

# **Free XOOPS Services Documentation Series**



Free XOOPS Services  
**Modules - Themes - Hacks**

**Version 1.0**

**© 2011 Free XOOPS Services**

[www.freexoopsservices.com](http://www.freexoopsservices.com)

## Table des matières

<a href="#">1 Préambule</a>	3
<a href="#">2 Installation</a>	3
<a href="#">3 Utilisation</a>	3
<a href="#">3.1 Configuration de xoops_version.php</a>	3
<a href="#">3.2 Configuration pour l'utilisation du Frameworks</a>	6
<a href="#">3.3 Le menu d'administration</a>	7
<a href="#">3.4 La section d'aide du module</a>	8
<a href="#">3.5 Ajout d'une info box</a>	9
<a href="#">3.6 Ajout d'une navigation</a>	11
<a href="#">3.7 Ajout d'un bouton</a>	11
<a href="#">3.8 La page "index.php"</a>	12
<a href="#">3.9 La page "about.php"</a>	14

# 1 Préambule

Depuis Xoops 2.5, le module système est capable de créer automatiquement un menu d'administration pour n'importe quel module. Il suffit simplement de mettre la commande

```
$modversion['system_menu'] = 1;
```

dans le fichier xoops\_version.php. Le module système utilisera le fichier menu.php qui se trouve dans le dossier "admin" pour faire le menu.

Dans le cadre du "bluemove", l'équipe de développement de xoops a décidé d'uniformiser l'administration de tous les modules compatibles avec xoops 2.5. Afin de faciliter ce travail, nous avons créé un Framework nommé "moduleadmin". Ce Framework permet de gérer la partie visuelle de l'administration de n'importe quel module.

# 2 Installation

Pour installer le Framework "moduleadmin" il faut:

1. Télécharger le Framework sur [FreeXoopService](#).
2. Mettre le Framework dans le dossier "Frameworks/moduleclasses" du site.

```
sitefolder
├── Framework
│   ├── moduleclasses
│   │   └── moduleadmin
```

# 3 Utilisation

## 3.1 Configuration de xoops\_version.php

Pour pouvoir utiliser le Framework, il faut ajouter les lignes suivantes dans xoops\_version.php :

Pour lister les auteurs :

```
$modversion['author'] = 'Author1,Author2,Author3' ;
```

Vous pouvez mettre autant d'auteurs du module que vous le voulez, mais il faut les séparer avec une virgule.

```
$modversion['nickname'] = 'Pseudo1,Pseudo1,Pseudo3';
```

Vous devez mettre le même nombre de pseudos que d'auteurs *Pseudo1 = Auteur1*, etc. Les pseudos sont également séparés avec une virgule.

```
$modversion['module_website_url'] = 'www.website.com';  
$modversion['module_website_name'] = 'website';
```

Pour la variable *module\_website\_url*, vous devez mettre l'url de votre site et pour la variable *module\_website\_name*, il faut mettre le nom de votre site

```
$modversion['license'] = 'GNU GPL';  
$modversion['license_url'] = 'www.gnu.org/licenses/gpl-2.0.html/';
```

Pour la variable *license*, vous devez mettre le type de license et pour la variable *license\_url*, il faut mettre l'url de la license.

```
$modversion['module_status'] = 'statut';
```

Vous devez indiquer le statut de votre module. Les statuts sont les suivants :

- Alpha : Les versions Alpha sont les premières versions d'un module, il reste encore beaucoup de bugs et il manque encore des fonctionnalités. Il ne faut pas mettre ces versions sur un site en production.
- Beta : Les versions Beta sont plus avancées que les Alpha. Des bugs existent encore, mais quasiment toutes les fonctionnalités du module sont présentes. Il ne faut pas mettre ces versions sur un site en production.
- RC : Qui veut dire "release candidate", les versions RC sont suffisamment avancées pour qu'elles soit distribuées à n'importe quel utilisateur. Il est déconseillé de les mettre sur un site en production.
- Final : C'est la version finale du module. Cette version peut être installée sur un site en production.

Toutes les versions, exceptée la finale, peuvent être suivies par un chiffre afin de différencier plusieurs versions.

```
$modversion['release_date'] = 'year/month/day';
```

Date de parution de la version du module. Cette date doit avoir le format suivant : **année/mois/jour**

Le Framework s'occupera automatiquement de traduire la date dans le format adéquat par rapport à la langue du site.

```
$modversion['help'] = 'page=help';
```

Permet de spécifier le template d'aide du module et de créer le menu d'aide.

```
$modversion['min_php'] = '5.2';
```

Vous devez spécifier la version minimum de php avec laquelle votre module fonctionne. Dans tous les cas, vous devez mettre au minimum 5.2, car le Framework ne fonctionne pas sous une version antérieure.

```
$modversion['min_xoops'] = '2.5';
```

Vous devez spécifier la version minimum de xoops avec laquelle votre module fonctionne. Dans tous les cas, vous devez mettre au minimum 2.5, car le Framework ne fonctionne pas sous une version antérieure.

```
$modversion['system_menu'] = 1;
```

Depuis la version 2.5 de xoops, le module système génère automatiquement le menu d'administration si on met cette ligne.

**Tous ces paramètres sont obligatoires pour utiliser le Framework.**

### 3.2 Configuration pour l'utilisation du Frameworks

Pour utiliser de manière adéquate le Framework, il faut créer une fonction qui permette de vérifier s'il est présent sur le site. Cette précaution évitera une page blanche à l'utilisateur si le Framework n'est pas présent. Cette fonction permet également d'inclure le fichier "moduleadmin.php".

```
function modulename_checkModuleAdmin()
{
    if ( file_exists($GLOBALS['xoops']-
>path('/Frameworks/moduleclasses/moduleadmin/moduleadmin.php'))){
        include_once $GLOBALS['xoops']-
>path('/Frameworks/moduleclasses/moduleadmin/moduleadmin.php');
        return true;
    }else{
        echo xoops_error("Error: You don't use the Frameworks \"adminmodule\".
Please install this Frameworks");
        return false;
    }
}
```

*modulename* doit être remplacé par le nom du module.

Cette fonction est placée dans le fichier functions.php:

```
modulename
└ include
  └ functions.php.
```

Cette fonction est utilisée de la manière suivante : à chaque fois que l'on veut utiliser le framework:

```
if (modulename_checkModuleAdmin()){
    $variable_name = new ModuleAdmin();
}
```

**Il n'est pas obligatoire d'utiliser cette fonction. Si vous êtes certain que le framework est utilisé, il suffit juste de mettre cette ligne pour l'utiliser :**

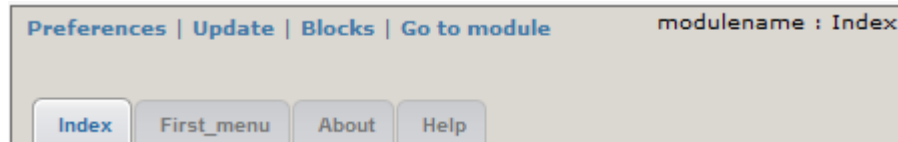
```
include_once $GLOBALS['xoops']-
>path('/Frameworks/moduleclasses/moduleadmin/moduleadmin.php');
```

### 3.3 Le menu d'administration

Pour créer le menu d'administration, nous avons mis dans *xoops\_version.php* :

```
$modversion['system_menu'] = 1;
```

Cette instruction permet d'activer automatiquement un menu de navigation dans l'administration du module :



La partie haute est générée automatiquement et les onglets sont créés par le fichier *menu.php* qui se trouve dans :

```
modulename
├─ admin
├─ menu.php.
```

Le fichier *menu.php* doit avoir la structure suivante :

```
$adminmenu[1]['title'] = _MI_MODULENAME_INDEX;
$adminmenu[1]['link'] = "admin/index.php";
$adminmenu[1]['icon'] = "images/admin/home.png";
$adminmenu[2]['title'] = _MI_MODULENAME_FIRSTMENU;
$adminmenu[2]['link'] = "admin/firstmenu.php";
$adminmenu[2]['icon'] = "images/admin/firstmenu.png";
$adminmenu[3]['title'] = _MI_MODULENAME_ABOUT;
$adminmenu[3]['link'] = "admin/about.php";
$adminmenu[3]['icon'] = "images/admin/about.png";
```

```
$adminmenu[id]['title']
```

Variable dans laquelle vous indiquez le titre du lien.

```
$adminmenu[id]['link']
```

Variable dans laquelle vous indiquez le chemin du lien

```
$adminmenu[id]['icon']
```

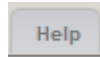
Variable dans laquelle vous indiquez le chemin de l'icône. Cette icône est utilisée par le thème de l'administration ainsi que par le menu d'index généré par le Framework. Les icônes pour la page "index" et "à propos" doivent toujours être les mêmes, pour les autres icônes vous êtes libres. Toutes les icônes doivent avoir une taille 32px / 32px.

### 3.4 La section d'aide du module

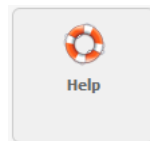
Pour créer une section d'aide, nous avons mis dans xoops\_version.php :

```
$modversion['help'] = 'page=help';
```

Cette instruction permet d'activer automatiquement un menu d'aide dans les onglets de navigation de l'administration du module :



Elle permet aussi de créer un menu dans la page d'index de l'administration :



Le template d'aide se trouve dans le dossier de langue afin de pouvoir le traduire pour chaque langue. La structure est la suivante :

```
modulename
├─ language
│   └─ english
│       └─ help
│           └─ help.html
```

Veuillez regarder le template d'aide du module de TDMDownloads pour voir comment il est construit.



### 3.5 Ajout d'une info box

Une info box est une zone de texte qui est composée d'un titre et d'un texte.

Pour créer une info box, on utilise `addInfoBox()` :

```
$variable_name->addInfoBox($title);
```

\$title : Titre de l'info box.

Pour ajouter une ligne dans l'info box , on utilise `addInfoBoxLine()` :

```
$variable_name->addInfoBoxLine($title, $text, $value, $color, $type);
```

\$title : Titre de l'info box (identique à celui utiliser avec "addInfoBox").

\$text : Texte à faire apparaître dans l'info box.

\$value : Valeur à faire apparaître dans le texte.

\$color : Couleur de la valeur.

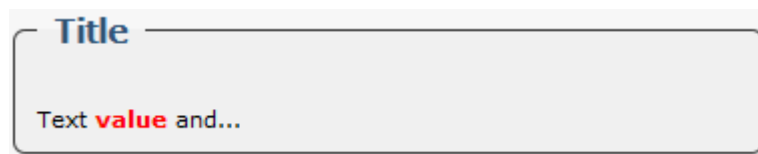
\$type : Valeurs: 'default' ou 'information'.

Le type 'default' utilise tous les paramètres. Il affiche un texte qui comprend la variable `$value` de la couleur 'color'.

Exemple :

```
$variable_name->addInfoBox("Title");
$variable_name->addInfoBoxLine("Title", 'Text %s and...', 'value', 'red', 'default');
```

On obtient :



Le type 'information' utilise juste le paramètre '\$text'. Il affiche donc juste un texte.

Exemple :

```
$variable_name->addInfoBox("Title");
$variable_name->addInfoBoxLine("Title", 'Text', "", "", 'information');
```

On obtient :



Pour que les infos box s'affichent, on utilise *renderInfoBox()* :

```
$variable_name -> renderInfoBox();
```

Les infos box sont automatiquement affichés dans les pages "index.php" et "about.php" si vous utilisez les fonctions qui gèrent ces parties.

On peut ajouter autant de lignes que l'on veut dans une info box et on peut mettre autant d'info box que l'on désire. Ce sont les titres des infos box qui permettent de lier les bonnes lignes avec les bonnes info box.

Exemple :

```
$variable_name = new ModuleAdmin();  
$variable_name->addInfoBox("bike");  
$variable_name->addInfoBox("car");  
$variable_name->addInfoBoxLine("bike", 'Text1', "", "", 'information');  
$variable_name->addInfoBoxLine("bike", 'Text2', "", "", 'information');  
$variable_name->addInfoBoxLine("car", 'Text3', "", "", 'information');  
echo $variable_name -> renderInfoBox();
```

Résultat :



### 3.6 Ajout d'une navigation

La navigation permet de mettre une image avec le titre de la section dans laquelle on se trouve. L'image et le titre sont automatiquement créés.

Pour ajouter la navigation, on utilise *addNavigation()* :

```
$variable_name->addNavigation($file);
```

\$file : Nom de la page avec l'extension (Attention, mettre les mêmes noms que ceux qui sont dans menu.php).

Exemple :

```
$variable_name = new ModuleAdmin();  
$variable_name->addNavigation("fist_menu.php");  
echo $variable_name -> renderLabel();
```

Résultat :



### 3.7 Ajout d'un bouton

Le bouton est composé d'une image et d'un titre, il permet de faire un lien.

Pour créer un bouton, on utilise *addItemButton()* :

```
$variable_name->addItemButton($title, $link, $icon, $extra);
```

\$title : Titre du lien.

\$link : Url du lien.

\$icon : Image qui se trouve avant le lien. Valeurs: 'add' ou 'list'.

\$extra : Texte après le titre.

Pour afficher les boutons, on utilise *renderButton()* :

```
$variable_name -> renderButton($position, $delimiter);
```

\$position : Position des boutons. Valeurs: 'left', 'center' ou 'right'.

\$delimiter : Délimitation entre les boutons. Par défaut ' '.

On peut ajouter autant de boutons que l'on désire.

Exemple:

```
$variable_name = new ModuleAdmin();
$variable_name->addItemButton('Link title', 'index.php', 'add' , '');
$variable_name->addItemButton('Link title', 'index.php', 'add' , 'extra');
$variable_name->addItemButton('Link title', 'index.php', 'list' , '');
echo $variable_name->renderButton('left', '');
```

Résultat:



### 3.8 La page "index.php"

La page d'index est composée de 3 parties :

1. Un menu (généré automatiquement).
2. Des infos box personnalisées si on le désire.
3. Une info box de configuration.

Pour créer l'index, on utilise *renderIndex()* :

```
$variable_name->renderIndex();
```

Par défaut il y a juste 2 configurations qui sont listées :

1. Version php (reprise de la variable \$modversion['min\_php'] qui est dans xoops\_version.php).
2. Version de xoops (reprise de la variable \$modversion['min\_xoops'] qui est dans xoops\_version.php).

Si les deux variables sont définies dans xoops\_version.php, on obtient automatiquement cette info box



On peut ajouter 3 autres types de configuration.

1. Afficher un texte.
2. Vérification de l'existence d'un dossier.
3. Vérification du chmod d'un dossier

Pour ajouter une ligne de configuration, on utilise *addConfigBoxLine()* :

```
$variable_name->addConfigBoxLine($value, $type);
```

\$value : Avec le type 'folder' il faut mettre le chemin du dossier. Avec le type 'chmod' il faut mettre un tableau qui a comme premier paramètre le chemin du dossier et en deuxième le chmod voulu.

\$type : 'default', 'folder' ou 'chmod'.


Exemple :


```
$variable_name = new ModuleAdmin();
$variable_name->addInfoBox("Title");
$variable_name->addInfoBoxLine("Title", 'Text %s ...', 'value', 'red', 'default');
$variable_name->addConfigBoxLine('----Hello world----', 'default');
$folder = XOOPS_ROOT_PATH . '/uploads/modulename/';
$variable_name->addConfigBoxLine($folder, 'folder');
$variable_name->addConfigBoxLine(array($folder, '777'), 'chmod');
echo $variable_name->addNavigation('index.php');
echo $variable_name->renderIndex();
```

Résultat :


[Preferences](#) | [Update](#) | [Blocks](#) | [Go to module](#)
modulename : Index

[Index](#)
[First\\_menu](#)
[About](#)
[Help](#)



**Index**



First\_menu



About



Help

**Title**

Text **value** ...

**Configurations**

- ✓ You must have at least 5.2 as a php version (your current version is 5.3.0)
- ✓ You must have at least the version 2.5 (your current version is 2.5.1)
- Hello world----
- ✓ The folder 'C:/wamp/www/Modules/2.5.1/uploads/modulename/' exists
- ✓ The folder 'C:/wamp/www/Modules/2.5.1/uploads/modulename/' must be with a chmod '777' (it's now set on 777).

### 3.9 La page "about.php"

La page "A popos" est composée de 4 parties :

1. Un résumé des informations du module(généré automatiquement depuis xoops\_version-php) sous forme d'une info box.
2. Des infos box personnalisées si on le désire.
3. Une info box avec des informations concernant le module(généré automatiquement depuis xoops\_version-php).
4. Le changelog du module.

Pour créer la page " A propos" on utilise *renderabout()* :

```
$variable_name->renderabout($paypal, $icon);
```

\$paypal: Votre numéro paypal si vous voulez un bouton "DONATE" pour que les utilisateurs puissent faire des dons.

\$icon: Valeurs 'True' si vous voulez mettre l'image xoops en pied de page et 'False' si vous ne la voulez pas.

Le changelog sera automatiquement chargé mais vous devez le mettre dans :

```
modulename
├ docs
└ changelog.txt
```

Et dans :

```
modulename
├ language
├ Langue
└ changelog.txt
```

Le changelog sera donc automatiquement chargé dans la langue du site. S'il n'existe pas, il sera chargé depuis le dossier "docs".

Vous pouvez aussi ajouter autant d'infos box que vous le désirez dans cette section.

Exemple

```
$variable_name = new ModuleAdmin();
$variable_name->addInfoBox("Title");
$variable_name->addInfoBoxLine("Title", 'Text %s ...', 'value', 'red', 'default');
echo $variable_name->addNavigation('about.php');
echo $variable_name->renderabout('9MYQB7GUK5MCS', true);
```

Résultats: