

Prise en main d' OPENMAIRIE à l'aide de l'exemple openPresse 20/03/2006

version: 1.00: Laurent POUCHOULOU pouchoulolaurent@free.fr

Ce document a pour objet la PRISE EN MAIN du composant openMairie à l'aide d'un exemple d'application « openPresse »

Ce document s'adresse aux développeurs php ayant les connaissances suivantes

- notion UML
- notion de programmation objet
- connaissances SQL

La structure de ce document est la suivante:

1. Description UML du projet openPresse
2. Mise en place d'écrans d'affichage
3. Formulaire de saisie journal
4. Formulaire article
5. Affichage revue de presse
6. Description des menus et des droits

1. Description UML du projet openPresse

Il est décrit ici une analyse UML sommaire de la mise en oeuvre de la revue de presse. L'objectif est de présenter le passage de cette analyse UML à l'implémentation sur PHP avec le composant openMairie

1.1 Objet de l'application

openPresse doit permettre à une organisation de constituer une revue de presse consultable en ligne par intranet

1.2 Ressources consommées et acteurs

Ressources consommées:

- journaux locaux et nationaux

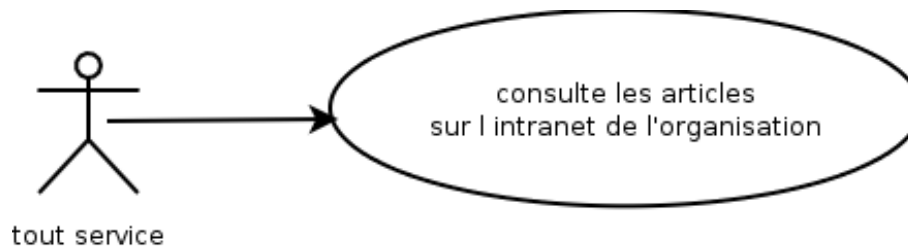
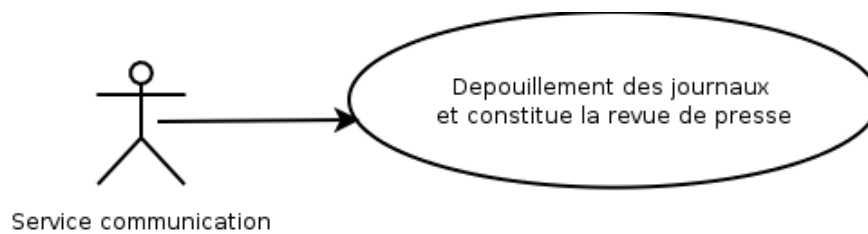
Les acteurs internes:

- le service de communication
- tout service
- administrateur

1.3 Cas d'utilisation

Ces diagrammes sont des diagrammes statiques qui étudient le lien entre les acteurs (internes) et les actions:

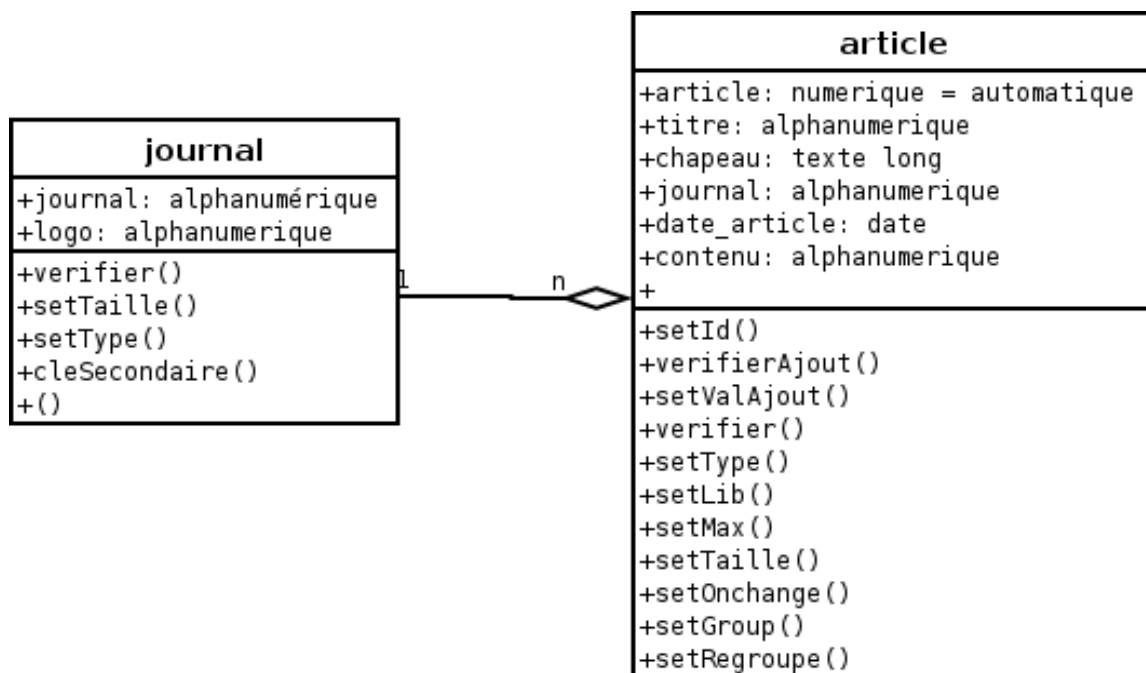
(Le cas de l'administrateur qui attribue les droits correspond à un cas de paramétrage et n'est pas traité dans ces diagrammes)



1.4 Diagramme de classe

Nous définissons 2 objets métier:

- article
- journal



2. Mise en place des écrans d'affichage des objets « métier »

Pour mettre en place l'affichage en table des objets métier, il faut utiliser:

- scr/tab.php : interface PHP en basculant la variable obj
 - exemple : tab.php?obj=journal
- tab.php appelle la classe générique tab.class.php qui est dans openMairie
 - explications : <http://www.openmairie.org/folder.2005-11-05.5614546554/folder.2005-11-05.7277343956/folder.2005-11-05.2965896480/>
- le paramétrage de l'objet se fait suivant la base de données dans un fichier sql/nom_de_base/nom_d_objet.inc
 - exemple : sql/mysql/journal.inc

2.1 affichage de la table journal

Description de sql/mysql/journal.inc

```
<?php
$DEBUG=0;
$serie=10;
$ent = "Parametrage -> Journal";
$ico = "../img/ico_parametrage.png";
$table="journal";
$methode="afficherImage";
$edition="";
//
$champAffiche=array("journal","logo");
$champRecherche=array("journal");
$tri= "";
$selection="";
?>
```

Variables	Description
\$DEBUG	Permet de débbugger l'application. Valeur 0 ou 1
\$serie	Contient le nombre de journaux que l on souhaite afficher dans la page.
\$ent	Entête.
\$ico	C'est le chemin de l'image affichée en haut a droite.
\$table	C'est la table sur laquelle on va faire la requête.
\$methode	Permet d'insérer la méthode d affichage existant dans tab.class.php afficherImage() dans notre cas par défaut : afficher()
\$edition	Permet de faire un lien sur édition pdf.
\$champAffiche	Correspond aux champs de la table que l'on va afficher.
\$champRecherche	Permet de mettre les champs sur lequel on souhaite effectuer une recherche. [recherche unique sur plusieurs champs].
\$tri	Permet de trier la table.
\$selection	Permet la sélection de champs.

2.2 affichage de la table article

Description de sql/mysql/article.inc

```
<?php
$DEBUG=0;
$serie=15;
$ent = "Article -> Saisie";
$ico = "../img/ico_presse.png";
$edition="";
$table="article left join journal on article.journal=journal.journal";
$champAffiche=array("article",
                    "titre",
                    "concat(substring(date_article,9,2),'/',substring(date_article,6,2),'/',substring(date_article,1,4)) as date",
                    "journal.logo");
$champRecherche=array("titre");
$tri= "";
$selection="";
?>
```

Il est important de signaler que le code les requêtes d'affichage sont externalisées dans des fichiers spécifiés pour chaque base de données, permet la compatibilité entre base de données qui n'est pas effectuée totalement par rapport au connecteur dbpear:

Pour l'affichage de la date_article, la concaténation et la troncature est différente:

avec mysql

```
"concat(substring(date_article,9,2),'/',substring(date_article,6,2),'/',substring(date_article,1,4)) as date",
```

avec postgre

```
"(substring(date_article,9,2)||'/'||substring(date_article,6,2)||'/'||substring(date_article,1,4)) as date",
```

La jointure dans \$table "article left join journal on article.journal=journal.journal"; permet de pouvoir afficher

3. Mise en place du formulaire de mise à jour de journal

Pour faire un formulaire

- il faut construire l'objet avec une requête
 - exemple : la requête sql/mysql/journal.form.inc
- il faut décrire un objet métier
 - exemple : obj/journal.class.php

Les objets métiers surchargent les méthodes de la classe générique dbform.class.php (openMairie).

Les informations sur dbform.class.php sont à l'adresse suivante:

<http://www.openmairie.org/folder.2005-11-05.5614546554/folder.2005-11-05.7277343956/folder.2005-11-05.3998173735/>

3.1 Le formulaire « journal »

- la requête sql/mysql/journal.form.inc

```
<?php
$DEBUG=0;
$tableSelect="journal";
$champs=array("journal","logo");
$selection="";
?>
```

\$tableSelect est la table de selection.

\$champ est un tableau qui contient les champs qui seront les propriétés de l'objet journal lors de la construction de l'objet.

– l'objet métier « journal » : obj/journal.class.php

journal.class.php hérite de dbform qui permet l'interface avec la base de données et formulaire.class.php qui permet l'affichage des composants du formulaire

```
require_once ("formulaire.class.php");
require_once ("dbform.class.php");
```

```
class journal extends dbForm {
```

Il est définie 3 variables globales:

- table
- clé primaire (nom de la clé)
- type de clé (N)umérique ou (A)lphabétique

```
var $table="journal";
var $clePrimaire= "journal";
var $typeCle= "A" ;
```

Le constructeur de dbform est utilisé pour construire une instance de l'objet.

```
function journal($id,$db,$debug) {
$this->constructeur($id,$db,$debug);}
```

Par défaut, l'écran suivant s'affiche:

les contrôles sont des champs « text » de longueur et maximum saisie égal à la longueur du champ dans la base de données, le libelle est le nom du champ

Il s'agit de le personnaliser et de définir les méthodes métier de cet objet.

3.2 La personnalisation du formulaire « journal »

La surcharge des méthodes de dbform relative au formulaire permet de personnaliser le formulaire :

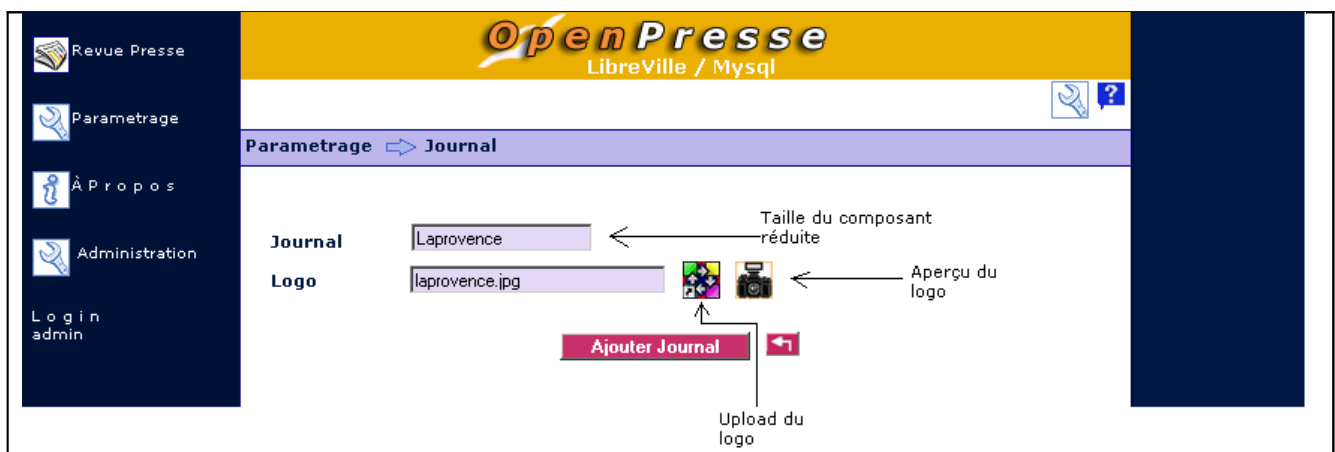
La fonction setTaille() permet de changer la taille des composants de la page. Ici on change la taille du champ journal : il était à 30 dans la base et maintenant il sera à 20.

```
function setTaille(&$form,$maj){
$form->setTaille('journal', 20);
}
```

La fonction setType() permet de mettre le type que l'on souhaite aux champs de la page. Ainsi on aura, lorsqu'on sera en mode « ajouter » ou « modifier », le champ logo qui sera de type « upload » (avec la recherche). En ce qui concerne le champ journal, lorsqu'on sera en mode « modifier » ou « supprimer », on verra le journal sur lequel on effectue l'opération mais on ne pourra pas changer son nom (« hiddenstatic »).

```
function setType(&$form,$maj) {
if ($maj < 2) { //ajouter et modifier
$form->setType('logo', 'upload');
if ($maj==1){ //modifier
$form->setType('journal', 'hiddenstatic');
}
}else{ // supprimer
$form->setType('journal', 'hiddenstatic');
}}
```

Ecran personnalisé:



3.3 Les règles métiers de l'objet « journal »

La surcharge des méthodes de dbform relative à la base de données permet de prendre en charge les règles métiers de journal

La méthode verifier() permet la vérification des données entrées. On peut rendre alors des champs obligatoires comme ici le remplissage du logo. S'il manque une entrée ou si une entrée est incorrecte, la fonction renvoie « false » et ne permet pas l'enregistrement. Si tout est correct, alors la fonction renvoie « true » et l'enregistrement peut avoir lieu.

```
function verifier(){
$this->correct=True;
if ($this->valF['logo']=="){
$this->correct=false;
$this->msg= $this->msg."<br>logo obligatoire";
}
}
```

La méthode `cleSecondaire()` permet de vérifier l'intégrité référentielle. Si l'on souhaite supprimer un journal, cette fonction va vérifier qu'il n'existe pas d'article qui correspond à ce journal. S'il en trouve un, il ne peut pas y avoir de suppression, il faudra d'abord supprimer les articles correspondants à ce journal.

```
function cleSecondaire($id,&$db,$val,$DEBUG) {
    $this->correct=True;
    $sql = "select * from article where journal = '".$id."'";
    $res = $db->query($sql);
    if($debug==1) echo $sql;
    if (DB :: isError($res))
        die($res->getMessage(). " => Echec ". $sql);
    else{
        $nbligne=$res->numrows();
        if ($nbligne>0){
            $this->msg = $this->msg."<br> il y a ".$nbligne.
            " enregistrement(s) en lien dans la table article pour le journal [" .
            $id."] <br> VEUILLEZ SUPPRIMER LES ARTICLES DE CE JOURNAL<BR>";
            $this->correct=false;
        }
    }
}
```

4 la mise en place du formulaire de mise à jour « article »

4.1 Le formulaire « article »

– la requête `sql/mysql/article.form.inc`

```
<?php
$DEBUG=0;
$tableSelect="article";
$champs=array("article",
    "titre",
    "chapeau",
    "journal",
    "date_article",
    "contenu"
);
$selection="";
$sql_journal = "select journal, concat(journal,' ',logo) as lib from journal";
?>
```

`$sql_journal` contient la requête permettant l'affichage des journaux dans un champ « select »

– l'objet métier « article » : `obj/article.class.php`

`article.class.php` hérite de `dbform` qui permet l'interface avec la base de données et `formulaire.class.php` qui permet l'affichage des composants du formulaire

```
require_once ("formulaire.class.php");
require_once ("dbform.class.php");
```

```
class article extends dbForm {
```

```
    var $table="article";
    var $clePrimaire= "article";
    var $typeCle= "N" ;
```

```
function article($id,$db,$debug) {
    $this->constructeur($id,$db,$debug);
}
```

Par défaut, l'écran suivant s'affiche:

les contrôles sont des champs « text » de longueur et maximum saisie égal à la longueur du champ dans la base de données, le libelle est le nom du champ

Il s'agit de personnaliser le formulaire.

4.2 La personnalisation du formulaire « article »

La surcharge des méthodes de dbform relative au formulaire permet de personnaliser le formulaire

stValF est la méthode qui permet le passage des valeurs du formulaire (\$val) dans la base (\$valF).

Il est précisé la transformation de la date si date_article n est pas vide en format de la base de donnée: 10/12/2005 devient 2005-12-10 (avec mysql)

```
if($val['date_article']!="")
    $this->valF['date_article'] = $this->dateDB($val['date_article']);
```

```
function setvalF($val){
    $this->valF['article'] = $val['article'];
    $this->valF['titre'] = $val['titre'];
    $this->valF['chapeau'] = $val['chapeau'];
    $this->valF['journal'] = $val['journal'];
    if($val['date_article']!="")
        $this->valF['date_article'] = $this->dateDB($val['date_article']);
    $this->valF['contenu'] = $val['contenu'];
}
```

Cette méthode permet la vérification des données.

Elle permet de vérifier que le champ titre est saisi

```
function verifier(){
    $this->correct=True;
    // obligatoire
    if ($this->valF['titre']==""){
        $this->correct=false;
        $this->msg= $this->msg."<br>titre obligatoire";
    }
}
```

Cette méthode permet de changer le type du contrôle:

- contenu est transformé en « upload » avec la recherche,
- article en « hidden » c'est-à-dire en contrôle caché,
- date est du type « date » c'est-à-dire zone date avec le calendrier,
- chapeau en type « textarea » donc en champ texte,
- journal en type « select » qui est un contrôle liste déroulante.

```
function setType(&$form,$maj) {
    if ($maj < 2) { //ajouter et modifier
        $form->setType('contenu', 'upload');
```



```

$form->setType('article', 'hidden');
$form->setType('date_article', 'date');
$form->setType('chapeau', 'textarea');
$form->setType('journal', 'select');
if ($maj==1){ //modifier
    $form->setType('article', 'hiddenstatic');
}
}else{ // supprimer
    $form->setType('article', 'hiddenstatic');
}}

```

La méthode `setSelect()` permet de passer des paramètres pour une combo, un select, etc.... Ici on passe comme paramètres les noms des journaux et on affiche le nom et le logo, en exécutant la requête `$sql_journal` défini en `sql/mysql/article.form.inc`

```

function setSelect(&$form, $maj,$db,$debug) {
include ("../dyn/connexion.php");
include ("../sql/".$dsn['phptype']."/".$this->table.".form.inc");
if($maj<2){
$res = $db->query($sql_journal);
if (DB :: isError($res))
    die($res->getMessage().$sql_journal);
}else{
if ($debug == 1)
    echo " la requete ".$sql_journal." est exécutée<br>";
$contentu[0][0]="";
$contentu[1][0]=" Choisir un journal ";
$k=1;
while ($row=& $res->fetchRow()){
    $contentu[0][$k]=$row[0];
    $contentu[1][$k]=$row[1];
    $k++;
}
$form->setSelect("journal",$contentu);
}}

```

Cette méthode permet de fixer la taille du contrôle. Le chapeau sera alors de 70

```

function setTaille(&$form,$maj){
$form->setTaille('chapeau', 70);
}

```

.

La méthode `setMax()` permet de fixer la longueur de saisie. Dans le cas particulier d'un textarea, `setMax` définit le nombre de lignes

```

function setMax(&$form,$maj){
$form->setMax('chapeau', 7);
}

```

La méthode `setLib()` permet de changer le libellé du champ. Ici le champ « `date_article` » devient « `Date de l'article` ».

```

function setLib(&$form,$maj) {
$form->setLib('date_article','Date de l'article');
}

```

La méthode `setGroupe()` permet de grouper des contrôles sur une même ligne. Ici on groupe le champ de la date et le contenu.

```

function setGroupe(&$form,$maj){
$form->setGroupe('date_article','D');    => D = debut
$form->setGroupe('contenu','F');          => F = Fin

```

```
}
```

La méthode `setRegroupe()` permet de regrouper des contrôles dans un fieldset dans un sous tableau. Ici on regroupe les champs `journal`, la date et le contenu dans un même fieldset.

```
function setRegroupe(&$form,$maj){  
  
$form->setRegroupe('journal','D',' Le journal '); => D =debut / 'Le journal' = titre du fieldset  
$form->setRegroupe('date_article','G','');      => G = groupé  
$form->setRegroupe('contenu','F','');           => F = Fin  
}
```

4.3 Les règles métiers de l'objet « journal »

La méthode `setId()` permet la numérotation automatique d'après une séquence. Dans le cas de `mysql`, la numérotation automatique de la clé primaire se fait dans la table `article_seq`.

Avec `postgre` ou `oracle`, les séquences sont utilisés

`$db->nextId` est une méthode de `dbpear` qui assure la portabilité entre différentes bases.

```
function setValFAjout($val){  
}  
function setId($db){  
$this->valF['article'] = $db->nextId($this->table);  
}  
function verifierAjout(){  
}
```

La méthode `setOnChange()` permet le traitement javascript lorsqu'une mise à jour de la zone est faite (vérification sur le poste client). Dans notre cas, la date saisie est vérifiée (par exemple on ne peut pas mettre le jour 32 ou le mois 13).

```
function setOnChange(&$form,$maj){  
  
// saisie de la date *****  
// si vide et 6 caractères => 211082 = 21/10/1982  
// si vide et 8 caractères => 21101982 = 21/10/1982  
// contrôle 31 max pour jour et 12 max pour mois  
// *****  
$form->setOnChange("date_article","fdate(this)");  
?>  
<script language="javascript">  
function fdate(champ){  
if(champ.value.lastIndexOf("/")!=-1){  
if (champ.value.substring(0,2)>32){  
champ.value="";  
alert("jour > 32")  
return  
}  
if (champ.value.substring(2,4)>13){  
champ.value="";  
alert("mois > 12")  
return  
}  
if(champ.value.length==6)  
champ.value=champ.value.substring(0,2)+"/"+champ.value.substring(2,4)+"/19"+champ.value.substring(4,6);  
if(champ.value.length==8)  
champ.value=champ.value.substring(0,2)+"/"+champ.value.substring(2,4)+"/"+champ.value.substring(4,8);  
}  
}  
</script>  
<?php  
}
```

5. l'affichage de la Revue de presse

L'affichage de la revue de presse doit être originale et n'est pas générique. Il est donc proposer de surcharger la méthode `tab.class.php` par une méthode spécifique stockée en `obj/revue.class.php` appelée par l'interface `revue_tab.class.php`

5.1 la modification de l interface php

`revue.php` appelle la méthode `afficher` de `revue_tab.class.php` qui permet l'affichage de la pagination, la possibilité de recherche, le titre de l'article avec le chapeau, la date de l'article, le logo du journal et le lien vers l'article entier.

5,2 surcharge de la méthode `afficher` de la classe `tab.class.php` dans `obj/tabrevue.class.php`

`obj/tabrevue.class.php` hérite de la classe `tab.class.php`

```
require_once ("tab.class.php");
class tabrevue extends tab{
```

Trois méthodes sont surchargées

- entête
- afficher
- pagination (retour `revue_tab.class.php`)



6. Mise en place du menu et gestion utilisateurs

Le menu est paramétré en dyn/menu.inc.

Ce menu appelle la fonction spg/menu.js: les explications sont visibles sur les liens suivants:

<http://www.openmairie.org/folder.2005-11-05.5614546554/folder.2005-11-05.0523803223/document.2005-11-05.7642373780/>

<http://iubito.free.fr/prog/menu.php>

6.1 Personnalisation du menu et gestion de droits

L'appel de la page « scr/tab.php?obj=journal » fait dans « dyn/menu.inc ».

```
echo "<a href='../scr/tab.php?obj=journal' >";
```

La première variable « \$titre » permet d'indiquer le titre qui sera affiché sur le menu. Lorsqu'on passe la souris sur le titre, on affiche le sous-menu.

Exemple : \$titre= " Revue Presse";



Le sous-menu possède aussi un titre (ici « Presse ») et contient des rubriques (« Revue de presse », et « Saisie article »).

6.2 administration des droits et des accès

L'administrateur possède donc un menu différent qui lui permet notamment de gérer les droits, les utilisateurs et les profils.

```
$titre= "Administration";  
echo "<li>";  
echo " <font id='titremenu'>".$titre."<br></font>";  
echo "</li>";  
echo "<li><a href='../scr/tab.php?obj=collectivite'>Collectivité</a></li>";  
echo "<li><a href='../scr/tab.php?obj=utilisateur'>Gestion Utilisateur</a></li>";  
echo "<li><a href='../scr/tab.php?obj=profil'>Gestion de Profil</a></li>";  
echo "<li><a href='../scr/tab.php?obj=droit'>Gestion Droits Accès</a></li>";  
echo "</ul>";
```



Chaque utilisateur a un numéro de profil qui lui permet des droits sur les objets qui ont eux aussi un numéro minimum de profil pour la mise à jour.