[系统架构 2](#_Toc266533854)

[一基本概念 2](#_Toc266533855)

[1.1 2](#_Toc266533856)

[1.2 2](#_Toc266533857)

[二架构设计 2](#_Toc266533858)

[2.1系统特性 2](#_Toc266533859)

[2.2目录结构 2](#_Toc266533860)

[2.2.1目录结构 2](#_Toc266533861)

[2.2.2重要文件说明 3](#_Toc266533862)

[2.3 MVC分层 4](#_Toc266533863)

[2.4执行流程 4](#_Toc266533864)

[2.5命名规范 4](#_Toc266533865)

[2.5.1类文件 4](#_Toc266533866)

[2.5.2函数 4](#_Toc266533867)

[2.5.3配置文件 5](#_Toc266533868)

[2.5.4内部用函数 5](#_Toc266533869)

[2.5.5变量名 5](#_Toc266533870)

[2.5.6数组 5](#_Toc266533871)

[2.5.7类对象存储方式 5](#_Toc266533872)

[2.6入口文件 5](#_Toc266533873)

[2.6.1.后台入口 5](#_Toc266533874)

[2.6.2后台程序文件(类似xxx\_admin.php) 5](#_Toc266533875)

[2.6.3前台文件规范 5](#_Toc266533876)

[2.7程序初始化 6](#_Toc266533877)

[2.7.1加载初始化文件 6](#_Toc266533878)

[2.7.6加载模块和插件配置 6](#_Toc266533879)

[2.8程序url访问 6](#_Toc266533880)

[2.8.1后台地址admincp.php 6](#_Toc266533881)

[2.9控制器( control)与模块(module) 6](#_Toc266533882)

[2.9.1前台和后台 6](#_Toc266533883)

[2.9.2模块 6](#_Toc266533884)

[2.10模板引擎 6](#_Toc266533885)

[2.10.1模板查找顺序 6](#_Toc266533886)

[2.10.2调用范例 7](#_Toc266533887)

[2.10.3基本标签调用 7](#_Toc266533888)

[三.构建应用 7](#_Toc266533889)

[四.开发指南 7](#_Toc266533890)

[4.1基本要求 7](#_Toc266533891)

[4.1.2菜单定义 7](#_Toc266533892)

[4.2模块开发 8](#_Toc266533893)

[4.2.1文件要求 8](#_Toc266533894)

[4.2.2目录要求 8](#_Toc266533895)

[4.2.3范例配置文件 9](#_Toc266533896)

[4.3插件开发 10](#_Toc266533897)

[五.模板引擎 10](#_Toc266533898)

[5.1基本功能 10](#_Toc266533899)

[5.2注意事项 10](#_Toc266533900)

[六.开发技巧 11](#_Toc266533901)

[6.1后台开发 11](#_Toc266533902)

[6.2目录绑定子域名 11](#_Toc266533903)

# 系统架构

一基本概念

1.1单一入口后台程序统一入口admincp.php

1.2MVC采用类discuz的模板引擎进行简单mvc分离

二架构设计

## 2.1系统特性

a.整个系统采用模块概念划分,不同模块后台管理在自己目录内.

b.模块插件安装卸载简单只需要一个install.php/unstall.php文件

c.权限管理.系统实现的权限管理采用mod,action,op来区分不同权限,无需每个文件加对应标识来区分不同权限

d. 数据调用,自定义位置.可自写sql来增加数据调用模块,生成缓存,且数据调用模板可用任何现有模板标签(此处不同于现有大部分cms,现在大部分都是只能用规定的几个标签,然后采用替换方式实现)

e.模板引擎修改的uchome的,优化过，调用方式稍有变化.

当程序模块多时不同目录绑定不同域名是更加容易实现

f.数据库读写分离,缓存方式：memcache和普通文件缓存切换

g.程序配置采用继承覆盖方式，例如模板查找起始路径，不定义自动找/template/default/下定义了就按对应目录找，找不到在找默认目录

## 2.2目录结构

后台模块定义:模块(mod)默认为admin,动作(ac),操作(op),后台admin目录也是一个模块

### 2.2.1目录结构

admin/ 管理后台核心程序

data/ 缓存目录

data/backup 程序用保存日志等

data/cache\_data 插件配置,网站配置文件

data/plugin 插件自己用的缓存目录,下边的目录按插件标识创建

data/tpl\_cache 模板缓存目录,文件规律为模板所在目录把路径中的”/”替换成”\_”

distribfile/ 程序用的图片,和js等,可以分发到其他服务器

include/ 程序核心文件

language/ 语言包文件,暂时没用到

member/ 会员模块

member/admin 会员相关后台管理程序

member/admin/template 会员后台管理程序模板,具体请看系统架构中模块文档

plug/ 插件目录,此目录下的目录名为对应的各个插件

public/ 网站本身用的公共文件,例如web编辑器,js库等

template/ 网站前台模板文件目录,

admincp.php 后台入口文件

config.php 网站数据库配置等信息

editor\_upload.php 编辑器用来上传图片的代码

### 2.2.2重要文件说明

**include/目录**

cache.class.php 缓存类,可切换memcache或一般文件缓存,具体看代码

common.inc.php 网站公共包含文件,包括初始化程序,进行的操作有

创建数据库对象,缓存对象

读取网站cookie

判断用户是否登录,登录则把用户信息读取存储到公共变量中

获取当前操作的url……

function\_cache.php 缓存更新函数包括配置缓存,插件缓存等

function\_common.php 网站公共文件

function\_member.php 判断会员登录和更新信息相关函数

function\_template.php 模板引擎相关函数

mysql.class.php 数据库链接类,可实现读写分离

tools/ 工具类和函数,例如图片上传处理类

**admin/目录**

include/ 后台用到的函数库

include/function\_admin.php 后台判断登录,权限检查

images/ 后台用到的图片

js/ 后台用到的js库

template/ 后台程序模板目录

admin\_admin.php 后台管理员相关操作,管理员列表,权限分配等

config\_admin.php 网站配置相关

datacall\_admin.php 自定义调用数据相关操作,

help\_admin.php 显示数据库表结构等

main\_admin.php 后台登录界面流程控制模块

menuinfo.php 菜单配置文件

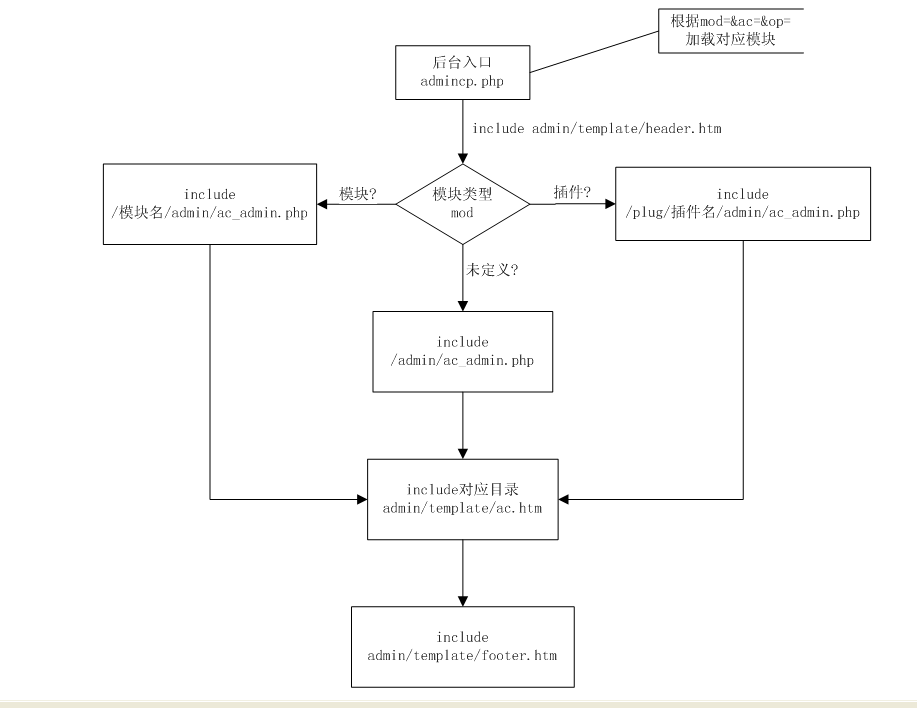
modules\_admin.php 模块和插件安装卸载加载相关

upfile\_admin.php 上传文件管理

## 2.3 MVC分层

程序中 采用简单的mvc分层架构,view层指的是template/目录下文件.control一般是admincp.php或对应action的一个文件,然后采用opertion区分不同操作

## 2.4执行流程



## 2.5命名规范

2.5.1类文件

类文件以 .class.php结尾内部类名和文件名一致例如mysql.class.php

2.5.2函数

函数按功能以function\_xxx.php结尾,公共函数文件为function\_common.php存放目录为对应模块下的include目录,分为系统公共函数文件和模块的公共函数文件,函数名全部小写,且用有意义的单词

2.5.3配置文件

程序配置文件cofnig.php,各种配置请存储到$\_SCONFIG数组中,例如:$\_SCONFIG[‘dbhost’]=’locahost’;

2.5.4内部用函数

一批同功能函数中若有些函数不提供给外部调用以\_开头,例如function \_init()

2.5.5变量名

变量统一小写,中间用”\_”分隔,但尽量不要出现超过三个”\_”,用有意义的单词或缩写命名

例如$username

2.5.6数组

数组以”arr”结尾.例如$userlistarr

2.5.7类对象存储方式

类对象尽量存储到$\_SGLOBAL全局变量中例如$\_SGLOBAL[‘db’],$\_SGLOBAL[‘cache’];或以”obj”开头例如$obj\_sql = new mysql();

## 2.6入口文件

### 2.6.1.后台入口

admincp.php?mod=xxx&ac=xxx&op=xxx

代表mod=模块,ac=对应模块下文件(mod/ac\_admin.php),op=对应操作(例如op=loglist)

### 2.6.2后台程序文件(类似xxx\_admin.php)

原则上一类功能都在一个文件里用op区分例如log\_admin.php是与日志相关的功能,若觉得文件可能太大,可自己在写成另一个.

### 2.6.3前台文件规范

同样的功能尽量写道一个文件里,然后采用标识区分可参考uchome的方式

## 2.7程序初始化

2.7.1加载初始化文件

a.所有程序加载include/common.inc.php进行系统初始化

操作包括, 对post,get,数据过滤,针对后台前台的sql语句过滤,包含功能函数文件.

b.初始化$\_SGLOBAL,$\_SCONFIG,$\_TPL, $\_SCOOKIE数组

c.读取本站所有cookie存储到$\_SCOOKIE中,初始化数据库对象$\_SGLOBAL[‘db’]

初始化缓存对象$\_SGLOBAL[‘cache’].

d.判断用户是否登录,初始化此变量$\_SGLOBAL[‘super\_uid’],若登录获得用户信息存储到$\_SGLOBAL['member']中,代码在include/function\_member.php中

e.获得用户当前操作url,存储到$\_SGLOBAL[‘operate\_nowurl’],获得用户ip

### 2.7.6加载模块和插件配置

## 2.8程序url访问

### 2.8.1后台地址admincp.php

## 2.9控制器( control)与模块(module)

2.9.1前台和后台

控制器一种为admincp.php根据mod和ac分发(dispatch ),另一个就是mod/admin/xxx\_admin.php(根据op区分不同操作)

前台尽量也用单一入口方式通过mod和op区分操作

### 2.9.2模块

对应模块调用自己目录mod/include/下的函数库和.class类文件

## 2.10模板引擎

### 2.10.1模板查找顺序

1. 自定义模板路径/template/$\_TPL[‘defaulttmp’]/;
2. 默认模板路径/template/default/
3. 自定义模板查找起始文件夹WEB\_ROOT.$\_TPL[‘tpldirbase’]/;
4. 自定义查找目录未找到时缺省目录WEB\_ROOT.$\_TPL[‘basedefault’]/;

后边设置会覆盖前边

### 2.10.2调用范例

方法一:

<?

$a=’hello world’;

include template(‘index’);//此时加载的是/template/default/index.htm

?>

方法二(可参加后台或程序模块的开发方式)

$\_TPL[‘tpldirbase’] = ‘member/weber/template/’;

include\_once template(‘index’);//此时加载的是/member/weber/template/index.htm

### 2.10.3基本标签调用

此模板引擎修改自uchome,语法除模板调用和日期调用外基本相同

标签都以<!--{ }-->开头结尾

三.构建应用

四.开发指南

## 4.1基本要求

不管模块还是插件尽量采用单一入口模式.

同样功能都放到一个文件里利用op=xxx区分

模块或插件管理后台文件命名xxx\_admin.php 例如admin/memberlist\_admin.php

模块或插件管理后台模板命名xxx.htm例如admin/template/memberlist.htm

### 4.1.2菜单定义

二维数组定义菜单

array(

‘title’=>’一级菜单名’,

‘模块标识|模块名|action|operate’,

)

其中action 对应此模块下的action\_admin.php



$arr\_menu['sysadmin'] = array(

array(

'title' => '网站工具',

'admin|更新缓存|help|updatecache',

'admin|文件管理|upfile|filelist'

),

array(

'title'=>'管理员',

'admin|管理员列表|admin|adminlist',

'admin|用户组列表|admin|usergroup',

'admin|动作列表|admin|actionlist',

'admin|修改密码|admin|updatepwd',

),

);

## 4.2模块开发

a.模块和插件分核心和非核心,核心模块不允许删除

b.表前缀用#@\_\_代替例如#@\_\_member 数据表以模块标识开头例如#@\_\_member\_adlist

c.模块/插件管理后台文件要求

mod/admin/config.php若存在则加载模块后台管理文件时则会自动加载

d.开发范例参考/member，插件参考/plus/board

### 4.2.1文件要求

必须文件:

module/install.php 安装文件

module/unstall.php 卸载文件

### 4.2.2目录要求

模块加载起始路径为网站根目录

例如模块文件夹为member则对应目录为ROOT/member/

后台管理目录和名称module/admin/

后台模板文件 module/admin/template/

其他目录不做要求

### 4.2.3范例配置文件

<?

//插件安装文件install.php

!defined('IN\_ADMIN') && exit('Access Denied');//防止直接打开或恶意包含执行

$mconfigarr = array(

'name'=>'公告板',//模块名

'introduce'=>'简单显示公告',// 介绍

'author'=>'bobboy007',// 作者

'version'=>'1.0',// 版本

'moduleconfig'=>array(//模块默认配置信息

),

'menuconfig'=>array(//后台菜单

'title'=>'公告板',

'board|公告列表|boardlist',

'board|添加公告|boardlist|addboard'

),

);

//需要运行的sql,表前缀用#@\_\_代替

$sqlbatch = <<<EOT

CREATE TABLE `#@\_\_plug\_board` (

`id` INT( 11 ) NOT NULL AUTO\_INCREMENT PRIMARY KEY ,

`title` VARCHAR( 250 ) NULL ,

`content` TEXT NULL ,

`dt` INT( 11 ) NOT NULL DEFAULT '0',

`adminid` INT( 11 ) NOT NULL DEFAULT '0'

) ENGINE = MYISAM ;

EOT;

runquery($sqlbatch); //执行批量sql语句,语句中用;分隔

?>

插件卸载文件unstall.php

<?

!defined('IN\_ADMIN') && exit('Access Denied');//防止恶意包含或执行

$sqlbatch = <<<EOT

DROP TABLE `#@\_\_member\_board`;

EOT;

runquery($sqlbatch);//执行批量sql语句,语句中用;分隔

?>

## 4.3插件开发

起始目录root/plug/

数据表都以plug开头例如#@\_\_plug\_friendlink

后台目录,模板文件与模块开发规范一样

五.模板引擎

## 5.1基本功能

修改自uchome的模板引擎,但内部代码有所改动,调用方式基本没变.

改动:

1. 包含子模板方式,uchome原来方式是用file\_get\_contents抓取子模板,这函数稳定性差,故本次舍弃换用其他方式解析子模板.
2. 日期调用方式date
3. 自定义数据调用datacall

## 5.2注意事项

a.模板嵌套, 最多允许两层嵌套.及例如index.htm包含header.htm此文件还可包含其他模板b.htm,但此时b.htm不允许再包含其他模板

b.foreach循环嵌套最多允许6层

c.子模板(例如header.htm)修改后需要删除模板缓存,其他模板修改后会自动更新

d.数据调用模块,可使用任何模板标签和foreach循环等.

六.开发技巧

## 6.1后台开发

a.后台表单定义采用二维数组例如

<input type=’text’ name=’userinfo[username]’ value=’$userinfo[username]’/>

b.程序用id标识区分是添加还是编辑数据,因编辑数据时id必定大于0

<input type=’text’ name=’userid’ value=’$userinfo[uid]’/>

c.通过submit的name属性,区分不同表单

<?

if(submitcheck(‘userinfosubmit’)){

if($\_POST[‘userid’]>0){//更新用户信息

updatetable(‘member\_info’, $\_POST[‘userinfo’]’, id=’.$\_POST[‘userid’]);

}else{//添加用户信息

inserttable(‘member\_info’, $\_POST[‘userinfo’]’);

}

}

?>

d.用cookie保存当前操作url,例如当显示完列表时,setcookie(‘nowurl’);

用途:当编辑或删除操作成功时,读取cookie中的url,返回用户上次操作页面

e.可看下系统管理菜单里边的功能,基本包含了整个程序的开发风格

## 6.2目录绑定子域名

程序架构完全可以在一个子目录中实现单独一个站点,和单独的管理后台,只需要定义模板起始加载路径即可