

## 1. Introduction

ACM (a card CMS) est une mini-CMS sans base de données, orienté fiche ou carte. Il puise sa lointaine inspiration dans HyperCard, et il emprunte quelques concepts à la POO (programmation orienté objet).

L'idée de base est la suivante : on manipule des données sous forme de fiche et de liste. Suite à une recherche, on obtient une liste de réponse, cette liste affichant quelques champs (données) choisis. Puis quand on clique sur un élément de cette liste, on affiche alors une fiche contenant en général toutes les données.

Certains champs peuvent donner accès à leur tour à une autre liste ou une autre fiche, cela de façon la plus simple possible, sans devoir recourir à trop de code.

Les enregistrements (groupes de champs) sont répartis en plusieurs types (familles, catégories). A chaque type correspond 2 types d'affichage : Liste et Fiche. Par exemple, les dessinateurs sont un type de données. Les BD sont un autre type de données. Il va de soi que le contenu de la fiche d'un dessinateur n'est pas identique à celui de la fiche d'une BD. Néanmoins, il est possible d'afficher la fiche du dessinateur d'une BD. De même, il est possible de lister les BD d'un dessinateur.

Ce présent manuel est beaucoup plus un mémento qu'autre chose.

## 2. Data.txt

Le contenu du site est stocké dans le fichier data.txt

Toutes les fiches sont identifiées par un identifiant, un index (\$IDX)  
En dehors des identifiants de fiche, toutes les données du fichier data.txt obéissent au couple **champ:valeur**.

Les champs liés au système interne sont en majuscules et sur 3 caractères.

Un identifiant de fiche, qui est aussi un indicateur de début de fiche commence toujours par 3 tirets suivis du numéro d'identifiant (ou d'index). Ex : **---123**

### Systeme – 0

---

---0	Fiche système dont l'identifiant est 0 Cette fiche représente le site lui-même, c'est donc ici que tout ce qui concerne la globalité du site est stocké.
<b>TTL</b>	Titre du site
<b>TPL</b>	Template par défaut du site
<b>IDX</b>	Identifiant de la fiche à afficher par défaut au lancement du site

### Requête – 10 à 98

---

---12	Requête dont l'identifiant est 12 Les identifiants de requête ont 2 chiffres et commencent à 10. Une requête est une recherche dont le résultat est une liste. 🚫 Ne pas utiliser l'identifiant 99 qui est réservé à un usage interne
<b>WHR:xxx</b>	Condition "Where" de la requête ✍ WHR:getField('TYP')==1 && getField('dessinateur')==301 <i>le type est égal à 1 et l'identifiant du dessinateur est égal à 301</i>
<b>ORD:sxxx</b>	Ordre de tri. +champ : trier par ordre croissant selon le champ -champ : trier par ordre décroissant selon le champ ✍ ORD:+TTL → trier les titres de A à Z
<b>TYP:nnn</b>	Type de template utilisé pour lister les résultats de la requête Ce numéro de type est utilisé par les templates 2 types de fichier : single# et list# (# remplace ici un chiffre) single1.php : template de simple fiche et de type 1 list3.php : template de liste de fiches et de type 3

## **Fiche – 100 à 998**

---

---123	Fiche dont l'identifiant est 123. Les identifiants de fiche ont 3 chiffres et commencent à 100 ●* Ne pas utiliser l'identifiant 999 qui est réservé à un usage interne
<b>TTL:xxx</b>	Titre (title) de la fiche
<b>TYP:nnn</b>	Type de la fiche et donc type de template (un chiffre)
<b>DSC:xxx</b>	Description
<b>DTC:xxx</b>	Date de création de la fiche (date create)
<b>DTU:xxx</b>	Date de mise à jour de la fiche (date update)
<b>XTD:nnn</b>	Parent de la fiche Le terme "extends" désigne le parent d'un objet dans divers langages informatiques. Par exemple, l'appel de la fonction getField('monchamp') va d'abord rechercher la valeur de 'monchamp' dans la fiche en cours. Si cette valeur n'existe pas, cette recherche se fait alors dans le parent de la fiche en cours grâce à XTD, et ainsi de suite.

## 3. Templates

L'affichage des informations est déléguée à des templates.

A un type de fiche (TYP) correspond un template "single" et un template "list"

Les fonctions `getLink()`, `getVal(...)` et `getField(...)` sont utilisables dans ce template

### ***page.php***

---

Ce template affiche la page elle-même, l'armature générale du site.

La fonction `showContent()` affiche la fiche ou la liste de fiches désirée (ne pas utiliser `echo` devant le nom de la fonction)

### ***single#.php***

---

Ce template affiche le contenu d'une simple fiche avec la fonction `getField()` pour chaque champ.

Ce template est un sous-template du template "page"

Ex : Le fichier `single5.php` est le template qui affiche les fiches de type 5

### ***list#.php***

---

Ce template affiche la liste de différentes fiches suite à une requête.

Ce template est un sous-template du template "page"

Ex : Le fichier `list4.php` est le template qui affiche les fiches de type 4

## 4. PHP - Variables et Fonctions

<b>\$DTA</b>	Tableau associatif issu du fichier data.txt
<b>\$TPL</b>	Template, équivalent à : <code>getVal(0,'TPL')</code>
<b>\$IDX</b>	Identifiant de la fiche en cours
<b>\$IDXARR</b>	Tableau associatif contenant le résultat d'une requête <code>\$IDXARR[\$idt]=\$valeur_du_champ_à_trier</code>

### **getField(\$champ)**

Retourne la valeur du champ de la fiche en cours  
 Equivalent à : `getVal($IDX,$champ)`  
 ✎ `echo getField('TTL')` → titre de la fiche en cours

### **getField(\$champ,\$indirect)**

Retourne la valeur du champ indirect  
 Equivalent à : `getVal(getField($champ),$indirect)`  
 ✎ `echo getField('dessinateur','TTL')` → titre de la fiche du dessinateur

### **getLink(\$texte,\$idt)**

### **getLink(\$texte,\$idt,\$commentaire)**

Retourne le lien ainsi créé  
 ✎ `echo getLink('Fiche 456',456,'commentaire sur la fiche 456')`  
 ❖ Un lien est obligatoirement un numéro d'index  
 ✎ `index.php?456` → appelle la fiche n°456  
 ✎ `index.php?23` → appelle la requête/liste n°23

### **getVal(\$idt,\$champ)**

Retourne la valeur associée au champs dans la fiche identifiée par l'idt  
 ✎ `echo getVal(0,'TTL')` → titre du site

### **setReq(\$req,\$ordretri,\$type)**

Programme la requête n°99 afin de personnaliser une demande de liste  
 ✎ `setReq("getField('TYP')==4 && getField('client')==500",' +TTL',4)`  
*Lister toutes les fiches de type 4 ayant pour client le n°500 par ordre croissant de titre et avec le template n°4*

### **showContent()**

Affiche la fiche ou la liste des fiche  
 Cette fonction est obligatoirement appelée dans le template 'page'  
 Il est possible de lancer une requête dans une simple fiche :  
 ✎ `setReq("getField('TYP')==4 && getField('client')==500",' +TTL',4);`  
 ✎ `showContent();`

## 5. Plugins - Variables

On peut afficher dans le corps du texte les diverses variables contenues dans le CMS, sans devoir passer forcément par un plugin fonction (voir section suivante)

3 syntaxes sont possibles :

### ***[[idx:champ]]***

---

Affiche le champ lié à l'index

-  `[[0:TTL]]` → affiche le titre du CMS
-  `[[123:DTU]]` → affiche la date de mise à jour de la fiche n°123

### ***[[ -1:champ]]***

---

Affiche le champ de la fiche en cours

-  `[[ -1:TTL]]` → affiche le titre de la fiche actuelle

### ***[[ -2:champ1, champ2]]***

---

Affiche le champ2 indirect lié au champ1 de la fiche en cours

-  `[[ -2:ville, TTL]]` → affiche le titre de la fiche de la ville de la présente fiche

### ***Appel du plugin variable***

---

C'est la fonction `getReg($txt)` qui s'occupe de la détection puis de l'appel de tous les types de plugin.

Cette fonction doit être appelée explicitement :

-  `echo getReg(getField('TXT'));`

## 6. Plugins - Fonctions

Les plugins liés aux fonctions sont mis dans le dossier "plugins"

Chaque fonction possède son propre dossier dont le nom doit être identique au nom de la fonction, nom de la fonction dont on ôte le préfixe PLUGIN.

Dans le dossier, se trouve le fichier index.php qui contient la ou les fonctions liées au plugin.

Le nom d'une fonction plugin doit commencer par PLUGIN (en majuscules)

<b>Plugin</b>	mafonction
<b>Dossier</b>	plugins/mafonction
<b>Fichier</b>	plugins/mafonction/index.php
<b>Code PHP</b>	function PLUGINmafonction(\$arg){...}

La fonction doit renvoyer un flux HTML.

### ***Programmation dans le fichier index.php***

---

```
function PLUGINminigal($args){
    $folder=$args[0];           // 1er argument de la fonction
    $title=$args[1];           // 2ème argument, et ainsi de suite...
    ...
    return $html;
}
```

\$args est un tableau qui contient les différents arguments de la fonction.

La fonction doit retourner un contenu html.

### ***Appel du plugin***

---

Soit directement à partir de la fonction :

```
 echo PLUGINminigal(array('images','Photos'));
```

Soit à partir du fichier data.txt avec un # devant le nom de la fonction :

```
 ... #PLUGINminigal(images,Photos) ...
```

*Ne pas encadrer les textes par des guillemets ou des apostrophes*

C'est la fonction getReg(\$txt) qui s'occupe de la détection puis de l'appel des plugins.

Cette fonction doit être appelée explicitement :

```
 echo getReg(getField('TXT'));
```