



API Développeur



SOMMAIRE

A - Comment installer Isens Evolution sur votre serveur ?	4
B - Comment faire des requêtes SQL ?.....	6
C - Comment gérer les gabarits graphiques ?.....	7
D - Comment gérer une base de données ?.....	14
1. API « Base de données »	14
dbConnect(\$change_base)	14
dbQuery(\$sql,\$connection)	14
dbInsert(\$sql,\$connection)	14
dbNumrows(\$result)	14
dbResOk(\$result)	15
dbClose(\$connection)	15
dbFreeRes(\$result)	15
dbFetch(\$result,\$i)	15
dbFetchArray(\$result,\$i)	16
dbDataSeek(\$result,\$num).....	16
dbError()	16
dbNumFields(\$result)	16
dbNameFields(\$result,\$i).....	17
dbLookup(\$champ, \$table, \$condition="", \$connection).....	17
dbOptimize(\$table, \$connection)	17
dbDelete(\$table, \$where, \$connection)	17
2. API « Gestion visuelle des bases de Données ».....	18
Type cle	19
Type text.....	20
Type password	21
Type textarea	21
Type richtext.....	21
Type smalledit	22
Type radio	22
Type checkbox.....	22
Type checkbox-single	23
Type select	23
Type date	23
Type calendrier.....	24
Type table	24
Type fichier	24
Type fichier simple	25
Type hidden	27
Type jointure	27

Type list	28
Type ordre	28
Type titre	28
Type modifie	29
Type supprime	29
Type copie	29
Type ligne	29
Type test	29
Type imprime	30
Type on-delete	30
Type on-delete-cascade	30
Type on-after-create	31
Type on-before-change	31
Type on-change, on-after-change	31
Type on-delete-file	32
Type on-activate	32
Type on-export	32
Type workflow	33
Type stats	33
Type stats-unique	33
Type stats-unique-multiligne	34
Type stats-file	34
Type stats-file-unique	34
Type tri alphabétique	35
Type indexation	35
Type display	35
Type obligatoire	36
Type filtre	36
Type liste-multiple	37
Type google_map	39
Type flash	40
3. API « Moteur de recherche »	40

A - Comment installer Isens Evolution sur votre serveur ?

Isens Evolution™ nécessite :

- Apache
- PHP 5.2 ou supérieur
- MySQL 5 ou supérieur

Cette configuration est proposée par de nombreux hébergeurs professionnels.

Installez Apache: <http://www.apache.org/> (téléchargement gratuit).

Installez PHP : <http://www.php.net/>

Installez MySQL depuis : <http://www.mysql.com/>

Apache

Assurez-vous que le module mod_rewrite est activé en modifiant le cas échéant le fichier de configuration du serveur Apache (httpd.conf).

Vérifiez que les deux lignes suivantes ne soient pas mises en commentaires :

```
LoadModule rewrite_module modules/mod_rewrite.so
AddModule mod_rewrite.c
```

Vérifiez les variables suivantes :

```
ServerRoot "C:/Program Files/EasyPHP1-7/apache"
DocumentRoot "F:/" (à faire pointer vers votre répertoire racine)
<Directory "F:/">
AllowOverride All
AccessFileName .htaccess
```

PHP

Pour des raisons de sécurité, vérifiez que register_globals soit sur « Off » dans le fichier php.ini :

```
register_globals = Off
```

Vérifier la variable "error_reporting" :

```
error_reporting = E_ALL & ~E_NOTICE
```

Décommenter la ligne suivante :

```
extension=php_gd2.dll
```

Vérifier les lignes suivantes :

```
display_startup_errors = Off
magic_quotes_gpc = On
magic_quotes_runtime = Off
```

Installation

Une fois les logiciels paramétrés :

- 1- Copier les fichiers par FTP
- 2- Il faut changer les attributs de répertoires (Chmod 777) sur le répertoire '/tmp', '/tmp/upload' et '/admin'
- 3- Lancer depuis un navigateur l'adresse suivante : <http://DOMAINE/install/>
Spécifier le serveur de base de données, le nom de la base de données, le login et le mot de passe vers MySQL, ainsi que les accès FTP. En option, spécifiez la clé Google Map correspondant à votre nom de domaine

Votre site est installé.

Adresse du site :

<http://DOMAINE/>

Adresse du module d'administration :

<http://DOMAINE/admin/>

Par défaut, un utilisateur est créé avec les accès suivant :

Login : **admin**, Mot de passe : **isens**

Après l'installation, supprimer le répertoire /install/ et changer les attributs du répertoire /admin/ en 755 (Chmod 755)

Fonction copier / coller sous Mozilla / Firefox

Pour résoudre le problème de sécurité du copier / coller depuis l'éditeur d'Isens Evolution (pour Firefox / Mozilla), créez un fichier user.js dans le répertoire "profile" contenant le texte suivant :

```
user_pref("capability.policy.policynames", "allowclipboard");
user_pref("capability.policy.allowclipboard.sites", "http://www.votre-site.com");
user_pref("capability.policy.allowclipboard.Clipboard.cutcopy", "allAccess");
user_pref("capability.policy.allowclipboard.Clipboard.paste", "allAccess");
```

Modifier le fichier user.js en remplaçant <http://www.votre-site.com> par votre adresse URL.

Sur Windows, le chemin est généralement %AppData%\Mozilla\Firefox\Profiles\xxxxxxx.default\, ou xxxxxxxx est une chaîne aléatoire de 8 caractères. Allez voir C:\Documents and Settings\[User Name]\Application Data\Mozilla\Firefox\Profiles\

Sur Linux, le chemin est généralement ~/.mozilla/firefox/xxxxxxx.default/

Sur Mac OS X, le chemin est généralement ~/Library/Application Support/Firefox/Profiles/xxxxxxx.default/

Point sur la sécurité dans Isens Evolution :

- Isens Evolution est basé sur le système LAMP (Linux, Apache, MySQL, PHP)
- Fonctionne en register_globals à OFF
- Fonctionne en SAFE MODE
- Fonctionne en magic_quotes_gpc à ON (Pour éviter les injections MySQL)
- Mot de passe crypté en MD5
- SSL pour la partie administration si l'hébergement le permet
- Code en 100% PHP, pas de modules compilés.
- Fonctionne sur serveur mutualisé ou dédié.

B - Comment faire des requêtes SQL ?

Icône « Outils » > Onglet « SQL »

`show tables` : lister les tables existantes

`desc nom_table` : visualise la description d'une table

`select * from nom_table` : affiche le contenu d'une table

C - Comment gérer les gabarits graphiques ?

API Template

Les template sont stockés dans le répertoire /template/.

L'ensemble des images et animations FLASH sont stockées dans le répertoire /template/img/.

L'ensemble des thèmes sont stockés dans le répertoire /template/theme/.

Chaque répertoire contient différents fichiers, décrits ci-dessous :

index.dwt

Ce fichier correspond au template de la page d'accueil

page.dwt

Ce fichier correspond au template d'une page intérieure type. Vous pouvez à tout moment rajouter de nouveaux fichiers DWT. Il vous suffit de créer un fichier HTML, d'insérer les balises Isens Evolution™ et de le renommer en DWT.

Liste des balises Isens Evolution™

{title}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Titre". Cette balise correspond au méta tag title
{title_ou_titre}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Titre" ou par le nom de la page si ce premier est vide
{description}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Description". Cette balise correspond au Meta tag description
{keywords}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Mots clés". Cette balise correspond au méta tag keywords
{version}	Pour chaque page, cette balise sera remplacée à la publication par le numéro de la version courante.
{id_pages}	Pour chaque page, cette balise sera remplacée à la publication par le numéro d'identifiant de la page courante. Cette balise est utilisée pour lancer un lien direct vers l'éditeur.
{ordre}	Pour chaque page, cette balise sera remplacée à la publication par le numéro d'ordre de la page courante

{resume}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Résumé".
{indexation}	Pour chaque page, cette balise sera remplacée à la publication par 'noindex' dans le cas d'une exclusion sur les moteurs, ou 'index,follow' dans le cas d'une inclusion.
{mot}	Cette balise sera remplacée lors d'une recherche par mots clés. Elle sera remplacée par le mot entré par l'utilisateur. Cette balise s'utilise à l'intérieur du champ de recherche pour rappeler à l'utilisateur le mot entré.
{numero}	Pour chaque page, cette balise sera remplacée à la publication par le numéro de la page courante.
{sid}	Pour chaque page, cette balise sera remplacée à la publication par le SID de la page courante.
{theme}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Thème". Par défaut, le thème spécifié dans le fichier template.inc.php est pris.
{style}	Pour chaque page, cette balise sera remplacée à la publication par le contenu du champ "Style".
{navigation}	Pour chaque page, cette balise sera remplacée à la publication par le chemin parcouru. Exemple : Accueil > Produits > Fonctionnalités Le gabarit est géré dans le fichier template.inc.php
{menu}	Cette balise sera remplacée par le menu général (1° niveau). Le gabarit est géré dans le fichier template.inc.php
{sousmenu}	Cette balise sera remplacée par le sous menu (2° niveau). Le gabarit est géré dans le fichier template.inc.php
{soussousmenu}	Cette balise sera remplacée par le sous sous menu (3° niveau). Le gabarit est géré dans le fichier template.inc.php
{soussoussousmenu}	Cette balise sera remplacée par le sous sous sous menu (4° niveau). Le gabarit est géré dans le fichier template.inc.php

{menu_dynamique}	Cette balise sera remplacée par l'ensemble du menu comprenant tous les niveaux
{home_menu_dynamique}	Cette balise sera remplacée par l'ensemble du menu comprenant tous les niveaux, en utilisant le template de menu défini pour la page d'accueil
{menu_dynamique_niveau1}	Cette balise sera remplacée par l'ensemble du menu à partir du niveau 1
{menu_dynamique_niveau2}	Cette balise sera remplacée par l'ensemble du menu à partir du niveau 2
{menu_dynamique_niveau3}	Cette balise sera remplacée par l'ensemble du menu à partir du niveau 3
{titre}	Cette balise sera remplacée par le titre de la page courante.
{titre_niveau0}	Cette balise sera remplacée par le titre de la page mère (à la racine de l'arborescence).
{titre_niveau1} {titre_niveau2} {titre_niveau3} {titre_niveau4} {titre_niveau5}	Ces balises seront remplacées par les titres des pages mères, de niveau 1 à 5
{titre_parent}	Cette balise sera remplacée par le libellé de la page mère.
{titre_parent_sous_menu}	Cette balise sera remplacée par le libellé de la page mère, uniquement pour les pages de niveau 2 ('sous menu').
{titre_parent_sous_sous_menu}	Cette balise sera remplacée par le libellé de la page mère, uniquement pour les pages de niveau 3 ('sous sous menu').
{titre_parent_sous_sous_sous_menu}	Cette balise sera remplacée par le libellé de la page mère, uniquement pour les pages de niveau 4 ('sous sous sous menu').
{corps}	Cette balise sera remplacée par le corps de la page courante.
{zone1} {zone2} {zone3} {zone4} {zone5} {zone6}	Ces balises seront remplacées par le contenu du bloc 1,2,3,4,5 ou 6
{marker}	Cette balise sera remplacée par le marqueur des statistiques.
{xiti}	Génère le chemin suivant le format XITI . Ex : accueil ::produits ::documentation (...)
{url_template}	Cette balise sera remplacée par

	« template/nom_du_repertoire_courant/ » à la publication
{language}	Cette balise sera remplacée par le contenu du champ correspondant dans la gestion des versions (icône Versions)

popup.dwt

Ce fichier correspond au template d'une fenêtre popup.

style.css / form.css

Ces fichiers correspondent aux feuilles de styles. Ils doivent impérativement se nommer de cette façon.

listeData.inc.php

Ce fichier correspond à la description des tables pour la gestion de base données. Il doit impérativement se nommer de cette façon.

template.inc.php

Ce fichier contient les gabarits d'affichage pour les menus et sous menus, la navigation, le moteur de recherche, le plan du site et le thème par défaut.

template.jpg

Ce fichier permet de visualiser les différents template depuis le module de gestion des versions. C'est une image au format JPG de taille 205 x 149 pixels. Il représente la page d'accueil du site.

Template d'inclusion

Vous pouvez faire appel à un autre template depuis un fichier DWT en appelant cet autre template avec la syntaxe : {nom_du_fichier.dwt}

Cela permet d'éviter les doublons de code HTML ou PHP dans les différents fichiers DWT.

Structure du fichier template.inc.php

Le fichier template.inc.php contient un ensemble de variables permettant d'afficher les menus, sous menus, sous sous menus, sous sous sous menus, chemin parcouru (navigation), moteur de recherche et plan du site.

Voici sa structure :

```
// CHAINE PUBLIEE AVANT LE SOUS MENU
$start_sousmenu_tpl = "";

// CHAINE PUBLIEE APRES LE SOUS MENU
$end_sousmenu_tpl = "";

// CHAINE PUBLIEE AVANT LE SOUS SOUS MENU
$start_soussousmenu_tpl = "";

// CHAINE PUBLIEE APRES LE SOUS SOUS MENU
$end_soussousmenu_tpl = "";

// CHAINE PUBLIEE AVANT LE SOUS SOUS SOUS MENU
$start_soussoussousmenu_tpl = "";

// CHAINE PUBLIEE APRES LE SOUS SOUS SOUS MENU
$end_soussoussousmenu_tpl = "";

// CHAINE PUBLIEE APRES LE TITRE PARENT (POUR LE SOUS MENU)
$start_titre_parent_sous_menu = "";

// CHAINE PUBLIEE APRES LE TITRE PARENT (POUR LE SOUS SOUS MENU)
$start_titre_parent_sous_sous_menu = "";

// CHAINE PUBLIEE APRES LE TITRE PARENT (POUR LE SOUS SOUS SOUS MENU)
$start_titre_parent_sous_sous_sous_menu = " : ";

// Libellé des onglets pour les blocs
$libelle_zone1 = "...";
$libelle_zone2 = "...";
$libelle_zone3 = "...";
$libelle_zone4 = "...";
$libelle_zone5 = "...";
$libelle_zone6 = "...";

// Type des onglets pour les blocs
$type_zone1 = "textarea";
$type_zone2 = "smalldit";
$type_zone3 = "textarea";
$type_zone4 = "textarea";
$type_zone5 = "textarea";
$type_zone6 = "textarea";
```

```

// PAGE D'ACCUEIL - Menu
$home_menu_tpl = <a href="{fichier_sortie}">{libelle}</a><br />";

// PAGE INTERIEURE - Menu
$menu_tpl = <a href="{fichier_sortie}">{libelle}</a><br />";

// PAGE D'ACCUEIL - Menu sélectionné
$home_menu_select_tpl = "";

// PAGE INTERIEURE - Menu sélectionné
$menu_select_tpl = "";

// PAGE D'ACCUEIL - Sous menu
$home_sousmenu_tpl = "";

// PAGE INTERIEURE - Sous menu
$sousmenu_tpl = "<a href="{fichier_sortie}">{libelle}</a><br />";

// PAGE D'ACCUEIL - Sous menu sélectionné
$home_sousmenu_select_tpl = "";

// PAGE INTERIEURE - Sous menu sélectionné
$sousmenu_select_tpl = "";

// PAGE D'ACCUEIL - Sous sous menu
$home_soussousmenu_tpl = "";

// PAGE INTERIEURE - Sous sous menu
$soussousmenu_tpl = "";

// PAGE D'ACCUEIL - Sous sous menu sélectionné
$home_soussousmenu_select_tpl = "";

// PAGE INTERIEURE - Sous sous menu sélectionné
$soussousmenu_select_tpl = "";

// PAGE D'ACCUEIL - Sous sous sous menu
$home_soussoussousmenu_tpl = "";

// PAGE INTERIEURE - Sous sous sous menu
$soussoussousmenu_tpl = "";

// PAGE D'ACCUEIL - Sous sous sous menu sélectionné
$home_soussoussousmenu_select_tpl = "";

// PAGE INTERIEURE - Sous sous sous menu sélectionné
$soussoussousmenu_select_tpl = "";

// NAVIGATION
$navigation_avant_tpl = "<a href='\"index.php\"'>Accueil</a> > ";
$navigation_tpl = "<a href="{fichier_sortie}">{libelle}</a> ";
$navigation_courant_tpl = "<a href="{fichier_sortie}">{libelle}</a> ";
$separateur_tpl = "> ";
$navigation_apres_tpl = "";

```

```
// THEME PAR DEFAULT
$type_theme = "gif"; // swf, gif, jpg uniquement
$theme_default = "";
```

```
// COULEUR DE FOND DE L'EDITEUR
$background_site = "#E3E5F2";
$background_home = "#E3E5F2";
```

```
// LARGEUR EN PIXEL DE L'EDITEUR
$largeur_editeur = 486;
```

Intégration d'un plan de site

Il suffit de créer une page contenant le texte suivant :

```
<div id="votre-style">
{site_map}
</div>
```

D - Comment gérer une base de données ?

Il faut au préalable inclure le fichier db.inc.php dans votre script PHP :
`include("admin/db.inc.php");`

1. API « Base de données »

dbConnect(\$change_base)

Crée une connexion vers la base de données

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";  
$connection = dbConnect();  
$result = dbQuery($sql,$connection);  
  
// Pour se connecter à une base différente  
$connection = dbConnect("autre_base");
```

dbQuery(\$sql,\$connection)

Lance une requête SQL vers la base de données

Exemple d'utilisation

```
$result = dbQuery($sql,$connection);
```

dbInsert(\$sql,\$connection)

Lance une requête SQL d'insertion vers la base de données et renvoie l'identifiant du nouvel enregistrement

Exemple d'utilisation

```
$sql = "INSERT INTO table_exemple(id_table_exemple) VALUES('Nouveau libellé');"  
$new_id = dbInsert($sql,$connection);
```

dbNumrows(\$result)

Récupère le nombre de résultats d'une requête

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";  
$result = dbQuery($sql,$connection);  
$k = dbNumrows($result);
```

dbResOk(\$result)

Indique si le nombre de résultat d'une requête est supérieur à 0

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";
$connection = dbConnect();
$result = dbQuery($sql,$connection);

if (dbResOk($result)) echo "Il existe des résultats";
```

dbClose(\$connection)

Ferme la connection vers la base de données

Exemple d'utilisation

```
dbClose($connection);
```

dbFreeRes(\$result)

Libère la mémoire allouée pour stocker les résultats d'une requête

Exemple d'utilisation

```
dbFreeRes($result);
```

dbFetch(\$result,\$i)

Récupère une ligne de résultat dans un tableau accessible par numéro de champs

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";
$connection = dbConnect();
$result = dbQuery($sql,$connection);
$k = dbNumrows($result);

for($j = 0;$j < $k;$j++)
{
    $row = dbFetch($result,$j);
    echo $row[0];
    echo $row[1];
}
```

dbFetchArray(\$result, \$i)

Récupère une ligne de résultat dans un tableau accessible par nom de champs

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";
$connection = dbConnect();
$result = dbQuery($sql, $connection);
$k = dbNumrows($result);

for($j = 0; $j < $k; $j++)
{
    $row = dbFetchArray($result, $j);
    echo $row["libelle"];
    echo $row["date_enregistrement"];
}
```

dbDataSeek(\$result, \$num)

Déplace le pointeur sur le tableau des résultats

Exemple d'utilisation

```
dbDataSeek($result, 45);
```

dbError()

Retourne le message d'erreur de la dernière opération effectuée

Exemple d'utilisation

```
$result = dbQuery($sql, $connection);
echo dbError();
```

dbNumFields(\$result)

Récupère le nombre de champs du résultat courant

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";
$connection = dbConnect();
$result = dbQuery($sql, $connection);
$nombre_champs = dbNumFields($result);
```


dbNameFields(\$result,\$i)

Récupère le nom du champs suivant son numéro d'ordre

Exemple d'utilisation

```
$sql = "SELECT * FROM table_exemple";
$connection = dbConnect();
$result = dbQuery($sql,$connection);
for ($j = 0;$j < dbNameFields($result);$j++)
{
    $liste_champ .= dbNameFields($result,$j);
}
```

dbLookup(\$champ, \$table, \$condition="", \$connection)

Récupère la valeur d'un champ suivant le nom d'une table et une condition

Exemple d'utilisation

```
$ma_valeur = dbLookup("mon_champ","ma_table_lieu", "IdLieu = $IdLieu", $connection);
$ma_valeur = dbLookup("count(IdLieu)","ma_table_lieu", "", $connection);
```

dbOptimize(\$table, \$connection)

Optimise une table suivant son nom et une condition. A lancer après une suppression

Exemple d'utilisation

```
dbOptimize("ma_table", $connection);
```

dbDelete(\$table, \$where, \$connection)

Supprime le contenu d'une table suivant son nom et une condition

Exemple d'utilisation

```
dbDelete("ma_table", "where id_table=1", $connection);
dbDelete("ma_table", "", $connection);
```

2. API « Gestion visuelle des bases de Données »

Dans chaque template, vous pouvez définir une gestion de base de données différentes.

Par exemple, pour une version française utilisant le template "Français", la gestion de base de données permettra l'accès à une table EXEMPLE_1.

Pour une version anglaise utilisant le template "English", la gestion de base de données permettra l'accès à une table EXEMPLE_2.

Il vous suffit de placer un fichier appelé listeData.inc.php dans chaque template (ou répertoire).

Les template sont stockés dans le répertoire /template/.

Exemple de fichier listeData.inc.php :

Ce fichier décrit une ou plusieurs tables SQL.

Un tableau principal nommé \$listeData contient différents sous tableaux.

Pour chaque sous tableau :

- la première ligne indique le libellé de la table : "Actualités"
- la deuxième ligne indique le nom de la table : "isens_actu"
- la troisième ligne et les suivantes décrivent les champs de la table : "date", "calendrier", "text", "richtext" etc.

```
$listeData = array(
  0 => array(
    "Actualités",
    "isens_actu",
    array("id_isens_actu","Id","cle",1),
    array("date_creation","Date","date",0,"d-m-Y H:i:s"),
    array("date_debut","Début","calendrier",0),
    array("date_fin","Fin","calendrier",0),
    array("titre","Titre","text",1),
    array("corps","Corps","richtext",0),
    array("ordre","SELECT id_isens_actu, concat(id_isens_actu,\" / \",titre) FROM
isens_actu ORDER BY ordre","ordre",0),
    array("", "", "modifie", 1),
    array("", "", "supprime", 1),
    array("en_ligne", "", "ligne", 1)
  ),
  1 => array(
    "Espace documentations",
    "demo_fichier",
    array("id_fichier","Identifiant","cle",1),
    array("id_groupe","Groupe","table",1,"SELECT nom FROM demo_groupe WHERE
id_groupe=", "SELECT id_groupe, nom FROM demo_groupe"),
```

```

        array("nom","Nom du fichier","text",1),
        array("fic_demo","Fichier","fichier",1,"_demo.doc","../doc/","Format DOC"),
        array("commentaires","Commentaires","textarea",0),
        array("", "", "modifie", 1),
        array("", "", "supprime", 1)
    )
);

$max = sizeof($listeData);

```

Cet exemple correspond au code SQL suivant :

```

CREATE TABLE isens_actu
(
    id_isens_actu bigint NOT NULL auto_increment,
    titre text DEFAULT "",
    corps text DEFAULT "",
    date_creation int,
    date_debut int,
    date_fin int,
    en_ligne smallint,
    ordre bigint,
    PRIMARY KEY (id_isens_actu)
);

CREATE TABLE demo_fichier
(
    id_fichier bigint NOT NULL auto_increment,
    id_groupe bigint DEFAULT "",
    nom text,
    fic_demo smallint,
    commentaires text,
    PRIMARY KEY (id_fichier)
);

```

Type cle

Ce type est utilisé pour les champs représentant une clé primaire, déclarés en **bigint DEFAULT '0' NOT NULL auto_increment**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = cle

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Exemple d'utilisation

```
array("id","Identifiant","cle",1)
```

Type text

Ce type est utilisé pour les champs textuels, déclarés en `text` ou `varchar`

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = `text`

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Contrôle = Type du contrôle pour la saisie : `controle-entier`, `controle-flottant`, `controle-heure`, `controle-email`, `controle-url`, `controle-date`, `controle-majuscule`, `controle-minuscule`, `controle-premiere-majuscule`, `controle-code-postal`, `controle-telephone`

Limite inférieure = Nombre, date ou heure pour la limite inférieure de saisie

Limite supérieure = Nombre, date ou heure pour la limite supérieure de saisie

Précision = Dans le cas d'un contrôle de flottant, spécification de la précision après virgule

Valeur par défaut = Valeur affiché par défaut lors d'un ajout

Exemple d'utilisation

```
array("nom","Nom","text",1)
```

```
array("titre","Titre","text",1,"controle-entier","10","50")
```

```
array("titre","Titre","text",1,"controle-entier","10")
```

```
array("titre","Titre","text",1,"controle-entier")
```

```
array("titre","Titre","text",1,"controle-flottant","0.23","5420.12","4")
```

```
array("titre","Titre","text",1,"controle-flottant","0.23","5420.12")
```

```
array("titre","Titre","text",1,"controle-flottant","0.23")
```

```
array("titre","Titre","text",1,"controle-flottant")
```

```
array("titre","Titre","text",1,"controle-heure","09:00","15:00")
```

```
array("titre","Titre","text",1,"controle-heure","09:00")
```

```
array("titre","Titre","text",1,"controle-heure","","15:00")
```

```
array("titre","Titre","text",1,"controle-heure")
```

```
array("titre","Titre","text",1,"controle-email")
```

```
array("titre","Titre","text",1,"controle-url")
```

```
array("titre","Titre","text",1,"controle-date","1970/04/20","2003/01/31")
```

```
array("titre","Titre","text",1,"controle-date")
```

```
array("titre","Titre","text",1,"controle-majuscule")
```

```
array("titre","Titre","text",1,"controle-minuscule")
```

```
array("titre","Titre","text",1,"controle-premiere-majuscule")
```

```
array("titre","Titre","text",1,"controle-code-postal")
```

```
array("titre","Titre","text",1,"controle-telephone")
```

```
array("titre","Titre","text",1,"controle-telephone")
```

```
array("titre","Titre","text",1,"","","","Valeur par défaut")
```

Type password

Ce type est utilisé pour les champs de type mot de passe, déclarés en **text** ou **varchar**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = password

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Exemple d'utilisation

```
array("passwd", "Mot de passe", "password", 1)
```

Type textarea

Ce type est utilisé pour les champs textuels, déclarés en **text**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = textarea

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Zone agrandie = xxl (Cette option permet d'agrandir la zone de saisie)

Valeur par défaut = Valeur affichée par défaut lors d'un ajout

Exemple d'utilisation

```
array("adresse", "Adresse", "textarea", 0)
```

```
array("adresse", "Adresse", "textarea", 0, "xxl")
```

```
array("adresse", "Adresse", "textarea", 0, "xxl", "Valeur par défaut")
```

Type richtext

Ce type est utilisé pour les champs textuels, déclarés en **text** ou **longtext**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = richtext

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Exemple d'utilisation

```
array("descriptif", "Descriptif", "richtext", 0)
```

```
array("fiche_technique", "Fiche technique", "richtext", 0)
```

Type smalletedit

Ce type est utilisé pour les champs textuels (éditeur réduit par rapport à un type richtext), déclarés en **text** ou **longtext**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = smalletedit

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Exemple d'utilisation

```
array("descriptif","Descriptif","smalletedit",0)
```

```
array("fiche_technique","Fiche technique","smalletedit",0)
```

Type radio

Ce type est utilisé pour des boutons radios, pour des champs déclarés en **text** ou **varchar**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = radio

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Liste 1 séparé par des # = Liste des libellés

Liste 2 séparé par des # = Liste des valeurs envoyés (optionnel)

Exemple d'utilisation

```
array("civillite","Civilité","radio",1,"Monsieur#Madame#Mademoiselle")
```

```
array("choix","Choix","radio",1,"Oui#Non#N-A", "0#1#2")
```

```
array("sexe","Sexe","radio",1,"Masculin#Féminin")
```

Type checkbox

Ce type est utilisé pour des cases à cocher, pour des champs déclarés en **text** ou **varchar**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = checkbox

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Liste séparé par des # = Liste de choix

Exemple d'utilisation

```
array("choix","Choix","checkbox",1,"Célibataire#Marié#Homme")
```

Type checkbox-single

Ce type est utilisé pour créer une case à cocher unique, pour des champs déclarés en smallint

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = checkbox-single

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Exemple d'utilisation

```
array("administrateur","Administrateur","checkbox-single",0)
```

Type select

Ce type est utilisé pour créer une boîte de sélection, pour des champs déclarés en smallint

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = checkbox-single

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Liste séparé par des # = Liste de choix

Valeur correspondante séparée par # = Liste de valeurs correspondantes aux choix

Exemple d'utilisation

```
array("categorie","Catégorie","select",0,"Abonné#Enfant#Carte  
PASS#Standard","0#1#2#3")
```

Type date

Ce type est utilisé pour créer un champ d'insertion automatique de la date du jour, déclaré en int

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = date

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Format d'affichage = d/m/Y H:i (Pour d'autres formats, voir la fonction date sur www.php.net)

Exemple d'utilisation

```
array("date_creation","Date de création","date",1,"d/m/Y H:i")
```

Type calendrier

Ce type est utilisé pour créer un champ avec sélection à partir d'un calendrier, déclaré en **int**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = calendrier

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Heure = Spécifie si l'heure doit être affichée

Limite inférieure = Date pour la limite inférieure de saisie

Limite supérieure = Date pour la limite supérieure de saisie

Exemple d'utilisation

```
array("date_creation", "Date de création", "calendrier", 0)
```

```
array("date_debut", "Début", "calendrier", 0, "", "1970/04/20", "2003/01/31")
```

```
array("date_fin", "Fin", "calendrier", 0, "heure")
```

Type table

Ce type est utilisé pour un champ de clé étrangère, déclaré en **bigint**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = table

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Requête SQL = Requête qui sélectionne la liaison courante

Requête SQL = Requête qui sélectionne l'ensemble de la table

Exemple d'utilisation

```
array("pere", "Père", "table", 1, "SELECT nom FROM test1 WHERE id=", "SELECT id, nom FROM test1")
```

```
array("id_groupe", "Groupe", "table", 1, "SELECT nom FROM demo_groupe WHERE id_groupe=", "SELECT id_groupe, nom FROM demo_groupe")
```

Type fichier

On stocke 0 ou 1 en base de données.

Ce type est utilisé pour un champ lié à un fichier sur disque. Correspond à un champ déclaré en **smallint**

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = fichier

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Nom de l'image = Isens Evolution™ place automatiquement l'identifiant devant pour créer le nom du fichier (Exemple : 124_technic.pdf)

Emplacement = Emplacement de stockage des fichiers sur le serveur

Format pour affichage = Affichage du format pour les utilisateurs

Contrôle = Type du contrôle pour la saisie : controle-file

Format pour affichage = Extension du format désiré : doc, xls, pdf, gif, jpg, rtf...

Exemple d'utilisation

```
array("photo","Photo","fichier",0,"_actu.gif","../actu/","Format GIF")
```

```
array("document_pdf","Document  
technique","fichier",0,"_technic.pdf","../files/","Format PDF")
```

```
array("document_word","Document  
commerciale","fichier",0,"_commerciale.doc","../files/","Format DOC")
```

```
array("actu","Actualités","fichier",0,"_actu.doc","../actu/","Format DOC","controle-  
file","doc")
```

```
array("photo","Photo","fichier",0,"_actu.gif","../photo/","Format JPG","controle-  
file","gif")
```

```
array("photo","Photo","fichier",0,"_actu.jpg","../photo/","Format JPG","controle-  
file","jpg")
```

Type fichier simple

On ne stocke rien en base de données.

Ce type est utilisé pour un champ lié à un fichier sur disque. Il n'y a pas de liaison directe avec la base de données (uniquement sur le nom du fichier qui comporte l'identifiant de l'enregistrement).

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = fichier-simple

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Nom de l'image = Isens Evolution™ place automatiquement l'identifiant devant pour créer le nom du fichier (Exemple : 124_technic.pdf)

Emplacement = Emplacement de stockage des fichiers sur le serveur

Format pour affichage = Affichage du format pour les utilisateurs

Contrôle = Type du contrôle pour la saisie : controle-file

Format pour affichage = Extension du format désiré : doc, xls, pdf, gif, jpg, rtf...

Exemple d'utilisation

```
array("photo","Photo","fichier-simple",0,"_actu.gif","../actu/","Format GIF")
```

```
array("document_pdf","Document technique","fichier-  
simple",0,"_technic.pdf","../files/","Format PDF")
```

```
array("document_word","Document commerciale","fichier-  
simple",0,"_commerciale.doc","../files/","Format DOC")
```

```
array("actu", "Actualités", "fichier-simple", 0, "_actu.doc", "../actu/", "Format  
DOC", "controle-file", "doc")
```

Type hidden

Ce type est utilisé pour un champ caché, déclaré avec le format adéquat (text, bigint, smallint etc.)

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = hidden

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Contenu = Contenu du champ caché

Exemple d'utilisation

```
array("coord_x","Coordonnées X","hidden",0,"136")
```

Type jointure

Ce type est utilisé pour une jointure entre table. A partir de la version 7.7, il est conseillé d'utiliser le type liste-multiple.

Nom SQL = Nom du champ pour le formulaire (ce n'est pas un champ SQL)

Titre = Libellé explicatif pour affichage

Type = jointure

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Table de jointure = Nom de la table de jointure de deux tables (Exemple : isens_user_pages)

Clé table 1 = Nom du champ de clé étrangère vers la première table (Exemple : id_user)

Clé table 2 = Nom du champ de clé étrangère vers la deuxième table (Exemple : id_pages)

Requête 1 = Requête qui sélectionne l'ensemble des possibilités de jointure entre les deux tables

Requête 2 = Requête qui sélectionne les jointures liées à l'enregistrement courant

Clé jointure = Nom de la clé primaire de la table de jointure

Type = jointure-isens-page (uniquement dans le cas de liaison avec les pages d'isens)

Exemple d'utilisation

```
array("pages","Pages","jointure",0,"isens_user_pages","id_user","id_pages","SELECT b.id_pages, a.libelle FROM isens_version a, isens_pages b ORDER BY a.id_version ASC, b.ordre ASC","SELECT id_pages FROM isens_user_pages WHERE id_user=", "id_user_pages")
```

```
array("pages","Pages","jointure",0,"isens_user_pages","id_user","id_pages","SELECT b.id_pages, concat(\"Version \",a.libelle,\" > \",b.nom) FROM isens_version a, isens_pages b ORDER BY a.id_version ASC, b.ordre ASC","SELECT id_pages FROM isens_user_pages WHERE id_user=", "id_user_pages")
```

```
array("pages","Droit d'accès","jointure",1,"isens_annuaire_pages_jn","id_annuaire","id_pages", "", "SELECT id_pages FROM isens_annuaire_pages_jn WHERE id_annuaire=", "id_isens_annuaire_pages_jn","jointure-isens-pages")
```

Type list

Ce type est utilisé pour un champ lié à un fichier disponible dans le répertoire `/files/`.
Correspond à un champ déclaré en `text` ou `varchar`

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = list

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Emplacement = Emplacement de stockage des fichiers sur le serveur

Exemple d'utilisation

```
array("document","Document technique","list",0)
```

```
array("document","Document technique","list",0,"../document")
```

Type ordre

Pour lancer le système virtuel d'ordonnancement d'enregistrement.

Ce type est utilisé pour trier visuellement des enregistrements. Correspond à un champ déclaré en `bigint`.

Nom SQL = Nom du champ dans la table SQL

Requête SQL = Requête permettant d'afficher les champs sélectionnés dans la fenêtre de tri (titre, libellé, nom, numéro...)

Type = ordre

Présent accueil = 0

Exemple d'utilisation

```
array("ordre","SELECT id_isens_actu, concat(id_isens_actu,\' / \',titre) FROM  
isens_actu ORDER BY ordre","ordre",0),
```

Type titre

Pour affichage seulement.

Ce type est utilisé pour améliorer la présentation des formulaires. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide `""`

Titre = Titre du bloc

Type = titre

Présent accueil = 0

Exemple d'utilisation

```
array("", "Coordonnées de livraison", "titre", 0)
```

```
array("", "Explication", "titre", 0, "Après avoir cliqué sur 'enregistrer', un email sera envoyé à la personne inscrite")
```

```
array("", "Coordonnées personnelles", "titre", 0)
```

Type modifie

Ce type est obligatoire et non modifiable.

Exemple d'utilisation

```
array("", "", "modifie", 1)
```

Type supprimer

Ce type est obligatoire et non modifiable.

Exemple d'utilisation

```
array("", "", "supprime", 1)
```

Type copie

Ce type est optionnel et non modifiable. Il permet de dupliquer l'enregistrement sélectionné.

Exemple d'utilisation

```
array("", "", "copie", 1)
```

Type ligne

Ce type est optionnel. Il permet l'activation ou la désactivation d'un enregistrement. Correspond à un champ déclaré en **smallint**.

Exemple d'utilisation

```
array("en_ligne", "En ligne", "ligne", 1)
```

Type test

Ce type est optionnel. Il permet de faire un lien direct vers une page qui permettra de tester l'enregistrement. Il n'y a pas de liaison directe avec la base de données.

Exemple d'utilisation

```
array("", "", "test", 1, "http://www.votre-site.com/ma-page.php?id=")
```

Type imprime

Ce type est optionnel. Il permet de faire un lien vers un état d'impression. Il n'y a pas de liaison directe avec la base de données.

Exemple d'utilisation

```
array("", "", "imprime", 1)
```

Type on-delete

Lancé lors d'une suppression

Ce type est lancé lors d'une suppression. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Requête SQL = Requête permettant la suppression de jointure simple

Type = on-delete

Nom de la table = Indique le nom de la table ou supprimer

Exemple d'utilisation

```
array("", "DELETE FROM isens_user_pages WHERE id_user=", "on-delete", "isens_user_pages")
```

Type on-delete-cascade

Lancé lors d'une suppression.

Ce type est lancé lors d'une suppression. Il n'y a pas de liaison directe avec la base de données.

Requête SQL = Sélection des champs à supprimer

Requête SQL = Requête permettant la suppression de jointure multiple, suivant la sélection ci-dessus

Type = on-delete-cascade

Exemple d'utilisation

```
array("SELECT id_pages FROM isens_pages WHERE id_version=", "DELETE FROM isens_user_pages WHERE id_pages=", "on-delete-cascade")
```

Type on-after-create

Lancé après une création

Ce type est lancé après une création. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = on-after-create

Exemple d'utilisation

```
array("", "email.inc.php", "on-after-create")
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type on-before-change

Lancé avant une modification

Ce type est lancé avant une modification. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = on-before-change

Exemple d'utilisation

```
array("", "email.inc.php", "on-before-change")
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type on-change, on-after-change

Lancé après une modification

Ce type est lancé après une modification. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = on-change ou on-after-change

Exemple d'utilisation

```
array("", "email.inc.php", "on-change")
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type on-delete-file

Lancé avant une suppression

Ce type est lancé avant une suppression. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = on-delete-file

Exemple d'utilisation

```
array("", "poubelle.inc.php", "on-delete-file")
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type on-activate

Lancé après une activation d'enregistrement (type ligne)

Ce type est lancé après une activation. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = on-activate

Exemple d'utilisation

```
array("", "alerte.inc.php", "on-activate")
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type on-export

Lancé depuis la page 'Exporter > Au format Excel'.

Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = on-export

Exemple d'utilisation

```
array("", "export.inc.php", "on-export")
```


Type workflow

Lancé depuis l'interface d'accueil.

Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Champ vide = Doit comprendre uniquement une chaîne vide ""

Type = workflow

Présent accueil = 1 obligatoirement

Nom du fichier = Nom du fichier à placer dans le répertoire template

Exemple d'utilisation

```
array("", "", "workflow", 1, "workflow.inc.php"),
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type stats

Ce type est lancé sur la page d'accueil de la gestion courante. Il n'y a pas de liaison directe avec la base de données.

Requête SQL = Sélection dans une base de données (Données statistiques)

Titre = Libellé explicatif pour affichage

Type = stats

Exemple d'utilisation

```
array("SELECT count(id_prestataire) FROM prestataire WHERE dep_intervention like\n'%09%' ", "Nombre de prestataires pour 31 - Haute-Garonne", "stats")
```

Type stats-unique

Ce type est lancé lors d'une modification d'un enregistrement. Il n'y a pas de liaison directe avec la base de données.

Requête SQL = Sélection dans une base de données suivant l'identifiant courant (Données statistiques)

Titre = Libellé explicatif pour affichage

Type = stats-unique

Exemple d'utilisation

```
array("SELECT count(*) FROM jointure_reservation_manifestation WHERE\nid_manifestation=", "Nombre de réservation effectuée pour cette\nmanifestation", "stats-unique")
```

Type stats-unique-multiligne

Ce type est lancé lors d'une modification d'un enregistrement. Il n'y a pas de liaison directe avec la base de données.

Requête SQL = Sélection dans une base de données suivant l'identifiant courant (Données statistiques)

Titre = Libellé explicatif pour affichage

Type = stats-unique-multiligne

Liste séparé par des # = Libellé des champs de la requête

Exemple d'utilisation

```
// Détail du contenu d'une commande (jointure et affichage de la liste des produits)
array("SELECT b.libelle, b.reference, l.quantite, l.prix_ttc FROM ligne_commande l,
bouteille b WHERE b.id_bouteille=l.id_bouteille AND l.id_commande=", "Détails
commande", "stats-unique-multiligne", "Libellé#Référence#Quantité#Prix € TTC")
```

Type stats-file

Ce type est lancé sur la page d'accueil de la gestion courante (icône Audience). Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = stats-file

Exemple d'utilisation

```
// Statistiques sur les commandes
array("", "stats_commande.inc.php", "stats-file")
```

Le script contient ensuite une variable `$date` qui permet d'utiliser le calendrier

Type stats-file-unique

Ce type est lancé lors d'une modification d'un enregistrement. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Nom du fichier = Nom du fichier à placer dans le répertoire template

Type = stats-file-unique

Exemple d'utilisation

```
// Statistiques sur une commande sélectionnée
array("", "stats_commande_select.inc.php", "stats-file-unique")
```

Vous disposez ensuite de la variable `$id_courant` dans votre script. Cette variable contient l'identifiant de l'enregistrement courant.

Type tri alphabétique

Ce type permet de donner un accès alphabétique aux enregistrements. Il n'y a pas de liaison directe avec la base de données.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Champ vide = Doit comprendre uniquement une chaîne vide ""

Type = tri-alphabétique

Présent accueil = 0

Nom SQL = Nom du champ sur lequel est fait le tri alphabétique

Exemple d'utilisation

```
array("mot", "Mot", "text", 1),  
array("", "", "tri-alphabétique", 0, "mot")
```

Type indexation

Ce type permet d'indexer des fichiers PDF ou DOC en base de données. Correspond à un champ déclaré en **longtext**

Nom SQL = Nom du champ dans la table SQL

Champ vide = Doit comprendre uniquement une chaîne vide ""

Type = indexation

Présent accueil = 0

Nom SQL = Nom du champ où est stocké le fichier à indexer

Exemple d'utilisation

```
array("chemin", "Fichier", "list", 1),  
array("recherche", "", "indexation", 0, "chemin")
```

Type display

Ce type permet d'afficher un champ sans pouvoir le modifier

Nom SQL = Nom du champ dans la table SQL

Titre = Libellé explicatif pour affichage

Type = display

Présent accueil = 0

Format = vide ou date

Formatage = format documenté sur php.net, fonction date

Exemple d'utilisation

```
array("version", "Version du document", "display", 1)  
array("date_creation", "Date du document", "display", 1, "date")  
array("date_creation", "Date du document", "display", 1, "date", "d-m-Y H:i:s")
```

Type obligatoire

Ce type est optionnel. Il permet d'indiquer les champs obligatoires en listant les noms des champs SQL séparés par le caractère #. Pour une jointure (dans l'exemple le champ theme), ajouter [] après le nom du champ.

Exemple d'utilisation

```
array("", "", "obligatoire", 0, "nom#email#theme[]#ville")
```

Type filtre

Ce type est optionnel. Il permet d'ajouter un filtre sur un ou plusieurs champs SQL. L'intérêt est de pouvoir gérer une table sur plusieurs onglets différents. Il peut être couplé avec le type hidden.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Champ vide = Doit comprendre uniquement une chaîne vide ""

Type = filtre

Présent accueil = 0

Code SQL = Condition du filtre

Liste = Id des groupes, séparés par #, qui ne doivent pas appliquer le filtre

Exemple d'utilisation

```
array("categorie", "Catégorie", "hidden", 1, 6),  
array("", "", "filtre", 0, "id_categorie=6")
```

```
array("id_groupe_user", "Groupe", "hidden", 1, $_SESSION["droit_id_groupe_user"]),  
array("", "", "filtre", 0, "id_groupe_user=" . $_SESSION["droit_id_groupe_user"])
```

```
array("id_user", "Utilisateur", "hidden", 1, $_SESSION["droit_id_user"]),  
array("", "", "filtre", 0, "id_user=" . $_SESSION["droit_id_user"], "1#2")
```

Type liste-multiple

Ce type permet de gérer des jointures entre tables, sur un, deux ou trois niveaux. Il permet de gérer par exemple depuis une fiche actualité, les catégories, sous catégories ou sous sous catégories par un système arborescent.

Nom SQL = Nom du champ pour le formulaire (ce n'est pas un champ SQL)

Titre = Libellé explicatif pour affichage

Type = liste-multiple

Présent accueil = 0 (Non implémenté pour ce type)

Tableau 1 = Tableau contenant la table de niveau 1 (catégorie), la clé et le libellé

Jointure sur tableau 1 = Tableau contenant la table de jointure de niveau 1 (entre actualité et catégorie), la clé de la table de niveau 1 et la clé de table principale (actualité)

Tableau 2 (optionnel) = Tableau contenant la table de niveau 2 (sous catégorie), la clé et le libellé

Jointure sur tableau 2 = Tableau contenant la table de jointure de niveau 2 (entre actualité et sous catégorie), la clé de la table de niveau 2 et la clé de table principale (actualité)

Tableau 3 (optionnel) = Tableau contenant la table de niveau 3 (sous sous catégorie), la clé et le libellé

Jointure sur tableau 3 = Tableau contenant la table de jointure de niveau 3 (entre actualité et sous sous catégorie), la clé de la table de niveau 3 et la clé de table principale (actualité)

Formatage (optionnel) = Hauteur en pixel de la zone de saisie

Exemple d'utilisation pour une jointure sur 3 niveaux

```
array("categorie","Catégorie","liste-multiple",0,
      array("categorie","id_categorie","libelle_categorie"),
      array("actu_categorie_jn","id_categorie","id_actualites"),

      array("sous_categorie","id_sous_categorie","libelle_sous_categorie"),
      array("actu_sous_categorie_jn","id_sous_categorie","id_actualites"),

      array("sous_sous_categorie","id_sous_sous_categorie",
            "libelle_sous_sous_categorie"),
      array("actu_sous_sous_categorie_jn","id_sous_sous_categorie",
            "id_actualites"),

      "200px"
)
```

Exemple d'utilisation pour une jointure simple avec une hauteur de 200 pixels maximum

```
array("categorie","Catégorie","liste-multiple",0,
      array("categorie","id_categorie","libelle_categorie"),
      array("actu_categorie_jn","id_categorie","id_actualites"),

      "200px"
)
```

Exemple d'utilisation pour une jointure simple sans limite de hauteur

```
array("categorie","Catégorie","liste-multiple",0,
      array("categorie","id_categorie","libelle_categorie"),
      array("actu_categorie_jn","id_categorie","id_actualites")
)
```

Exemple d'utilisation pour une jointure sur 2 niveaux

```
array("categorie","Catégorie","liste-multiple",0,
      array("categorie","id_categorie","libelle_categorie"),
      array("actu_categorie_jn","id_categorie","id_actualites"),

      array("sous_categorie","id_sous_categorie","libelle_sous_categorie"),
      array("actu_sous_categorie_jn","id_sous_categorie","id_actualites"),
)
```

Code SQL correspondant aux exemples

```
CREATE TABLE actualites
(
  id_actualites bigint NOT NULL auto_increment,
  titre text DEFAULT "",
  corps text DEFAULT "",
  PRIMARY KEY (id_actualites)
);

CREATE TABLE categorie
(
  id_categorie bigint NOT NULL auto_increment,
  libelle_categorie varchar(255) DEFAULT "",
  PRIMARY KEY (id_categorie)
);

CREATE TABLE sous_categorie
(
  id_sous_categorie bigint NOT NULL auto_increment,
  id_categorie bigint DEFAULT "",
  libelle_sous_categorie varchar(255) DEFAULT "",
  PRIMARY KEY (id_sous_categorie)
);

CREATE TABLE sous_sous_categorie
(
  id_sous_sous_categorie bigint NOT NULL auto_increment,
  id_sous_categorie bigint DEFAULT "",
  libelle_sous_sous_categorie varchar(255) DEFAULT "",
  PRIMARY KEY (id_sous_sous_categorie)
);

CREATE TABLE actu_categorie_jn
(
  id_actu_categorie_jn bigint NOT NULL auto_increment,
  id_categorie bigint DEFAULT "",
  id_actualites bigint DEFAULT "",
  PRIMARY KEY (id_actu_categorie_jn)
```

```

);

CREATE TABLE actu_sous_categorie_jn
(
  id_actu_sous_categorie_jn bigint NOT NULL auto_increment,
  id_sous_categorie bigint DEFAULT "",
  id_actualites bigint DEFAULT "",
  PRIMARY KEY (id_actu_sous_categorie_jn)
);

CREATE TABLE actu_sous_sous_categorie_jn
(
  id_actu_sous_sous_categorie_jn bigint NOT NULL auto_increment,
  id_sous_sous_categorie bigint DEFAULT "",
  id_actualites bigint DEFAULT "",
  PRIMARY KEY (id_actu_sous_sous_categorie_jn)
);

```

Type google_map

Ce type est utilisé pour ajouter un outil cartographique basé sur Google MAP.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Titre = Libellé explicatif pour affichage

Type = google_map

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Centre = Coordonnées X,Y du centre par défaut de la carte

Largeur = Largeur de la carte

Hauteur = Hauteur de la carte

Zoom = Niveau par défaut du zoom, valeur comprise entre 0 et 17

Champ X = Nom du champ ou enregistrer la latitude, de type FLOAT(10,6)

Champ Y = Nom du champ ou enregistrer la longitude, de type FLOAT(10,6)

Clé Google MAP = Numéro de la clé fournie par Google à l'adresse suivante :

<http://www.google.com/apis/maps/signup.html>

Exemple d'utilisation

```
array("", "Localisation", "google_map", 0, "48.8737479930069, 2.2950267791748047",
"100%", "400", "10", "latitude", "longitude", "Clé Google MAP"),
```

```
array("latitude", "Latitude", "text", 0),
array("longitude", "Longitude", "text", 0),
```

Type flash

Ce type est utilisé pour ajouter un objet FLASH.

Champ vide = Doit comprendre uniquement une chaîne vide ""

Titre = Libellé explicatif pour affichage

Type = flash

Présent accueil = 0 non présent ou 1 présent en page d'accueil

Fichier = Nom du fichier SWF

Couleur de fond = Couleur de fond (blanc)

Largeur = Largeur de l'objet FLASH

Hauteur = Hauteur de l'objet FLASH

Code Javascript = Code Javascript appelé éventuellement par le FLASH

Exemple d'utilisation

```
array("", "Carte", "flash", 0, "../media/
cartographie_otivh_admin.swf?version=100&id_courant=", "#FFFFFF", "500", "500",
"function savepos(x,y) { document.formulaire.Posx.value = x;
document.formulaire.Posy.value = y; } ");

array("Posx", "POS X", "text", 0),
array("Posy", "POS Y", "text", 0),
```

3. API « Moteur de recherche »

Pour mettre en place un moteur de recherche, il suffit d'instancier l'objet depuis une page dynamique contenant le code suivant (disponible depuis la librairie) :

```
<?php
include_once("admin/db.inc.php");
include_once("admin/lib_search.inc.php");

$connection = dbConnect();

$search = new isens_search($connection, "{version}", $_GET["mot"], 10);
?>
```

Les paramètres de la classe `isens_search` sont :

Connection = Connection vers la base de données

Version = Version courante, le tag `{version}` est remplacée par `isens` à la publication

Mot = Mot passé en paramètre, POST ou GET

Nombre = Nombre de réponse par page

Stopwords = Tableau contenant des stopwords spécifiques à un domaine

Global = True pour une recherche globale ou false pour une recherche dans la version courante uniquement