

# Les Médecins Maîtres-Toile

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

**Publié le : 9 juillet 2005**

 [Imprimer cet article](#)

**Auteur :**  
**Alain Sichel †**



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

### 3) Utilisation d'un cache

Après les articles précédents sur l'utilisation de **fil\_LE** version 2.1 de Dominique WOJYLAC dans **les situations courantes**, puis **les adaptations pour améliorer la présentation**, nous allons voir comment mettre en place un système de cache [1].

Ceci concerne les fils de syndication régulièrement présentés sur votre site et offre comme avantages un accès plus rapide au fichier quand celui-ci est dans le cache et moins de demandes à votre serveur.

#### Sommaire

- [Le principe \[2\]](#)
- [Premier exemple avec cache](#)
- [Deuxième exemple avec cache](#)
- [Autre solution](#)
- [Conclusion provisoire](#)

### Le principe [2]

Lorsqu'on syndique sur son site un flux RSS ou Atom d'une autre source, celle-ci pour des raisons diverses peut se trouver indisponible ou lente d'accès. Pour ne pas avoir à recalculer la page de votre site et à faire une requête sur le flux distant à chaque visite de celle-ci, il suffit d'en mettre tout ou partie en cache. Pour cela on génère un fichier contenant le code HTML résultat de l'analyse du flux distant à intervalles réguliers. Le reste du temps on se contente d'inclure le code généré précédemment.

Voici le déroulement des opérations :

```
Si (le fichier n'existe pas en cache ou il est trop ancien) {  
  parser le flux distant;  
}
```

```
g n rer le fichier en cache;
} // fin
inclure le fichier en cache;
```

Le fichier est-il en cache ? [file\_exists(string filename) nous permet de le savoir ; son  ge est donn  par filemtime(string filename) qui retourne la date de sa derni re modification sous forme d'un timestamp (nombre de secondes)]

Cette solution pr sente le gros avantage de ne pas avoir   intervenir sur la configuration serveur (impossible pour un serveur mutualis ) ou   avoir recours   un serveur tiers d clenchant la requ te sur un fichier PHP design  sur votre site (solution du type cron.hiwit.org ou www.webcron.org). C'est la requ te elle-m me sur la page qui d clenche sa mise   jour ou non.

Pour parser le flux distant la m thode est la m me que celle d crite dans le fichier exemple. Au lieu de faire des echo successifs pour afficher le r sultat de l'analyse, on stocke les chaines g n r es dans une variable unique \$data qui sera ensuite  crite dans un fichier plac  en cache.

---

## Premier exemple avec cache

Cet exemple est pr sent  par Dominique WOJYLAC dans [exemple-3.php](#) [3]

Dans le fichier o  doit s'afficher le flux RSS il suffira de faire une inclusion du fichier [parse\\_flux.php](#)  
<?php include 'parse\_flux.php'; ?>

N'oubliez pas de cr er un dossier "cache" ayant tous les droits en  criture (  adapter suivant l'h bergeur).

Cet exemple suppose que l'on affiche simplement le titre du fil RSS et les titres de chacun des items, il peut  tre adapt  pour un affichage plus complet (date, auteur, description...).

---

## Deuxi me exemple avec cache

Deuxi me exemple de Dominique WOJYLAC : [exemple-4.php](#).

Ici, la mise en cache a  t  transform e en une fonction :

```
affiche_fil($url_fil, $fichier_cache, $delai, $nombre_items);
```

avec :

- ▶ \$url\_fil = URL du fil rss   syndiquer
- ▶ \$fichier\_cache = nom du fichier mis en cache
- ▶ \$delai = dur e en heure entre deux rafraichissement du fichier en cache (possibilit  de mettre

des heures décimales - 1.5 par exemple -)

► \$nombre\_items = nombre maximum d'items à afficher.

Il est également possible d'afficher plusieurs fils syndiqués sur la même page avec mise en cache.

---

## Autre solution

Ces exemples fonctionnent très bien, mais ne répondent pas exactement à ce que je voulais faire, je vous propose donc une adaptation d'*exemple-3.php* qui permet une présentation différenciée pour chaque fichier (selon les éléments que l'on veut présenter et le nettoyage adapté du code de l'élément *description* pour avoir un affichage satisfaisant).

Le résultat est illustré dans *exemple\_e.php*, téléchargeable en bas de cette page.

Je vais partir du code du fichier *exemple\_d.php* présenté dans l'article précédent et expliquer ce que je change :

► Tout d'abord je décide d'afficher 2 des fils de syndication du site allergique.org :

<http://www.allergique.org/backunan.php3> et <http://www.allergique.org/backend-breves.php3>.

► La fonction nettoyage() [4] est plus simple :

```
function nettoyage($description) {
    $avant = array
    ("<p.*?>'si", "</p>'si", "<hr.*?>'si", "'align=\"left\"'si");
    $apres = array (" ", "<br />", "<br />", "align=\"right\"");
    $description = preg_replace($avant, $apres, $description);
    return $description;
}
```

► Ensuite, çà change :

```
// chemin relatif vers le fichier cache (le répertoire "cache" doit être
autorisé en écriture)
$file_cache = 'cache/allergique.html';
// le delai entre deux rafraichissements 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 considéré comme n'étant pas en cache on le génère
    include 'rss_read.inc.php';
    $rss = new rss_read();
}
```

```
$data = '';
```

Les commentaires dans le code sont assez clairs pour qu'il n'y ait rien à ajouter.

► Puis on continue comme on le faisait dans *exemple\_d.php* à indiquer les informations que l'on veut montrer de *\$channel \$image* et de *\$items*, à ceci près qu'on lieu de l'afficher dans la page avec `echo` on le met dans la variable `$data` [5]

► On continue de la même manière pour un autre fil (les brèves d'allergique.org), et après la fin :

```
$data .= "</ul>\n";  
} // fin if $fil  
else {  
    exit("Le fichier RSS sur les brèves des MMT n'est pas disponible  
actuellement.");  
}
```

on ajoute pour finir et permettre la mise en cache des données :

```
$fd = fopen($file_cache, "w");  
fputs($fd, $data);  
fclose($fd);  
  
} // fin if $en_cache  
  
include $file_cache
```

Le dernière ligne de code permet l'affichage des informations dans la page.

---

## Conclusion provisoire

Voilà donc trois solutions de mise en cache, à vous de choisir.

Volontairement dans cet article en 3 parties, je n'ai pas cru nécessaire d'expliquer tout le script des exemples, car les commentaires de Dominique WOJYLAC dans le code sont clairs, et il y a aussi la [documentation](#). Si on me signale des points à éclaircir, je modifierais les articles en conséquence.

---

[1] Pour bien comprendre cet article, il faut avoir lu les deux précédents.

[2] Ces explications reprennent celles de Dominique WOJYLAC dans [exemple-3.php](#).

[3] Rappel, tous les fichiers du lecteur fil\_LE version 2.1 peuvent être [téléchargés ici](#).

[4] La fonction nettoyage() permet un meilleur affichage et la réparation de certaines erreurs, voir [l'article précédent](#).

[5] Pour ajouter au fur et à mesure, il suffit d'écrire `$data .= "élément à mettre à la suite";` c'est le point avant le signe égal qui indique que cela s'ajoute à la valeur précédente de `$data`. Attention, si `echo` était écrit avec des virgules pour ajouter des éléments, il faut remplacer par des points, sinon vous aurez une erreur PHP.

### Document joint à l'article



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



 [Imprimer cet article](#)

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

---