

The cloning of a module - Hacks

NEWS_PDF_AUTHOR: JMorris

NEWS_PDF_DATE: 2004/9/16 14:19:53

This article is duplicated/translated with permission from the author.

Translation performed using Translate v0.5.5 for Mozilla Firefox by Paul Grave.

Author: thecat (Claude)

Author's URL: <http://thecat.ouvaton.org/>

Original Article: [The cloning of a module](#)

Article:

This guide is written for the users who want to install the same module twice. It will be necessary cloner for them the original module in order to be able to install it in double.

1) Préliminaire

The cloning of a module is not an extremely complicated operation, but can appear tiresome. The difficulty will be very variable from one module to another:

- certain modules will be more difficult with cloner, in particular if they do not respect conventions of writing, are not homogeneous in this writing, do not use the functions of xoops for the access to the bdd, etc. It is not certain that they are worth the blow to be clonés!
- other modules, coded well, will be rather easy with cloner.

Better is worth to have a good editor allowing to make the replacements on a whole repertory: the risk of error is more significant, but in the event of success, it is definitely faster. Make this replacement in step by step by activating the option of respect of breakage.

If you choose this solution, think on the contents and the order of your replacements, in order not to cumulate them (kind ' news' transformed into ' news0202 ' instead of ' news02 ').

This tuto explains you the procedure while being based on the module of news of xoops.

In theory all the modifications are described, by ommettant those which one pourrait to believe necessary, and who are not it!

Maintaining with you to adapt it!

And wait to have finished to install and test your clone.

2) the repertory

- * To make a copier/coller repertory of the module and to re-elect it in news02

* in xoops_version to change the value of the 'dirname':

```
$$modversion['dirname'] = "news02";
```

* seek and replace in all the files the references to the repertory:

replace /modules/news/ by /modules/news02/

NB: For certain modules systematically using \$\$xoopsModule->getvar('dirname') this last operation will not be necessary.

3) tables

* to publish the file mysql.sql L of the repertory sql and re-elect the tables for each instruction sql:

```
CREATE COUNTS STORIES02 (
```

```
CREATE COUNTS TOPICS02 (
```

```
INSERT INTO TOPICS02
```

* in xoops_version modify the references to the tables:

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

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

```
$$modversion['tables'][1] = "topics02";
```

* seek and replace in all the files the references to the tables.

If you make a total replacement, do it with prefix('nomdelatable' rather ') than the name of the table alone; you would be likely unnecessarily to change values of variables which would bear the same name.

In this case, made research several times: with simple and doubles quotes, with and without space:

replace prefix(' stories ') , prefix("stories") , prefix(' stories') ... by prefix(' stories02 ')

replace prefix(' topics ') , prefix("topics") , prefix(' topics') ... by prefix(' topics02 ')

NB: certain modules have a file include/read_config.php (or equivalent) in which the names of the tables are assigned with variables.

In this case, only this file is to be modified.

4) definitions of language (modinfo.php)

You will have to re-elect all the definitions of language of the file modinfo.php , to prevent in particular that they are declared 2 times and generate of Warning in mode debug php.

In theory, the other files of languages do not need to be modified, but you can do it for questions of coherence.

Not very easy to describe a method, all will depend on the module.

If all the variables are prefixed (ex) _MI_NEWS_ made a total replacement of _MI_NEWS_ by _MI_NEWS02_

Idem if they all are prefixed _SEMI_ made a total replacement of _SEMI_ by _MI_NEWS02_ p.ex.

If it is mixed one, attention not to find you with _MI_NEWS02_NEWS02_ p.ex.

NB: if you entirely do not carry out this part, at the end of the cloning, install and launch the module in mode debug php: correct the defines generating a warning.

5) Function seeks

- * in xoops_version only modify the noun of the function
`$$modversion['search']['func '] = "news02_search";`
- * publish the file `include\search.inc.php` and modify the noun of the function
`function news02_search($queryarray, $$andor, $$limit, $$offset, $$userid)`

6) Blocs

- * in xoops_version modify for each block the noun of the functions show and edict.
`$$modversion['blocks'][3]['show_func '] = "b_news02_top_show";`
`$$modversion['blocks'][3]['edit_func '] = "b_news02_top_edit";`
- * publish each file of the repertory blocks referred by `$$modversion['blocks']`['file '] = and modify the noun of the functions.
 Ex with the file `news_top.php`
`function b_news02_top_show($options) {`
`function b_news02_top_edit($options) {`
- * Nonessential, but preferable, also change the names of the files in xoops_version, without forgetting to re-elect correctly those in the repertory blocks
`$$modversion['blocks'][3]['file '] = "news02_block_top.html";`

7) Commentaires

- * in xoops_version modify the noun of the functions approve and update:
`$$modversion['comments']['callbackFile '] = ' include/comment_functions.php ';`
`$$modversion['comments']['callback']['approve '] = ' news02_com_approve ';`
`$$modversion['comments']['callback']['update '] = ' news02_com_update ';`

8) Notification

- * in xoops_version modify the noun of the function look_up:
`$$modversion['notification']['lookup_func '] = ' news02_notify_iteminfo ';`

9) templates

- * in xoops_version modify the noun of the function look_up:
`$$modversion['notification']['lookup_func '] = ' news02_notify_iteminfo ';`
- * Nonessential, but preferable, also change the names of the files in xoops_version, without forgetting to re-elect correctly those in the repertory templates
`$$modversion['templates'][1]['file '] = "news02_archive.html";`

10) the logo

* It is not useless to modify the logo to be found there more easily in the administration of the modules.

Once modified, think of re-electing it and of changing the name indicated in xoops_version.

```
$$modversion['image ' ] = "images/news02_slogo.png";
```

11) to finish

Go in the administration of the modules: your clone owes apparaitre without error message. If it is the case, useless to go further!

Install your clone. If you have error messages to the install, note to correct.

Launch your clone. Activate the mode debug php to correct the errors.

Test all the configurations of blocks and functionalities (research, comments, notification).

These tests must be carried out in the 2 cases of figure:

- with the original version installed, to detect conflicts or interactions
- without the original version installed (removed, or temporarily famous repertory) to check that your clone is autonomous, and that certain files do not point on those of origin.

12) current Problems

- Check well that the data of the original are not posted in your clone, or that the insertion of data of the clone is not done in the original. You would have in this case forgotten to re-elect certain references to the tables.

- Fatal error: cannot redeclare... This case was encountered with WFSection at the time of the access to userinfo: the original and its clone do each one a ' include' of groupaccess.php, and the functions of this file bearing the same name, Php refuses of the redéclarer.

It is possible, either to make point the clone on the file groupacces.php original, or to re-elect all the functions of this file, as well as the calls to these functions. This second solution is preferable, because it makes your clone autonomous, and avoids a problem in the event of desinstallation and suppression of the original.

This article is duplicated/translated with permission from the author.

Translation performed using Translate v0.5.5 for Mozilla Firefox by Paul Grave.

Author: thecat (Claude)

Author's URL: <http://thecat.ouvaton.org/>

Original Article: [The cloning of a module](#)

Article:

This guide is written for the users who want to install the same module twice. It will be necessary cloner for them the original module in order to be able to install it in double.

1) Préliminaire

The cloning of a module is not an extremely complicated operation, but can appear tiresome. The difficulty will be very variable from one module to another:

- certain modules will be more difficult with cloner, in particular if they do not respect conventions of writing, are not homogeneous in this writing, do not use the functions of xoops for the access to the bdd, etc. It is not certain that they are worth the blow to be clones!
- other modules, coded well, will be rather easy with cloner.

Better is worth to have a good editor allowing to make the replacements on a whole repertory: the risk of error is more significant, but in the event of success, it is definitely faster. Make this replacement in step by step by activating the option of respect of breakage.

If you choose this solution, think on the contents and the order of your replacements, in order not to cumulate them (kind ' news' transformed into ' news0202 ' instead of ' news02 ').

This tuto explains you the procedure while being based on the module of news of xoops.

In theory all the modifications are described, by ommettant those which one pourrait to believe necessary, and who are not it!

Maintaining with you to adapt it!

And wait to have finished to install and test your clone.

2) the repertory

* To make a copier/coller repertory of the module and to re-elect it in news02

* in xoops_version to change the value of the ' dirname':

```
$$modversion['dirname ' ] = "news02";
```

* seek and replace in all the files the references to the repertory:

replace /modules/news/ by /modules/news02/

NB: For certain modules systematically using \$\$xoopsModule->getvar('dirname ') this last operation will not be necessary.

3) tables

* to publish the file mysql.sql L of the repertory sql and re-elect the tables for each instruction sql:

```
CREATE COUNTS STORIES02 (
CREATE COUNTS TOPICS02 (
INSERT INTO TOPICS02
```

* in xoops_version modify the references to the tables:

```
$$modversion['sqlfile']['mysql ' ] = "sql/mysql.sql";
$$modversion['tables'][0 ] = "stories02";
$$modversion['tables'][1 ] = "topics02";
```

* seek and replace in all the files the references to the tables.

If you make a total replacement, do it with prefix(' nomdelatable rather ') than the name of the table alone; you would be likely unnecessarily to change values of variables which would bear the same name.

In this case, made research several times: with simple and doubles quotes, with and without space:

replace prefix(' stories ') , prefix("stories") , prefix(' stories') ... by prefix(' stories02 ')

replace prefix(' topics ') , prefix("topics") , prefix(' topics') ... by prefix(' topics02 ')

NB: certain modules have a file include\read_config.php (or equivalent) in which the names of the tables are assigned with variables.

In this case, only this file is to be modified.

4) definitions of language (modinfo.php)

You will have to re-elect all the definitions of language of the file modinfo.php , to prevent in particular that they are declared 2 times and generate of Warning in mode debug php.

In theory, the other files of languages do not need to be modified, but you can do it for questions of coherence.

Not very easy to describe a method, all will depend on the module.

If all the variables are prefixed (ex) _ MI_NEWS _ made a total replacement of _ MI_NEWS _ by _ MI_NEWS02 _

Idem if they all are prefixed _ SEMI _ made a total replacement of _ SEMI _ by _ MI_NEWS02 _ p.ex.

If it is mixed one, attention not to find you with _ MI_NEWS02_NEWS02 _ p.ex.

NB: if you entirely do not carry out this part, at the end of the cloning, install and launch the module in mode debug php: correct the defines generating a warning.

5) Function seeks

* in xoops_version only modify the noun of the function

```
$$modversion['search']['func ' ] = "news02_search";
```

* publish the file include\search.inc.php and modify the noun of the function
function news02_search(\$queryarray, \$\$andor, \$\$limit, \$\$offset, \$\$userid)

6) Blocs

* in xoops_version modify for each block the noun of the functions show and edict.
`$$modversion['blocks'][3]['show_func'] = "b_news02_top_show";`
`$$modversion['blocks'][3]['edit_func'] = "b_news02_top_edit";`
 * publish each file of the repertory blocks referred by `$$modversion['blocks']`['file'] = and modify the noun of the functions.

Ex with the file news_top.php
`function b_news02_top_show($options) {`
`function b_news02_top_edit($options) {`
 * Nonessential, but preferable, also change the names of the files in xoops_version, without forgetting to re-elect correctly those in the repertory blocks
`$$modversion['blocks'][3]['file'] = "news02_block_top.html";`

7) Commentaires

* in xoops_version modify the noun of the functions approve and update:
`$$modversion['comments']['callbackFile'] = 'include/comment_functions.php';`
`$$modversion['comments']['callback']['approve'] = 'news02_com_approve';`
`$$modversion['comments']['callback']['update'] = 'news02_com_update';`

8) Notification

* in xoops_version modify the noun of the function look_up:
`$$modversion['notification']['lookup_func'] = 'news02_notify_iteminfo';`

9) templates

* in xoops_version modify the noun of the function look_up:
`$$modversion['notification']['lookup_func'] = 'news02_notify_iteminfo';`
 * Nonessential, but preferable, also change the names of the files in xoops_version, without forgetting to re-elect correctly those in the repertory templates
`$$modversion['templates'][1]['file'] = "news02_archive.html";`

10) the logo

* It is not useless to modify the logo to be found there more easily in the administration of the modules.
 Once modified, think of re-electing it and of changing the name indicated in xoops_version.
`$$modversion['image'] = "images/news02_slogo.png";`

11) to finish

Go in the administration of the modules: your clone owes apparaitre without error message. If it is the case, useless to go further!

Install your clone. If you have error messages to the install, note to correct.

Launch your clone. Activate the mode debug php to correct the errors.

Test all the configurations of blocks and functionalities (research, comments, notification).

These tests must be carried out in the 2 cases of figure:

- with the original version installed, to detect conflicts or interactions
- without the original version installed (removed, or temporarily famous repertory) to check that your clone is autonomous, and that certain files do not point on those of origin.

12) current Problems

- Check well that the data of the original are not posted in your clone, or that the insertion of data of the clone is not done in the original. You would have in this case forgotten to re-elect certain references to the tables.

- Fatal error: cannot redeclare... This case was encountered with WFSection at the time of the access to userinfo: the original and its clone do each one a 'include' of groupaccess.php, and the functions of this file bearing the same name, Php refuses of the redéclarer.

It is possible, either to make point the clone on the file groupacces.php original, or to re-elect all the functions of this file, as well as the calls to these functions. This second solution is preferable, because it makes your clone autonomous, and avoids a problem in the event of desinstallation and suppression of the original.