

Réécriture d'URL [URL Rewriting sous XOOPS] **XOOPS**

Posté par: Ash

Publiée le : 08/03/2007 12:10:00

Ce sujet ne vante absolument pas les mérites de l'URL rewriting, de nombreux sites sont très bien référencés sans URL rewriting, on le sait. L'URL rewriting favorise le référencement mais ça ne suffit pas, il faut du contenu, on le sait. Donc, pas besoin de débattre sur l'utilité ou pas de ce type de technique

Je fais l'impasse pour les termes et insultes techniques, si vous souhaitez plus d'informations sur les expressions régulières, installation serveur et autre, direction :

J'ai décidé de mettre sur papier toutes les infos sur ce thème, afin de les retrouver si ma mémoire devait se percer un jour, en même temps, j'en fais profiter les autres. Tous les exemples sont réalisés avec le forum newbb (cbb) valable jusqu'à la version 1.16. De toute façon, ces méthodes sont les mêmes pour tous les modules, du moins leurs principes. Une autre technique alternative et plus douce est expliquée [ici](#)

Mais pour les psychopathes endurcis c'est par ici que ça commence : 1. Première méthode, le hack **Pour l'URL rewriting j'utilise le très vieux hack de kenogui :**

<http://www.frxoops.org/modules/news/article.php?storyid=602>

<http://kenogui.free.fr/XoopsModRewrite-1.0.zip> Le principe de ce hack est vraiment super, il consiste à réécrire les URL de façon dynamique et à la volée. De plus, vous pouvez personnaliser vos règles de réécriture, exemple :

mon-url-c-est-la-plus-belle-mais-avec-50-mots-cles-elle-nest-pas-tres-pertinente.coca ou un simple t45-28-2555.html et encore, ce que vous voulez ! Avantage, il évite de modifier les fichiers php des modules. Problème, les règles de ce hack sont très vieilles et des modules n'existent même plus, il faut donc tout retravailler. Dans un premier temps il faut lire le fichier readme.txt et ajouter les morceaux de codes proposés. Faites une traduction des commentaires avec google, si vous avez des problèmes avec l'anglais. Si vous avez des problèmes avec le fichier header.php vous avez une réponse ici : [header.php Ensuite tout se joue avec les fichiers ModRegExp.php et .htaccess Pour simplifier au maximum Le fichier ModRegExp.php, niveau script php, sert à dire : cette URL doit ce transformer en cette URL Le fichier .htaccess, niveau serveur, sert à appliquer la règle de réécriture et la rendre compréhensible pour le serveur, blablabla, passe détails Pour commencer, il faut s'occuper des URL accessibles à la racine \(en page d'accueil\), donc, un fichier ModRegExp.php sera placé à la racine. A savoir que je ne place pas les extensions .html ou .php je préfère les virer car inutiles. En plus, on peut mettre ce que l'on veut comme extensions. Si vous souhaitez les rajouter il faut juste placer .html ou .cocacola à la fin des URL que l'on souhaite obtenir. Dans ModRegExp.php il y a deux zones La première : urlin où il faut placer les URL non-réécrites :](#)

[Dans cet exemple, on va réécrire deux URL différentes :](#)

-
"(!?,

est premier dans urlin et que :

-
"modules/newbb/topic-\1-forum-\2".

doit être premier dans urlout, ainsi de suite. La moindre erreur, virgule, slash en moins, etc, peut faire planter le site. En principe, c'est une page blanche que vous avez quand il y a une erreur dans une règle de réécriture. De toute façon, cela n'est pas un problème puisque vous travaillez sur un site de test et non celui en production, enfin, j'espère!!Vérification de l'application des règles On clique sur un des sujets du forum ou sur une catégorie qui se trouvent dans un bloc en page d'accueil et si les adresses sont bien de type :

<http://www.example.com/modules/newbb/topic-724-forum-16>

et

http://www.example.com/modules/newbb/forum_blabla-11

c'est gagné ! Pour la réécriture coté php uniquement. Car l'adresse renvoie une page 404. Ce qui est excellent. En effet, l'url est réécrite. Il ne reste plus qu'à passer à la suite. Maintenant, il faut que le serveur apache puisse comprendre cette URL afin de servir la page, c'est le rôle du fichier .htaccess. Toujours à la racine, puisque l'on souhaite réécrire dans un premier temps les URL de la page d'accueil, on crée un fichier .htaccess :

C'est très simple, voici le contenu de notre fichier .htaccess :

(Attention : Les règles de réécriture d'URL peuvent cohabiter avec les options de configuration déjà mis en place. Si un fichier .htaccess est déjà présent, il suffit juste de l'éditer et d'ajouter les règles suivantes)

Options +FollowSymlinks
RewriteEngine On

```
RewriteRule ^modules/newbb/topic-([0-9+])-forum-([0-9+])$ modules/newbb/viewtopic.php?topic_id=$1&forum=$2 [L]
RewriteRule ^modules/newbb/forum_blabla-([0-9+])$ modules/newbb/viewforum.php?forum=$1 [L]
```

En langage humain ça veut dire :

L'adresse : modules/newbb/viewtopic.php?topic_id égale variable 1 / forum égale variable 2 /

sera réécrite ainsi : modules/newbb/topic-([0-9+])-forum-([0-9+])

Si la première partie avec le fichier ModRegExp.php est un succès et que le serveur apache autorise la réécriture d'URL, soit ça fonctionne et dans ce cas bravo, soit il affiche une erreur 404 et là

toujours bravo. Si une page 404 s'affiche, il faut revoir tranquillement les règles de réécriture du fichier .htaccess, il y a forcément une erreur de frappe, syntaxe. En revanche si une erreur code 500 s'affiche, là c'est un peu plus chiant, le serveur n'est sûrement pas configuré pour la réécriture d'URL ou une grosse erreur est présente dans le fichier .htaccess. Dans ce cas 2 solutions, vous essayez de corriger et si ça ne fonctionne toujours pas alors direction "son hébergeur" si on est en mutualisé pour obtenir des renseignements ou en cuisine si on a un serveur dédié pour configurer le tout :

Une petite solution (ultra douce) :

http://www.webrankinfo.com/forums/viewtopic_4291.htm

Si elle ne fonctionne pas, alors rendez-vous à la fin de ce document (Chapitre 4) pour les sites sur serveur dédié. Une partie peut également intéresser les sites sur hébergement mutualisé Voir (phpinfo) **Maintenant que les URL sont réécrites en page d'accueil, il faut passer aux URL profondes.** En effet, j'ai traité les URL en page d'accueil uniquement, donc quand je clique sur un lien l'URL est bien réécrite et le sujet est accessible, mais ça ne fonctionnera que dans le sens page d'accueil vers le modules/newbb. Une fois arrivé sur ce module, plus aucune règle ne fonctionnera. Il faut juste créer un fichier ModRegExp.php et créer un fichier .htaccess à la racine de modules/newbb :

La seule différence sera que les liens seront relatifs à ce dossier, donc écrits ainsi :

Les lignes :

```
RewriteRule ^topic-([0-9+)-forum-([0-9+)]$ viewtopic.php?topic_id=$1&forum=$2 [L]
RewriteRule ^forum_blabla-([0-9+)]$ viewforum.php?forum=$1 [L]
```

remplaceront pour le fichier .htaccess les lignes :

```
RewriteRule ^modules/newbb/topic-([0-9+)-forum-([0-9+)]$ modules/newbb/viewtopic.php?topic_id=$1&forum=$2 [L]
RewriteRule ^modules/newbb/forum_blabla-([0-9+)]$ modules/newbb/viewforum.php?forum=$1 [L]
```

Et dans les zones urlin et urlout du fichier ModRegExp.php, les lignes :

```
"(?,"
"(","topic-1-forum-2",
"forum_blabla-1",
```

remplaceront les lignes :

```
"(?,"
```

```
"(?,
```

```
"modules/newbb/topic-1-forum-2",
```

```
"modules/newbb/forum_blabla-1",
```

Bonus : Remplacer modules/newbb par cequejeveux/Il faut déjà être à l'aise avec le hack kenogui, avoir compris 2 ou 3 trucs, mais c'est pas sorcier.

L'astuce c'est de placer une règle de réécriture dans le fichier ModRegExp.php à la racine / :

```
//Dans la zone urlin
```

```
//Ajoutez
```

```
"(?//Pour le reste normalement si vous avez bien suivi le début
```

```
//vous n'avez aucun changement à faire ici.
```

```
"(?,
```

```
"(?,
```

```
//Dans la zone urlout
```

```
//Ajoutez
```

```
"cequejeveux",
```

```
//Et il faut remplacer les anciennes lignes par celles-ci
```

```
"cequejeveux/topic-1-forum-2",
```

```
"cequejeveux/forum_blabla-1",
```

On fait la même chose dans le fichier ModRegExp.php à la racine de /modules/newbb/ :

```
//Dans la zone urlin
```

```
//Ajoutez
```

```
"(?"(?//)index.php""",
```

```
"(?,
```

```
//Normalement si vous avez bien suivi le début
```

```
//vous n'avez aucun changement à faire ici.
```

```
"(?,
```

```
"(?,
```

```
//Dans la zone urlout
```

```
//Ajoutez
```

```
"../cequejeveux/""",
```

```
"../cequejeveux/""",
```

```
"../cequejeveux/""",
```

```
//Et il faut remplacer les anciennes lignes par celles-ci
```

```
"../cequejeveux/topic-1-forum-2",
```

```
"../cequejeveux/forum_blabla-1",
```

On teste et on passe ensuite au fichier .htaccess :

En fait, on va créer un dossier virtuel qui ne doit absolument pas exister physiquement ^^ ! Alors ne pas le créer. Et comme d'habitude, c'est le fichier .htaccess qui va se charger de cuisiner tout ça :

Donc à la racine de notre site / on ajoute dans le fichier .htaccess ceci

```
RewriteRule ^cequejeveux$ modules/newbb/ [L]
RewriteRule ^cequejeveux/(.*)-(.*)$ modules/newbb/$1-$2 [L]
RewriteRule ^cequejeveux/(.*)$ modules/newbb/$1 [L]
```

On teste et on obtient ces URL :

www.example.com/cequejeveux/
www.example.com/cequejeveux/topic-2-forum-3
www.example.com/cequejeveux/forum_blabla-3

Si vous obtenez des erreurs 404 mais que les URL sont bien de ce type c'est déjà très bien, il faut juste relire votre fichier .htaccess pour chasser les erreurs de frappe ^^

Oui, j'avais écrit que (.*) ou autre truc du genre ne devaient pas être utilisés par principe, car si on a besoin de chiffres uniquement, pourquoi tout accepter ? Sauf qu'ici, nous avons déjà fait une bonne partie du boulot, les URL sont déjà traitées si vous avez bien suivi tout le début de la première méthode. Il n'y a donc aucun problème, seuls les caractères déjà autorisés dans votre fichier modules/newbb/.htaccess seront utilisés, les autres seront interdits. Notez que le fichier modules/newbb/.htaccess ne sera pas modifié pour cet exercice. Il a déjà été mis au point en début de partie de la première méthode. Vous devez modifier uniquement celui de la racine de votre site.

2. Seconde méthode : Modification des fichiers php et templates Tout fonctionne, le hack de kenogui est opérationnel et vous avez déjà créé plusieurs règles de réécriture d'URL. Malheureusement, ce n'est pas très souple et une réécriture totale de votre site est impossible. Ou bien, vous souhaitez des trucs un peu plus exotiques du genre, réécrire les URL des profils et ajouter le nom, etc... Il y a donc une autre solution qui consiste à modifier les URL directement dans les fichiers php ou templates. **Attention !** Modifier les sources de xoops rend très difficile les mises à jour par la suite. Il faut les faire à la main comme avec du phpbb sauf que pour l'instant il n'existe pas de document pour faire les mises à jour pas à pas chez xoops (Pas à ma connaissance). Si vous faites une mise à jour en automatique de xoops, vos fichiers seront donc écrasés et vos modifications perdues. Vous êtes prévenus. **Je prends toujours comme exemple le module newbb (cbb) :** C'est encore plus simple que le hack de kenogui, cependant ce dernier avait le mérite d'éviter de modifier les fichiers php des modules. Il faut juste chercher les lignes qui commencent par href dans les fichiers php :

Exemple viewtopic.php :

Je fais une recherche de href ou viewtopic.php? :

je tombe sur :

```
$xoopsTpl->assign(array('topic_title' => ".$bbUrl['root'].viewtopic.php?viewmode=')
```

.\$viewmode.&topic_id='\$topic_id.'&forum='.\$forum.'">'.\$forumdata['topic_title'].",

Cette version me donne :

http://www.bulma-animation.org/module ... lat&topic_id=730&forum=67

Je désire ce type d'URL :

<http://www.bulma-animation.org/modules/newbb/viewf-flat-730-67>

Je vais donc modifier cette ligne ainsi :

```
$xoopsTpl->assign(array('topic_title' => '.$bbUrl['root'].  
'viewf-'.$viewmode.'-'. $topic_id.'-'. $forum.'">'.$forumdata['topic_title'].",
```

viewf et son tiret le séparateur >> - Cependant il faut qu'il y est la même chose dans votre .htaccess qui va venir)Le 2 ème tiret remplace &topic_id= (Pareil j'ai mis un tiret, mais je peux mettre -cocacola- (Mais il faut toujours qu'il y ait une séparation, donc pas de cocacola sans un tiret avant et un tiret après)Le 3 ème tiret remplace &forum= (idem)

J'enregistre les modifications et je teste l'URL sur la page :

http://www.bulma-animation.org/module ... php?topic_id=730&forum=67Dans cet exemple on a modifié le lien qui se trouve 2 niveaux en dessous de index des forums (Attention, dans cet exemple d'adresse vous obtiendrez une version différente sur Bulma Animation). J'ai fait exprès de choisir celui-ci car en plus de chiffres on a également des lettres dans cette URL "viewf-flat-730-67"flat = la valeur de la variable viewmode= que nous avons transformé en viewf-

Cliquez et si vous obtenez <http://www.bulma-animation.org/modules/newbb/viewf-flat-730-67> avec une belle erreur 404 c'est parfait !Pour qu'Apache puisse la servir il faut donc créer une règle dans un fichier .htaccess déjà présent ou créer ce fichier pour l'occasion et qui servira pour tout le dossier modules/newbb (y a du boulot) Dans notre exemple, le fichier .htaccess doit se trouver dans le dossier modules/newbb soit modules/newbb/.htaccessIl doit contenir ces lignes :

```
Options +FollowSymlinks  
RewriteEngine On
```

```
RewriteRule ^viewf-([a-z]+)-([0-9]+)-([0-9]+)$ viewtopic.php?viewmode=$1&topic_id=$2&forum=$3 [L  
]
```

viewf-([a-z]+) ==> ([a-z]+) car la valeur ne contiendra que des lettres minuscules de a à z, donc pas besoin de faire compliquer. Elle correspond à la variable numéro 1 (\$1) soit l'ensemble viewtopic.php?viewmode=\$1 (Dans le cas où vous avez besoin de réécrire une URL avec une variable qui contient des chiffres et des lettres minuscules et majuscules, il faut donc placer ([a-z,A-Z,0-9]+)

-([0-9]+)- ==> -([0-9]+)- car la valeur ne contiendra que des chiffres de 0 à 9. Les tirets avant et

après sont les séparateurs. Elle correspond à la variable numéro 2 (\$2) soit l'ensemble topic_id=\$2

Etc...

viewf-([a-z]+)-([0-9]+)-([0-9]+) sont séparés par des tirets >> -

viewtopic.php?viewmode=\$1&topic_id=\$2&forum=\$3 sont séparés par des esperluettes >> &

Toujours dans l'ordre viewf-([a-z]+)-([0-9]+)-([0-9]+) | \$1 \$2 \$3 Notre exemple comporte une URL à trois variable, mais le principe est identique avec 5/6/7 variables, etc... Vous avez compris le système avec les scripts php ? Vous avez aussi la solution (parfois) de modifier les templates. Oui encore de la souffrance en vue :

Exemple :

Une ligne de ce genre dans le templates newbb_block_new.html :

```
a href="/modules/newbb/viewtopic.php?topic_id=&forum=">
```

Peut être réécrite ainsi :

```
a href="/modules/newbb/sujet--forum-">
```

Ensuite il faut créer la règle dans un fichier .htaccess. Dans ce cas, il s'agit d'un bloc que j'ai placé en page d'accueil. **Je dois donc éditer le fichier .htaccess situé à la racine :**

Options +FollowSymlinks

RewriteEngine On

#newbb

```
RewriteRule ^modules/newbb/sujet-([0-9]+)-forum-([0-9]+)$ modules/newbb/viewtopic.php?topic_id=$1&forum=$2 [L]
```

C'est enfin terminé, a vous de mettre en place vos propres règles de réécriture d'URL afin d'éviter des trucs du genre :

[http://www.bulma-animation.org/module ... since=1000&sortorder=DESC](http://www.bulma-animation.org/module...since=1000&sortorder=DESC) â€¦ a prend du temps, il existe des méthodes automatiques pour la réécriture d'URL, mais moins souples, a vous de choisir. Conclusion, l'URL rewriting c'est pas magique et automatique. Un .htaccess bien rempli ne suffit pas pour réécrire de façon dynamique les URL. Il faut soit utiliser des hacks pour rendre la tâche automatique ou bien bricoler dans le code source des fichiers xoops. La seconde méthode est utilisable sur n'importe quel site php. Bon courage

3. Evitez le "Duplicate content" ou "Pages similaires" **Indispensable pour un URL rewriting avancé** : Le fichier robots.txt est à prendre en compte très sérieusement pour faire copain-copain avec les bons moteurs de recherche Exemple, je souhaite interdire l'accès de ces URL aux moteurs parce que :

[http://www.bulma-animation.org/module ... bb/viewf-flat-DESC-363-29](http://www.bulma-animation.org/module...bb/viewf-flat-DESC-363-29)

et inutile car c'est un doublon de la page :

<http://www.bulma-animation.org/module ... b/page-363-29-flat-DESC-0>

qui est inutile aussi car c'est un doublon de la page :

<http://www.bulma-animation.org/module ... tion-363-5285-0-flat-0-29>

Pareil pour cette page :

http://www.bulma-animation.org/module ... php?topic_id=363&forum=29

qui est inutile car c'est un doublon de :

<http://www.bulma-animation.org/modules/newbb/viewf-flat-363-29>Etc.... à €ja vous fait rire hein ?

Ben pas les moteurs. C'est le meilleur moyen de les énerver car trop de pages identiques. Alors la seule et unique version de page à faire manger aux moteurs c'est :

<http://www.bulma-animation.org/modules/newbb/topic-363-forum-29> / Bien sûr, on peut très bien supprimer ces liens doublons, mais il faut penser aux utilisateurs aussi. Certaines de ces URL sont très pratiques, notamment : viewf-flat-ASC-719-58 et viewf-flat-DESC-719-58 qui permettent à l'utilisateur de choisir l'ordre des messages en un clique (Plus récent, moins récent). Il faut alors faire deux types de tri : pour les moteurs et utilisateurs et pour les utilisateurs uniquement. Remarquez que je place des préfixes différents à ces URL pour mieux gérer le tri. Je souhaite interdire les liens de sélection uniquement et non les liens officiels. Le début du nom doit être différent d'oà¹ le préfixe viewf pour ne pas exclure topic. Si leur nom commençait de la même façon, le tri serait impossible. Je vais donc ajouter des interdictions dans mon fichier robots.txt. Les 5 dernières lignes signalent aux moteurs que toutes les pages qui commencent par viewf, contribution, page, viewforum, viewtopic doivent être ignorées !

User-agent: *

Disallow: /cgi-bin/

Disallow: /tmp/

Disallow: /cache/

Disallow: /class/

Disallow: /images/

Disallow: /include/

Disallow: /install/

Disallow: /kernel/

Disallow: /language/

Disallow: /templates_c/

Disallow: /themes/

Disallow: /uploads/

Disallow: /modules/newbb/viewf

Disallow: /modules/newbb/contribution

Disallow: /modules/newbb/viewforum.php

Disallow: /modules/newbb/page

Disallow: /modules/newbb/viewtopic.php

4. Quelques infos supplémentaires **Très important** : Ne travaillez jamais sur un site en production et sauvegardez toujours vos fichiers avant de faire une modification, même minime soit-elle ! N'hésitez pas à commenter votre travail : Vous pouvez mettre des commentaires ou désactiver des lignes à problèmes dans les fichiers php ou .htaccess Si une ligne cause problème, pas besoin de désactiver le fichier entier. En plus commenter un fichier peut rendre son organisation plus claire. Pour les fichiers .htaccess toutes les lignes qui commencent par # sont ignorées. Pour les fichiers php toutes

les lignes qui commencent par // sont ignorées.

Exemple de commentaires et de lignes dans un fichier php :

```
//Réécritures URL forum

//Ces lignes ne seront pas pris en compte par le système,
//mais vous permettront d'avoir une information quand vous éditez le fichier
//la prochaine fois.
//$xoopsTpl->assign(array('topic_title' => ".$forumdata['topic_title'].",
// Liens Bonus :Si mon article vous donne envie de stopper à la première ligne, je ne suis pas jaloux
:
```

<http://www.webrankinfo.com/analyses/autres/url-rewriting.php>

http://www.illiweb.com/manuel/Apache_1.3_VF/mod/mod_rewrite.html

[http://www.frxoops.org/modules/news/a ... id=971&keywords=rewriting](http://www.frxoops.org/modules/news/a...id=971&keywords=rewriting) Savoir si rewrite est présent sur son serveur : **Hébergement Mutualisé ou Serveur Dédié (Méthode douce) :**

Créez un fichier à la racine de votre site (par exemple) totalinfo.php ou ce que vous voulez.php et ajoutez ceci :

()?> **Important:** Quand vous n'avez plus besoin de ce fichier, supprimez le ! ou protégez le par mot de passe.

Ensuite lancez cette page en tapant son adresse <http://example.com/totalinfo.php> Recherchez la zone Apache et précisément Loaded Modules : Vous trouvez mod_rewrite, ouf, c'est bon. Dans le cas contraire passez à la méthode un peu moins douce pour les serveurs dédiés uniquement, pour les autres, contacter votre hébergeur :

Si on a un serveur dédié avec Debian GNU/linux il faut activer le mod rewrite :

```
dpkg-reconfigure apache
```

A la page des mods il faut cocher mod_rewrite

Vous avez maintenant Apache avec le mod_rewrite. Vérifiez le avec le fichier <http://example.com/totalinfo.php> Si vous avez une autre distribution alors cherchez sur google : rewrite apache xoops ou bien passez à la méthode dure qui consiste à chercher si apache a de quoi activer ce pu@!n de mod_rewrite avant de passer à une recompilation du serveur http. C'est pas difficile, mais vaut mieux faire quelques recherches avant de se lancer avec la trinité .configure make make install :

Vous pouvez lancer une recherche avec cette commande, si vous ne trouvez pas le fichier modules.conf qui peut avoir un emplacement ou un nom différents en fonction des systèmes (Valable pour toutes les distributions, normalement) :

```
bulma2:~# find /etc . -type f -print | xargs grep LoadModule
```

Dés que vous avez le résultat appuyez sur control + c (ctrl c) pour arrêter la recherche.

Ok si il répond des trucs du genre (on a donc trouvé son emplacement : /etc/apache/modules.conf) :

```
/etc/apache/modules.conf:LoadModule config_log_module /usr/lib/apache/1.3/mod_log_config.so
/etc/apache/modules.conf:LoadModule mime_magic_module
/usr/lib/apache/1.3/mod_mime_magic.so
/etc/apache/modules.conf:LoadModule mime_module /usr/lib/apache/1.3/mod_mime.so
/etc/apache/modules.conf:LoadModule negotiation_module /usr/lib/apache/1.3/mod_negotiation.so
/etc/apache/modules.conf:LoadModule status_module /usr/lib/apache/1.3/mod_status.so
/etc/apache/modules.conf:LoadModule autoindex_module /usr/lib/apache/1.3/mod_autoindex.so
/etc/apache/modules.conf:LoadModule dir_module /usr/lib/apache/1.3/mod_dir.so
/etc/apache/modules.conf:LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so
/etc/apache/modules.conf:LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
/etc/apache/modules.conf:LoadModule alias_module /usr/lib/apache/1.3/mod_alias.so
/etc/apache/modules.conf:LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so
/etc/apache/modules.conf:LoadModule access_module /usr/lib/apache/1.3/mod_access.so
/etc/apache/modules.conf:LoadModule auth_module /usr/lib/apache/1.3/mod_auth.so
/etc/apache/modules.conf:LoadModule expires_module /usr/lib/apache/1.3/mod_expires.so
/etc/apache/modules.conf:LoadModule setenvif_module /usr/lib/apache/1.3/mod_setenvif.so
/etc/apache/modules.conf:LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

Si vous êtes pressé et qu'il est 2 heures du matin lancez la commande de recherche avec un nice -20 (Priorité maximum pour le processus) :

```
bulma2:~# nice -n -20 find /etc . -type f -print | xargs grep mod_rewrite
```

grep LoadModule va rechercher le mot LoadModule dans tous les fichiers de /etc

Dés que vous avez la réponse vous pouvez mettre fin à la recherche avec la combinaison : Control + c (ctrl c)

Vous avez donc trouvé le fichier /etc/apache/modules.conf. Il ne reste plus qu'à voir si la ligne LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so est bien présente.

Si vous ne trouvez pas la ligne, c'est que le mod_rewrite n'est pas activé, alors cherchez dans le dossier /usr ce fameux module mod_rewrite :

```
bulma2:~# nice -n -20 find /usr . -type f -print | xargs grep mod_rewrite
```

Ok si il répond par :

```
/usr/lib/apache/1.3/200mod_rewrite.info:LoadModule: rewrite_module
/usr/lib/apache/1.3/mod_rewrite.so
Fichier binaire /usr/lib/apache/1.3/mod_rewrite.so concorde
```

La recherche dit que mod_rewrite.so se trouve dans /usr/lib/apache/1.3/mod_rewrite.so il faut donc ajouter dans le fichier /etc/apache/modules.conf la ligne : LoadModule rewrite_module /usr/lib/apache/1.3/mod_rewrite.so et redémarrer Apache. Ouf...

Vous ne trouvez toujours pas l'endroit où se trouvent les fichiers :

Tentez une recherche dans tous les dossiers du serveur :

```
bulma2:~# find / . -type f -print | xargs grep LoadModule
```

```
bulma2:~# find / . -type f -print | xargs grep mod_rewrite
```

Plus rapide mais ne pas utiliser cette commande aux heures de pointe (N'oubliez pas que nice -n -20 va donner une priorité maxi à votre processus de recherche très gourmand en ressource) :

```
bulma2:~# nice -n -20 find / . -type f -print | xargs grep LoadModule
```

```
bulma2:~# nice -n -20 find / . -type f -print | xargs grep mod_rewrite
```

Vous avez tout trouvé et modifié, le fichier <http://example.com/totalinfo.php> vous indique que le mod_rewrite est présent, bravo. Toutes ces recherches n'ont rien donné ? Pfffffff bordel...

Alors vous êtes bon pour une compilation d'apache :

http://www.linuxfrench.net/article.php3?id_article=926

Tout est ok mais ça ne fonctionne toujours pas ? Alors il faut peut-être réaliser une petite modification dans le fichier de conf httpd.conf qui se trouve en principe dans /etc/apache/httpd.conf

Cherchez la ligne et remplacez AllowOverride none par AllowOverride All :

```
#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride All
```

Pour les autres distributions cherchez sur google : rewrite apache xoops

Et si ça fonctionne toujours pas, ne jouez pas au loto samedi. shutdown et dodo.

5. C'est terminé **Encore une fois** : le fait de modifier le code source de xoops rend les mises à jour difficiles. Tant pis, s'il y a des mises à jour à faire il faut les réaliser quand même ! Au pire, vous refaites les règles de réécriture après une remise à niveau automatique du cms, c'est plus rapide que de mettre à jour xoops à la main. Lignes foireuses : Si vous souhaitez signaler des fotes ou otre :

[Direction profil Ash, pour me contacter Conclusion de cet article Vous pouvez le modifier, l'améliorer, le mettre sur votre site, mais n'oubliez pas de citer la source](#)

<http://www.bulma-animation.org>