

# Les Médecins Maîtres-Toile

[Accueil](#) ► [Boîte à Outils](#) ► [Syndication de contenu](#) ► [Afficher plusieurs flux RSS/Atom sur une page](#)

**Publié le : 29 avril 2007**

 **Imprimer cet article**

**Auteur :**  
**Alain Sichel †**



## Afficher plusieurs flux RSS/Atom sur une page

Cet article propose d'utiliser `Fil_LE` pour recueillir les items en provenance de plusieurs fichiers de syndication et de les trier par date pour les présenter du plus récent au plus ancien.

### Sommaire

- Principe
- Encodage des caractères
- Nettoyer les fils
- Associer les fils
- Tri par date
- Suite du script

### Principe

Cet article suppose que vous avez pris connaissance des articles précédents [Afficher un fil RSS/Atom sur votre site 1/3](#), [2/3](#), [3/3](#) et [Lecture des caractères spéciaux et d'éléments d'un fil RSS/Atom](#) ; je suppose que vous utilisez `Fil_LE` avec le système de cache décrit dans l'article [Afficher un fil RSS/Atom sur votre site 3/3](#) avec en plus l'astuce décrite dans l'article [Lecture des caractères spéciaux et d'éléments d'un fil RSS/Atom](#), je reparts donc de ce script en l'adaptant.

Je prends pour exemple les fichiers de syndication de Femiweb. Il y a en a un pour les articles et un autre pour les brèves, on va les lire tous les deux :

```
// chemin relatif vers le fichier cache (le repertoire "cache" doit etre
autorise en ecriture)
$file_cache = 'cache/femiweb.html';
// le delai entre deux rafraichissement en secondes
$delai = 12*3600;
```

```

// le fichier est-il en cache et suffisamment jeune
$en_cache = file_exists($file_cache);
if ($en_cache) {
    $en_cache = (time() - filemtime($file_cache) < $delai);
}

if (!$en_cache) { // il est considere comme n'etant pas en cache on le
genere
    include 'rss_read.inc.php';
    // le fichier est-il en cache et suffisamment jeune
$en_cache = file_exists($file_cache);
if ($en_cache) {
    $en_cache = (time() - filemtime($file_cache) < $delai);
}

if (!$en_cache) {
    // il est considere comme n'etant pas en cache on le genere
    include 'rss_read.inc.php';
    $flux1 = "http://www.femiweb.com/backend.php3";
    $flux2 = "http://www.femiweb.com/backend-breves.php3";
}
}

```

## Encodage des caractères

Ensuite, j'ai supposé que nous pouvions avoir deux flux dont l'encodage des caractères est différent [1]. On va donc d'abord regarder quel est l'encodage de chaque flux :

```

function encodage($flux) { // détermination de l'encodage et ecriture de
celui-ci dans chaque item
    $fil = file_get_contents($flux); // articles
    if (preg_match('/encoding="([^\"]*)"/i', $fil , $temp)) {
        $encode = strtolower($temp[1]);
    } elseif (preg_match("/encoding='([^\']*)' /i", $fil , $temp)) {
        $encode = strtolower($temp[1]);
    }
    $apres = array("@@", '-
', "<encode>$encode</encode></entry>", "<encode>$encode</encode></item>");
    return $apres;
}

```

et noter celui-ci dans chaque item avant que la fonction de récupération des données intervienne, en utilisant l'astuce décrite dans [Lecture des caractères spéciaux et d'éléments d'un fil RSS/Atom](#) :

```

$apres = encodage($flux1); // articles
$fill = new rss_read();
$savant = array('&#', '&bull;', '</entry>', '</item>');
$fill -> to_replace_with($savant, $apres);
$fill -> parsefile($flux1);
$data = '';
$items1 = $fill -> get_items();

```

```

$apres = encodage($flux2); // breves
$fil2 = new rss_read();
$fil2 -> to_replace_with($avant, $apres);
$fil2 -> parsefile($flux2);
$items2 = $fil2 -> get_items();

```

## Nettoyer les fils

Comme je l'ai indiqué dans [Afficher un fil RSS/Atom sur votre site 2/3](#), la mise en forme de l'élément *description* nécessite parfois une mise en forme pour améliorer la présentation, voire une véritable correction pour les fils qui n'indiquent pas les adresses absolues des images ou qui utilisent des tags HTML de mise en forme du texte non suivis de tags de fermeture [2]. Ce nettoyage peut être différent pour chaque fil, on va donc préparer des fonctions de nettoyage :

```

function nettoyage($description, $options = '') {
    if (ereg("p", $options)) {
        $avant = array ("<p.*?>'si","</p>'si");
        $apres = array (" ", "<br />");
        $description = preg_replace($avant,$apres,$description);
    }
    if (ereg("h", $options)) { $description = preg_replace("<hr.*?>'si","<br />",$description); }
    if (ereg("r", $options)) { $description =
preg_replace("'align=\\"left\\"'si,"align=\\"right\\"",$description); }
    if (ereg("d", $options)) { $description =
str_replace('rel="directory"', 'title="Rubrique"', $description); }
    if (ereg("m", $options)) { $description =
str_replace('rel="tag"', 'title="Mot cl&eacute;', $description); }
    return $description;
}

```

Selon les lettres indiquées dans `$options` on va réaliser [3] :

- ▶ la suppression des paragraphes (remplacé par un retour à la ligne simple),
- ▶ la suppression des lignes horizontales (remplacées par un retour à la ligne simple),
- ▶ l'alignement à droite des images à gauche de la page (ce qui perturbe l'alignement des items),
- ▶ l'indication d'une rubrique,
- ▶ l'indication de mots clés.

Pour indiquer les opérations nécessaires pour chaque item, on utilise la même astuce que pour l'encodage des caractères. La fonction `encodage()` devient :

```

function encodage($flux, $options) { // détermination de l'encodage et
écriture de celui-ci dans chaque item
    $fil = file_get_contents($flux); // articles
    if (preg_match('/encoding="([\^"]*)"/i', $fil , $temp)) {
        $encode = strtolower($temp[1]);
    } elseif (preg_match("/encoding='([\^']*)'/i", $fil , $temp)) {
        $encode = strtolower($temp[1]);
    }
}

```

```
$apres = array("@@", '-  
, "<entry><encode>$encode</encode><options>$options</options>",  
          "<item><encode>$encode</encode><options>$options</options>");  
return $apres;  
}
```

L'appel se fait en précisant les options de nettoyage souhaitées :

```
$apres = encodage($flux1, "dm");
```

## Associer les fils

On prévient si un fil est absent :

```
if (!$fil1) {  
    $data = "<span class=\"rouge\">Le fichier $flux1 est introuvable ou  
incorrect.</span>\n\n";  
}  
if (!$fil2) {  
    $data.= "<span class=\"rouge\">Le fichier $flux2 est introuvable ou  
incorrect.</span>\n\n";  
}
```

On place les 2 fils dans le même tableau associatif [4] :

```
$items = array_merge($items1, $items2);
```

## Tri par date

Pour trier les items par date, j'ai créé la fonction `tri_date()` qui récupère les informations d'une date au format anglais US, la transformer en timestamp Unix (le nombre de secondes depuis le 1er Janvier 1970 à 00:00:00 GMT). On retire ensuite -3000000000 pour obtenir un nombre négatif, afin que le tri se fasse dans le bon sens (du plus récent au plus ancien) :

```
function tri_date($date) { // pour le tri par date  
    if (ereg("[0-9]", $date) and ereg("([[:digit:]]|-  
)*)T([[:digit:]]|:)*[^[[:digit:]].*", $date, $temp)) {  
        $date = $temp[1].' '.$temp[3];  
    }  
    $date = strtotime($date)-3000000000;  
    return $date;  
}
```

```
}
```

Le tri des items par date peut alors se faire par [5] :

```
usort($items, create_function('$a,$b',' return  
strcmp(tri_date($a["pubdate"]),tri_date($b["pubdate"]));' ));
```

## Suite du script

Le reste du script est très proche de celui pour parser un seul fil, à ceci près que l'on récupère \$encode que l'on avait écrit tout à l'heure [[span class="base64" title="PGNvZGUgY2xhc3M9J3NwaXBfY29kZScgZGlyPSdsdHInPiRlbnNvZGU8L2NvZGU+" data-bbox="113 338 812 365">span class="base64" title="PGNvZGUgY2xhc3M9J3NwaXBfY29kZScgZGlyPSdsdHInPiRlbnNvZGU8L2NvZGU+"](#)] :

```
$data.= "<ul>\n";  
for($i = 0; $i < 20; $i++) {  
    if (trim($items[$i]['title'])!='') {  
        $encode = trim($items[$i]['encode']);  
        $data .= '<li>';  
        if (trim($items[$i]['pubdate'])!='') {  
            $data .=  
'<b>'.clean_date(trim($items[$i]['pubdate'])).'&nbsp;:</b> '  
        }  
        if (trim($items[$i]['author'])!='') {  
            $data .= '<span  
class="orange">'.clean_text($items[$i]['author'], $encode).'&nbsp;:</span>  
'  
        }  
        $lien = '<a href="'. $items[$i]['link'].'">';  
        $data .= $lien.clean_text($items[$i]['title'],  
$encode)."</a>&nbsp;:<br />\n";  
        if (trim($items[$i]['description'])!='') {  
            $description = nettoyage(clean_text($items[$i]['description'],  
$encode), $options);  
            $data .= str_replace("(...)",[".$lien."Lire la  
suite...</a>"],$description);  
        }  
        $data .= "</li>\n";  
    }  
}
```

A noter,

► la boucle `for($i = 0; $i < 20; $i++)` c'est là qu'on indique le nombre d'items que l'on veut afficher dans la page (ici 20),

► la condition `if (trim($items[$i]['title'])!='')` permet de ne pas afficher la ligne si on a récupéré moins d'items que le nombre prévu par la boucle,

---

► la ligne `$data .= str_replace("(...)", "[".$lien."Lire la suite...</a>]", $description);` est placée ici plutôt que dans la fonction nettoyage() afin de permettre l'écriture du lien vers l'article (sur le site d'origine du flux).

[1] Ce n'est pas le cas pour les flux de syndication de [Femiweb](#) qui sont tous les deux encodés iso-8859-1 mais en testant ce script j'ai vu la nécessité de cette astuce quand les encodages sont différents.

[2] On peut avoir ainsi la mise en forme des autres items perturbée avec tout le texte en gras, en petits caractères, etc.

[3] D'autres exemples de corrections sont donnés dans [Lecture des caractères spéciaux et d'éléments d'un fil RSS/Atom](#).

[4] Pour 3 fils il suffit d'écrire `$items = array_merge($items1,$items2,$items3);` etc...

[5] Il y a sûrement plus simple, mais ça marche. Si vous avez une formule plus élégante, je suis preneur...

`[span class="base64" title='PGNvZGUGY2xhc3M9J3NwaXBfy29kZScgZGlyPSdsdHIInPirIbmNvZGUG8L2NvZGUG+' ]` est utilisé pour décoder les caractères encodés en utf-8 :  
`if ($encode == 'utf-8') { $text = utf8_decode($text); }`

### Document joint à l'article



exemple\_f.zip - Ce fichier contient exemple\_f.php décrit dans cet article.



 [Imprimer cet article](#)

Copyright Médecins Maîtres-Toile francophones  
[Espace membres](#) - [Administration](#) - [Crédits](#)

---