Content

Table des matières

1 Installation

- Wiki

```
$ wget http://prdownloads.sourceforge.net/xoops/XOOPS2_mod_wiwimod_0.8.1_Xavier.zip
$ unzip XOOPS2_mod_wiwimod_0.8.1_Xavier.zip
$ mv wiwimod/ htdocs/modules/
```

  Pour créer des pages, il faut créer un nouveau lien.

- Editeurs pour le wiki ? ()

```
$ wget http://xoops.org.cn/uploads/mydownloads/xoops/xoops-class-xoopseditor.zip
$ mv tgz/xoops-class-xoopseditor/class/xoopseditor/ htdocs/modules
$ rm -fr tgz/xoops-class-xoopseditor
```

Administration->Système->Module->Mise à jour Wiwi Administration->wiwi->préférences->Xoops éditeurs+TinyMCE

2 Cloner

- Copier le répertoire **htdocs/modules/content** en le renomant

```
$ cp -fr wiwimod _CL1_wiwimod
$ cd _CL1_wiwimod
```

- Modifier le logo.

```
$ gimp images/wiwilogo.gif
```

2.1 Les variables

2.1.1 Théorie

Les définitions de language (*modinfo.php*). Vous devrez renommer toutes les définitions de langage du fichier modinfo.php, pour éviter notamment qu'elles soient déclarées 2 fois et génèrent des Warning en mode debug php. En principe, les autres fichiers de langues n'ont pas besoin d'être modifiés, mais vous pouvez le faire pour des questions de cohérence. Pas très facile de décrire une méthode, tout dépendra du module. Si toutes les variables sont préfixées (ex) `_MI_NEWS_` faites un remplacement global. NB : si vous ne réalisez pas entièrement cette partie, à la fin du clonage, installez et lancez le module en mode debug php : corrigez les defines générant un warning.

2.1.2 Pratique

- On énumère toutes les variables définies.

```
$ grep -R 'define[[:blank:]]*(' * | sed 's/['\''"]/%/g' | cut -d% -f 2 | sort | uniq | less
```

- On les préfixes. (copiez/collez la ligne suivante)

```
PFX='_CL1_' ;\
VAR=$(grep -R 'define[[:blank:]]*(' * | sed 's/['\''"]/%/g' | sed "s/$PFX//g" | cut -d% -f 2 | sort | uniq) ;\
CPT=0; SUM=$(echo $VAR | wc -w) ;\
for VAR in $VAR ; do \
  echo -ne "\r\r\r\r\r\r\r\r\r$[CPT+=1]/$SUM" ;\
  find . -type f -exec sed -i -e "s%\($PFX\)*\($VAR\)%$PFX\2%g" {} \; ;\
done
```

- On Change le nom du module.

```
$ find language -exec grep -n "define('_CL1_MI_WIWIMOD_NAME','.*');" {} /dev/null \;

$ find language -type f -exec sed -i -e \
  "s%\(define('_CL1__MI_WIWIMOD_NAME',\)'.*');%\1'Wiwi clone 1');%g" {} \;
```

2.2 Les tables

2.2.1 Théorie

Editer le fichier mysql.sql du répertoire sql et renommez les tables pour chaque instruction sql.

```
CREATE TABLE ...
```

Dans *xoops_version* modifiez les références aux tables :

```
$modversion['tables'][0] = ...
```

Recherchez et remplacez dans tous les fichiers les références aux tables. Si vous faites un remplacement global, faites-le avec prefix('nomdelatable'). 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') NB : certains modules ont un fichier include *ead_config.php* (ou équivalent) dans lequel les noms des tables sont assignés à des variables. Dans ce cas, seul ce fichier est à modifier.

2.2.2 Pratique

- On prefixe le nom des tables dans le fichier *xoops_version*.

```
$ grep -n "^\$modversion\['tables'\]\[.\] = \".*\";" xoops_version.php
$ sed -i -e "s%\(^\$modversion\['tables'\]\[.\] = \"\)\(.*\";\)%\1_CL1_\2%" xoops_version.php
```

- On prefixe les tables dans les fichiers SQL.

```
$ find sql/* -exec grep -n '\(TABLE \|INTO \|%%\)\(_CL1_\)*wiwimod' {} /dev/null \; | cut -d: -f1 | sort | uniq
$ sed -i -e 's!\(TABLE \|INTO \|%%\)\(wiwimod\)!\1_CL1_\2!g' sql/*
```

- On prefixe les tables dans les fichiers PHP.

```
$ grep -R "\->prefix.*wiwimod" * | wc -l
$ grep -R "\->prefix(['\"]wiwimod" * | wc -l
$ find . -type f -exec sed -i -e 's%\(->prefix(\)\(["'\'']\)\(wiwimod\)%\1\2_CL1_\3%g' {} \;
```

2.3 Fonction recherche

2.3.1 Théorie

Dans *xoops_version* modifiez le nom de la fonction uniquement

```
$modversion['search']['func'] = ...
```

Editez le fichier includesearch.inc.php et modifiez le nom de la fonction

```
function ...($queryarray, $andor, $limit, $offset, $userid)
```

2.3.2 Pratique

```
$ find . -exec grep -n 'wiwimod_search' /dev/null {} \;
$ find . -type f -exec sed -i -e 's!wiwimod_search!_CL1_wiwimod_search!g' {} \;
```

2.4 Les blocs

2.4.1 Théorie

Dans *xoops_version* modifiez pour chaque bloc le nom des fonctions show et edit.

```
$modversion['blocks'][?]['show_func'] = ...
$modversion['blocks'][?]['edit_func'] = ...
```

Editez chaque fichier du répertoire blocks référencé par `$modversion['blocks'][]['file'] =` et modifiez le nom des fonctions.

```
function ..._show($options) {
function ..._edit($options) {
```

Non indispensable, mais préférable, changez également les noms des fichiers dans *xoops_version*, sans oublier de renommer correctement ceux-ci dans le répertoire blocks.

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

2.4.2 Pratique

1. On prefixe le nom des blocks dans le fichier *xoops_version*.

```
$ find . -exec \
  grep -n "$modversion\['blocks'\]\[.*]\['\(show\|edit\)_func'\] = \"\(.*\)\"" \
  /dev/null {} \;

$ sed -i -e "s\
%\$modversion\['blocks'\]\[\(.*\)]\['\(show\|edit\)_func'\] = \"\(.*\)\"\
%\$modversion['blocks'][\1]['\2_func'] = \"_CL1_\3\"\
%" xoops_version.php
```

2. On prefixe les fonctions dans le répertoire *blocks*.

```
$ grep -n 'function wiwimod_' blocks/*
$ find blocks/* -exec sed -i -e 's!\(function \)\(wiwimod_\)!\1_CL1_\2!' {} \;
```

3. On change le nom des fichiers blocks dans *xoops_version*,

```
$ find . -exec grep -n "$modversion\['blocks'\]\[.*]\['file'\] = \"\(.*\)\"" /dev/null {} \;

$ sed -i -e "s\
%\$modversion\['blocks'\]\[\(.*\)]
\['file'\] = \"\(.*\)\"\
%\$modversion['blocks'][\1]['file'] = \"_CL1_\2\"\
%" xoops_version.php

$ cd blocks ;\
  for f in *; do mv $f _CL1_$f; done ;\
  cd ..
```

2.5 Les templates

2.5.1 Théorie

Non indispensable, mais préférable, changez également les noms des fichiers dans *xoops_version*, sans oublier de renommer correctement ceux-ci dans le répertoire templates

```
$modversion['templates'][?]['file'] = ...
```

2.5.2 Pratique

1. On prefixe le nom des templates dans le fichier *xoops_version*.

```
$ find . -exec grep -n "$modversion\['templates'\]\[.*]\['file'\] = '\(.*\)'" /dev/null {} \;

$ sed -i -e "s\
%\$modversion\['templates'\]\[\(.*\)]\['file'\] = '\(.*\)'\
%\$modversion['templates'][\1]['file'] = '_CL1_\2'\
%" xoops_version.php
```

2. On change le nom des fichiers blocks dans *xoops_version*,

```
$ cd templates ;\
  for f in wiwimod*.html; do mv $f _CL1_$f; done ;\
  cd ..
```

3. Idem pour les templates de blocks :

```
$ find . -exec grep -n "$modversion\['blocks'\]\[.*]\['template'\] = '\(.*\)'" /dev/null {} \;

$ sed -i -e "s\
%\$modversion\['blocks'\]\[\(.*\)]\['template'\] = '\(.*\)'\
%\$modversion['blocks'][\1]['template'] = '_CL1_\2'\
%" xoops_version.php

$ cd templates/blocks ;\
  for f in wiwimod*.html; do mv $f _CL1_$f; done ;\
  cd ../..
```

2.6 Répertoire

2.6.1 Théorie

Dans *xoops_version* changer la valeur du 'dirname' :

`$modversion['dirname'] = ...`

Recherchez et remplacez dans tous les fichiers les références au répertoire : remplacez `'/modules/.../`
NB : Pour certains modules utilisant systématiquement `'\$xoopsModule->getVar('dirname')'`
cette dernière opération ne sera pas nécessaire.

2.6.2 Pratique

- On prefixe le nom du répertoire dans le fichier *xoops_version*.

```
$ grep -n "\$modversion\['dirname'\]" xoops_version.php
$ sed -i -e "s%\(\$modversion\['dirname'\] * =\) \".*\"%\1 \"_CL1_wiwimod\"%" xoops_version.php
```

- On prefixe le répertoire partout où il apparaît.

```
$ find . -exec grep -n '/modules/wiwimod/' {} /dev/null \; | cut -d: -f1,2
$ find . -type f -exec sed -i -e 's!/modules/wiwimod!/modules/_CL1_wiwimod/!g' {} \;
```

2.7 Les commentaires

2.7.1 Théorie

Dans *xoops_version* modifiez le nom des fonctions approve et update :

```
$modversion['comments']['callbackFile'] = ...
$modversion['comments']['callback']['approve'] = ...
$modversion['comments']['callback']['update'] = ...
```

2.7.2 Pratique

Rien à faire.

```
$ grep -R callback *
```

2.8 Les notifications

2.8.1 Théorie

Dans *xoops_version* modifiez le nom de la fonction `look_up:`

```
$modversion['notification']['lookup_func'] = ...
```

2.8.2 Pratique

Rien à faire.

```
$ grep -R notification *
```

2.9 Pour finir

2.9.1 Théorie

Allez dans l'administration des modules : votre clone doit apparaitre sans message d'erreur. Si c'est le cas, inutile d'aller plus loin ! Installez votre clone. Si vous avez des messages d'erreur à 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.

2.9.2 Pratique

- TinyMCE (l'éditeur) ne fonctionne pas.
- Voir son compte Il faut aussi cloner la classe 'WiwiProfile' qui est appelée pour chacun des clones lorsqu'on veut afficher son compte.
  - On prefixe le répertoire partout où il apparaît.

    ```
    $ find . -exec grep '[wW]iwiProfile' {} /dev/null \;
    $ find . -type f -exec sed -i -e 's/[wW]iwiProfile/_CL1_WiwiProfile/g' {} \;
    ```

  - On renome le fichier où est défini la classe.

    ```
    $ mv class/wiwiProfile.class.php class/_CL1_WiwiProfile.class.php
    ```

3 Cloner à nouveau

A priori c'est beaucoup plus simple :

- Changer le logo

```
$ cp -fr _CL1_wiwimod _CL2_wiwimod
$ cd _CL2_wiwimod
$ gimp images/wiwilogo.gif
```

- Changer le nom des variable, procédures, tables :

```
$ find . -exec grep -n "_CL1_" {} /dev/null \; | cut -d: -f 1,2 | wc -l
$ find . -type f -exec sed -i -e 's/_CL1_/_CL2_/g' {} \;
```

- Changer le nom des fichiers :

```
$ find . -type f -name '_CL1_*'

$ find . -type f -name '_CL1_*' -exec sh -c \
  'f=$(echo $1 | sed -e 's/_CL1_/_CL2_/'); mv $1 $f' \
  {} {} \;
```

- On Change le nom du module.

```
$ find language -exec grep -n "define('_CL2__MI_WIWIMOD_NAME','.*');" {} /dev/null \;

$ find language -type f -exec sed -i -e \
  "s%\(define('_CL2__MI_WIWIMOD_NAME',\)'.*');%\1'Wiwi clone 2');%g" {} \;
```