

## Normalisation et état des lieux de la prise en charge de l'amazighe et des tifinaghes<sup>1</sup>

Patrick Andries  
Conseils Hapax, Québec, Canada  
Membre du consortium Unicode

*In this paper, we look at the progress made since Tifinagh characters were encoded in ISO/IEC 10646 and Unicode in 2005. Eight years ago, it was impossible to send documents written in Tifinagh without reference to a private encoded font. Today, you can create HTML pages, XML documents, email in Tifinagh. Now, There is a standard keyboard approved by the Moroccan standard body to enter Tifinagh text, a standard for sorting Tifinagh strings, and Microsoft provides by default in its newest versions a font that supports the Tifinagh. users can now view HTML pages without having to explicitly install a Tifinagh font on their system. Software libraries like ICU also support Tifinagh and it is possible in theory to have internet domain names in Tifinagh.*

*Dans cette contribution, nous nous pencherons sur le chemin parcouru depuis le codage des caractères tifinaghes dans l'ISO 10646 et Unicode en 2005. Il y a huit ans, il était impossible d'envoyer des documents en tifinaghes sans se référer à un codage de police privé. Aujourd'hui, on peut créer des pages HTML, des documents XML en tifinaghes, envoyer des courriels. Il existe un clavier normalisé pour saisir des textes tifinaghes, une norme de tri, Microsoft fournit une police qui prend en charge les tifinaghes. L'utilisateur peut désormais voir des pages HTML sans qu'il n'ait à explicitement installer de polices tifinaghes sur son système. Des bibliothèques logicielles comme ICU prennent également en charge les tifinaghes et il est possible, en théorie, d'avoir des noms de domaine Internet en tifinaghes.*

---

<sup>1</sup> Nous tenons à vivement remercier l'IRCAM et plus particulièrement le directeur du CEISIC, Youssef Aït Ouguengay, pour leur accueil chaleureux et l'organisation du colloque international à Rabat au cours duquel une première version de cet article a été présentée. Nous voulons également ici rendre hommage au prédécesseur de M. Aït Ouguengay, le professeur Lahbib Zenkour, sans lequel la normalisation informatique du tifinaghe n'aurait été ni aussi rapide ni aussi complète.

## Introduction

Le 31 mars 2005, Unicode 4.1 était publié. Il comprenait tous les tfinaghés normalisés de l'IRCAM ainsi que les principaux caractères touaregs et kabyles.

Avant cette date, il n'existait aucune façon de coder de manière normalisée des textes tfinaghés. Chaque producteur utilisait un codage, le plus souvent de son cru, lié à une police particulière. Il était donc virtuellement impossible d'échanger des documents produits par des personnes qui utilisaient des polices tfinaghés différentes. C'était vrai pour les courriels, les documents Word, les pages HTML en général et, bien sûr, celles d'une encyclopédie en ligne naissante comme Wikipédia.

Depuis huit ans bien des choses ont changé : les tfinaghés normalisés sont désormais de plus en plus présents sur Internet et dans d'autres produits et normes informatiques. Nous allons brièvement passer en revue ci-dessous ces heureuses améliorations.

## Les caractères Unicode

Le lecteur de cette contribution connaît certainement les premiers caractères tfinaghés qui ont été codés dans Unicode 4.1 en mars 2005, car ils ont fait l'objet de plusieurs communications de l'IRCAM et de ses chercheurs<sup>2</sup>. Il ignore peut-être certaines précisions apportées depuis 2005 qui se retrouvent dans les dernières versions d'Unicode.

### *Les diacritiques*

Plusieurs variantes tfinaghés modernes utilisent des diacritiques pour compléter les lettres du bloc tfinaghe. C'est ainsi que la notation Hawad utilise des diacritiques du bloc [U+0300-U+036F] commun à de nombreuses transcriptions latines. Ces signes s'utilisent pour représenter des voyelles ou des consonnes étrangères. Dans cette notation, <U+2D35, U+0307> ǣ représente un « a » court, <U+2D49, U+0304> ȓ un « i » long /i:/ et <U+2D31, U+0302> ê permet d'écrire un « p ». On indique certaines voyelles longues à l'aide de deux signes diacritiques, un « é » long /e:/ s'écrit <U+2D49, U+0307, U+0304> ȓ̄. Ces signes sont affichés côte à côte, et non empilés, au-dessus de la lettre de base dans l'ordre d'apparition dans la chaîne codée.



### *Quatre caractères ajoutés dans Unicode 6.0 et 6.1*

Deux caractères tfinaghés ont été ajoutés dans Unicode 6.0 (et bien sûr dans la version correspondante de l'ISO/CEI 10646) :

- U+2D70 𠵰 SÉPARATEUR TIFINAGHE  
= tazaraste
- U+2D7F 𠵿 LIANT DE CONSONNES TIFINAGHE

---

<sup>2</sup> Voir, par exemple, Lahbib ZENKOUAR (2004 : 173-175).

Le caractère U+2D70  est ce signe targui dont Prasse dit<sup>3</sup> « Au Hoggar on nous a donné le séparateur , à l'intérieur duquel s'écrivait la dernière lettre de chaque mot phonétique. »

Les deux figures ci-dessous illustrent l'utilisation de ce séparateur. On remarquera que le signe est réfléchi quand il s'écrit dans un contexte de droite à gauche.

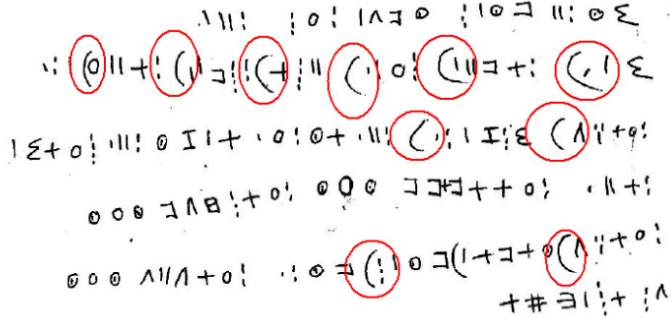


Figure 1 : Exemple d'utilisation du séparateur de mot, Sud de l'Algérie

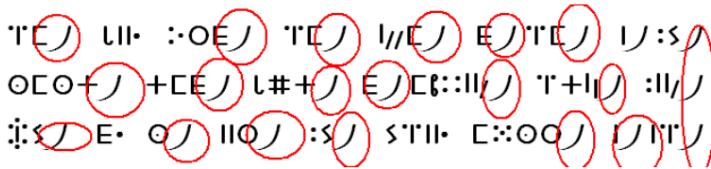



Figure 2 : Utilisation au Niger dans un contexte de gauche à droite


Le *liant de consonnes* U+2D7F  est un caractère de commande qui permet de forcer la formation d'une biconsonne. Nous étudierons son utilisation dans la section 2.3. *Ligatures*.

Unicode 6.1 a par la suite ajouté deux caractères :

- U+2D66 < LETTRE TIFINAGHE YÉ
- U+2D67 -- LETTRE TIFINAGHE YO

YÉ et YO sont simplement deux voyelles préconisées par l'APT (Association pour la promotion des tifinaghs) au Niger pour transcrire les voyelles « é » et « o ».

### **Ligatures**

Dès le codage des tifinaghes dans Unicode 4.1, il était possible de préciser que l'on préférerait obtenir une biconsonne (ou une triconsonne) en insérant un U+200D  *liant sans chasse* entre les consonnes qui constituent la ligature souhaitée. On peut

<sup>3</sup> Karl-G. PRASSE (1972 : 152).

également utiliser U+200C *antiliant sans chasse* entre deux consonnes pour décourager la formation d'une ligature à partir de ces deux lettres.

C'est ainsi que l'on peut demander la formation de la biconsonne « rt » en écrivant <U+2D54, U+U200D, U+2D5C> (Ⓞ<sub>54</sub>Ⓞ<sub>5C</sub>). Les polices pourvues d'une telle ligature devraient en présence de ces caractères afficher le glyphe de celle-ci, par exemple ⊕, ⊞, ⊕Ⓞ, ou ⊕Ⓞ selon la police. Il existe également quelques cas de triconsonnes : parmi celles-ci notons le groupe consonantique « nft » | ⓃⓄⓄ dont la ligature est parfois ⓃⓄⓄ et « nkn » | ⓃⓄⓄ représenté dans certaines régions par ⓃⓄⓄ. Pour bien fixer les idées, on demandera la formation optionnelle de la ligature « nkn » à l'aide de la suite suivante de caractères : <U+2D4F, U+200D, U+2D3E, U+200D, U+2D4F>.

Certaines polices pourront être dépourvues de ligatures, d'autres n'en inclure que pour certaines variantes géographiques. Si une police venait à ne pas avoir de ligature correspondant à la suite de caractères liée par un liant sans chasse, la police devrait simplement afficher les deux consonnes de base, à savoir ⓄⓄ dans notre exemple ci-dessus.

Toutefois, aux yeux d'aucuns, il est apparu que le liant sans chasse qui n'indique que la formation facultative d'une biconsonne ne suffisait pas. Il fallait pouvoir préciser qu'une biconsonne devait impérativement être formée car la présence de cette ligature, dans une graphie non voyellée, indique l'absence d'une voyelle implicite entre les consonnes qui forment la ligature. C'est à cet effet qu'a été introduit dans Unicode 6.0 U+2D7F Ⓞ<sub>7F</sub> *liant de consonnes tiffinaghe*. Ce caractère de commande impose la formation de la ligature. À ce titre, il joue un rôle similaire au U+0652 Ⓞ *soukoûn*, le signe de quiescence arabe. Le tableau ci-dessous illustre l'utilité d'une telle convention, les exemples sont tirés du dictionnaire de Foucauld (Ch. de Foucauld, 1951).

Lettres de base	Graphie touarègue	Translittération	Glose en français
: Ⓞ	Ⓞ <sub>7F</sub> Ⓞ	ənkər	se lever (inhabituel)
	: Ⓞ	nāk:ər	se lever (habituel)
+ : ⓄⓄ	+ : ⊕	tɛɣert	marmite en terre
	+ : ⓄⓄ	tɛɣərit	cri strident, très perçant
ⓄⓄ :	ⓄⓄ <sub>7F</sub> : <sup>4</sup>	istəɣ	chasser, pousser devant soi
	ⓄⓄ :	ɔsatəɣ	chasse, poursuite
+ ⓃⓄⓄ	+ ⓃⓄⓄ	təfərt	mot, proposition, vers
	+ ⓃⓄⓄ	təfərit	petite aiguille rocheuse
∨ Ⓞ	∨Ⓞ	əndər	excéder les forces
	∨ Ⓞ	ənadar	fait d'être en chaleur, en rut

Dans les exemples ci-dessus, comme les biconsonnes sont considérées comme obligatoires, on n'utilisera pas de U+200D *liant sans chasse*, mais bien un

<sup>4</sup> On aurait aussi pu mettre ⊕ à la place de : .



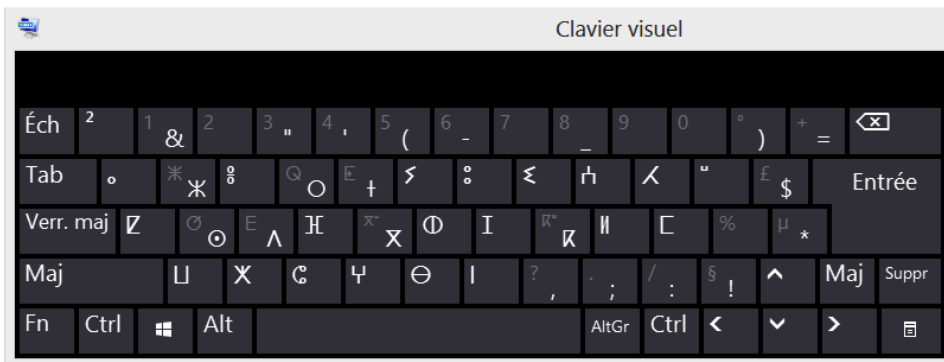
## Claviers

### *Clavier national normalisé*

Depuis la normalisation des tfinaghges dans l'ISO 10646 et Unicode, un clavier normalisé marocain pour la saisie de ces caractères a été homologué par le SNIMA<sup>6</sup>. Ce clavier respecte la norme internationale en la matière, l'ISO/CEI 9995. Il a été repris par plusieurs fabricants de Linux et inclus par Microsoft dans Windows 8. Normalisé il y a plusieurs années, ce clavier ne comprend pas les caractères tfinaghges introduits par Unicode 6.0 et 6.1.

### *Clavier tfinagh sur Windows 8*

Microsoft a adopté le clavier SNIMA dans Window 8 :



Pour le sélectionner, l'utilisateur doit ajouter un profil linguistique à partir du Panneau de configuration de Windows. Le profil linguistique regroupe une langue et une méthode d'entrée correspondante. La langue correspond à un indicatif ISO 639. Dans Windows 8, le seul indicatif « berbère » disponible est [tzm], à savoir le parler amazighe de l'Atlas central. À partir de Windows 8.1, une nouvelle langue et son indicatif sont disponibles : l'amazighe standard marocain [zgh].

---

<sup>6</sup> <http://hapax.qc.ca/pdf/NM%2017.6.000.pdf>



d'élément et d'attribut ainsi que dans certaines<sup>7</sup> valeurs d'attribut dans les documents XML 1.0. En effet, à l'époque, XML 1.0 ne permettait pour ces noms que des caractères appartenant au répertoire d'Unicode 2.0. Cela signifiait que ces noms ne pouvaient contenir des caractères tiffinaghes, ou d'autres provenant d'écritures comme l'éthiopien également introduits après Unicode 2.0, pas plus que des caractères importants récemment ajoutés à des écritures qui existaient déjà dans Unicode 2.0 comme le latin, l'arabe, le cyrillique ou le chinois.

Il n'était donc pas permis d'écrire en XHTML :

```
<a name="⊕" id="⊕">attribut touareg</a>
```

Pas plus que ceci n'était permis en XML 1.0 :

```
<⊕ : ⊕ >...</⊕ : ⊕ >
```

Au vu de cette incongruité, le W3C chargé de la normalisation de XML a décidé de publier à la fin 2008 (le 26 novembre très précisément) la cinquième édition de XML 1.0<sup>8</sup>. Outre la mise à jour de quelques références bibliographiques et la correction de quelques errata, le grand changement introduit par cette édition est de permettre la quasi-totalité des caractères Unicode et notamment les tiffinaghes dans les noms d'élément et d'attribut ainsi que les valeurs d'attribut.

En théorie, la pratique est identique à la théorie, mais en pratique cela peut bien sûr être différent. Pour qu'un nom d'élément en tiffinaghes soit accepté par un logiciel de traitement de documents XML, il faut tout de même que les analyseurs (parseurs) XML soient modifiés pour mettre en œuvre la nouvelle règle de formation des noms. Il faudra donc encore attendre quelques années avant que les analyseurs en place soient mis à jour pour que l'échange de documents XML/XHTML avec des tiffinaghes dans les noms d'éléments et d'attributs puisse se faire sans encombre. Par contre, si les documents XML ne sont utilisés qu'en interne où sont contrôlés les analyseurs XML, il se peut que, en mettant à jour ceux-ci pour choisir une version qui prend en charge la cinquième édition de XML 1.0, vous puissiez dès aujourd'hui utiliser des documents XML dont les noms d'élément, d'attribut et les valeurs d'attributs contiennent des tiffinaghes.

## Les polices

### *Ebrima sur Windows*

Depuis Windows 7, Microsoft inclut en série dans son système d'exploitation une police « panafricaine », la police Ebrima. Elle est fournie en deux graisses : normale et grasse. Ebrima a été conçue pour prendre en charge un grand nombre de langues africaines. Elle

---

<sup>7</sup> Les attributs CDATA peuvent contenir n'importe quoi, la restriction était sur les attributs déclarés ID, IDREF, NMTOKEN, etc.

<sup>8</sup> <<http://www.w3.org/TR/2008/REC-xml-20081126/>>

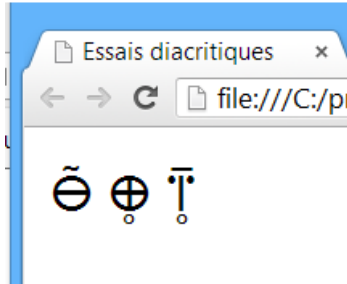


contient des glyphes pour le n'ko, les tfinaghes, le vaï et l'osmanya. Les glyphes latins de la police sont accompagnés des diacritiques et autres signes utilisés pour représenter les langues africaines.

Les œils tfinaghes d'Ebrima sont directement inspirés de ceux utilisés dans les tableaux Unicode (c'est-à-dire de Hapax Berbère).

La police Ebrima ne comprend pas les deux voyelles nigériennes (yé et yo) introduites par Unicode 6.1, elle comprend cependant depuis Windows 8 plusieurs innovations par rapport aux versions précédentes :

1. On peut ajouter des diacritiques à toute lettre tfinaghe<sup>9</sup> :



2. De nombreuses biconsonnes ont été ajoutées à la police, elles sont formées à l'aide du U+2D7F LIANT DE CONSONNES TIFINAGHE, notamment

| + <U+2D7F> + ∷ → ∷

3. Quand une biconsonne ne peut être formée parce qu'elle n'est pas prise en charge par la police, Ebrima affiche une série de points souscrits sous la paire de consonnes :

Λ∨

### ***Les polices incorporées dans les pages HTML***

Avec une écriture aussi récente – en termes de normalisation informatique – que le tfinaghe, il n'est pas rare qu'un utilisateur qui désire afficher une page Internet qui contient des caractères tfinaghes n'ait pas de police tfinaghe à sa disposition.

Malgré la diffusion récente d'Ebrima, le problème demeure entier aujourd'hui, car il existe encore de nombreux utilisateurs qui n'utilisent pas Windows et *a fortiori* Windows 7.

En outre, que faire quand on veut être sûr qu'une page s'affiche dans un style

---

<sup>9</sup> Ceci fonctionne bien avec les trois grands navigateurs (Chrome, Firefox et Internet Explorer) et dans Windows 8.1 avec MS Word 2013.

tifinaghe particulier ? On risque donc dans ces cas-là d'être confronté à des pages remplies de petits rectangles blancs comme dans l'illustration ci-dessous, chaque rectangle y représente un caractère (tifinaghe ici) qui ne peut être représenté par manque de police adéquate.

Il existe plusieurs remèdes possibles à ce désagrément. Une solution consiste à prévenir les lecteurs des pages en tifinaghe qu'ils doivent installer une ou plusieurs polices en fournissant un lien permettant de télécharger ces polices. Il existe cependant une autre solution : les polices dynamiquement téléchargeables. Cette solution consiste à envoyer les glyphes nécessaires à l'affichage d'une page HTML avec la page en question. On parle alors de polices embarquées ou incorporées.

---

**Extrait du dictionnaire touareg-français de Charles de Foucauld (page 339)**

□□ éferi □□□ sm. φ (pl. iferân □□□), dar āferi (ēferi), dar ferân || aiguille rocheuse (quelconque) || diffère d'āḍaouḍa « aiguille rocheuse très mince (ressemblant à un doigt) ». Tout āḍaouḍa est un éferi, mais non inversement.

— téferit □□□□ sf. φ (pl. tiferâtîn □□□□□), dar tāferit (tēferit), dar tferâtîn || dim. du pr.

□□ tāfirt □□□ sf. φ (pl. tifir □□□), dar tfir || mot (une syllable ou plusieurs syllabes réunies exprimant une idée) || p. ext. « quelques paroles (paroles en petit nombre) » || p. ext. « vers (assemblage de mots rythmés d'après des règles déterminées en poésie) » Le sing. tāfirt signifie un vers, le pl. tifir signifie 2 ou plusieurs vers.

□□ aferra || v. □□□ afri .

*Figure 3 : Police manquante dans une page HTML*

Il y a près de deux ans, en 2008, nous avons déjà abordé les techniques alors disponibles (Andries, *ibid.*). Rappelons que les polices incorporables dans les pages HTML ne sont pas une idée neuve. Dès 1998, CSS 2 permettait de préciser un lien vers une police que les fureteurs internet pouvaient télécharger. Microsoft et Netscape prirent en charge cette syntaxe et permettaient de télécharger des polices. Malheureusement, aucun des deux fabricants ne prenait en charge directement le format de police le plus populaire : TrueType. Microsoft choisit le format de police EOT (un format propriétaire) dans Internet Explorer 4.0 alors que Netscape 4.0 jeta son dévolu sur un format rival TrueDoc qu'il abandonna deux versions plus tard, car Mozilla ne pouvait rendre public le code source de TrueDoc, propriété de Bitstream.

Depuis 2008, cette question a connu un vif regain d'intérêt. À partir de 2010, on a assisté à un véritable engouement pour les polices internet. Subitement, coup sur coup, on a assisté à l'élaboration d'un format de police conçue pour le téléchargement sur Internet (WOFF), la prise en charge de ce standard naissant par

les quatre grands concepteurs de moteurs de composition HTML<sup>10</sup> et enfin la mise en place de nombreux sites de partage ou de vente de polices incorporables dans des pages HTML<sup>11</sup>.

- **WOFF**

WOFF est un format de police embarquée qui a été proposé au W3C par Microsoft, Mozilla et Opera. Il a été conçu pendant l'année 2009. Le 27 juillet 2010, le W3C publiait un « projet de travail ». Le 13 décembre 2012, WOFF devenait une « recommandation » W3C, un standard Internet.

Fondamentalement, WOFF est une enveloppe qui recouvre les polices de type *sfnt* (à savoir TrueType, OpenType ou Open Font Format) qui ont été comprimées à l'aide d'un outil qui leur permet d'être incorporées dans des pages HTML. Ce format utilise une compression *zlib* qui assure habituellement une réduction de plus de 40 % par rapport à la police TTF équivalente.

WOFF est pris en charge par Firefox depuis sa version 3.6 et par Chrome de Google depuis la version 6.0. Microsoft, pour sa part, la prend en charge depuis la version 9 de son navigateur. Enfin Opera, le permet depuis sa version 11.10.

WOFF en tant que tel ne contient aucun mécanisme de sécurité qui empêcherait la copie de la police. Certains navigateurs, toutefois, ne téléchargent que les polices qui sont hébergées dans le même domaine que la page qui y fait référence. De même, l'optimisation qui consiste à ne sélectionner et envoyer sur le réseau que le sous-ensemble de glyphes correspondant aux caractères utilisés dans la page HTML demandée (ou un ensemble de pages) est laissée à des outils tiers qui ne font pas partie de la norme.

- **Autres formats**

Malgré le fait que les nouvelles versions des grands navigateurs prendront toutes en charge le format WOFF, il est important de considérer le parc actuel des navigateurs internet, trois autres types de polices téléchargeables y sont utilisés :

TTF	Fonctionne bien avec la majorité des fureteurs (mais pas IE, ni l'iPhone), volumineux car non comprimé.
EOT	Uniquement sur IE, nécessaire pour IE 5 à 8, comprimé, permet de ne sélectionner qu'un sous-ensemble de glyphes qu'on retrouve dans une

---

<sup>10</sup> Les quatre grands moteurs de composition sont *Trident* de Microsoft, *Gecko* de Mozilla, *WebKit* utilisé notamment par Google et Safari et *Presto* d'Opera. Chacun de ces moteurs de rendu est utilisé dans plusieurs applications : Presto se retrouve ainsi, non seulement dans le fureteur Opera, mais aussi dans des produits Nintendo, Nokia, Sony, Adobe et Macromedia.

<sup>11</sup> Parmi ces sites, on peut citer : <typekit.com>, <fontsquirl.com>, <typotheque.com>, <openfontlibrary.org>, <fontshop.com> et <fontfont.com>.

page HTML donnée.

SVG

Format XML nécessaire pour l'iPhone et l'iPad avant la version 4.2, volumineux.

- ***Hapax Berbère et Hapax Tifinar Carrée***

Les polices Hapax Berbère allégée et Hapax Tifinar Carrée sont disponibles sous quatre formats (TTF, EOT, SVG et WOFF) respectivement aux adresses suivantes <hapax.qc.ca/essai-polices-incor/hapaxber-sousensemble-webfont.zip> et <hapax.qc.ca/extrait-foucault/hapaxtifinarcarrée.zip>.

La police Hapax Berbère allégée ne comprend qu'un sous-ensemble restreint de la police Hapax Berbère alors que la police Hapax Tifinar Carrée est complète (et donc assez volumineuse). On peut les copier et les utiliser à loisir. Deux pages de démonstration permettent de voir ces polices Web en action : <hapax.qc.ca/essai-polices-incor/HapaxBerbereRegular-demo-hex.html> et <hapax.qc.ca/extrait-foucault/foucault-p339.html>.

- ***@font-face et CSS***

Revenons à notre extrait du dictionnaire touareg de Charles de Foucauld. Comment s'assurer que la page s'affiche correctement tant sur les anciennes versions de MSIE, les iPhone, les iPad que les nouvelles versions de Firefox ?

La première étape est de créer un ensemble de polices TTF, SVG, EOT et WOFF. Un site comme Font Squirrel<sup>12</sup> permet de fournir sa police en entrée et de créer des versions SVG, EOT, TTF et WOFF de celle-ci ainsi que de sélectionner qu'un sous-ensemble des glyphes de la police d'origine afin de réduire encore la taille des fichiers produits.

Dans notre cas, la police Hapax Tifinar Carrée utilisée dans la page du dictionnaire Charles de de Foucauld a déjà été transformée en ces quatre formats. Il suffit de télécharger l'ensemble<sup>13</sup>, de les dézipper et de les placer dans un répertoire du serveur. Le code fourni ci-dessous assume que les polices sont présentes dans le même répertoire que les feuilles de style CSS.

Il faut ensuite définir la police Hapax Tifinar Carrée dans CSS à l'aide d'une déclaration @font-face. C'est ce que fait le code ci-dessous. Il définit cette même police dans deux déclarations que les principaux navigateurs comprendront. Si les polices ne sont pas stockées dans le même répertoire que le code CSS, on n'oubliera pas de changer les url() pour y indiquer l'URL qui correspond à leur position.

---

<sup>12</sup> Voir <fontquirrel.com/fontface/generator>. Font Squirrel fournit également les @font-face à inclure dans vos feuilles de style.

<sup>13</sup> Disponible ici : <hapax.qc.ca/extrait-foucault/hapaxtifinarcarrée.zip>.

```
@font-face {
    /* Déclaration destinée à Internet Explorer
*/

    font-family: 'HapaxTifinarCarree';
    src: url('hapaxtifinarcree.eot') ;
}

@font-face {
    /* Déclaration destinée à tous les autres */

    font-family: 'HapaxTifinarCarree';
    src: url(//:) format('pasde404'),
        url('hapaxtifinarcree.woff') format ('woff'),
        url('hapaxtifinarcree.ttf')
format('truetype'),
        url('hapaxtifinarcree.svg#webfontYQ4E2jU')
        format('svg') ;

    font-weight: normal ;
    font-style: normal ;
}
}
```

---

**Extrait du dictionnaire touareg-français de Charles de Foucauld (page 339)**

- **éferi** □□□ sm. φ (pl. **iferân** □□□), **dar āferi** (ēferi), **dar ferân** || aiguille rocheuse (quelconque) || diffère d'**āḍaouḍa** « aiguille rocheuse très mince (ressemblant à un doigt) ». Tout **āḍaouḍa** est un **éferi**, mais non inversement.
- **tēferit** □□□□ sf. φ (pl. **tiferâtîn** □□□□), **dar tāferit** (tēferit), **dar tferâtîn** || dim. du pr.
- **tāfirt** □□□ sf. φ (pl. **tifir** □□□), **dar tfir** || mot (une syllable ou plusieurs syllabes réunies exprimant une idée) || p. ext. « quelques paroles (paroles en petit nombre) » || p. ext. « vers (assemblage de mots rythmés d'après des règles déterminées en poésie) » Le sing. **tāfirt** signifie un vers, le pl. **tifir** signifie 2 ou plusieurs vers.
- **aferra** || v. □□□ **afri** .

Nous reviendrons sur ces déclarations par la suite. Pour l'instant, il suffit de savoir que ces déclarations permettent aux navigateurs de savoir où trouver la définition d'une police nommée HapaxTifinarCarree.

Pour employer la police dans des documents HTML, il nous reste à préciser en CSS quand l'utiliser. Le code ci-dessous précise, par exemple, que les paragraphes

(<p>) devront d'abord utiliser la police HapaxTifinarCarree, puis si des glyphes manquent d'employer la police Arial et enfin, par défaut, n'importe quelle police sans empattements.

```
p {font-family: 'HapaxTifinarCarree', Arial, sans-serif;}
```

Dans le cas de l'extrait du dictionnaire touareg, nous désirons d'abord utiliser des polices à empattements hautement optimisées comme Times New Roman pour le texte latin avant d'utiliser les glyphes de la police Hapax Tifinar Carree dont les glyphes latins sont moins esthétiques.

```
body {  
    font-family : Times New Roman, HapaxTifinarCarree,  
    serif;  
    width: 650px ;  
    text-align: justify;  
    margin-left: auto ; margin-right: auto ;  
}
```

Ensuite dans le corps de la page HTML, il suffit d'écrire par exemple

```
<p>- <u>téferit</u>+ ⵏⵔ+ sf. φ (pl.  
<u>tiferâtîn</u> + ⵏⵔ+ | ), <u>daṛ tăferit</u>  
(<u>tēferit</u>), <u>daṛ tferâtîn</u> || dim. du  
pr.</p>
```

pour que les caractères latins s'écrivent à l'aide de la police Times New Roman et les caractères tfinaghes avec la police Hapax Tifinar Carrée comme dans la figure ci-après.

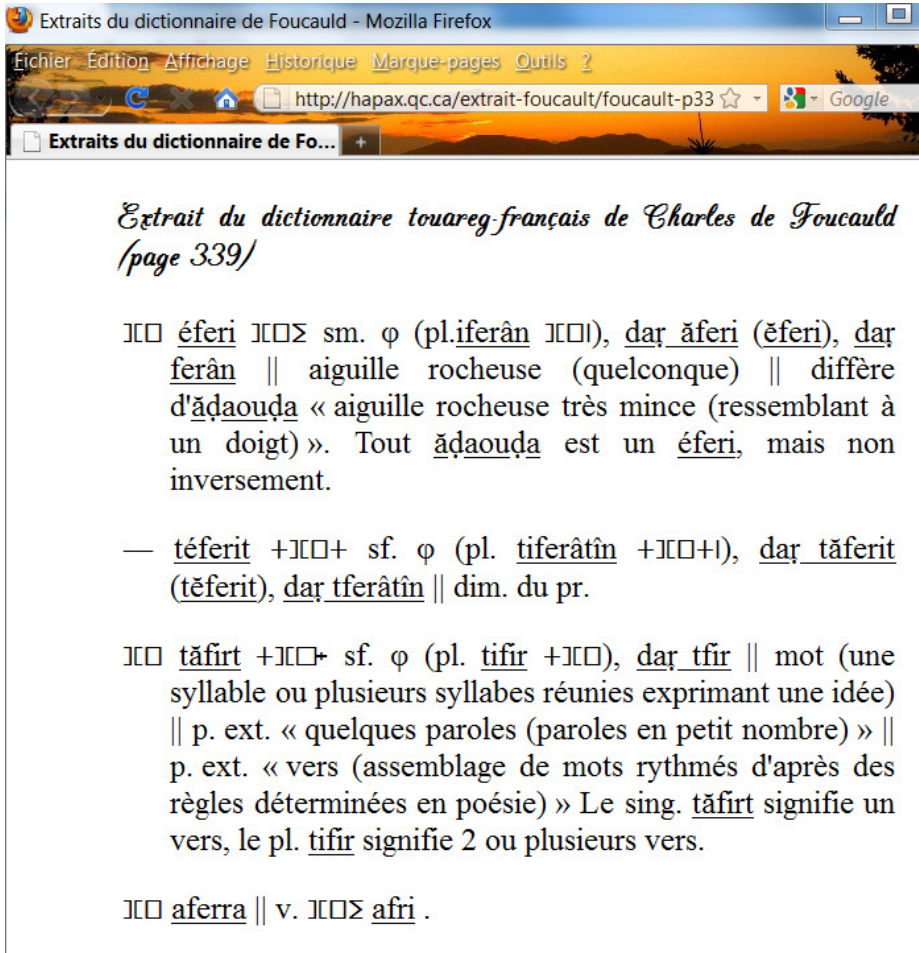


Figure 4 : Page avec des polices Internet incorporées

- **Retour sur les différents @font-face**

Chaque @font-face permet de déclarer plusieurs fichiers alternatifs pour un même nom de police, en séparant chacun des emplacements, les `url()`, par une virgule :

```
@font-face{
  font-family : 'maPolice';
  src : url('maPolice.woff') format('woff'),
        url('maPolice.svg#abcd') format('svg'),
        url('maPolice.ttf') format('truetype') ;
}
```

La propriété `src` est suivie de l'URL d'un fichier de police puis d'un format optionnel. Déclarer le format permet de préciser explicitement aux navigateurs le type de police associé à l'URL qui précède et d'éviter ainsi au fureteur d'avoir à télécharger le fichier en question pour en vérifier le format.

L'URL de la police SVG se termine par un nom de fragment après le croisillon (« # »). Ce nom précise où, dans le fichier SVG, se trouve la déclaration de police. À première vue, cela peut paraître superflu, il faut toutefois se rappeler qu'un fichier SVG peut contenir bien d'autres choses qu'une police de caractères ou même en contenir plusieurs.

L'ordre de déclaration des formats est également important : d'abord WOFF, puis selon la taille des fichiers, la version SVG et enfin TrueType. Il est important de considérer l'ordre de déclaration, car certains navigateurs prennent en charge plusieurs de ces formats et utiliseront le premier qu'ils comprennent dans la liste fournie. Or, dans cette liste, WOFF est – comme nous l'avons vu – nettement plus léger que TTF. Le SVG étant un format XML, les fichiers peuvent vite devenir volumineux. Pour palier cet inconvénient, il en existe une version comprimée qui porte le suffixe `.svgz`. Elle est toutefois peu prise en charge par les fureteurs. En outre, si votre serveur gère la compression des requêtes HTTP, cet artifice n'est guère utile. La version compressée d'une police SVG est, en règle générale, plus légère que sa version TrueType.

Pourquoi, dans notre code ne trouve-t-on pas de déclaration comme ci-dessous ?

```
src : url('maPolice.eot') format('eot') ;
```

La raison en est simple : Internet Explorer ignore la propriété `format`<sup>14</sup> et ne prend en charge qu'une valeur d'URL(). En outre, confronté à la ligne

```
src : url('maPolice.woff') format('woff') ;
```

Internet Explorer cherche à télécharger une police du nom suivant : « `maPolice.woff` » `format('woff')`. MSIE envoie une requête dans ce sens au serveur et attend que le serveur lui réponde. Il le fera à l'aide d'un 404 (pas trouvé), car bien évidemment il n'existe pas de telle police sur le serveur. Cet appel au serveur est inutile et inefficace.

C'est pourquoi<sup>15</sup>, pour les versions 4 à 8 de MSIE, on ajoute une déclaration supplémentaire :

---

<sup>14</sup> Ceci est corrigé à partir de la future version 9 d'Internet Explorer.

<sup>15</sup> Pour plus de détails sur les bogues d'IE en la matière, voir <<http://covertprestige.info/css/font-face/>>.



```
@font-face {
    /* Déclaration destinée à Internet Explorer
*/

    font-family: 'HapaxTifinarCarree';
    src: url('hapaxtifinarcree.eot');
}
```

et l'on ajoute cette première ligne dans la deuxième déclaration<sup>16</sup> :

```
src: url(//:) format('pasde404'),
```

afin que la requête inutile d'IE échoue avant l'envoi de celle-ci, car aucun protocole ni aucun serveur ne correspond à l'url fourni.

## Les bibliothèques logicielles

Grâce à la normalisation des tifinaghes il y a 8 ans et à leur inclusion dans Unicode, aujourd'hui les bibliothèques logicielles permettent de traiter les textes écrits en tifinaghes. Si Java 6 ne connaît pas les propriétés des caractères tifinaghes, ce n'est pas le cas d'ICU<sup>17</sup>, une bibliothèque Unicode<sup>18</sup> qui sert de terrain d'essai à Java et dont les fonctionnalités sont habituellement intégrées avec un certain retard dans Java.

On trouvera ci-dessous un exemple de programme Java qui permet de découper un texte Unicode en « mots ». On remarquera que la syntaxe du BreakIterator ICU utilisée ici est identique à celle de Java. Il suffit souvent pour accéder aux fonctionnalités d'ICU de remplacer l'import de Java (ici `java.text.BreakIterator`) par son équivalent ICU (`com.ibm.icu.text.BreakIterator`) :

```
import com.ibm.icu.text.BreakIterator;

// import java.text.BreakIterator;

public class AnalyseMotsTifinaghes
{
    public static void main(String[] args)
    {
```

---

<sup>16</sup> On préférera cette astuce à celle qui consiste à ajouter une propriété `local()` qui plante le navigateur du système d'exploitation Android utilisé sur des téléphones intelligents : [readableweb.com/best-practice-for-font-face-css-takes-a-turn](http://readableweb.com/best-practice-for-font-face-css-takes-a-turn/).

<sup>17</sup> On peut télécharger ICU pour Java ici : [ici](http://download.icu-project.org/files/icu4j/) <download.icu-project.org/files/icu4j/>

<sup>18</sup> Pour une introduction à ICU, voir le chapitre 12 d'*Unicode 5.0 en pratique*, Dunod (2008), Paris.

```

String texteÀExaminer = "。⊙H: XH: | : ⊙XX" 。 ⊙
。 C. *ξ Ψ ⊙ H C | Λ | : ⊙XX" 。 ⊙。 C. *ξ Ψ 。 C. ρ | : ." ;
    int début=0;

    BreakIterator frontière =
BreakIterator.getWordInstance();

    frontière.setText(texteÀExaminer);

    début = frontière.first();

    for ( int fin = frontière.next();
        fin != BreakIterator.DONE;
        début = fin, fin = frontière.next())
    {

        System.out.println(texteÀExaminer.substring(début, fin)
);
    }
}
}
}

```

Ce programme divise le texte en mots en se basant sur les propriétés<sup>19</sup> Unicode des caractères (est-ce une lettre, un signe de ponctuation, un chiffre, un espace ?) et produit le résultat suivant :

```

。 ⊙H: XH:
|
: ⊙XX" 。 ⊙
。 C. *ξ Ψ
⊙
H C | Λ
|
: ⊙XX" 。 ⊙
。 C. *ξ Ψ
。 C. ρ | :
.

```

ICU comprend bien plus que cette fonctionnalité, il permet notamment de trier des textes tfinaghes et définit des données de localisation (nom de mois, format des dates, etc.) en tfinaghes. .NET de Microsoft offre des fonctionnalités similaires.

<sup>19</sup> Voir le chapitre 4 d'*Unicode 5.0 en pratique*, Dunod (2008), Paris.

## **Le répertoire de données de localisation CLDR**

Depuis l'homologation de l'indicatif [zgh] par l'ISO 639-2, il est désormais possible de l'utiliser dans des dépôts de données numériques ou un « Répertoire de données de paramètres régionaux classiques » comme CLDR. Le CLDR contient des informations spécifiques aux paramètres régionaux (les « locales » en jargon informatique) fournies pour faciliter l'adaptation des applications informatiques à des langues ou des cultures différentes. Le CLDR comprend donc :

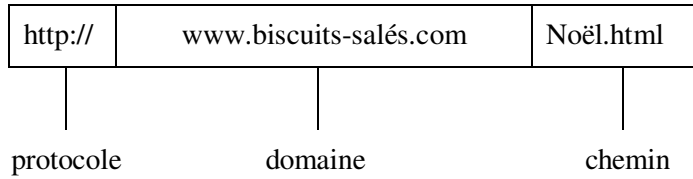
- des traductions pour les noms de langues ;
- des traductions pour les noms de territoires et de pays ;
- des traductions pour les noms de monnaies, y compris les modifications singulier/pluriel ;
- des traductions pour les jours de la semaine, les mois, dans des formes complètes et abrégées ;
- des traductions pour les fuseaux horaires et des exemples de villes (ou similaire) pour les fuseaux horaires ;
- des traductions pour les périodes du calendrier ;
- des motifs pour formater/reconnaître des dates ou des heures, etc.

Grâce à CLDR (dont les données sont notamment intégrées dans ICU) il devient facile pour un programmeur (même s'il ne connaît pas l'amazighe) de concevoir des programmes qui pourront afficher, par exemple, des dates en français et en amazighe standard du Maroc. Pour afficher la date en français, il précisera simplement la valeur ISO 639 « fr » comme « locale », pour l'afficher en amazighe standard il utilisera « zgh ».

Des données initiales pour [zgh] ont été versées dans le répertoire CLDR, elles devraient, toutefois, être complétées et révisées dans les mois à venir.

## **Adresses internet, courriel et noms de domaine internationalisés (NDI)**

Lors du colloque de l'IRCAM en 2008 (Andries, *op.cit.*), nous avons déjà abordé le sujet des adresses internet internationalisées du type <http://www.biscuits-salés.com/Noël.html>. Techniquement on nomme ce type d'adresse des identificateurs de ressource internationalisée (IRI). Ces adresses se divisent en trois parties principales, la première avant le « :// » indique le protocole à utiliser (ici http), ensuite vient le nom de domaine proprement dit puis, après un « / », le chemin de la ressource sur le serveur identifié par le nom de domaine.



Tous les domaines de tête (le *.ma*, le *.fr* ou *.com* final du nom de domaine) n'acceptent pas ces adresses internationalisées. Certains domaines de tête comme la Suisse (*.ch*) acceptent des accents français (et plus généralement les lettres accentuées de ses langues nationales) alors que la France n'accepte pas de tels accents dans les domaines qui se terminent par *.fr*. La politique d'attribution de ces noms est décidée par l'autorité d'enregistrement responsable d'un domaine de tête particulier. Au Canada, il s'agit d'une agence sans but lucratif (l'ACEI) régie par les lois canadiennes. Les domaines de tête génériques (*.com*, *.org*, *.biz*) sont administrés aux États-Unis en vertu des lois américaines.

Comme nous le mentionnions déjà en 2008, les navigateurs modernes transformeront de manière transparente les noms de domaine comme *biscuits-salés* en des noms de domaine compatibles avec l'infrastructure actuelle prévue pour des adresses ASCII.

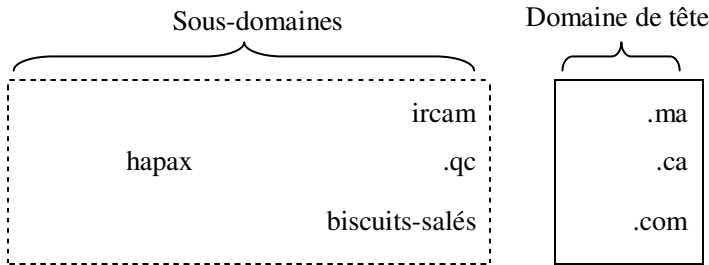


Figure 5 – Parties d'un nom de domaine

Ce qui a changé depuis ce colloque, c'est l'approbation d'IDNA 2008 qui a mis à jour le répertoire des caractères permis, passant d'Unicode 3.2 à Unicode 5.2. Par ce fait même, tous les caractères tfinaghés marocains peuvent, en théorie, être utilisés dans les adresses Internet. Il est important d'insister sur la dimension théorique de cette avancée, car l'acceptation de ces caractères dans les noms de domaine dépend des autorités d'enregistrement de noms de domaine. Pour que des sous-domaines en tfinaghés soient acceptés dans des adresses du domaine de tête *.ma*, il faudra donc que l'organisme responsable de l'attribution des noms de domaine au Maroc le permette.

## Conclusion

Depuis 2005 et la normalisation du tfinagh dans Unicode et l'ISO/CEI 10646, beaucoup de choses ont changé. Il est de plus en plus facile d'utiliser des tfinagh sur internet et dans les applications informatiques.

Il reste encore, bien sûr, dans ce domaine quelques améliorations possibles. On peut citer des pistes comme le fait de s'assurer que des claviers et des polices tfinagh soient disponibles sur les téléphones intelligents et les tablettes informatiques, la diffusion et l'emploi de CSS 3 qui permettra une typographie fine des polices, embarquées ou non, la mise en place concrète d'IDNA 2008, la création et la diffusion de ressources de localisation en tfinagh et amazighe, la mise à jour du clavier tfinagh pour inclure les nouveaux caractères tfinagh Unicode, etc. Dans ces domaines, les choses avancent et des solutions techniques existent déjà ou se dessinent pour que l'utilisation des tfinagh et de l'amazighe dans tous ces domaines se fasse désormais sans entrave. En huit ans, que de chemin parcouru !

## Références bibliographiques

Foucauld Ch.-de (1951), *Dictionnaire touareg-français, dialecte de l'Ahaggar*. 4 volumes, Imprimerie nationale, Paris.

Karl-G. Prasse (1972), *Manuel de Grammaire touarègue*, Éditions de l'Université de Copenhague.

Zenkouar L. (2004), « L'Écriture amazighe tfinagh et Unicode », *Études et Documents Berbères*, vol. 22, p. 173-175.

Andries P. (2008), *Demain encore plus de tfinagh sur Internet*, Actes du colloque du CEISIC (NTIC), Rabat.