



# Présentation de l'acceptation universelle

Mark Svancarek et Luisa Villa

# À propos du présent document

## Objectif

Les technologies de l'Internet, dont ses composantes de nommage, sont en perpétuelle évolution. Ces dernières années, un grand nombre de nouveaux TLD dotés de caractères ASCII et de noms de domaine de premier niveau internationalisés ont été introduits par l'ICANN. Par exemple, on peut citer `.nyc`, `.hsbc`, `.eco` et `.ストア`. Toutefois, la réponse apportée à l'évolution en matière de nommage n'a pas été assez rapide. Bon nombre d'applications et de services n'ont pas été mis à jour afin de pouvoir assurer la gestion de ces nouveaux TLD. Cela a des conséquences sur l'expérience utilisateur. Par exemple :

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

---

**Bon nombre d'applications et de services n'ont pas été mis à jour afin de pouvoir assurer la gestion des nouveaux TLD. Cela a des conséquences sur l'expérience utilisateur.**

---

À moins que le logiciel ne reconnaisse et ne puisse traiter les nouveaux domaines, état connu sous le nom d'**acceptation universelle**, il sera impossible d'offrir une expérience cohérente et positive aux internautes. Par conséquent, le présent document propose une introduction générale au concept d'acceptation universelle afin de faciliter le développement de logiciels intégrant l'acceptation universelle.

## Public cible

- Développeurs de logiciels
- Directeurs de la technologie
- Ensemble de la communauté technique

## Structure du document

- Partie 1 **Concepts de base liés à l'acceptation universelle** tels que les noms de domaine, le système des noms de domaine, l'ASCII et l'Unicode, le Punycode, l'internationalisation des adresses de courrier électronique et autres concepts fondamentaux.
- Partie 2 Les **cinq critères de l'acceptation universelle** ainsi que les **bonnes pratiques** pour chacun de ces critères. Comprend également des **scénarios utilisateurs** et des **pratiques de non-conformité** en matière d'acceptation universelle, les exigences techniques et les défis actuels.
- Partie 3 **Sujets pointus** tels que les scripts écrits de droite à gauche, l'algorithme Bidi, la normalisation et la casse.
- Partie 4 Contient le **glossaire** ainsi que de précieuses **ressources en ligne**.

**Vous souhaitez davantage d'informations ?**

L'UASG et la communauté sont en mesure, si besoin est, de donner des conseils aux développeurs de logiciels et aux responsables de l'implémentation.

- **Contactez-nous** afin de partager vos idées et suggestions en la matière sur [info@uasg.tech](mailto:info@uasg.tech)
- **Rejoignez la discussion relative à l'acceptation universelle** sur <http://tinyurl.com/ua-discuss>
- Pour en **savoir davantage** sur les travaux menés, veuillez consulter <http://www.icann.org/universalacceptance>

**Sommaire**

Introduction..... 6

    Bref historique de l'internationalisation des noms de domaine..... 6

    L'acceptation universelle, un élément indispensable ..... 6

Partie 1 : Concepts de base liés à l'acceptation universelle ..... 8

    Nom de domaine ..... 8

    Système des noms de domaine (DNS)..... 8

    Domaines de premier niveau (TLD) ..... 8

    Domaines génériques de premier niveau (gTLD)..... 9

    Jeux de caractères et scripts..... 9

    ASCII et Unicode ..... 9

    Noms de domaine internationalisés (IDN) et Punycode..... 10

    Adresses de courrier électronique et ..... 11

    internationalisation des adresses de courrier électronique (EAI) ..... 11

    Génération de lien dynamique (linkification) ..... 12

Partie 2 : Les effets de l'acceptation universelle ..... 13

    Les cinq critères de l'acceptation universelle ..... 13

    Scénarios utilisateurs ..... 14

    Non-respect des pratiques en matière d'acceptation universelle ..... 16

Exigences techniques pour l'intégration de l'acceptation universelle ..... 17

    Critères de haut niveau ..... 17

    Impératifs pour les développeurs ..... 18

        Principe directeur pour la réalisation de l'acceptation universelle : la loi Postel..... 19

        Bonnes pratiques en matière de développement et de mise à jour de logiciels à des fins d'intégration de l'acceptation universelle ..... 19

    Sources faisant autorité pour les noms de domaine ..... 26

        Zone racine du DNS ..... 26

        Liste publique de suffixes ..... 26

Autres défis..... 27

    Généralités ..... 27

    E-mail de type IDN et différences avec l'EAI ..... 27

    Défis en matière de linkification ..... 28

Troisième partie : Sujets pointus..... 30

    Scripts complexes..... 30

        Langues écrites de droite à gauche et conformité à Unicode ..... 30

        L'algorithme de Bidi ..... 30

La règle Bidi pour les noms de domaine.....	31
Liants.....	32
Homoglyphe et caractères prêtant à confusion .....	33
Normalisation et casse.....	33
Normalisation .....	33
Casse .....	35
Partie 4 : Glossaire et autres ressources .....	37
Glossaire.....	37
RFC .....	39
Principales normes.....	42
Ressources en ligne.....	43
Remerciements .....	45
Changements de version .....	46

## Introduction

### Bref historique de l'internationalisation des noms de domaine

Dans les années 1970, les caractères disponibles à des fins d'enregistrement des noms de domaine étaient limités à un sous-ensemble de caractères **ASCII** (lettres a-z, chiffres 0-9 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 afin de tenir compte des besoins liés à l'utilisation croissante au niveau mondial de l'Internet en tant que ressource collective. Aujourd'hui, la majorité des internautes ne sont pas anglophones. Toutefois, la principale langue utilisée sur l'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 afin de soutenir des représentations non ASCII de noms de domaine dans des scripts locaux géographiquement diversifiés (par exemple 普遍接受-□□.世界, ua-test.世界, etc.).

Le Conseil d'administration de la Société pour l'attribution des noms de domaine et des numéros sur Internet (**ICANN**) a approuvé le processus visant à introduire de nouveaux noms de domaine de premier niveau géographiques (ccTLD) IDN en octobre 2009, le premier ajout de ccTLD IDN dans la zone racine ayant eu lieu en mai 2010. En juin 2011, le Conseil d'administration a approuvé et autorisé le lancement du **programme des nouveaux domaines génériques de premier niveau (gTLD)**, qui comprenait de nouveaux TLD ASCII et IDN. Le premier lot de TLD issu de ce programme a été ajouté à la zone racine en 2013. L'ajout de ccTLD IDN et de nouveaux TLD a considérablement augmenté le rythme d'ajout de TLD dans la zone racine.

Dix ans après, l'IETF a publié ses directives relatives aux IDN et grâce au programme des nouveaux TLD, plus de mille nouveaux TLD ont désormais été introduits. En dépit de tous ces efforts, bon nombre de logiciels et applications n'intègrent pas l'acceptation universelle. Cela cause des problèmes aux internautes, notamment à ceux dont les langues sont écrites dans des scripts comprenant des caractères non ASCII.

### L'acceptation universelle, un élément indispensable

Afin de suivre la cadence de ce nouvel univers de TLD, de nouveaux logiciels doivent être créés et les anciens logiciels et anciennes applications doivent être mis à jour. L'état consistant à respecter pleinement ce nouvel univers de TLD est appelé **acceptation universelle**.

L'**acceptation universelle** correspond à 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. En d'autres termes, chaque adresse web valide renvoie au site web recherché et chaque adresse de courrier électronique valide envoie des messages à la destination escomptée. En raison de la rapide évolution du panorama des noms de domaine, de nombreux systèmes ne reconnaissent pas ou ne traitent pas convenablement les nouveaux noms de domaine, principalement du fait qu'ils peuvent se présenter sous un format non ASCII, du fait que les logiciels n'ont pas intégré les TLD récemment introduits, ou du fait que la durée de leur TLD varie. Il en va de même pour les adresses de courrier électronique qui incorporent ces nouvelles extensions.

Le **Groupe directeur sur l'acceptation universelle (UASG)**, une initiative menée par la communauté s'appliquant à toute l'industrie et soutenue par l'ICANN, tâche de sensibiliser ainsi que d'identifier et de

---

**L'acceptation universelle correspond à 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.**

---

## Présentation de l'acceptation universelle (UASG 007)

résoudre les problèmes associés à l'acceptation universelle des noms de domaine, afin de garantir une expérience cohérente et positive pour les internautes du monde entier.

## Partie 1 : Concepts de base liés à l'acceptation universelle

Cette section vise à donner un aperçu des principaux termes et concepts qu'il est nécessaire de comprendre afin de consulter les sections plus détaillées du présent document.

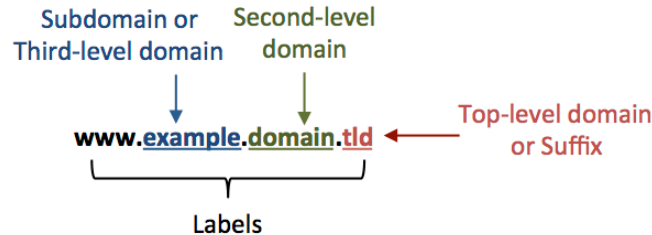
### Nom de domaine

Un nom de domaine est une chaîne de texte avec des points utilisée comme identifiant technique convivial pour les ordinateurs et réseaux sur l'Internet. Par exemple :

`www.domain.tld`

Comment lire un nom de domaine :

- Chaque point représente un **niveau** dans la hiérarchie du système des noms de domaine (DNS).
- Un domaine de premier niveau (TLD) est souvent qualifié de **suffixe** à la fin d'un nom de domaine.
- Les termes ou caractères individuels entre les points sont appelés **étiquettes**. Pour les langues ou scripts écrits de **gauche à droite (LTR)**,<sup>1</sup> l'étiquette la plus à droite représente le domaine de premier niveau.
- La deuxième étiquette en partant de la fin représente le **domaine de second niveau**.
- Toute étiquette figurant avant le domaine de second niveau est considérée comme un **sous-domaine** du domaine de second niveau (parfois appelé **domaine de troisième niveau**).



### Système des noms de domaine (DNS)

Chaque ressource sur l'Internet se voit attribuer une adresse qui sera utilisée par le protocole Internet (IP). Dans la mesure où il est difficile de se souvenir des adresses IP, le DNS fournit un mappage entre les adresses IP et les noms de domaine lisibles par une personne. Les serveurs fournissant collectivement un DNS public existent à des adresses bien connues sur l'Internet.

### Domaines de premier niveau (TLD)

Les noms de domaine lisibles par une personne sont gérés par des organisations connues sous le nom de **registres**. Lorsqu'un nom de domaine est enregistré, il comprend plusieurs chaînes de texte représentant plusieurs niveaux de domaines, chacun séparé par un « . ». Dans les scripts LTR, le

---

<sup>1</sup> Les langues ou scripts écrits de droite à gauche (RTL) seront abordés ultérieurement dans le présent document.



niveau de domaine le plus à droite est le domaine de premier niveau (TLD). Certains TLD sont délégués vers des pays ou des territoires spécifiques. On parle de TLD géographiques (ccTLD).

### Domaines génériques de premier niveau (gTLD)

En 2013, l'ICANN (l'organisation chargée de créer et mettre à jour les cessions de TLD) a approuvé la création d'un grand nombre de nouveaux TLD. Ces nouveaux TLD peuvent représenter des marques, des communautés d'intérêt, des communautés géographiques (villes, régions) ainsi que des concepts plus génériques. Collectivement, l'ensemble de ces nouveaux TLD sont connus sous le nom de domaines génériques de premier niveau (gTLD).

Exemples de TLD communs	Exemples de ccTLD	Exemples de nouveaux gTLD
.com	Chine = .cn	.app
.gov	Allemagne = .de	.lawyer
.info	États-Unis = .us	.shopping
.org		.panasonic
		.osaka

### Jeux de caractères et scripts

Les langues sont écrites à l'aide de systèmes d'écriture. La plupart des systèmes d'écriture utilisent un script, c'est-à-dire un jeu de caractères graphiques utilisé pour la forme écrite d'une ou de plusieurs langues. Un faible nombre de systèmes d'écriture ont recours à plus d'un script à la fois. Ces caractères ou scripts peuvent être reconnus par les personnes. Toutefois, ils ne servent pas aux ordinateurs. Un ordinateur doit disposer d'un script codé afin de pouvoir assurer un traitement (par exemple afin de résoudre une adresse web). Le mécanisme utilisé à cet effet porte le nom de **mappage de caractères** ou **jeu de caractères codés (CCS)** ou **page de code**.<sup>2</sup> Un mappage de caractères associe des caractères avec des nombres spécifiques. Un grand nombre de pages de code différentes ont été créées au fil du temps à différentes fins, mais eu égard au sujet qui nous concerne, nous nous concentrerons uniquement sur deux d'entre elles : l'ASCII et l'Unicode.

### ASCII et Unicode

Dans les exemples de TLD ci-dessus, l'ensemble des chaînes de texte sont représentées à l'aide du jeu de caractères latins. Ce jeu de caractères est inclus dans le schéma de codage de caractères connu sous le nom de Code américain normalisé pour l'échange d'information (ASCII). L'ASCII est un schéma de codage ancien basé sur l'anglais. Pour des raisons historiques, c'est devenu le schéma de codage de caractères standard sur l'Internet. L'ASCII utilise uniquement 7 bits par caractère, ce qui limite le jeu à 128 caractères, tous ne pouvant être utilisés dans les noms de domaine. Les noms de domaine sont limités aux caractères A-Z, aux chiffres 0-9 et au tiret « - ».

---

<sup>2</sup> Il existe certaines subtilités concernant ces termes qui ne relèvent pas directement de l'acceptation universelle. Si vous souhaitez en savoir davantage sur la terminologie, voici un bon point de départ : <https://tools.ietf.org/html/rfc6365>

Tableau ASCII - ISO 8859-1 (latin-1) <sup>3</sup>

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
20		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
30	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

Étant donné que la plupart des systèmes d'écriture n'ont pas recours au jeu de caractères latins, d'autres schémas de codage ont été adoptés. Unicode, également connu sous le nom de **jeu de caractères codés universel (UCS)**, est en mesure de coder plus d'un million de caractères. Chacun des caractères Unicode est appelé **point de code**. La forme la plus courante d'Unicode est appelée **format de transformation du jeu de caractères codés universel 8-bits (UTF-8)**.

Afin de consulter tous les tableaux de codes de caractères Unicode, cliquez ici : <http://unicode.org/charts>

### Noms de domaine internationalisés (IDN) et Punycode

L'utilisation d'Unicode permet aux noms de domaine de contenir des caractères non ASCII. Comme vu précédemment dans le présent document, les noms de domaine qui utilisent des caractères non ASCII sont appelés noms de domaine internationalisés (IDN).<sup>4</sup> La partie internationalisée d'un nom de domaine peut correspondre à tout niveau, pas uniquement le TLD mais aussi les autres étiquettes.

Dans la mesure où le DNS lui-même n'utilisait avant que des caractères ASCII,<sup>5</sup> il est devenu nécessaire de créer un schéma de codage supplémentaire afin de permettre aux points de code d'Unicode non ASCII d'être convertis en chaînes ASCII, et vice versa. L'algorithme qui met en œuvre ce schéma de codage 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 chaînes sont appelées **préfixes ACE**.<sup>6</sup>

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

<sup>3</sup> Source : université d'État de Californie. 1997. *Tableau ASCII - ISO 8859-1 (latin-1) avec noms des entités HTML*.

[http://web.calstatela.edu/faculty/jchen13/Docs/CS120/Lectures/ASCIITable\\_with\\_HTML\\_Entity\\_Names.htm](http://web.calstatela.edu/faculty/jchen13/Docs/CS120/Lectures/ASCIITable_with_HTML_Entity_Names.htm)

<sup>4</sup> Il convient de noter que tous les caractères non ASCII ne constituent pas un IDN.

<sup>5</sup> Pour le statut actuel, voir <http://tools.ietf.org/html/rfc6055#section-3>

<sup>6</sup> Le préfixe encodage compatible ASCII (ACE) est utilisé afin de distinguer les étiquettes codées par Punycode des étiquettes ASCII ordinaires.

La seule utilisation définie par<sup>7</sup> les RFC de l'algorithme Punycode consiste à exprimer des domaines internationalisés. Toutefois, plutôt que de mettre en œuvre Unicode, certains développeurs choisissent d'appliquer Punycode à d'autres scénarios.

#### Exemples d'IDN (imaginaires)

exemple.みんな (codage Punycode = exemple.xn--q9jyb4c)  
大坂.info (codage Punycode = xn--uesx7b.info)  
みんな.大坂 (codage Punycode = xn--q9jyb4c.xn--uesx7b)

Pour en savoir davantage, consultez la FAQ sur les IDN : <http://unicode.org/faq/idn.html>

#### Adresses de courrier électronique et

#### internationalisation des adresses de courrier électronique (EAI)

Les adresses de courrier électronique sont constituées de deux parties :

1. Une partie locale (le nom d'utilisateur avant le caractère « @ »)
2. Un domaine (après le caractère « @ »).

La partie domaine peut contenir n'importe quel TLD, y compris un nouveau TLD. Les deux parties peuvent être des étiquettes U Unicode.

#### Scripts Left to Right (LTR)

User Domain TLD  
name part ↓  
↓ ↓ ↓  
user@example.app

#### Scripts Right to Left (RTL)

TLD Domain User  
↓ part name ↓  
↓ ↓ ↓  
app.مثال@مستخدم

NOTE : Un autre format, à savoir les adresses de courrier électronique de type IDN, sera abordé par la suite.

<sup>7</sup> RFC : appel à commentaires. Pour de plus amples informations, consultez le glossaire des termes à la partie 4 du présent document.

#### Exemples d'adresses de courrier électronique (imaginaires) contenant des IDN

<code>utilisateur@exemple.</code>	(utilise un TLD internationalisé)
<code>みんな</code>	(utilise un domaine de second niveau internationalisé)
<code>utilisateur@大坂.info</code>	(utilise un nom d'utilisateur internationalisé et un nouveau gTLD)
<code>用門@exemple.avocat</code>	

L'internationalisation des adresses de courrier électronique (EAI) oblige à utiliser Unicode dans toutes les parties de l'adresse de courrier électronique. Chacun des exemples ci-dessus pourrait être exprimé en tant qu'EAI, et il s'agit du format préféré.

#### 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 uniquement en saisissant une chaîne qui ressemble à une adresse web, une adresse de courrier électronique ou un chemin d'accès réseau. Par exemple, saisir « `www.icann.org` » dans le corps d'un courrier électronique peut se transformer en un lien cliquable `http://www.icann.org`, automatiquement créé si l'application traite « `www.` » comme un préfixe spécial ou « `.org` » comme un suffixe spécial.

La linkification devrait marcher pour l'ensemble des adresses web, adresses de courrier électronique ou chemins d'accès réseau correctement saisis.

## Partie 2 : Les effets de l'acceptation universelle

### Les cinq critères de l'acceptation universelle

Comme indiqué dans la section précédente, l'acceptation universelle correspond à 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. Ces cinq critères sont décrits ci-dessous.

<p><b>1. Accepter<sup>8</sup></b></p>	<p><b>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 utilisé par une application logicielle ou un service en ligne.</b></p> <p>Les applications et services permettent aux noms de domaine et adresses de courrier électronique d'être :</p> <ul style="list-style-type: none"> <li>• Saisis dans des interfaces utilisateur, ET/OU</li> <li>• Reçus d'autres applications et services via des API.</li> </ul>
<p><b>2. Valider<sup>9</sup></b></p>	<p><b>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.</b></p> <p>La validation a pour but de veiller à ce que les informations saisies sont valides ou au moins manifestement pas invalides. En d'autres termes, la validation garantit l'exactitude syntaxique des informations fournies.</p> <p>Pour les noms de domaine et les adresses de courrier électronique, bon nombre de programmeurs utilisent des heuristiques (par exemple ils vérifient qu'un TLD a le « bon » nombre de caractères, ou que les caractères appartiennent au jeu de caractères ASCII). Toutefois, ces heuristiques ne sont plus applicables car l'Internet est en train de changer :</p> <ul style="list-style-type: none"> <li>• Les noms de domaine et les adresses de courrier électronique peuvent désormais inclure des caractères Unicode (non ASCII)</li> <li>• La liste de TLD s'allonge</li> <li>• Un TLD peut avoir jusqu'à 63 caractères.</li> </ul>
<p><b>3. Stocker</b></p>	<p><b>Le stockage est le processus via lequel une adresse de courrier électronique ou un nom de domaine est stocké en tant que chaîne de caractères dans une base de données ou un dossier utilisé par une application logicielle ou un service en ligne.</b></p> <p>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</p>

<sup>8</sup> Dans le présent document, on distingue l'acceptation de la validation. Dans la pratique, ces deux étapes peuvent se chevaucher.

<sup>9</sup> Dans le présent document, on distingue l'acceptation et le traitement de la validation. Dans la pratique, ces deux étapes peuvent se chevaucher.

	<p>courrier électronique. Indépendamment de la durée de vie des données, ils peuvent être stockés sous :</p> <ul style="list-style-type: none"> <li>• Des formats définis par les RFC, OU</li> <li>• D'autres formats pouvant être facilement traduits à partir de ou vers des formats définis par les RFC (c'est beaucoup moins souhaitable).</li> </ul> <p>Bien que les RFC imposent l'utilisation de l'UTF-8, d'autres formats sont disponibles en vertu d'un codage historique. Voir ci-dessous la section consacrée aux « Bonnes pratiques ».</p>
<b>4. Traiter<sup>10</sup></b>	<p><b>Le <i>traitement</i> intervient lorsqu'une adresse de courrier électronique ou un nom de domaine est utilisé par une application ou un service afin de mener une activité (par exemple la recherche ou le tri d'une liste) ou est transformé en un format distinct (par exemple le stockage ASCII en tant qu'Unicode).</b></p> <p>Le traitement implique l'utilisation de noms de domaine et de chaînes de courrier électronique dans une fonctionnalité. Une validation supplémentaire peut être réalisée lors du traitement. Il n'y a pas de limite quant au nombre de modes de traitement des noms de domaine et des adresses de courrier électronique (exemples : « Identifier toutes les personnes associées à la Nouvelle-Zélande car elles ont un nom avec un TLD <code>.nz</code> » ; « Identifier tous les pharmaciens car ils ont une adresse de courrier électronique <code>utilisateur@exemple.pharmacie</code> » ; « Identifier les pare-feux susceptibles de filtrer les requêtes DNS qui ne relèvent pas de leurs politiques »).</p>
<b>5. Afficher</b>	<p><b>L'<i>affichage</i> intervient dès qu'une adresse de courrier électronique ou un nom de domaine est délivré au sein d'une interface utilisateur.</b></p> <p>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 dans Unicode. Si ces conditions ne sont pas respectées, des transformations propres aux applications peuvent être requises.</p>

### 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 :

<b>Enregistrement d'un nouveau TLD</b>	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 <code>nomduclient@exemple.marque</code> .
--	---

<sup>10</sup> Dans le présent document, on distingue le traitement de la validation. Dans la pratique, ces deux étapes peuvent se chevaucher.

	<p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>• Les applications web acceptent que ces nouvelles adresses de courrier électronique « @exemple.marque » soient tout aussi valides que celles ayant des TLD tels que .com, .net, .org.</li> </ul>
<b>Accès à un gTLD</b>	<p>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.</p> <p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>• Même s'il s'agit d'un nouveau TLD, tout navigateur que l'utilisateur souhaite utiliser doit afficher l'adresse web dans sa forme originelle et accéder au site comme demandé par l'utilisateur. Le navigateur n'affiche pas du texte en Punycode à l'utilisateur à moins que cela ne profite, d'une quelconque façon, à ce dernier.</li> </ul>
<b>Utilisation d'une adresse de courrier électronique contenant un nouveau gTLD en tant qu'identité numérique</b>	<p>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.</p> <p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>• Même si le domaine utilisé dans l'adresse de courrier électronique est nouveau, 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.</li> </ul>
<b>Accès à un IDN</b>	<p>Un utilisateur accède à l'URL d'un IDN en saisissant une adresse dans un navigateur ou en cliquant sur un lien dans un document.</p> <p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>• 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.</li> </ul>
<b>Utilisation d'une adresse de courrier électronique internationalisée pour les courriers électroniques</b>	<p>Un utilisateur a acquis plusieurs adresses de courrier électronique, certaines internationalisées (par exemple <a href="mailto:Info@普遍接受-□□.世界">Info@普遍接受-□□.世界</a>).</p> <p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>• L'utilisateur peut envoyer des messages à et recevoir des messages de toute adresse de courrier électronique à l'aide d'un client de messagerie.</li> </ul>
<b>Utilisation d'une adresse de courrier électronique en tant qu'identité numérique</b>	<p>Un utilisateur acquiert une adresse de courrier électronique internationalisée et l'utilise en tant qu'identité pour accéder à sa banque et au programme de fidélité d'une compagnie aérienne.</p> <p>L'acceptation universelle implique que :</p>

	<ul style="list-style-type: none"> <li>La banque ou le site de la compagnie aérienne accepte l'identité EAI exactement comme s'il s'agissait d'une autre identité de courrier électronique.</li> </ul>
<b>Création dynamique d'un hyperlien dans une application</b>	<p>Un utilisateur saisit une adresse web dans un document ou dans le corps d'un courrier électronique.</p> <p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>Les règles utilisées par l'application afin de générer automatiquement un hyperlien sont identiques même si l'adresse est une EAI ou contient un nouveau TLD.</li> </ul>
<b>Développement d'une application</b>	<p>Un développeur écrit une application qui accède à des ressources web.</p> <p>L'acceptation universelle implique que :</p> <ul style="list-style-type: none"> <li>Les outils utilisés par les développeurs incluent des bibliothèques qui permettent l'acceptation universelle en prenant en charge Unicode, les IDN et l'EAI.</li> </ul>

### Non-respect des pratiques en matière d'acceptation universelle

Les pratiques suivantes sont considérées comme étant de **mauvaises pratiques** :

❑	<p>Afficher un texte en Punycode à l'utilisateur sans qu'il n'en tire un bénéfice.</p> <p>Par exemple, montrer le mappage entre une étiquette U et une étiquette A.</p>
❑	<p>Obliger un utilisateur à saisir un texte en Punycode lorsqu'il crée une nouvelle adresse de courrier électronique ou obliger un utilisateur à saisir un texte en Punycode lorsqu'il enregistre un nouveau domaine hébergé.</p>
❑	<p>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.</p>
❑	<p>Utiliser une ancienne liste de TLD même si les nouveaux TLD y sont régulièrement ajoutés.</p>
❑	<p>Montrer l'utilisation interne de texte en Punycode aux utilisateurs.</p> <p>Par exemple, convertir une adresse de courrier électronique internationalisée en une adresse de courrier électronique de type IDN lors de l'envoi d'une réponse à un utilisateur EAI.</p>
❑	<p>Traiter certains noms de domaine comme des termes de recherche et non pas comme des noms de domaine car l'application ne les reconnaît pas en tant que tels.</p>
❑	<p>Configurer des bloqueurs de spam afin de bloquer automatiquement des TLD dans leur intégralité.</p>



# Exigences techniques pour l'intégration de l'acceptation universelle

## Critères 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 du jeu de caractères.**

Voir le [RFC 5892](#).

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

C'est-à-dire qu'il permet les points de code Unicode.

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

Voir le [RFC 3490](#).

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 dans des formats prenant en charge Unicode et pouvant être convertis en/de l'UTF-8.**

Pour de plus amples informations sur l'UTF-8, voir le [RFC 3629](#).

6. **Propose des API publiques qui prennent en charge Unicode et l'UTF-8.**

7. **Propose des API privées qui prennent en charge Unicode et l'UTF-8.**

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

8. **Stocke des données d'utilisateurs dans des formats prenant en charge Unicode et pouvant être convertis en/de l'UTF-8.**

De telles conversions ne seraient visibles que par le propriétaire du produit/service.

9. **Prend en charge toutes les chaînes de nom de domaine figurant dans la liste de TLD faisant autorité de l'ICANN et la liste publique de suffixes destinée à la communauté, indépendamment de la longueur ou du jeu de caractères.**

Voir <https://newgtlds.icann.org/en/program-status/delegated-strings>.

10. **Peut envoyer des courriers électroniques à des destinataires et en recevoir de ces derniers indépendamment du nom de domaine ou du jeu de caractères.**

Voir le [RFC 6530](#).

11. **Traite les adresses EAI de la même façon que leurs équivalents en Punycode (format de courrier électronique IDN).**

## Impératifs 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 et les nouveaux TLD. Cette section aborde la façon dont les développeurs peuvent intégrer des changements de code qui permettront l'acceptation universelle de tous les nouveaux TLD.

### Principe directeur pour la réalisation de l'acceptation universelle : la loi Postel

Dans le RFC 793, Jon Postel a formulé le principe de robustesse, également connu sous le nom de loi Postel, censé faciliter la mise en œuvre du TCP qui venait alors d'être créé. En informatique, le principe de robustesse est une ligne directrice générale pour la conception des logiciels :

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

En d'autres termes, soyez conservateur dans ce que vous envoyez et libéral dans ce que vous acceptez. Il s'agit également d'une bonne approche eu égard à la gestion des subtilités de l'acceptation universelle actuellement mises en œuvre dans l'écosystème.

### Bonnes pratiques en matière de développement et de mise à jour de logiciels à des fins d'intégration de l'acceptation universelle

Accepter	
□	Toujours proposer des équivalents Unicode.  Les utilisateurs doivent être autorisés, mais non obligés, à saisir du texte à encodage compatible ASCII (soit « Punycode ») au lieu de son équivalent Unicode. Toutefois, Unicode doit être affiché par défaut, du texte en Punycode devant être affiché à l'utilisateur uniquement lorsque cela est avantageux.
!	<b>Ne pas</b> générer d'adresses de courrier électronique de type IDN mais <b>être en mesure</b> 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 de courrier électronique doit prendre en charge le format Unicode, les étiquettes allant jusqu'à 63 caractères, et les chaînes de nom de domaine allant jusqu'à 253 caractères. <ul style="list-style-type: none"><li>• Voir le RFC 1035.</li></ul>

Valider	
□	Valider uniquement ce qui est nécessaire.  <b>Valider uniquement si cela est nécessaire à l'exploitation de l'application ou du service.</b> C'est la façon la plus fiable de garantir que tous les noms de domaine valides sont acceptés dans vos systèmes.
□	Reconnaître que des saisies syntaxiquement correctes peuvent ne pas représenter des noms de domaine ou des adresses de courrier électronique actuellement utilisés sur l'Internet.
!	Si la validation est requise, examiner les éléments suivants : <ul style="list-style-type: none"><li>• Vérifier la partie TLD d'un nom de domaine par rapport à un tableau faisant autorité. Voici quelques exemples de tableaux faisant autorité que vous pouvez utiliser :<ul style="list-style-type: none"><li>○ <a href="http://www.internic.net/domain/root.zone">http://www.internic.net/domain/root.zone</a></li><li>○ <a href="http://data.iana.org/TLD/tlds-alpha-by-domain.txt">http://data.iana.org/TLD/tlds-alpha-by-domain.txt</a></li></ul></li></ul> <p>Voir aussi : <a href="https://www.icann.org/en/system/files/files/sac-070-en.pdf">https://www.icann.org/en/system/files/files/sac-070-en.pdf</a></p>

- Interroger le nom de domaine par rapport au DNS.
  - Envisager d'utiliser l'API GETDNS (<http://getdnsapi.net/>)
- Exiger qu'une adresse de courrier électronique soit répétée plusieurs fois afin d'éviter les erreurs de frappe.
- Valider les caractères sur les étiquettes uniquement afin de déterminer que l'étiquette U ne contient pas de points de code « NON AUTORISÉS » ou de points de code qui ne sont pas attribués dans sa version d'Unicode.
  - Voir le [RFC 5892](#)
- Limiter la validation d'étiquettes à un nombre restreint de règles s'appliquant à toutes les étiquettes définies dans les RFC.
  - Voir le [RFC 5894](#)
- Si une chaîne ressemblant à un nom de domaine contient le caractère arabe point « - » (U+06D4), ou le caractère idéographique point « 。- Veiller à ce que le produit ou la fonctionnalité prend correctement en charge les chiffres.
  - Par exemple : les chiffres ASCII et les représentations de nombres idéographiques asiatiques doivent être traités en tant que nombres.

## Stocker

□	Les applications et services doivent répondre aux normes Unicode en vigueur.
□	<p>Les informations doivent être stockées sous le format UTF-8 (format de transformation Unicode) autant que possible.</p> <p>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. Les formats UTF-7 et UTF-32 doivent être évités.</p>
!	<p>Envisager tous les scénarios dans leur globalité avant de convertir des étiquettes A (Punycode) en étiquettes U et vice versa lors du stockage.</p> <p>Il peut être souhaitable de ne conserver que les étiquettes U dans un dossier ou une base de données, cela simplifiant 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. Envisager de stocker et d'indexer sous les deux formats.</p>
□	<p>Indiquer clairement les adresses de courrier électronique et les noms de domaine lors du stockage afin d'y accéder plus facilement.</p> <p>Les cas où les adresses de courrier électronique et les noms de domaine ont été déposés en utilisant le champ « auteur » d'un document ou « info contact » d'un journal ont entraîné la perte de l'adresse originale.</p>
□	<p>Si vous <i>ne</i> stockez <i>pas</i> sous le format Unicode, vous devez être en mesure de faire correspondre les chaînes stockées sous plusieurs formats.</p> <p>Par exemple, si l'on recherche exemple.みんな, on doit pouvoir également trouver exemple.xn--q9jyb4c.</p>

Traiter	
□	Veiller à ce que toutes les réponses du serveur disposent d'Unicode dans les types de contenu prévus.
□	Indiquer Unicode dans l'en-tête HTTP du serveur web et directement dans un dossier web. <ul style="list-style-type: none"> <li>• Chaque dossier web doit comprendre le jeu de caractères UTF-8.</li> <li>• Il est important de veiller à ce que l'encodage soit précisé sur chaque réponse.</li> </ul>
!	Envisager tous les scénarios dans leur globalité avant de convertir des étiquettes A (Punycode) en étiquettes U et vice versa lors du traitement. <p>Il peut être souhaitable de ne conserver que les étiquettes U dans un dossier ou une base de données, cela simplifiant 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. Envisager de stocker sous les deux formats.</p>
□	Veiller à ce que le produit ou la fonctionnalité prend en charge l'ordre de tri, les recherches et le classement en fonction de spécifications locales/linguistiques, et qu'il permet les recherches et le tri en plusieurs langues.
□	Ne pas utiliser d'encodage d'URL pour les noms de domaine : <ul style="list-style-type: none"> <li>• exemple.みんな est correct</li> <li>• exemple.%E3%81%BF%E3%82%93%E3%81%AA n'est pas correct</li> </ul>
□	Étant donné que la norme Unicode est en perpétuelle développement, 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 « cassent » pas l'expérience utilisateur. <p>Les polices manquantes dans le système d'exploitation sous-jacent peuvent conduire à des caractères non affichables (le caractère « □ » est souvent utilisé pour les représenter) mais cette situation ne devrait pas entraîner d'incident fatal.</p>
□	Utiliser les API sous format Unicode prises en charge.
□	Utiliser les documents relatifs au protocole portant sur les noms de domaine internationalisés dans les applications (IDNA) pour les IDN : <ul style="list-style-type: none"> <li>• le RFC 5891</li> <li>• le RFC 5892</li> </ul>
□	Traiter autant que possible sous le format UTF-8.
□	Mettre à jour les applications et serveurs/services ensemble. <p>Si le serveur est Unicode et que le client ne l'est pas, ou vice versa, les données devront être converties sur chaque page de code chaque fois que les données se déplacent entre le serveur et le client.</p>

□	<p>Effectuer des révisions de code afin d'éviter les attaques liées au dépassement de la mémoire tampon.</p> <p>Lors de la transformation de caractères, il se peut que des chaînes de texte grandissent ou rétrécissent considérablement.</p>
---	--

## Affichage

□	<p>Afficher tous les points de code Unicode qui sont pris en charge par le système d'exploitation sous-jacent.</p> <p>Si une application conserve ses propres ensembles de polices, une prise en charge Unicode complète doit être offerte à l'ensemble de polices proposé par le système d'exploitation.</p>
□	<p>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.</p>
□	<p>Convertir les données non Unicode en données Unicode avant l'affichage.</p> <p>Par exemple, l'utilisateur final doit voir « exemple.みんな » et non pas « exemple.xn--q9jyb4c ». (Cette conversion est un exemple de traitement intégrant l'UA).</p>
□	<p>Afficher Unicode par défaut.</p> <p>Afficher le texte en Punycode pour l'utilisateur uniquement lorsque cela est avantageux.</p>
!	<p>Garder à l'esprit que les adresses à plusieurs scripts deviendront plus courantes.</p> <ul style="list-style-type: none"> <li>• Il se peut que certains caractères Unicode se ressemblent à l'œil humain mais soient différents pour les ordinateurs.</li> <li>• Ne pas présumer que les chaînes à plusieurs scripts sont utilisées à des fins malveillantes, par exemple pour l'hameçonnage.</li> <li>• 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.</li> </ul> <p>Afin d'en savoir davantage sur les impératifs de sécurité Unicode, cliquez ici : <a href="http://unicode.org/reports/tr36">http://unicode.org/reports/tr36</a></p>
□	<p>Utiliser un traitement de compatibilité IDNA Unicode afin de satisfaire les attentes de l'utilisateur.</p> <p>Pour en savoir plus, consultez l'adresse suivante : <a href="http://unicode.org/reports/tr46">http://unicode.org/reports/tr46</a></p>
□	<p>Prendre connaissance des caractères non attribués et non autorisés pour les noms de domaine.</p> <ul style="list-style-type: none"> <li>• Voir le <a href="#">RFC 5892</a></li> </ul>

## Unicode

□	Utiliser les API sous format Unicode prises en charge.
□	Ne pas construire ses propres API pour : <ul style="list-style-type: none"> <li>• Les conversions des formats des chaînes</li> <li>• Déterminer quel script comprend une chaîne</li> <li>• Déterminer si une chaîne contient une combinaison de scripts</li> <li>• La normalisation/décomposition Unicode</li> </ul>
□	Ne pas utiliser l'UTF-7 ou l'UTF-32. <ul style="list-style-type: none"> <li>• L'UTF-7 n'est généralement pas utilisé comme représentation originale dans les applications car il s'agit d'un format très difficile à traiter. En dépit de son avantage en termes de taille par rapport à la combinaison UTF-8 avec Quoted-Printable ou base64, l'Internet Mail Consortium ne recommande son utilisation.</li> <li>• Le principal inconvénient de l'UTF-32 est son inefficacité en termes d'espace ; il utilise en effet quatre octets par point de code. Les caractères non BMP sont tellement rares dans la plupart des textes [précisions à apporter] qu'on pourrait décider de les ignorer pour des raisons de taille, l'UTF-32 étant ainsi deux fois plus grand que l'UTF-16 et quatre fois plus grands que l'UTF-8.</li> </ul>
□	Utiliser Unicode dans les cookies afin qu'ils puissent être lus correctement par les applications.
□	Utiliser le protocole IDNA 2008 et les documents suivants : <ul style="list-style-type: none"> <li>• <a href="#">le RFC 5891</a></li> <li>• <a href="#">le RFC 5892</a></li> </ul>
□	Ne pas utiliser le protocole IDNA 2003 ; dans presque tous les cas, il peut être remplacé par le protocole IDNA 2008.
□	Ne pas présumer automatiquement que les API externes peuvent consommer des données qui ont été converties en NFKC <sup>11</sup> .
!	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 les documents ( <a href="#">le RFC 5892</a> ), ses tableaux IDNA doivent être dérivés de la version d'Unicode prise en charge plus généralement sur le système. Quant à l'enregistrement, les tableaux ne doivent pas refléter la dernière version d'Unicode mais doivent être cohérents.
!	Valider les caractères sur les étiquettes uniquement afin de déterminer que l'étiquette U ne contient pas de points de code « NON AUTORISÉS » <sup>12</sup> ou de points de code qui ne sont pas attribués dans leur version d'Unicode.

<sup>11</sup> **NFKC** (*Forme de normalisation avec composition de compatibilité*) : les caractères sont décomposés par compatibilité, puis recomposés par équivalence canonique. Voir : <http://unicode.org/reports/tr15>

<sup>12</sup> **NON AUTORISÉS** : points de code ne devant être inclus dans les IDN. Voir : <https://tools.ietf.org/html/rfc5892>

□	<p>Limiter la validation d'étiquettes à un nombre restreint de règles s'appliquant à toutes les étiquettes :</p> <ul style="list-style-type: none"> <li>• Pas de principaux signes combinatoires</li> <li>• Les conditions bidirectionnelles sont respectées si des caractères écrits de droite à gauche apparaissent</li> <li>• Toutes règles contextuelles relatives aux caractères de liaison (et, plus généralement, aux caractères CONTEXTJ<sup>13</sup>) sont testées</li> </ul>
!	<p>Ne pas utiliser l'UTF-16 sauf en cas de disposition explicite contraire (comme pour certaines API de Windows).</p> <p>Si l'on utilise l'UTF-16, veuillez noter que 16 bits ne peuvent contenir que la gamme de caractères allant de 0x0 à 0xFFFF, et qu'un niveau de complexité supplémentaire est utilisé afin de stocker les valeurs inférieures à cette gamme (0x10000 à 0x10FFFF). On fait cela en utilisant des paires d'unités de code connues sous le nom de demi-codets. Si la gestion des paires de demi-codets n'est pas minutieusement testée, elle peut entraîner des bugs délicats et éventuellement des vulnérabilités.</p>

### Linkification

□	<p>Si une chaîne ressemblant à un nom de domaine contient le caractère arabe point « - » (U+06D4), ou le caractère idéographique point « ｡ » (U+3002), le convertir en point « . » (U+002E).</p>
---	--

### Généralités

□	<p>Utiliser des ressources faisant autorité afin de valider les noms de domaine.</p> <p>Ne pas formuler des hypothèses heuristiques telles que « tous les TLD ont une longueur de 2, 3, 4 ou 6 caractères ».</p>
□	<p>Veiller à ce que le produit ou la fonctionnalité prend correctement en charge les chiffres.</p> <p>Par exemple, les chiffres ASCII et les représentations de nombres idéographiques asiatiques doivent tous être traités en tant que nombres.</p>
!	<p>Rechercher des adresses de courrier électronique dans des endroits inattendus :</p> <ul style="list-style-type: none"> <li>• Métadonnées d'artistes, d'auteurs, de photographes ou relatives à des droits d'auteurs</li> <li>• Métadonnées relatives aux polices</li> <li>• Enregistrements de contacts du DNS</li> <li>• Informations binaires</li> <li>• Informations de soutien</li> <li>• Coordonnées de l'OEM</li> </ul>

<sup>13</sup> CONTEXTJ : règle contextuelle pour les contrôles de liaison. Voir : <https://tools.ietf.org/html/rfc5892>



	<ul style="list-style-type: none"> <li>• Enregistrement, feedback et autres formulaires</li> </ul>
!	<p>Rechercher d'éventuels chemins d'accès IRI<sup>14</sup> dans des endroits inattendus :</p> <ul style="list-style-type: none"> <li>• Noms de machine à une étiquette indépendamment de la page de code du système chargée</li> <li>• Noms de machine complets indépendamment de la page de code du système chargée</li> </ul>
□	Utiliser GB18030 (Chine) pour un soutien en chinois <sup>15</sup> en plus de l'UTF-8.
!	<p>Limiter les points de code autorisés lors de la génération de nouveaux noms de domaine et adresses de courrier électronique :</p> <p>Tous les produits qui utilisent des adresses de courrier électronique doivent accepter les adresses de courrier électronique internationalisées, en autorisant les caractères &gt; U+007f. C'est-à-dire qu'aucun caractère &gt; U+007f n'est interdit. 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 de courrier électronique. N'utiliser que la liste suivante de caractères autorisés pour les IDN : <a href="http://unicode.org/reports/tr36/idn-chars.txt">http://unicode.org/reports/tr36/idn-chars.txt</a></p> <p>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. (NOTE : la loi Postal imposerait tout de même aux logiciels d'accepter de telles chaînes si elles étaient présentées).</p>
!	<p>Garder à l'esprit que l'acceptation universelle ne peut pas toujours être mesurée uniquement via des tests automatisés.</p> <p>Par exemple, il n'est pas toujours possible d'évaluer la mesure dans laquelle une application ou un protocole gère les ressources réseau et il est parfois plus judicieux de vérifier la conformité via un examen spécifique fonctionnel et un examen de la conception.</p>
!	<p>Ne pas présumer 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.</p> <p>Comprendre comment les noms réseau 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 réseau :</p> <ul style="list-style-type: none"> <li>• Politique de groupe</li> <li>• Requête LDAP</li> <li>• Dossiers de configuration</li> <li>• Registre Windows</li> <li>• Transfert vers/depuis une autre composante/fonctionnalité</li> </ul>
□	Effectuer des révisions de code afin d'éviter les attaques liées au dépassement de la mémoire tampon.

<sup>14</sup> IRI Identificateurs de ressource internationalisés. Voir : <https://www.ietf.org/rfc/rfc3987.txt>

<sup>15</sup> GB 18030-2000 est une norme du gouvernement chinois qui prévoit l'utilisation d'une page de code étendue pour le marché chinois. Voir : <http://icu-project.org/docs/papers/unicode-gb18030-faq.html>

- |   |
|---|
| <ul style="list-style-type: none"><li>• Dans Unicode, la casse des chaînes peut se développer : Fluß → FLUSS → fluss</li><li>• Lors de la conversion de caractères, il se peut que le texte grandisse ou rétrécisse considérablement.</li></ul> |
|---|

## Sources faisant autorité pour les noms de domaine

### Zone racine du DNS

Il existe quelques options pour la liste de TLD faisant autorité. La première option est la zone racine du DNS même. Elle est signée DNSSEC ; la liste est donc dûment authentifiée. Vous pouvez obtenir la zone racine à partir de l'un des liens suivants :

- <http://www.internic.net/domain/root.zone>
- <http://www.dns.icann.org/services/authoritative-dns/index.html>
- <http://data.iana.org/TLD/tlds-alpha-by-domain.txt>

### Liste publique de suffixes

La liste publique de suffixes (PSL), gérée par des bénévoles de la Fondation Mozilla, constitue une liste précise des suffixes de noms de domaine. Cette liste est un ensemble de noms du DNS ou de caractères génériques concaténés avec des points et codés à l'aide de l'UTF-8. Si vous devez utiliser la PSL en tant que source faisant autorité pour les noms de domaine, votre logiciel doit recevoir régulièrement des mises à jour de la PSL. Ne pas conserver des copies statiques de la PSL dans votre logiciel sans mécanisme de mise à jour. Vous pouvez utiliser le lien ci-dessous afin que votre application télécharge périodiquement une liste mise à jour. La liste est mise à jour une fois par jour via Github :

- [https://publicsuffix.org/list/public\\_suffix\\_list.dat](https://publicsuffix.org/list/public_suffix_list.dat)

## Autres défis

### Généralités

<b>Encodages variables d'IDN</b>	<p>Dans certaines applications, les IDN sont encodés :</p> <ul style="list-style-type: none"> <li>• Dans Punycode, comme pour l'IDNA, si le nom est identifié en tant que nom Internet, MAIS</li> <li>• Dans l'UTF-8, si le nom est identifié en tant que nom sur le réseau local (« Intranet »).</li> </ul>
<b>Mécanisme de détection et de conversion de jeux de caractères</b>	<p>Certaines anciennes applications de courrier électronique étaient encodées dans une page de code locale et ne disposaient pas de mécanisme défini de détection et de conversion de jeux de caractères tel que requis. Cela était spécialement vrai pour les en-têtes de courrier électronique (À, CC, BCC, objet).</p>
<b>Incapacité à gérer les protocoles ne relevant pas du DNS</b>	<p>Certaines applications prenant en charge les IDNA (par exemple IE7+) tombent en panne pour les protocoles ne relevant pas du DNS. Cela pourrait affecter l'accès aux ressources à l'aide de protocoles ne relevant pas du DNS.</p>
<b>Mécanisme de gestion de multiples adresses de courrier électronique en une identité d'utilisateur unique</b>	<p>Lorsqu'un même utilisateur utilise des faux noms pour plusieurs adresses de courrier électronique, il peut être compliqué de gérer ces adresses en tant qu'identité d'utilisateur unique.</p> <p>Des programmes de courrier électronique peuvent diriger le trafic destiné à ces faux noms vers la même boîte de réception mais l'application peut toujours percevoir que ces courriers électroniques relèvent de différentes identités.</p>

### Conseils pour développeurs de logiciels



Lorsque l'on autorise un utilisateur à générer un nom de domaine ou une adresse de courrier électronique, penser à éviter d'utiliser des caractères prêtant visuellement à confusion afin d'empêcher les attaques homographes. N'utiliser que la liste suivante de caractères autorisés pour les IDN : <http://unicode.org/reports/tr36/idn-chars.txt>

### E-mail de type IDN et différences avec l'EAI

L'EAI est définie comme n'utilisant qu'Unicode ; les étiquettes A (Punycode) ne sont pas autorisées. Pourtant, les développeurs ont parfois adapté des logiciels et services de courrier électronique afin de gérer les adresses de courrier électronique de style IDN plutôt que de procéder à une pleine conversion vers Unicode.

Les IDN pouvant encodés par Punycode, certains logiciels existants permettent que la partie de l'IDN d'une adresse de courrier électronique 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) :

**Tous les logiciels ne traiteront pas ces deux courriers électroniques de type IDN comme équivalents d'un point de vue fonctionnel**

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

Toutefois, certains logiciels ne traiteront pas, en termes de robustesse, ces adresses de la même façon, même si elles sont toutes deux valides, car rien n'impose aux logiciels de convertir une étiquette A (c'est-à-dire « xn--q9jyb4c ») en son étiquette U équivalente (c'est-à-dire « みんな ») avant de comparer. Cela peut conduire à une expérience utilisateur imprévisible. L'expérience utilisateur peut devenir particulièrement troublante si certains logiciels convertissent des étiquettes U en étiquettes A dans un souci de « compatibilité » ; au fur et à mesure que des messages sont envoyés ou transmis, les adresses qui sont visuellement différentes pour un utilisateur, ou incapables de procéder à une recherche ou à un tri comme requis, peuvent augmenter.

Dans l'exemple ci-dessous, certains logiciels peuvent essayer de convertir même la partie locale de l'adresse de courrier électronique à l'aide du Punycode, créant ainsi quelque chose qui ressemble à une étiquette A dans la partie locale de l'adresse. Cela est interdit en vertu des RFC existants, et il est très probable que cela entraîne des ratés dans la réception de courriers électroniques par certains systèmes et des difficultés en

**Ne jamais convertir la partie locale d'une adresse de courrier électronique à l'aide de Punycode**

❑ `用□@exemple.みんな`  
❑ `xn--youq53b@exemple.xn--q9jyb4c`

matière de recherche et de tri comme expliqué ci-dessus.

Des logiciels et services robustes intégrant l'UA peuvent être en mesure de gérer et de traiter tous ces formats de manière identique, même ceux qui ne respectent pas les RFC. Toutefois, les logiciels intégrant l'UA ne doivent pas générer uniquement des adresses de courrier électronique EAI.

## 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](http://www.icann.org) » dans le corps d'un courrier électronique peut entraîner la création automatique d'un lien cliquable vers <http://www.icann.org> si l'application traite « [www.](http://www.icann.org) » comme un préfixe spécial ou « [.org](http://www.icann.org) » comme un suffixe spécial.

La linkification doit fonctionner de la même façon pour l'ensemble des adresses de courrier électronique, noms de courrier électronique ou chemins d'accès réseau bien formés.

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:>)

Elle a recours à des algorithmes et des règles créés par des développeurs de logiciels afin de déterminer si une chaîne doit 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 s'y atteler 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.
<a href="http://exemple.com">http://exemple.com</a>	Créer un hyperlien car le protocole est explicite.
<a href="http:exemple.com">http:exemple.com</a>	Pas de linkification car syntaxe incorrecte (il manque //).
<a href="http://example.a">http://example.a</a>	Pas de linkification car les politiques de l'ICANN imposent aux TLD d'avoir au moins deux caractères. NB : Cette syntaxe pourrait être prise en charge par un réseau interne.
<a href="http://exemple..ab">http://exemple..ab</a>	Pas de linkification car syntaxe incorrecte (points consécutifs).
<a href="http:// 普遍接受-□□.世界">http:// 普遍接受-□□.世界</a>	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](http://www) » implique « <http://www> »).

Exemple de chaîne	Comportement/résultat escompté
<a href="http://www.exemple.com">www.exemple.com</a>	Créer un hyperlien car le protocole est implicite. <sup>16</sup>
<a href="mailto:étiquette@exemple.com">étiquette@exemple.com</a>	Créer <a href="mailto:étiquette@exemple.com">mailto:étiquette@exemple.com</a> car le protocole est implicite.

3. Changer le point idéographique « 。 » (U+3002) et le point arabe « - » (U+06D4) en point « . » (U+002E) (par exemple <http://田中。com> => <http://田中.com>) si la chaîne est pour le reste bien formée.
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 doit régulièrement être mise à jour de manière dynamique.

<sup>16</sup> Note : il se pourrait que le site web impose aux utilisateurs finaux de saisir <https://> au lieu de <http://>. Si c'est le cas, l'hyperlien pourrait alors ne pas pouvoir résoudre ou renvoyer une page d'erreur.

## Troisième partie : Sujets pointus

### Scripts complexes

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

### Langues écrites de droite à gauche et conformité à Unicode

La plupart des scripts s'écrivent de gauche à droite lorsque le texte est présenté en lignes horizontales. Toutefois, pour plusieurs scripts tels que l'arabe et l'hébreu, le texte horizontal s'affiche de droite à gauche. Le texte peut également être bidirectionnel (de gauche à droite et de droite à gauche) 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.

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). La **sens de base**<sup>17</sup> attribué à la phrase déterminera l'ordre dans lequel le texte est affiché.

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)** en tant que LTR (de gauche à droite). Les lettres des scripts écrits de droite à gauche sont fortement typées en tant que RTL (de droite à gauche). 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 en tant que LTR ou RTL dans Unicode car elles peuvent être utilisées dans l'un ou l'autre des scripts. Elles sont ainsi qualifiées de **caractères neutres** ou **faibles**. Les caractères faibles sont dotés d'une orientation confuse. Des exemples de ce type de caractères comprennent :

- Les chiffres européens
- Les chiffres indo-arabes

---

<sup>17</sup> En HTML, le sens de base soit découle du sens par défaut du document, c'est-à-dire de gauche à droite, soit est explicitement défini par l'élément parent le plus proche qui utilise l'attribut `dir`.

- 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 insécable.

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

- Les tabulations
- Les 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 une 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 principal sens de base. Par exemple :

- exemple. مثال

Sauf en cas d'annulation directionnelle, les **nombres** sont toujours encodés (et saisis) gros-boutistes<sup>18</sup>, et les chiffres écrits LTR. La faible orientation s'applique uniquement au placement du nombre dans son intégralité.

Pour en savoir davantage sur l'algorithme Bidi, cliquez ici : <http://unicode.org/reports/tr9/tr9-11.html>

### 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. Il existe une règle qui détermine les conditions à respecter pour les étiquettes dans les noms de domaine Bidi. Cette règle est disponible à la section 2 du RFC 5893 : <https://tools.ietf.org/html/rfc5893>

---

<sup>18</sup> « Gros-boutiste et petit-boutiste sont des termes décrivant l'ordre dans lequel une séquence d'**octets** est stockée dans une mémoire informatique. Gros-boutiste est l'ordre dans lequel le « gros bout » (la valeur la plus significative de la séquence) est stocké en premier (au niveau de l'adresse de stockage la plus faible). Petit-boutiste est l'ordre dans lequel le « petit bout » (la valeur la moins significative de la séquence) est stocké en premier. »

Source : <http://searchnetworking.techtarget.com/definition/big-endian-and-little-endian>

## Liants

Certaines langues utilisent des scripts alphabétiques dans lesquels des phonèmes uniques sont écrits à l'aide de deux caractères que l'on appelle **digraphe**. En d'autres termes, un digraphe est un groupe de deux lettres successives qui représentent un son unique (ou **phonème**).

### Exemples de digraphes en anglais

*ch* (pour *church*)  
*ph* (*phone*)

*th* (*then*)  
*th* (*think*)

*sh* (*shoe*)

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 l'esperluette (&) qui vient de la jonction des lettres latines « e » et « t » (« et »).

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 être effectuée. Un exemple d'interprétation différente est disponible à la section 4.3 du RFC 5894 : <https://tools.ietf.org/html/rfc5894>

Le Consortium Unicode prévoit deux principales stratégies visant à déterminer le comportement liant d'un caractère donné après l'application de l'algorithme Bidi :

- « Lors de sa conception, une implémentation peut renvoyer au support de stockage original afin de voir s'il y avait des caractères ZWNJ ou ZWJ<sup>19</sup> adjacents.
- Sinon, l'implémentation 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 a été appliqué, ces informations hors bande peuvent alors être utilisées à des fins de conception adéquate. »<sup>20</sup>

Si les registres ne sont pas attentifs à la gestion des chaînes qui pourraient avoir différentes interprétations en vertu du protocole IDNA 2003 et de la spécification actuelle, il est possible que les différences puissent être utilisées en tant que composante pour des attaques de mise en correspondance de nom ou de confusion de nom. Il convient d'être attentif.

Pour en savoir davantage sur les liants, consultez la section 4.3 du RFC 5894 : <https://tools.ietf.org/html/rfc5894>

<sup>19</sup> Pour en savoir davantage sur les ZWNJ/ZWJ, cliquez ici : <http://www.unicode.org/L2/L2005/05307-zwj-zwnj.pdf>

<sup>20</sup> Source : Mark Davis, Aharon Lanin, Andrew Glass. 2015. *Unicode*. <http://unicode.org/reports/tr9>



## Homoglyphe et caractères prêtant à confusion

Les **homoglyphes** sont des caractères qui, en raison de leurs similitudes en termes de taille et de forme, peuvent sembler identiques au premier coup d'œil.

### Exemples d'homoglyphes

Caractère cyrillique а	=	Numéro Unicode 0430
Caractère latin a	=	Numéro Unicode 0061

Afin d'empêcher l'enregistrement des noms de domaine prêtant à confusion, les registres peuvent avoir recours à la procédure de « regroupement des homoglyphes ».<sup>21</sup>

Le **regroupement des homoglyphes** comprend les étapes suivantes : enregistrement d'un IDN, puis regroupement automatique par le système d'enregistrement de l'ensemble des homoglyphes de ce nom (s'il y en a). Cela signifie que plusieurs noms de domaine sont regroupés en une seule fois, et qu'aucun des autres noms de domaine de ce groupe ne peut être enregistré.

Le regroupement des homoglyphes constitue une bonne pratique permettant aux registres d'éviter d'éventuelles hameçonnages visant à tromper l'utilisateur avec des caractères prêtant visuellement à confusion.

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

- [http://www.unicode.org/reports/tr39/#Confusable\\_Detection](http://www.unicode.org/reports/tr39/#Confusable_Detection)

Pour une liste des homoglyphes, consultez :

- <http://homoglyphs.net>

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

- Aperçu des abus Unicode et tutoriel du M3AAGW  
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf>
- Meilleures pratiques du M3AAGW en matière de prévention des abus Unicode  
<https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf>

## Normalisation et casse

### Normalisation

La normalisation Unicode aide à déterminer si deux chaînes Unicode quelconques sont équivalentes. Certains caractères peuvent être représentés dans Unicode par plusieurs séquences de code. On parle d'**équivalence Unicode**. Unicode prévoit deux types d'équivalences :

- L'équivalence canonique (NFD)

<sup>21</sup> <https://www.icann.org/resources/pages/idn-guidelines-2011-09-02-en>

- L'équivalence de compatibilité (NFK)

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 :

**Exemples de caractères canoniquement équivalents**

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

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.

**Exemples de caractères présentant une équivalence de compatibilité**

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.

Afin d'éviter des 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<sup>22</sup> pour l'ensemble des contenus.

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 :

- Seules les chaînes NON transformées par la NFKC<sup>23</sup> sont valides.
- Lorsque deux applications partagent des données Unicode mais les normalisent différemment, des erreurs et des pertes de données peuvent se produire.

<sup>22</sup> NFC : décomposition canonique suivie de composition canonique.

<sup>23</sup> NFKC : décomposition de compatibilité suivie de composition canonique.

- Les formes de normalisation peuvent rester stables au fil du temps. Autrement dit, une chaîne doit rester normalisée conformément à toutes les futures versions d'Unicode (compatibilité à rebours).

### 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, l'extraction de données par des applications client et entraîner des pertes ou une corruption de données.

Pour en savoir plus sur les formes de normalisation, consultez : <http://www.unicode.org/reports/tr15>

### Casse

La **casse** est le processus consistant à élaborer deux textes à la casse différente mais sinon identiques. Le mappage de [a-z] en [A-Z] ne pose aucun souci pour la plupart des fichiers textes se présentant uniquement sous le format ASCII. Toutefois, des problèmes commencent à se poser avec des langues qui utilisent des caractères supplémentaires.

Unicode définit le mappage de casse par défaut pour chaque point de code Unicode. Il existe des **mappages de casse communs** et des **mappages de casse complets** :

- Les **mappages de casse communs** sont ceux qui présentent un mappage simple en un point de code (principalement en minuscules) à correspondance unique.
- Les **mappages de casse complets** sont ceux qui exigent en règle générale plus d'un caractère Unicode.

Pour le W3C, il est très important de déterminer<sup>24</sup> si les valeurs sont restreintes au sous-ensemble ASCII d'Unicode ou si le vocabulaire permet l'utilisation de caractères (tels que des accents sur des lettres latines ou une large gamme d'Unicode dont des scripts non latins) qui pourraient avoir des exigences complexes en matière de casse.<sup>25</sup>

### Conseils pour développeurs de logiciels



Envisager la normalisation Unicode en plus de la casse.

Pour en savoir davantage sur la normalisation Unicode, consultez :

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

<sup>24</sup> 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 au développement des **normes du web**. Voir : <https://www.w3.org>

<sup>25</sup> Source : A Phillips. 2015. *Character Model for the World Wide Web: String Matching and Searching*. <https://www.w3.org/TR/charmod-norm>

Pour des recommandations en matière de casse, consultez :

- [https://www.w3.org/International/wiki/Case\\_folding](https://www.w3.org/International/wiki/Case_folding)

## Partie 4 : Glossaire et autres ressources

### Glossaire

<b>Étiquette A</b>	Représentation avec un codage compatible ASCII (ACE) d'un nom de domaine internationalisé, c'est-à-dire la manière dont il est transmis en interne dans le protocole DNS. Les étiquettes A commencent toujours par le préfixe « xn- ». C'est la différence avec l'étiquette U.
<b>Préfixe ACE</b>	Préfixe avec un encodage compatible ASCII.
<b>Caractères ASCII</b>	Code américain normalisé pour l'échange d'information. Il s'agit des caractères de l'alphabet latin de base et des chiffres arabes. Ils sont aussi inclus dans la catégorie plus vaste de « caractères Unicode » qui constituent la base des IDN.
<b>API</b>	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.
<b>Espace code</b>	Gamme définissant les limites inférieure et supérieure d'un encodage.
<b>Points de code</b>	Un point de code ou une position de code est l'une des valeurs numériques qui constituent l'espace code. Ces points sont utilisés afin de distinguer à la fois le nombre d'un encodage en tant que séquence de bits, et le caractère abstrait d'une représentation graphique spécifique (glyphe).
<b>Zone racine du DNS</b>	La zone racine est l'annuaire central du DNS, c'est-à-dire la composante clé pour la traduction des noms d'hôtes lisibles en adresses IP numériques.
<b>EAI</b>	L'internationalisation des adresses de courrier électronique est une adresse de courrier électronique qui exige d'utiliser Unicode dans toutes les parties de l'adresse.
<b>IANA</b>	Autorité chargée de la gestion de l'adressage sur Internet. Ces fonctions comprennent : <ul style="list-style-type: none"> <li>• La tenue du registre des paramètres de protocole techniques de l'Internet</li> <li>• L'administration de certaines responsabilités associées à la gestion de la zone racine du DNS</li> <li>• L'attribution de ressources de numéros de l'Internet</li> </ul>
<b>ICANN</b>	La Société pour l'attribution des noms de domaine et des numéros sur Internet (ICANN) est une association internationale de droit privé à but non lucratif qui est chargée d'allouer l'espace des adresses du protocole Internet (IP), d'attribuer des identificateurs de protocole, de gérer le système de noms de domaine de premier niveau génériques (gTLD) et géographiques (ccTLD), et d'assurer les fonctions de gestion du système de serveurs racines.

<b>IDN</b>	Noms de domaine internationalisés. 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 « - ».
<b>IDNA</b>	Noms de domaine internationalisés dans les applications.
<b>IDN ccTLD</b>	Domaines de premier niveau géographiques 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 ». Exemples : <ul style="list-style-type: none"> <li>• .рф (Russie)</li> <li>• .صر (Égypte)</li> <li>• .السعودية (Arabie saoudite)</li> </ul>
<b>IETF</b>	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 relatives à l'Internet et, en particulier, des normes liées à la suite des protocoles d'Internet (TCP/IP).
<b>Langue</b>	Mode de communication humain, écrit ou parlé, consistant à utiliser des mots de manière structurée et conventionnelle.
<b>Punycode</b>	Il s'agit d'un algorithme représentant Unicode avec le sous-ensemble de caractères limité d'ASCII pris en charge par le système des noms de domaine. Punycode est utilisé pour l'encodage d'étiquettes dans le cadre des noms de domaine internationalisés dans les applications (IDNA).
<b>Bureau d'enregistrement</b>	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 ».
<b>Registre</b>	Base de données principale faisant autorité et regroupant tous les noms de domaine enregistrés dans chaque domaine de premier niveau.
<b>RFC</b>	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.
<b>Script</b>	Ensemble de lettres ou de caractères utilisés à l'écrit et représentant les sons d'une langue.
<b>Nom de domaine de second niveau</b>	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 <code>exemple.com</code> , <code>exemple</code> est le domaine de second niveau du TLD <code>.com</code> .

<b>Étiquette U</b>	Chaîne IDNA valide de caractères Unicode incluant au moins un caractère non ASCII. Les conversions entre étiquettes U et étiquettes A sont effectuées conformément à la spécification Punycode [RFC3492].
<b>Logiciel intégrant l'UA ou intégration de l'UA</b>	Logiciel intégrant l'acceptation universelle. Il s'agit d'un logiciel capable d'accepter, de stocker, de traiter, de valider et d'afficher, dans la même mesure, tous les domaines de premier niveau ainsi que tous les IDN, hyperliens et adresses de courrier électronique.
<b>Unicode</b>	Norme d'encodage de caractères universels. Elle définit la façon dont les caractères individuels sont représentés dans les fichiers textes, 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 prendre en charge environ 1 000 000 caractères et jusqu'à 4 octets pour chaque caractère. Voir : <a href="http://unicode.org">http://unicode.org</a>
<b>UTF</b>	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.
<b>M3AAWG</b>	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 : <a href="https://www.m3aawg.org/">https://www.m3aawg.org/</a>
<b>W3C</b>	Le Consortium mondial du web (W3C) est une communauté internationale où les <a href="#">organisations membres</a> , le <a href="#">personnel</a> à temps plein et le public travaillent ensemble au développement des <a href="#">normes du web</a> . Voir : <a href="https://www.w3.org/">https://www.w3.org/</a>
<b>ZWJ</b>	Le liant sans chasse est un caractère non imprimable utilisé dans la typographie informatisée de certains scripts complexes tels que le script arabe ou un script indien. Lorsqu'il est placé entre deux caractères qui autrement ne seraient pas connectés, un ZWJ permet de les imprimer sous leur forme connectée.
<b>ZWNJ</b>	L'antiliant sans chasse est un caractère non imprimable utilisé dans l'informatisation de systèmes d'écriture qui ont recours à des ligatures. Lorsqu'il est placé entre deux caractères qui autrement seraient connectés dans une ligature, un ZWNJ permet de les imprimer sous leur forme finale et initiale, respectivement. C'est également la fonction d'une espace, mais un ZWNJ est utilisé lorsqu'il est souhaitable que les termes soient rapprochés ou lorsqu'il est souhaitable de connecter un terme avec son morphème.

Consultez l'intégralité du glossaire de l'ICANN ici : <https://www.icann.org/resources/pages/glossary-2014-02-03-en>

## RFC

### RFC RELATIFS AU PUNYCODE

<b>RFC 3492</b>	<p><b>Punycode : Codage bootstring d'Unicode pour les noms de domaine internationalisés dans les applications (IDNA)</b></p> <p>Le RFC 3492 décrit Punycode comme :</p> <p style="text-align: center;"><i>« une syntaxe de codage de transfert simple et efficace conçue afin d'être utilisée avec les noms de domaine internationalisés dans les applications (IDNA) ».</i></p> <p>Punycode transforme exclusivement et de manière réversible une chaîne Unicode en une chaîne ASCII. Ce RFC définit un algorithme général appelé <b>bootstring</b>. 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.</p> <p><a href="https://tools.ietf.org/html/rfc3492">https://tools.ietf.org/html/rfc3492</a></p>
<b>RFC RELATIFS AUX IDN</b>	
<b>RFC 5890</b>	<p><b>Noms de domaine internationalisés dans les applications (IDNA) : définitions et document-cadre</b></p> <p>Ce RFC décrit le cadre et le protocole habituels pour la révision de noms de domaine internationalisés dans les applications (IDNA).</p> <p><a href="https://tools.ietf.org/html/rfc5890">https://tools.ietf.org/html/rfc5890</a></p>
<b>RFC 5891</b>	<p><b>Protocole portant sur les noms de domaine internationalisés dans les applications (IDNA)</b></p> <p>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.</p> <p><a href="https://tools.ietf.org/html/rfc5891">https://tools.ietf.org/html/rfc5891</a></p>
<b>RFC 5892</b>	<p><b>Points Unicode et noms de domaine internationalisés dans les applications (IDNA)</b></p> <p>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).</p> <p><a href="https://tools.ietf.org/html/rfc5892">https://tools.ietf.org/html/rfc5892</a></p>
<b>RFC 5893</b>	<p><b>Scripts écrits de droite à gauche pour les noms de domaine internationalisés dans les applications (IDNA)</b></p> <p>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.</p> <p><a href="https://tools.ietf.org/html/rfc5893">https://tools.ietf.org/html/rfc5893</a></p>
<b>RFC 5894</b>	<p><b>Noms de domaine internationalisés dans les applications (IDNA) : généralités, explications et fondements</b></p> <p>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.</p>



	<a href="https://tools.ietf.org/html/rfc5894">https://tools.ietf.org/html/rfc5894</a>
<b>RFC 5895</b>	<p><b>Mappage des caractères pour les noms de domaine internationalisés dans les applications (IDNA) 2008</b></p> <p>Ce RFC décrit les mesures pouvant être prises dans le cadre d'une implémentation 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 d'implémentation générale pour le mappage.</p> <p><a href="https://tools.ietf.org/html/rfc5895">https://tools.ietf.org/html/rfc5895</a></p>
<b>RFC RELATIFS AUX EAI</b>	
<b>RFC 6530</b>	<p><b>Aperçu et cadre des adresses de courrier électronique internationalisées</b></p> <p>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.</p> <p><a href="https://tools.ietf.org/html/rfc6530">https://tools.ietf.org/html/rfc6530</a></p>
<b>RFC 6531</b>	<p><b>Extension SMTP pour les adresses de courrier électronique internationalisées</b></p> <p>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.</p> <p><a href="https://tools.ietf.org/html/rfc6531">https://tools.ietf.org/html/rfc6531</a></p>
<b>RFC 6532</b>	<p><b>En-têtes d'adresses de courrier électronique internationalisées</b></p> <p>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 contenus de champs des en-têtes. 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.</p> <p><a href="https://tools.ietf.org/html/rfc6532">https://tools.ietf.org/html/rfc6532</a></p>
<b>RFC 6533</b>	<b>Statut d'envoi internationalisé et notifications relatives à l'élimination</b>

	<p>Cette spécification ajoute un nouveau type d'adresse pour les adresses de courrier électronique internationalisées afin que l'adresse de 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.</p> <p><a href="https://tools.ietf.org/html/rfc6533">https://tools.ietf.org/html/rfc6533</a></p>
--	---

## Principales normes

<p><b>ISO 10646 (Unicode)</b></p>	<p>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, dont 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 l'implémentation.</p> <p>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.</p> <p><a href="http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182">http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63182</a></p>
<p><b>GB18030 (Chine)</b></p>	<p>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.</p> <p><a href="http://icu-project.org/docs/papers/unicode-gb18030-faq.html">http://icu-project.org/docs/papers/unicode-gb18030-faq.html</a></p>
<p><b>Norme technique d'Unicode n° 46 : traitement de compatibilité IDNA Unicode</b></p>	<p>Cette spécification définit un mappage conforme aux exigences normatives du protocole IDNA 2008, et aussi compatible que possible avec le protocole IDNA 2003. Pour les logiciels clients, elle prévoit un comportement répondant le mieux possible aux attentes des utilisateurs eu égard à la gestion des noms de domaine avec les données existantes.</p> <p><a href="http://unicode.org/reports/tr46/">http://unicode.org/reports/tr46/</a></p>

## Ressources en ligne

<b>API</b>	<p>API Windows  <a href="https://www.msdn.microsoft.com/enus/library/windows/desktop/ff818516%28v=vs.85%29.aspx">https://www.msdn.microsoft.com/enus/library/windows/desktop/ff818516%28v=vs.85%29.aspx</a></p> <p>API SharePoint  <a href="https://msdn.microsoft.com/en-us/library/office/jj860569.aspx">https://msdn.microsoft.com/en-us/library/office/jj860569.aspx</a></p> <p>Liste publique de suffixes  <a href="https://publicsuffix.org/list/public_suffix_list.dat">https://publicsuffix.org/list/public_suffix_list.dat</a></p> <p>Liste de TLD de l'ICANN faisant autorité  <a href="http://data.iana.org/TLD/tlds-alpha-by-domain.txt">http://data.iana.org/TLD/tlds-alpha-by-domain.txt</a></p> <p>API Android  <a href="http://developer.android.com/guide/index.html">http://developer.android.com/guide/index.html</a></p> <p>API MAC IOS  <a href="https://developer.apple.com/library/mac/navigation">https://developer.apple.com/library/mac/navigation</a></p> <p>Cadre .Net  <a href="https://msdn.microsoft.com/en-us/library/system.text.encoding(v=vs.110).aspx">https://msdn.microsoft.com/en-us/library/system.text.encoding(v=vs.110).aspx</a></p>
<b>Sécurité Unicode</b>	<p>Impératifs de sécurité Unicode  <a href="http://www.unicode.org/reports/tr36">http://www.unicode.org/reports/tr36</a></p> <p>Mécanismes de sécurité Unicode  <a href="http://www.unicode.org/reports/tr39">http://www.unicode.org/reports/tr39</a></p>
<b>Groupements de caractères Unicode</b>	<p>Points de code Unicode  <a href="http://en.wikipedia.org/wiki/Mapping_of_Unicode_character_planes">http://en.wikipedia.org/wiki/Mapping_of_Unicode_character_planes</a></p> <p>Aperçu de la norme GB18030  <a href="http://en.wikipedia.org/wiki/GB_18030">http://en.wikipedia.org/wiki/GB_18030</a></p> <p>Tableau de mappage entre BG18030-2000 et Unicode faisant autorité  <a href="http://source.icu-project.org/repos/icu/data/trunk/charset/data/xml/gb-18030-2000.xml">http://source.icu-project.org/repos/icu/data/trunk/charset/data/xml/gb-18030-2000.xml</a></p> <p>Normalisation Unicode  <a href="https://en.wikipedia.org/wiki/Unicode_equivalence">https://en.wikipedia.org/wiki/Unicode_equivalence</a></p>
<b>Exploits Unicode</b>	<p>Section 3.1, « Exploits de l'UTF-8 » dans le rapport technique Unicode n° 36  <a href="http://unicode.org/reports/tr36/#UTF-8_Exploit">http://unicode.org/reports/tr36/#UTF-8_Exploit</a></p> <p>Meilleures pratiques du M3AAG en matière de prévention des abus Unicode  <a href="https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf">https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf</a></p> <p>Aperçu des abus Unicode et tutoriel du M3AAG  <a href="https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf">https://www.m3aawg.org/sites/default/files/m3aawg-unicode-tutorial-2016-02.pdf</a></p> <p>Voir aussi :  <a href="http://www.unicode.org">http://www.unicode.org</a></p>

<b>Divers</b>	URL <a href="http://tools.ietf.org/html/rfc3986">http://tools.ietf.org/html/rfc3986</a>  Le système des noms de domaine : une explication non technique - Pourquoi la résolubilité universelle est importante <a href="http://www.internic.net/faqs/authoritative-dns.html">http://www.internic.net/faqs/authoritative-dns.html</a>  Glossaire de l'ICANN <a href="https://www.icann.org/resources/pages/glossary-2014-02-03-en">https://www.icann.org/resources/pages/glossary-2014-02-03-en</a>
---------------	--

## Remerciements

Les auteurs remercient chaleureusement les personnes suivantes pour leurs contributions et leur collaboration au présent document :

Eleeza Agopian  
Gwen Carlson  
Edmon Chung  
Samantha Dickinson  
Don Hollander  
Chantal Lebrument  
Antonietta Mangiacotti  
Richard Merdinger  
Ram Mohan  
David Morrison  
Carolyn Nguyen  
Michael D. Palage  
Kurt Pritz  
André Schappo  
Zheng Song  
Lars Steffen  
Andrew Sullivan  
Dennis Tan  
Winnie Yu

## Changements de version

De la version 8 à la version 9

- Correction des transformations des points Unicode suggérées
- Suppression d'un lien sans intérêt sur les sources faisant autorité