https://www.rfc-editor.org/rfc/rfc5894.html#section-4.2

(Pour plus d'informations sur ces liants, consultez http://www.unicode.org/L2/L2005/05307-zwj-zwnj.pdf.)

- Lors de sa conception, une mise en œuvre peut renvoyer au support de stockage original afin de voir s'il y avait des caractères ZWNJ ou ZWJ adjacents.
- Sinon, la mise en œuvre peut remplacer le ZWJ et le ZWNJ par une propriété de caractère hors bande associée à ces caractères adjacents afin que les informations n'interfèrent pas avec l'algorithme Bidi et qu'elles soient préservées suite au réagencement de ces caractères. Une fois que l'algorithme aura été appliqué, ces informations hors bande pourront alors être utilisées à des fins de conception adéquate ».

Les opérateurs de registre de noms de domaine et toute autre entité permettant la création de noms de domaine (p. ex., les applications qui créent des étiquettes de troisième niveau ou inférieur) doivent suivre la règle Bidi pour les noms de domaine afin de garantir que les noms s'afficheront de forme cohérente et pour éviter la confusion des noms qui peuvent être utiliser pour perpétrer des attaques homographes.

Pour en savoir davantage sur les liants, consultez la section 4.3 du RFC 5894.

Homoglyphes et caractères similaires

Les homoglyphes sont des caractères qui, en raison de leurs similitudes en termes de taille et de forme, peuvent sembler identiques ou prêter à confusion au premier coup d'œil. Ils se produisent fréquemment lors du mélange entre scripts latin, cyrillique et grec. Par exemple, le « o » latin (code U+006f), la petite lettre « o » en cyrillique (code U+043e) et la petite lettre grecque omicron « o » (code U+03bf). Dans certains cas, il existe des homoglyphes dans un même script, tels que la petite lettre croate « lj » (code U+01c9) et les deux lettres « lj » (code U+006C U+006a). Pour d'autres exemples, consultez la table à http://homoglyphs.net/.

Pour éviter les noms de domaine avec des homoglyphes, les opérateurs de registre devraient appliquer les règles de génération d'étiquettes (LGR) qui limitent les points de code dans une étiquette à un ensemble d'un seul script ou de scripts compatibles. Chaque opérateur de registre devrait avoir des LGR pour chaque script dans lequel il accepte des enregistrements[17].

Afin d'en savoir plus sur les mécanismes de sécurité d'Unicode permettant de détecter une confusion, consultez :

- http://www.unicode.org/reports/tr39/#Confusable_Detection

Pour en savoir plus sur les caractères prêtant à confusion et les bonnes pratiques y afférentes, consultez :

- Didacticiel et aperçu des abus Unicode de M3AAWG https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf

[17] L'IANA a un ensemble de LGR des registres dans son référentiel de pratiques en matière d'IDN, disponible à https://www.iana.org/domains/idn-tables.

- Meilleures pratiques de M3AAWG en matière de prévention des abus Unicode https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf

Normalisation, casse des balises et préparation de chaînes

La normalisation Unicode permet de déterminer si deux chaînes Unicode sont équivalentes et fournit des formulaires standard à utiliser pour traiter et stocker des chaînes. Certains caractères peuvent être représentés dans Unicode par plusieurs séquences de code. On parle d'équivalence Unicode. Unicode prévoit deux types d'équivalences :

- Canoniques
- Compatibilité

Les séquences représentant le même caractère relèvent de l'équivalence canonique. Ces séquences ont le même aspect et la même signification lorsqu'elles sont imprimées ou affichées. Par exemple :

U+006E (« n » minuscule latin) suivi de U+0303 (tilde combinant « ◌̃ »)	= ñ
U+00F1 (« ñ » minuscule de l'alphabet espagnol)	= ñ

L'Unicode définit le NFC (la forme de normalisation C) comme une décomposition canonique, suivie de la composition canonique. Cela réduit le texte à un nombre minimal de points de code sans modifier son apparence. Il convient de noter que dans cet exemple, les trois caractères ci-dessus sont valides pour être utilisés conformément à l'IDNA2008.

Les équivalents de compatibilité sont des séquences qui présentent différents aspects mais qui, dans certains contextes, ont le même sens. Il s'agit d'un type d'équivalence plus faible entre caractères ou séquences de caractères. Par exemple :

U+FB00 (la ligature typographique « ff »)	= ff
U+0066 U+0066 (deux lettres latines « f »)	= ff

Dans l'exemple ci-dessus, le point de code U+FB00 est défini comme étant compatible mais pas canoniquement équivalent à la séquence U+0066 U+0066. Les séquences qui sont canoniquement équivalentes sont également compatibles, mais le contraire n'est pas forcément vrai.

Il convient de noter que le point de code U+FB00 n'est pas valide selon l'IDNA2008. L'Unicode définit le NFKC (la forme de normalisation KC) comme une décomposition de compatibilité, suivie de la composition canonique. Cela réduit le texte à un ensemble standard de points de code et pourrait modifier son apparence. Par exemple, le NKFC transforme la ligature « ﬀ » dans les deux lettres « f f » et le symbole ante meridiem ㏂ (symbole U+33C2) dans les quatre caractères « a.m. » (U+0061 U+002E U+006D U+002E).

Afin d'éviter les problèmes d'interopérabilité liés à l'utilisation de séquences de caractères canoniquement équivalentes bien que différentes, le W3C recommande d'utiliser la forme de normalisation C pour l'ensemble des textes.

Pour une liste de tous les caractères susceptibles de changer dans l'une des formes de normalisation, consultez : http://www.unicode.org/charts/normalization

Autres points à noter :

- Les caractères des étiquettes IDN doivent être en forme NFC.
- Lorsque deux applications partagent des données Unicode mais les normalisent différemment, des erreurs et des pertes de données peuvent se produire.
- Le consortium Unicode affirme que les formes de normalisation doivent rester stables au fil du temps. Autrement dit, une chaîne doit rester normalisée conformément à toutes les futures versions d'Unicode (compatibilité avec les versions précédentes).
- Tel que cela a été dit auparavant, soyez conservateur au moment de considérer quels points de code seront permis dans un nom de domaine.

Conseils pour développeurs de logiciels	
✗	Ne pas normaliser en convertissant en majuscules, ou en ignorant des caractères autres que les espaces, car cela pourrait aussi complexifier le tri, la copie de données, l'importation et l'exportation de données et l'extraction de données par des applications client et entraîner des pertes ou une corruption de données.
✗	Ne jamais permettre l'utilisation de points de code dans les noms de domaine n'étant pas autorisés conformément à l'IDNA2008.

Pour en savoir plus sur la normalisation Unicode, consultez :

- http://www.w3.org/TR/charmod-norm
- http://unicode.org/reports/tr15

Sensibilité à la casse (différentiation minuscules/majuscules) et correspondance

La casse des balises et la correspondance signalent le processus de convertir tous les caractères d'une chaîne soit en majuscules, soit en minuscules (généralement en minuscules). La correspondance entre majuscules [A-Z] et minuscules [a-z] fonctionne pour les documents exclusivement de texte en caractères ASCII, mais elle est bien plus complexe dans les langues qui utilisent des caractères supplémentaires. La correspondance entre majuscules et minuscules peut dépendre du contexte, où le caractère répertorié dépend du contexte dans lequel il apparaît, par exemple, les diverses formes du sigma grec. Il peut varier également en fonction des usages locaux, où le caractère répertorié dépend du contexte dans lequel le texte est interprété, par exemple, le I majuscule et le minuscule en turc avec et sans le point. La casse est indépendante des paramètres locaux dans le cas de chaînes qui seront interprétées par un logiciel, alors que la correspondance dépend du contexte local et est utilisée dans un texte qui sera lu par les gens. Enfin, la correspondance vers la majuscule et vers la minuscule **ne** sont **pas** des fonctions inverses.

Dans le cas des IDN, l'IDNA2008 permet aux applications d'utiliser soit des majuscules, soit des minuscules, parce que la correspondance se fait avant la validation des points de code. Dans la pratique, les correspondances d'identificateurs spécifiques au contexte local n'existent pas et tout le monde utilise les tables Unicode UTS#46[18].

Conseils pour développeurs de logiciels	
✔	Tenir compte de l'objectif souhaité avant de tenter la correspondance entre majuscules et minuscules : est-ce une liste générique pour les étiquettes, une chaîne dans une langue connue ou autre ?
✔	Effectuer la normalisation Unicode avant la casse.

Glossaire et autres ressources

Glossaire

Étiquette A	Représentation avec un encodage compatible ASCII (ACE) d'une étiquette dans un nom de domaine internationalisé, utilisé à l'interne dans le protocole DNS. Les étiquettes A commencent toujours avec le préfixe ACE « xn-- ». Une étiquette A peut être convertie en une étiquette U et vice-versa sans perte d'informations.
Préfixe ACE	Préfixe « xn-- » avec un encodage compatible ASCII.
ASCII	Code américain normalisé pour l'échange d'information. Les caractères ASCII comprennent des caractères latins sans accent et les chiffres européens-arabes. ASCII est un sous-ensemble d'Unicode : chaque caractère ASCII est également un caractère Unicode.

[18] UTS#46, *« Traitement de la compatibilité de l'IDNA d' Unicode »*, https://www.unicode.org/reports/tr46/#Mapping

API	Une interface de programmation applicative (API) constitue un ensemble de règles, de protocoles et d'outils pour la construction de logiciels et d'applications. Une API peut concerner un système web, un système d'exploitation ou un système de base de données, et elle fournit à ce système un lieu de développement d'applications à l'aide d'un langage de programmation donné.
Espace de code	Gamme définissant les limites inférieure et supérieure d'un encodage.
Point de code	Un point de code est une valeur numérique dans un espace de code. Les points de code sont utilisés afin de distinguer à la fois la valeur numérique de leur encodage en tant que séquence de bits, et le caractère abstrait d'une de ses représentations graphiques spécifiques (glyphes).
Zone racine du DNS	La zone racine est l'annuaire central du DNS, c'est-à-dire la composante clé pour la recherche dans le DNS ; par exemple, pour traduire des noms d'hôtes en adresses IP.
EAI	L'internationalisation des adresses de courrier électronique permet l'utilisation de caractères d'UTF-8 dans une adresse de courrier électronique : soit dans le nom de domaine, soit dans la partie locale, soit dans les deux.
IANA	Autorité chargée de la gestion de l'adressage sur Internet. Ses fonctions comprennent : ▪ La gestion de la zone racine, des domaines .int et .arpa, et des ressources pratiques des IDN. ▪ La coordination de l'ensemble total d'IP et de numéros AS, qui est fournie principalement aux registres Internet régionaux (RIR). ▪ Les systèmes de numérotation des protocoles Internet sont gérés ensemble avec les organismes de normalisation.
ICANN	La mission de l'ICANN est de garantir un Internet mondial sûr, stable et unifié. Pour contacter une personne sur Internet, vous devez saisir une adresse sur votre ordinateur ou autre dispositif : un nom ou un numéro. Cette adresse doit être unique pour permettre aux ordinateurs de s'identifier entre eux. L'ICANN coordonne ces identificateurs uniques à l'échelle mondiale. La société ICANN a été fondée en 1998 en tant qu'organisation à but non lucratif, reconnue d'utilité publique. Elle rassemble au sein de sa communauté des participants du monde entier.
IDN	Nom de domaine internationalisé Les IDN sont des noms de domaine incluant des caractères utilisés dans la représentation locale des langues autres que celles écrites avec les vingt-six lettres de l'alphabet latin de base « a – z », les chiffres « 0-9 » et le tiret « - ».
IDNA	Noms de domaine internationalisés dans les applications.

ccTLD IDN	Il s'agit d'un domaine de premier niveau géographique qui comprend des caractères au-delà des 26 lettres de l'alphabet latin de base « a-z ». Exemples : ▪ .рф (Russie) ▪ .مصر (Égypte) ▪ .السعودية (Arabie saoudite)
IETF	Le Groupe de travail de génie Internet (IETF) est une vaste communauté internationale ouverte qui regroupe des concepteurs de réseau, des opérateurs, des vendeurs et des chercheurs soucieux du bon fonctionnement de l'Internet et de l'évolution de son architecture. Il est ouvert à toute personne intéressée. L'IETF élabore des normes Internet, en particulier, les normes liées à l'ensemble de Protocoles Internet (TCP/IP) et aux protocoles utilisés dans le Web tels que le HTPP et le TLS.
Langue	Mode de communication humain, écrit ou parlé, qui consiste à utiliser des mots de manière structurée et conventionnelle.
Punycode	Il s'agit d'un algorithme qui représente l'UTF-8 dans les sous-ensemble de caractères limité de l'ASCII supporté par le système des noms de domaine (DNS). Punycode est utilisé dans les étiquettes A dans le cadre relatif aux noms de domaine internationalisés dans les applications (IDNA).
Bureau d'enregistrement	Organisation au sein de laquelle les noms de domaine sont enregistrés par les utilisateurs. Le bureau d'enregistrement conserve ces informations de contact et envoie les données techniques à un annuaire central connu sous le nom de « registre ».
Opérateur de registre	La base de données principale faisant autorité et regroupant tous les noms de domaine enregistrés dans chaque domaine de premier niveau.
RFC	Un appel à commentaires (RFC) est un document officiel du Groupe de travail de génie Internet (IETF) fruit des travaux de rédaction du comité et de l'examen ultérieur des parties intéressées. Certains (mais pas tous) des RFC documentent les normes Internet approuvées.
Script	Ensemble de lettres ou de caractères utilisés à l'écrit et représentant les sons d'une langue.
Nom de domaine de second niveau	Dans la hiérarchie du système des noms de domaine (DNS), un domaine de second niveau (SLD ou 2LD) est un domaine situé directement en dessous d'un domaine de premier niveau (TLD). Par exemple, dans exemple.com, exemple est le domaine de second niveau du TLD .com.

Étiquette U	L'étiquette U est une chaîne IDNA valide de caractères Unicode incluant au moins un caractère non ASCII. Elle peut être convertie en une étiquette A et vice-versa sans perte d'information.
Logiciel prêt à l'UA ou préparation pour l'intégration de l'UA	Logiciel capable d'accepter, stocker, traiter, valider et afficher tous les domaines de premier niveau ainsi que tous les IDN et adresses de courrier électronique de la même manière.
Unicode	Norme d'encodage de caractères universels. Elle définit la façon dont les caractères individuels sont représentés dans les fichiers de texte, les pages web et autres types de documents. Unicode a été conçue afin de prendre en charge des caractères des langues du monde entier. Elle peut supporter 1 000 000 de caractères environ. Voir : http://unicode.org.
UTF	Format de transformation Unicode. Il s'agit d'un mode de transformation des points de code Unicode en un flux d'octets. L'UTF-8 est l'UTF préféré pour la gestion des IDN et des EAI. L'UTF-8 convertit Unicode en octets à 8 bits.
M3AAWG	Le Groupe de travail anti-abus pour la messagerie, les programmes malveillants et les mobiles (M3AAWG) est le lieu où les membres du secteur se rassemblent afin d'organiser la lutte contre les réseaux zombies, les programmes malveillants, les spams, les virus, les attaques par déni de service et autres types d'exploitation en ligne. Voir : https://www.m3aawg.org/.
W3C	Le Consortium mondial du web (W3C) est une communauté internationale où les organisations membres, le personnel à temps plein et le public travaillent ensemble pour élaborer les normes du Web. Voir : https://www.w3.org/.
WHATWG	Le Groupe de travail sur la techologie de l'application d'hypertexte Web (WHATWG) est une communauté de personnes intéressées par l'évolution du Web à travers des normes et des essais. Le WHATWG a été fondé par des représentants d'Apple, la Fondation Mozilla et le logiciel Opera en 2004, après un atelier du W3C. Voir https://whatwg.org/.
ZWJ	Le liant sans chasse est un caractère non imprimable utilisé dans la typographie informatisée de certains scripts, tels que l'arabe ou l'hindi. Lorsqu'il est placé entre deux caractères qui autrement ne seraient pas connectés, un ZWJ permet de les imprimer sous leur forme connectée.

ZWNJ	L'antiliant sans chasse est un caractère non imprimable utilisé dans l'informatisation de systèmes d'écriture qui ont recours à des ligatures. Dans le cas de certaines langues et certains scripts, un bon nombre des lettres de l'alphabet connectent naturellement avec la lettre suivante lorsqu'elles sont écrites en un mot, formant une ligature. Afin d'afficher correctement certains préfixes, suffixes et mots composés, toutefois, le ZWNJ est utilisé pour remplacer ce comportement naturel de joindre des lettres et les empêcher de se joindre à la lettre suivante (mais sans l'ajout d'un espace entre les deux).

Consultez l'intégralité du glossaire de l'ICANN ici : https://www.icann.org/icann-acronyms-and-terms/.

RFC et normes clés

RFC RELATIFS AUX IDN	
RFC 3492	**Punycode : Codage bootstring d'Unicode pour les noms de domaine internationalisés dans les applications (IDNA)** Le RFC 3492 décrit Punycode comme : *« une syntaxe de codage de transfert simple et efficace conçue afin d'être utilisée avec les noms de domaine internationalisés dans les applications (IDNA) ».* Punycode transforme exclusivement et de manière réversible une chaîne Unicode en une chaîne ASCII. Ce RFC définit un algorithme général appelé Bootstring. Cet algorithme permet à une chaîne de points de code de base de représenter exclusivement toute chaîne de points de code tirée d'un ensemble plus large. https://tools.ietf.org/html/rfc3492
RFC 5890	**Noms de domaine internationalisés dans les applications (IDNA) : définitions et document-cadre** Ce RFC décrit le cadre et le protocole habituels pour la révision de noms de domaine internationalisés dans les applications (IDNA). https://tools.ietf.org/html/rfc5890

RFC 5891	**Protocole portant sur les noms de domaine internationalisés dans les applications (IDNA)** Ce RFC précise le mécanisme du protocole portant sur les noms de domaine internationalisés dans les applications (IDNA), eu égard à l'enregistrement et à la recherche d'IDN, de façon à ne pas avoir à modifier le DNS. https://tools.ietf.org/html/rfc5891
RFC 5892	**Points Unicode et noms de domaine internationalisés dans les applications (IDNA)** Le RFC 5892 indique les règles permettant de décider si un point de code, pris isolément ou dans son contexte, est candidat à l'inclusion dans un nom de domaine internationalisé (IDN). https://tools.ietf.org/html/rfc5892
RFC 5893	**Scripts écrits de droite à gauche pour les noms de domaine internationalisés dans les applications (IDNA)** Ce RFC fournit une nouvelle règle Bidi pour les étiquettes des noms de domaine internationalisés dans les applications (IDNA), eu égard à l'utilisation de scripts écrits de droite à gauche dans des noms de domaine internationalisés. https://tools.ietf.org/html/rfc5893
RFC 5894	**Noms de domaine internationalisés dans les applications (IDNA) : contexte, explications et fondements** Ce document informatif donne un aperçu d'un système révisé visant à gérer les nouvelles versions d'Unicode et fournit des supports explicatifs pour ses composantes. https://tools.ietf.org/html/rfc5894
RFC 5895	**Correspondance des caractères pour les noms de domaine internationalisés dans les applications (IDNA) 2008** Ce RFC décrit les mesures pouvant être prises dans le cadre d'une mise en œuvre entre la réception des retours des utilisateurs et l'adoption de points de code autorisés en vertu du nouveau protocole IDNA (2008). Il décrit une opération qui doit être effectuée sur les retours des utilisateurs afin de préparer ces retours à être utilisés dans un protocole « sur le réseau ». Il comprend également une procédure de mise en œuvre générale pour la correspondance. https://tools.ietf.org/html/rfc5895
RFC RELATIFS AUX EAI	

RFC 6530	**Aperçu et cadre des adresses de courrier électronique internationalisées** Cette norme introduit une série de spécifications qui définissent les mécanismes et les extensions de protocole requis afin de soutenir pleinement les adresses de courrier électronique internationalisées. Le présent document décrit la façon dont les différents éléments liés à l'internationalisation des adresses de courrier électronique s'imbriquent et les relations entre les principales spécifications liées au transport, aux formats de l'en-tête et à la gestion des messages. https://tools.ietf.org/html/rfc6530
RFC 6531	**Extension SMTP pour les adresses de courrier électronique internationalisées** Ce document définit une extension au protocole simple de transfert de courrier afin que les serveurs puissent communiquer sur leur capacité à accepter et traiter des adresses de courrier électronique internationalisées et des en-têtes d'adresses de courrier électronique internationalisées. https://tools.ietf.org/html/rfc6531
RFC 6532	**En-têtes d'adresses de courrier électronique internationalisées** Ce document rapporte l'amélioration apportée au format de message Internet et aux MIME qui permet d'utiliser Unicode dans des adresses de courrier électronique et la plupart des champs d'en-tête. Il rapporte l'amélioration apportée au format de message Internet (RFC 5322) et aux MIME qui permet d'utiliser directement l'UTF-8 et pas seulement l'ASCII dans les valeurs de champs d'en-têtes, y compris les adresses de courrier électronique. Un nouveau type de support, message/global, est défini pour les messages qui utilisent ce format étendu. Cette spécification lève aussi la restriction s'appliquant aux MIME eu égard au fait d'avoir des codages de transfert de contenu non identitaire sur tout sous-type du type de message de haut niveau afin que les parties message/global puissent être transmises en toute sécurité entre les infrastructures de courrier existantes . https://tools.ietf.org/html/rfc6532
RFC 6533	**Statut d'envoi internationalisé et notifications d'élimination** Cette spécification ajoute un nouveau type d'adresse pour les adresses de courrier électronique internationalisées afin que l'adresse du destinataire originale avec des caractères non ASCII puisse être correctement préservée même après un déclassement. Elle fournit également des types de support de renvoi de contenu mis à jour pour les notifications de statut d'envoi et les notifications d'élimination de messages afin d'encourager l'utilisation du nouveau type d'adresse.

	https://tools.ietf.org/html/rfc6533
RFC 8398	**Adresses de courrier électronique internationalisées dans les certificats X.509** Le présent document définit une nouvelle forme de nom pour inclure dans le champ « otherName » du certificat X.509 une extension de nom alternatif du détenteur et une extension de nom alternatif de l'émetteur permettant que le détenteur du certificat soit associé à une adresse de courrier électronique internationalisée. https://tools.ietf.org/html/rfc8398.
RFC 8399	**Mises à jour relatives à l'internationalisation du RFC 5290** Les mises à jour au RFC 5280 décrites dans le présent document fournissent une harmonisation avec la spécification de 2008 pour les noms de domaine internationalisés (IDN) et apportent le support des adresses de courrier électronique internationalisées dans les certificats X.509. https://tools.ietf.org/html/rfc8399

Principales normes

ISO 10646 (Unicode)	Afin de fournir une base technique commune pour le traitement d'informations électroniques dans plusieurs langues, l'Organisation internationale de normalisation (ISO) a élaboré une norme internationale de codage appelée ISO 10646. La norme ISO 10646 permet d'unifier les règles en matière de codage de caractères dans les principales langues du monde, y compris les caractères chinois traditionnels et simplifiés. Ce vaste jeu de caractères s'appelle jeu universel de caractères (UCS). Le même jeu de caractères est défini par la norme Unicode qui prévoit également de nouvelles propriétés des caractères ainsi que d'autres détails d'application présentant un grand intérêt pour les responsables de la mise en œuvre. Unicode est un système de codage de caractères conçu par le Consortium Unicode afin de prendre en charge l'échange, le traitement et l'affichage des textes écrits de toutes les principales langues du monde. La norme ISO 10646 et

	Unicode définissent plusieurs formes de codage de leur répertoire commun : UTF-8, UCS-2, UTF-16, UCS-4 et UTF-32. http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182
GB18030 (Chine)	GB 18030-2000 est une norme du gouvernement chinois qui prévoit l'utilisation d'une page de code étendue pour le marché chinois en plus de l'UTF-8. Le code de traitement interne pour le répertoire de caractères peut et doit être Unicode. Toutefois, la norme indique que les fournisseurs de logiciels doivent garantir une rotation complète entre la norme GB18030 et le code de traitement interne. Tous les produits actuellement vendus ou qui seront vendus en Chine doivent sans exception planifier la migration de la page de code afin de prendre en charge la norme GB18030. La norme GB18030 est une « norme obligatoire » et le gouvernement chinois réglemente le processus de certification afin de renforcer le déploiement de ladite norme. http://icu-project.org/docs/papers/unicode-gb18030-faq.html

Ressources en ligne

API	Interfaces de programmation d'applications (API) de Windows https://www.msdn.microsoft.com/enus/library/windows/desktop/ff818516%28v=vs.85%29.aspx API SharePoint https://msdn.microsoft.com/en-us/library/office/jj860569.aspx Liste publique de suffixes https://publicsuffix.org/list/public_suffix_list.dat Liste de TLD de l'ICANN faisant autorité http://data.iana.org/TLD/tlds-alpha-by-domain.txt API Android http://developer.android.com/guide/index.html API MAC IOS https://developer.apple.com/library/mac/navigation Cadre .Net https://msdn.microsoft.com/en-us/library/system.text.encoding(v=vs.110).aspx
Sécurité Unicode	Considérations de sécurité liées à Unicode http://www.unicode.org/reports/tr36 Mécanismes de sécurité Unicode http://www.unicode.org/reports/tr39
Groupements de caractères Unicode	Points de code Unicode https://www.unicode.org/versions/Unicode12.0.0/ch02.pdf; pages 44-54 Aperçu de la norme GB18030 http://icu-project.org/docs/papers/gb18030.html Tableau de correspondance entre BG18030-2000 et Unicode faisant autorité http://source.icu-project.org/repos/icu/data/trunk/charset/data/xml/gb-18030-2000.xml Normalisation Unicode https://unicode.org/reports/tr15/
Exploits Unicode	Section 3.1, « Exploits de l'UTF-8 » dans le rapport technique Unicode n° 36 http://unicode.org/reports/tr36/#UTF-8_Exploit Meilleures pratiques du M3AAWG en matière de prévention des abus Unicode https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf Aperçu des abus Unicode et tutoriel du M3AAWG

	https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf Voir aussi : http://www.unicode.org
Divers	URL http://tools.ietf.org/html/rfc3986 Le système des noms de domaine : une explication non technique - Pourquoi la résolvabilité universelle est importante http://www.internic.net/faqs/authoritative-dns.html Glossaire de l'ICANN https://www.icann.org/icann-acronyms-and-terms/

Vous souhaitez davantage d'informations ?

Le Groupe directeur sur l'acceptation universelle (UASG) et la communauté sont prêts à fournir leurs conseils aux développeurs de logiciels et aux responsables de leur mise en œuvre.

Contactez-nous afin de partager vos idées et suggestions en la matière sur info@uasg.tech.
Rejoignez la liste de diffusion sur l'acceptation universelle à http://tinyurl.com/ua-discuss
Pour d'autres informations sur cette initiative, consultez http://www.icann.org/universalacceptance