

[pour cloner un module](#)

Catégorie : Comment faire ...

Publié par [Fooups](#) le 16/04/2005

Ce guide est rédigé à l'intention des utilisateurs qui veulent installer deux fois le même module. Il leur faudra cloner le module original afin de pouvoir l'installer en double. **1) Préliminaire**
Le clonage d'un module n'est pas une opération extrêmement compliquée, mais peut se révéler fastidieuse.

La difficulté sera très variable d'un module à l'autre:

- certains modules seront plus difficiles à cloner, en particulier s'ils ne respectent pas les conventions d'écriture, ne sont pas homogènes dans cette écriture, n'utilisent pas les fonctions de xoops pour l'accès à la bdd, etc. Il n'est pas certain qu'ils valent le coup d'être clonés !
- d'autres modules, bien codés, seront assez faciles à cloner. Mieux vaut disposer d'un bon éditeur permettant de faire les remplacements sur tout un répertoire: le risque d'erreur est plus important, mais en cas de réussite, c'est nettement plus rapide. Faites ce remplacement en passant pas en activant l'option de respect de la casse.

Si vous optez pour cette solution, réfléchissez au contenu et à l'ordre de vos remplacements, afin de ne pas les cumuler (genre 'news' transformé en 'news0202' au lieu de 'news02'). Ce tuto vous explique la marche à suivre en se basant sur le module de news de xoops.

En principe toutes les modifications sont décrites, en omettant celles que l'on pourrait croire nécessaires, et qui ne le sont pas!

Maintenant à vous de l'adapter !

Et attendez d'avoir terminé pour installer et tester votre clone. **2) Le répertoire** Faire un copier/coller du répertoire du module et le renommer en `news02` dans `xoops_version` changer la valeur du 'dirname' :

`$modversion['dirname'] = "news02";` recherchez et remplacez dans **tous** les fichiers les références au répertoire:

remplacez `/modules/news/` par `/modules/news02/`

NB : Pour certains modules utilisant systématiquement `$xoopsModule->getVar('dirname')` cette dernière opération ne sera pas nécessaire. **3) Les tables** Éditer le fichier `mysql.sql` du répertoire sql et renommez les tables pour chaque instruction sql:

`CREATE TABLE stories02 (`

`CREATE TABLE topics02 (`

`INSERT INTO topics02` dans `xoops_version` modifiez les références aux tables:

`$modversion['sqlfile']['mysql'] = "sql/mysql.sql";`

`$modversion['tables'][0] = "stories02";`

`$modversion['tables'][1] = "topics02";` recherchez et remplacez dans **tous** les fichiers les références aux tables.

Si vous faites un remplacement global, faites-le avec prefix('nomdelatable') plutôt que le nom de la table seul; vous risqueriez de changer inutilement des valeurs de variables qui porteraient le même nom.

Dans ce cas, faites la recherche plusieurs fois: avec simples et avec doubles quotes, avec et sans espace:

remplacez `prefix('stories')`, `prefix("stories")`, `prefix('stories')`, ... par `prefix('stories02')`

remplacez `prefix('topics')`, `prefix("topics")`, `prefix('topics')`, ... par `prefix('topics02')`

NB : certains modules ont un fichier `include`

`ead_config.php` (ou Ã©quivalent) dans lequel les noms des tables sont assignÃ©s Ã des variables.

Dans ce cas, seul ce fichier est Ã modifier. **4) Les dÃ©finitions de langage** (`modinfo.php`)

Vous devrez renommer toutes les dÃ©finitions de langage du fichier `modinfo.php`, pour Ã©viter notamment qu'elles soient dÃ©clarÃ©es 2 fois et gÃ©nÃ©rent des Warning en mode debug php.

En principe, les autres fichiers de langues n'ont pas besoin d'Ãatre modifiÃ©s, mais vous pouvez le faire pour des questions de cohÃ©rence.

Pas trÃ©s facile de dÃ©crire une mÃ©thode, tout dÃ©pendra du module.

Si **toutes** les variables sont prÃ©fixÃ©es (ex) `_MI_NEWS_` faites un remplacement global de `_MI_NEWS_` par `_MI_NEWS02_`

Idem si elles sont **toutes** prÃ©fixÃ©es `_MI_` faites un remplacement global de `_MI_` par `_MI_NEWS02_` p.ex.

Si c'est un mixte, attention Ã ne pas vous retrouver avec des `_MI_NEWS02_NEWS02_` p.ex.

NB : si vous ne rÃ©alisez pas entiÃ©rement cette partie, Ã la fin du clonage, installez et lancez le module en mode debug php: corrigez les defines gÃ©nÃ©rant un warning. **5) Fonction recherche**

dans `xoops_version` modifiez le nom de la fonction uniquement

`$modversion['search']['func'] = "news02_search";` Ã©ditez le fichier `includessearch.inc.php` et modifiez le nom de la fonction

`function news02_search($queryarray, $andor, $limit, $offset, $userid)` **6) Blocs** dans `xoops_version` modifiez pour chaque bloc le nom des fonctions show et edit.

`$modversion['blocks'][3]['show_func'] = "b_news02_top_show";`

`$modversion['blocks'][3]['edit_func'] = "b_news02_top_edit";` Ã©ditez chaque fichier du rÃ©pertoire blocks rÃ©fÃ©rencÃ© par `$modversion['blocks']` et modifiez le nom des fonctions.

Ex avec le fichier `news_top.php`

`function b_news02_top_show($options) {`

`function b_news02_top_edit($options) {` Non indispensable, mais prÃ©fÃ©rable, changez

Ã©galement les noms des fichiers dans `xoops_version`, sans oublier de renommer correctement ceux-ci dans le rÃ©pertoire blocks

`$modversion['blocks'][3]['file'] = "news02_block_top.html";`

7) Commentaires dans `xoops_version` modifiez le nom des fonctions approve et update:

`$modversion['comments']['callbackFile'] = 'include/comment_functions.php';`

`$modversion['comments']['callback']['approve'] = 'news02_com_approve';`

`$modversion['comments']['callback']['update'] = 'news02_com_update';` **8) Notification** dans

`xoops_version` modifiez le nom de la fonction look_up:

`$modversion['notification']['lookup_func'] = 'news02_notify_iteminfo';` **9) Les templates** dans

`xoops_version` modifiez le nom de la fonction look_up:

`$modversion['notification']['lookup_func'] = 'news02_notify_iteminfo';`

Non indispensable, mais prÃ©fÃ©rable, changez Ã©galement les noms des fichiers dans `xoops_version`, sans oublier de renommer correctement ceux-ci dans le rÃ©pertoire templates

`$modversion['templates'][1]['file'] = "news02_archive.html";`

10) Le logo Il n'est pas inutile de modifier le logo pour s'y retrouver plus facilement dans l'administration des modules.

Une fois modifiÃ©, pensez Ã le renommer et changer le nom indiquÃ© dans `xoops_version`.

`$modversion['image'] = "images/news02_slogo.png";` **11) Pour finir**

Allez dans l'administration des modules: votre clone doit apparaître sans message d'erreur. Si c'est le cas, inutile d'aller plus loin !

Installez votre clone. Si vous avez des messages d'erreur Ã l'install, notez les pour corriger.

Lancez votre clone. Activez le mode debug php pour corriger les erreurs.

Testez toutes les configurations de blocks et fonctionnalités (recherche, commentaires, notification).

Ces tests doivent être réalisés dans les 2 cas de figure:

- avec la version originale installée, pour détecter des conflits ou interactions
- sans la version originale installée (répertoire supprimé, ou temporairement renommé) pour vérifier que votre clone est autonome, et que certains fichiers ne pointent pas sur ceux d'origine.

12) Problèmes courants

- Vérifiez bien que les données de l'original ne s'affichent pas dans votre clone, ou que l'insertion de données du clone ne se fasse pas dans l'original. Vous auriez dans ce cas oublié de renommer certaines références aux tables.

- Fatal error: cannot redeclare ... Ce cas a été rencontré avec WFSection lors de l'accès à userinfo: l'original et son clone font chacun un 'include' de groupaccess.php, et les fonctions de ce fichier portant le même nom, Php refuse de les redéclarer.

Il est possible, soit de faire pointer le clone sur le fichier groupaccess.php de l'original, soit de renommer toutes les fonctions de ce fichier, ainsi que les appels à ces fonctions. Cette seconde solution est préférable, car elle rend votre clone autonome, et évite un problème en cas de désinstallation et suppression de l'original.