

Les Médecins Maîtres-Toile

[Accueil](#) ► [Boite à Outils](#) ► [Syndication de contenu](#) ► [Lecture des caractères spéciaux et d'éléments d'un fil RSS/Atom](#)

Publié le : 25 avril 2007

**Publication antérieure :
8 octobre 2005**

 [Imprimer cet article](#)

Auteur :
Alain Sichel †



Lecture des caractères spéciaux et d'éléments d'un fil RSS/Atom

J'ai décrit dans différents articles [1] comment utiliser la classe `fil_LE` pour afficher des fichiers de syndication RSS ou Atom. Cela fonctionne parfaitement... sauf si le fil contient des caractères inhabituels (langue étrangère par exemple). En modifiant le script avec l'astuce décrite dans cette page, ces caractères particuliers seront affichés. Cette technique peut aussi permettre de décoder certains éléments que la classe `rss_read` n'arrive pas à lire [2].

Sommaire

- [Présentation du problème](#)
- [L'astuce](#)
- [Quelques exemples](#)
- [Application au décodage de certains éléments](#)

Présentation du problème

C'est grâce à [Bertrand Piret](#) que j'ai cherché une solution à ce problème. En effet, en voulant afficher le fichier de syndication de son site [Parole sans frontière](#) sur le [site du CODES 78](#), j'ai constaté que des phrases en arabe d'un des articles étaient remplacées par des " ??? ???".

Dans le fil RSS ces caractères sont codés sous la forme `&#xxxx;` (xxxx étant un nombre de 4 ou 5 chiffres). Si ces caractères sont intégrés dans du code HTML, ils seront normalement affichés par le navigateur [3]. Par contre lorsqu'on utilise une fonction PHP pour récupérer les informations d'un fil RSS ou Atom, ces caractères ne sont pas reconnus et sont remplacés par des "?".

L'astuce

J'ai donc cherché une solution pour que le code `&#xxxx;` soit modifié avant l'action de la fonction PHP pour parser les données, puis pour reconstituer ce code pour l'affichage dans la page. Voilà ce que je propose :

```
$rss -> to_replace_with("&#", "@@");
```

ce remplacement transforme `&#xxxx;` en `@@xxxx;` qui est considéré comme une chaîne de caractères banale et n'est pas remplacé par " ?" [4]. Attention, pour que cela fonctionne, il faut utiliser cette fonction avant que le fil soit envoyé à la fonction de récupération des données.

Pour reconstituer les caractères, on utilise la fonction :

```
function clean_text($text, $encodage = '') {
    $text = str_replace('&lt;', '<', $text);
    $text = str_replace('&gt;', '>', $text);
    $text = str_replace('&quot;', '"', $text);
    $text = str_replace('&', '&', $text);
    $text = str_replace("@@", "&#", $text);
    // pour le traitement suivant encodage du fil
    if ($encodage == 'utf-8') {
        $text = utf8_decode($text);
    }
    return $text;
}
```

qui reconstitue la forme `&#xxxx;` de ces caractères spéciaux. Cette fonction est déjà présente dans les scripts décrits, c'est simplement la ligne `$text = str_replace("@@", "&#", $text);` qui est ajoutée.

Quelques exemples

Après le principe, voici quelques exemples de son application. J'ai repris et modifié les fichiers à télécharger accompagnant les articles sur la classe `rss_read` :

Afficher un fil RSS/Atom sur votre site 1/3 :



exemples_modifie.zip (Ce fichier contient `exemple_a_modifie.php` et `exemple_b_modifie.php`).

Afficher un fil RSS/Atom sur votre site 2/3 :



Exemples_2_modifie.zip (Ce fichier contient exemple_c_modifie.php et exemple_d_modifie.php).

Afficher un fil RSS/Atom sur votre site 3/3 :



exemples_e_modifie.zip (Ce fichier contient exemple_e_modifie.php).

Lecteur RSS/Atom pour votre site 1/2 :



lecteur_modifie.zip (Ce fichier contient note.txt et script_modifie.txt).

Lecteur RSS/Atom pour votre site 2/2 :



Lecteur_plus_modifie.zip (Ce fichier contient note.txt et script2_modifie.txt).

Application au décodage de certains éléments

La classe `rss_read` a beau être performante, elle ne sait pas tout lire, en particuliers certains éléments dont la mise en forme est imposée pour avoir des fichiers de syndication valides.

Par exemple,

```
<author><name>Karim Lounici</name><uri>http://www.mmt-fr.org/article269.html</uri></author>
```

est une écriture valide pour le format Atom 1.0 pour indiquer un auteur, mais en raison des tags intriqués, `rss_read` ne sait pas les décoder.

La solution que je propose consiste à appliquer l'astuce décrite ci-dessus. On va la transformer en :

```
$avant =  
array('&#', '&bull;', '<author><name>', '</name><uri>', '</uri></author>');  
$apres = array("@@", '- ', '<author>', ' - ', '</author>');  
$rss -> to_replace_with($avant, $apres);
```

ceci aura pour conséquence de transformer l'élément auteur en :

```
<author>Karim Lounici - http://www.mmt-fr.org/article269.html</author>
```

et ça, rss_read peut le lire :

```
$items[$i]['author'] aura alors pour valeur Karim Lounici - http://www.mmt-fr.org/article269.html
```

pour le mettre sous une forme plus élégante, on écrit :

```
if ($items[$i]['author']) {  
    $tab = explode(" - ", clean_text(trim($items[$i]['author']), $encode));  
    $data .= '<b><a href="'. $tab[1].'" title="Pour contacter  
1\'auteur">'. $tab[0]. '</a>&nbsp;';</b> '  
}
```

ce qui donnera dans la page : [Karim Lounici](#)

Cette astuce peut largement être adaptée pour tous les éléments non lus par rss_read.

[1] Voir [Afficher un fil RSS/Atom sur votre site 1/3, 2/3, 3/3](#), [Lecteur RSS/Atom pour votre site 1/2 et 2/2](#).

[2] Cet article a été modifié le 25/04/2007 pour adapter cette astuce au décodage de certains éléments.

[3] Oui... si c'est un navigateur moderne (Firefox, Mozilla, Safari...), parce qu'avec Microsoft Internet Explorer pour Mac, c'est loin d'être évident !

[4] J'ai déjà proposé dans d'autres articles cette fonction pour certains caractères posant problème (voir [Lecteur RSS/Atom pour votre site 1/2 et 2/2](#)). S'il y a plusieurs types de caractères qu'il faut transformer, on peut utiliser la forme :

```
$avant = array('&#', '&bull;');  
$apres = array('@@', '- ');  
$rss -> to_replace_with($avant, $apres);
```



[Imprimer cet article](#)

Copyright Médecins Maîtres-Toile francophones
Espace membres - Administration - Crédits
