

Les Médecins Maîtres-Toile

[Accueil](#) ▶ [Boite à Outils](#) ▶ [Syndication de contenu](#) ▶ [Afficher un fil RSS/Atom sur votre site](#) 1/3

Publié le : 9 juillet 2005

 [Imprimer cet article](#)

Auteur :
Alain Sichel †



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

1) Présentation

Cet article fait suite à l'article "[La syndication de contenu RSS](#)" qui présente la syndication de contenu, ses avantages et quelques méthodes simples de mise en œuvre.

Dans un article précédent, j'ai présenté l'[utilisation de la classe easyRSS](#) pour permettre à des sites PHP la récupération des informations de fichiers RSS. Dominique WOJYLAC a repris le code et nettement amélioré ce lecteur en réalisant [fil_LE](#). Depuis la version 2.0 [1], cette classe est plus rapide et plus performante, elle reconnaît les formats RSS 0.91 1.0 et 2.0 et le format Atom 0.3.

Trois articles permettront de voir ses possibilités :

1. **Présentation** : utilisation de base du script,
2. **Adaptation du script**,
3. **Utilisation du script avec un cache**.

Sommaire

- [Le principe](#)
- [Premier essai](#)
- [Afficher le fichier de votre choix](#)
- [Mise en forme des données](#)
- [Un autre exemple de présentation](#)
- [Plusieurs syndications sur la même page](#)

Le principe

Le lecteur [fil_LE](#) version 2.1, [téléchargeable ici](#), comprend 12 fichiers :

- [read_rss.inc.php](#) : le fichier de la classe [read V2.1](#) de [fil_LE](#)
-

- *doc_rss_read.html* : la documentation des fonctions de la classe read
- *exemple-1.php* : un exemple général d'utilisation de la classe read
- *exemple-2.php* : un autre exemple avec "nettoyage des données"
- *exemple-3.php* : un exemple d'utilisation avec mise en cache du fichier obtenu après traitement pour limiter les requêtes vers le flux syndiqué. *exemple-3.php* utilise *parse_flux-1.php* et un dossier "cache" accessible en écriture doit être créé au même niveau que celui-ci.
- *exemple-4.php* : un exemple de lecture de plusieurs flux sur la même page avec mise en cache. *exemple-4.php* utilise *parse_flux-2.php* et un dossier "cache" accessible en écriture doit être créé au même niveau que celui-ci.
- *version.txt* : la suite des versions et modifications
- *gpl_us.txt* : version us de la licence "General Public Licence"
- *gpl_fr.txt* : version française non officielle de la licence "General Public Licence"
- et *Lisez-moi.txt* qui présente l'ensemble des fichiers.

Le lecteur Fil_LE utilise 2 fichiers :

- *rss_read.inc.php* : c'est la classe *rss_read* version 2.1 qui traite le fichier de syndication pour récupérer les informations, ce fichier doit être installé tel quel [2],
- le fichier qui affiche le fil de syndication choisi en mettant en forme les données récupérées, l'exemple de base s'appelle *exemple-1.php*.

Premier essai

Transférez les 2 fichiers *rss_read.inc.php* et *exemple-1.php* sur votre site. Attention, ces fichiers doivent être dans le même répertoire, sinon il faut modifier *exemple-1.php* à la ligne 54 :

```
include 'rss_read.inc.php';
```

en indiquant le chemin pour se rendre au fichier *rss_read.inc.php*

Affichez ensuite la page *exemple-1.php* mise sur votre site dans votre navigateur et regardez le résultat : l'affichage du fil de syndication du site MacDigit.

Afficher le fichier de votre choix

Pour afficher le fil de syndication de votre choix, ouvrez le fichier *exemple-1.php*, et indiquez, à la place des lignes 70 à 79 l'adresse de ce fichier, par exemple :

```
§rss -> parsefile('http://www.generalistes2002.net/backend.php3',15);
```

Le nombre 15 indique que seuls les 15 premiers items seront affichés. Si on n'indique pas de valeur, tous les items du fichier seront affichés.

Vous pouvez aussi renommer la page *exemple-1.php*, pourquoi pas *generalistes_2002.php* pour cet exemple ? Il ne reste plus qu'à transférer votre fichier sur votre serveur et regarder le résultat dans votre navigateur.

Mise en forme des données

La classe *rss_read* retourne les informations récupérées dans le fichier de syndication sous forme de tableaux associatifs :

- Le premier pour *channel* :

Une fois qu'on a écrit (ligne 103 de *exemple-1.php*) : `$channel = $rss -> get_channel();`

on peut récupérer tous les éléments en les écrivant tout simplement sous la forme :
`$channel['link'] $channel['title'] $channel['description']`

Cela fonctionne également s'il s'agit d'un fichier Atom, les éléments de `<feed>` sont présentés dans le tableau `$channel`.

- Le deuxième pour *image* :

De même, on écrit (ligne 108 de *exemple-1.php*) : `$image = $rss -> get_image();`

on peut ainsi retrouver tous les éléments : `$image['link'] $image['url'] $image['width'] $image['height'] $image['title']`

- Le troisième pour *item* :

Après avoir écrit (ligne 113 de *exemple-1.php*) : `$items = $rss -> get_items();`

on peut ainsi avoir tous les éléments : `$items['pubdate'] $items['author'] $items['category'] $items['link'] $items['title'] $items['description']`

Si un élément est appelé autrement, *rss_read* le récupère :

- `$items['pubdate']` pour `<date>` `<dc:date>` `<modified>` et `<published>`
- `$items['category']` pour `<dc:subject>`
- `$items['description']` pour `<summary>` et `<dc:description>`
- `$items['author']` pour `<dc:creator>`
- `$items['copyright']` pour `<dc:rights>` et `<rights>`

- Un dernier pour *textinput* :

Le principe est le même (voir les lignes 143 à 150 de *exemple-1.php*) pour afficher les éléments, mais le formulaire `textinput` n'est pas utilisé dans la plupart des fichiers de syndication.

Choisir son affichage :

Une fois compris la récupération des données, si le résultat obtenu n'est pas exactement ce que vous voulez, vous n'avez plus qu'à choisir les éléments que vous voulez présenter (et dans quel ordre ?), puis à les afficher dans un tableau, un div, une liste à puces, des paragraphes... et à les mettre en valeur (style, couleur...) en utilisant les moyens du HTML et des CSS.

Bien sûr, pour cela, il faut être un peu habitué à jongler avec le code, si ce n'est pas votre cas, je vous conseille plutôt de partir d'un exemple de présentation qui vous convient, *exemple-1.php* ou *exemple_a.php* qui est expliqué ci-dessous.

Un autre exemple de présentation

Le fichier *exemple_a.php* montre une autre façon de présenter un fichier de syndication. Je ne vais évoquer ici que les différences avec *exemple-1.php*.

► La présentation est ici sous forme de liste à puces et non pas comme précédemment dans un div, sous forme de paragraphes.

► Le format de date choisi est différent, sous forme chiffrée courte "JJ/MM/AAAA" (exemple : 23/06/2005), cette mise en forme est faite à la ligne 44 par : `$date = date('d/m/Y', strtotime($date));` au lieu de :

```
setlocale(LC_TIME, 'fr_FR');  
$date = strftime("%A %d %B %Y à %Hh", strtotime($date));
```

► Les conditions comme par exemple `if (!empty($items[$i]['pubdate']))` sont remplacées par `if (trim($items[$i]['pubdate'])!='')`. En apparence, le fonctionnement est le même, mais on évite ainsi l'affichage de caractères inutiles lorsque l'élément existe, mais est vide, ce qui arrive quand le fil contient `<pubDate />` `<author />` ou `<description />`

► A noter un choix tout à fait personnel pour les liens : `echo ''` les liens s'ouvrent dans une nouvelle fenêtre, `class="ext"` leur donne une couleur différente des liens internes au site, `title="Lien externe"` est destiné aux non voyants qui ne peuvent pas voir la couleur différente, ils sont ainsi informés par leur logiciel de lecture. Si vous préférez présenter les liens autrement, il est facile de supprimer `class="ext" title="Lien externe"` de même que le `target="_blank"`.

Si vous mettez le fichier *exemple_a.php* sur votre serveur, vous remarquerez sous le titre "*Articles de Fulmedico - Fédération des utilisateurs de logiciels médicaux et communicants* : " une répétition "*Fulmedico - Fédération des utilisateurs de logiciels médicaux et communicants*" qui correspond au titre et au lien de l'image, mais on ne voit pas de logo ! Ceci est assez fréquent avec des fils RSS générés automatiquement (comme avec SPIP) : le webmestre n'a pas remplacé l'image proposée par défaut par une image réelle. Dans ce cas, tant que le webmestre n'a pas modifié son fichier, il vaut mieux ne pas afficher ceci en bloquant les lignes 77 à 81 par exemple en tapant `/*` avant et `*/` après, ce qui neutralisera le code, mais vous permettra de le remettre par la suite.

Plusieurs syndications sur la même page

Comment faire si on veut afficher plusieurs fils sur la même page ?

Vous pourrez voir le code d'un tel fichier avec *exemple_b.php*. Ce fichier reprend le même code, mais au lieu de le répéter exactement :

```
$rss = new rss_read();
$encode = $rss -> get_encoding();
$rss -> parsefile('http://www.chemg.org/backend.php3',15);
if ($rss) {
$channel = $rss -> get_channel();
```

on remplace `$rss` par une autre variable `$fil` par exemple, en indiquant bien sûr l'URL du nouveau fil : `$fil -> parsefile('http://www.chemg.org/backend-breves.php3',15);`

Autres changements, par rapport à *exemple_a.php*, cette fois-ci :

- ▶ J'ai disposé titre et logo sur la même ligne (ce sont les 2 cellules d'un tableau).
- ▶ L'écriture de certains éléments de `$channel` (*description*) et de `$image` (*width*) et (*height*) n'a lieu que s'ils existent, par exemple :

```
if (trim($channel['description'])!='') {
echo" title=",clean_text($channel['description'],$encode);
}
```

- ▶ J'ai changé le format de date, pour montrer à quel point c'est simple, en remplaçant `$date = date('d/m/Y', strtotime($date));` par

```
setlocale(LC_TIME, 'fr_FR');
$date = ucfirst(strftime("%A %d %B %Y", strtotime($date)));
```

(lignes 43-44) [3].

Si vous [regardez le résultat](#), vous verrez que cela fonctionne très bien, mais que l'affichage des brèves n'est pas très esthétique.

Peux-t-on faire quelque chose pour éviter d'afficher tous ces espaces et les lignes horizontales ? Oui, bien sûr, je l'ai même [réalisé](#), mais si on veut modifier cet affichage, cela mérite bien un [deuxième article](#).

[1] La [dernière version disponible est la 2.1](#).

[2] Le code de *rss_read.inc.php* ne doit pas être modifié, il ne sera donc pas expliqué dans ces articles, sauf si j'ai plusieurs demandes sur ce sujet.


[3] La fonction `ucfirst()` met la première lettre de la chaîne en majuscule, pour voir les autres possibilités d'affichage avec la fonction `strftime()`, voir [cette page](#).

Document joint à l'article



exemples.zip - Ce fichier contient *exemple_a.php* et *exemple_b.php* décrits dans cet article.



 [Imprimer cet article](#)

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