

#### A. 入口文件

1. 新建个 makeConfig.php 文件,放在 joomla/文件夹下
2. 修改 joomla/index.php 文件,使其对域名做出判断,加入如下代码

```
require_once ( JPATH_BASE .DS.'makeConfig.php');
$makeConfig = new makeConfig();

if(!file_exists("./config/".$makeConfig->serverName().'_config.php'))
{
    if(file_exists("./installation_sites/index.php"))
    {
        header('Location: installation_sites/index.php');
        exit();
    }else{
        echo "No ".$makeConfig->serverName()."_config.php Exiting.....";
        exit();
    }
}
```

这样会根据域名查抄配置文件,如果不存在相应的配置文件就会进入安装配置界面

#### B. 安装文件

1. 复制 joomla 自带的安装文件夹,改名 installation\_sites/
2. 修改 installation\_sites/includes/framework.php

把原来的 if(file\_exists(...))注释掉,加入新的

```
require_once ( JPATH_ROOT.DS.'makeConfig.php');
$makeConfig = new makeConfig();
if (file_exists( JPATH_ROOT.DS."config".DS.$makeConfig->serverName()."_config.php" ) &&
(filesize(JPATH_ROOT.DS."config".DS.$makeConfig->serverName()."_config.php")>10))
{
    header('Location: ../index.php');
    exit();
}
```

3. 修改 installation\_sites/includes/defines.php

```
Define('JPATH_INSTALLATION', JPATH_ROOT.DS.'installation_sites');
```

4. 修改 installation\_sites/installer/installer.php

```
require_once ( JPATH_ROOT.DS.'makeConfig.php');
$makeConfig = new makeConfig();
if (file_exists( JPATH_ROOT.DS."config".DS.$makeConfig->serverName()."_config.php" )
&& (filesize(JPATH_ROOT.DS."config".DS.$makeConfig->serverName()."_config.php")>10) &&
file_exists( JPATH_INSTALLATION . DS . 'index.php' )) {
    $config['default_task'] = 'removedir';
} else {
    $config['default_task'] = 'lang';
}
```

5. Joomla/libraries/joomla/application/application.php

在\_\_construct(\$config = array())函数里,加入  
require\_once(JPATH\_ROOT.DS.'makeConfig.php')  
\$make = new makeConfig();  
Lines93 \$config['config\_file'] = 'configuration.php';修改为  
\$config['config\_file'] = \$makeConfig->serverName()." \_config.php";

Lines98 \$this->\_createConfiguration(JPATH\_CONFIGURATION.DS.\$config['config\_file']);修改为  
\$this->\_createConfiguration(JPATH\_ROOT.DS.'config'.DS.\$config['config\_file']);

6. Installation\_sites/installer/models/model.php

修改 function preInstall 函数,加入  
require\_once(JPATH\_ROOT.DS.'makeConfig.php');  
\$makeConfig = new makeConfig();  
修改\$cW 变量  
\$cW = (@ file\_exists("../config/".\$makeConfig->serverName()." \_config.php") && @  
is\_writable("../config/".\$makeConfig->serverName()." \_config.php")) || is\_writable('../config/'  
);

```
$phpOptions[] = array (  
    //'label' => 'configuration.php ' . JText::_('writable'),  
    'label' => $makeConfig->serverName()." _config.php" . JText::_('writable'),  
    'state' => $cW ? 'Yes' : 'No',  
    'notice' => $cW ? '' : JText::_('NOTICEYOUCANSTILLINSTALL')  
);
```

7. installation\_sites/installer/models/model.php

修改 function saveConfig()函数  
\$path = JPATH\_CONFIGURATION.DS.'configuration.php';改为  
\$path = JPATH\_ROOT.DS.'config'.DS.\$makeConfig->serverName()." \_config.php";

8. 因为会引起类的重定义问题所以必须要修改 libraries/joomla/application/application.php

修改 function &\_createConfiguration(\$file)函数,加入  
require\_once(JPATH\_ROOT.DS.'makeConfig.php');  
\$makeConfig = new makeConfig();  
\$className = \$makeConfig->serverName()."JConfig";

\$config = new JConfig();改为\$config = new \$className();

9. 再生成类的时候需要修改类名称这样配合第 8 点的修改

在 function saveConfig()函数里加入

\$tmpl->addVar('configuration', 'VAR\_CLASSNAME', \$makeConfig->serverName()."JConfig");

这样生成 config 文件的时候类名就匹配域名了,不会发生重定义类的问题了

10. 在完成配置文件后可以直接进后台,但是修改后的链接有问题,要修改  
Installation\_sites/installer/models/model.php  
在 function finish()函数里  
\$vars['siteurl'] = JURI::root();改为\$vars['siteurl'] = str\_replace("\_sites", "", JURI::root());
11. 在进入安装页面的时候会自动创建文件夹的方法:  
Installation\_sites/installer/models/model.php 文件  
Require\_once(JPATH\_ROOT.DS.'makeConfig.php');修改 Function finish()函数

#### C. 配置演示

1. 直接删除 config/ 对应\_config.php 文件即可
2. 删除数据库
3. 从入口文件进入,就会重新配置

#### D. 关于媒体文件

在前面的配置文件的时候最好加如代码使其在 images 文件夹下生产对应的域名文件夹,这样后期就不用手动创建文件,媒体库就不会产生警告!

1. /administrator/components/com\_media/models/list.php  
Function getList 函数是取得当前文件夹下的所有文件包括文件夹  
修改:  
在开头加入 require\_once ( JPATH\_ROOT .DS.'makeConfig.php');这样就可以对域名进行分析了  
Lines:96 \$basePath = COM\_MEDIA\_BASE;改为  
\$basePath = COM\_MEDIA\_BASE.DS.\$makeConfig->serverName();
2. /administrator/components/com\_media/models/manager.php  
Function getFolderTree 函数是生成目录树的函数供 js 使用  
修改:  
在函数开头加入 require\_once(JPATH\_ROOT.DS.'makeConfig.php');  
Lines:97 \$base = COM\_MEDIA\_BASE;改为  
\$base = COM\_MEDIA\_BASE.DS.\$makeConfig->serverName();
3. /administrator/components/com\_media/assets/mediamanager.js  
这个 js 文件很重要包括了创建文件夹删除文件夹等功能  
创建文件夹的时候要注意 class="update-folder"这个 input  
修改:  
Onloadframe:function()这个函数是得到当前文件夹路径以及创建文件夹的函数  
Lines:66  
This.updatepaths.each(function(path){ path.value="";});改为  
this.updatepaths.each(function(path){path.value=window.location.host.substring('0',  
window.location.host.indexOf('.');});这样就可以根据域名创建文件夹  
  
lines67  
this.folderpath.value = basepath;改为  
this.folderpath.value=basepath+'/' +window.location.host.substring('0',  
window.location.host.indexOf('.'));}

4. 添加文章的时候从媒体库取出图片的路径设置,对应的是  
administrator/components/com\_media/views/images/
5. 主要的路径设置变量全部在 administrator/components/com\_media/models/文件夹下  
List.php 文件是把文件或者文件夹图标以列表的形式显示出来,如果路径设置错误会显示  
jfolder,jfile is not a file or a directory,如果是列表文件显示不出来就修改这个文件

文件夹管理文件在 administrator/components/com\_media/models/文件夹下的  
Manager.php 这个文件为 js 文件准备,包括做了目录树,以及目录层次

#### E. Components 的安装

1. Administrator/components/com\_installer 这个文件夹下的是安装文件的全部
2. Admin.installer.php 入口文件处理 submenu
3. Install 部分是个 form 在 views/install/tmpl/文件夹下是处理该部分的代码,  
default\_form.php 是 install 部分 form 的主体代码文件,从主数据库(mi800.net)读取数  
据,用 submitbutton('\$var'),可以给 php 传递变量.  
对 default\_form.php 开头做了修改这样可以读取主数据库的 component 信息
4. 数据库的 jos\_components 里的 iscore 是显示的控制开关
5. Lines:76

```
<table class="adminform">
  <tr>
    <td>
      <th><a href="#"
onclick="javascript:document.adminForm.type.value='';submitbutt
on('components');">Components</a></th> #点击后可以在adminform里显
示相应的components或者modules或者plugins
    </td>
    <td>
      <th><a href="#"
onclick="javascript:document.adminForm.type.value='';submitbutt
on('modules');">Modules</a></th>
    </td>
    <th><a href="#"
onclick="javascript:document.adminForm.type.value='';submitbutt
on('plugins');">Plugins</a></th>
    </td>
  </tr>
</table>
```

6. 在 default\_form.php 的文件开头加入了数据库操作,是为了找出子站与主站的组建差别以便列出,然后读取相应的组件数据库文件,并在子站数据库里建立相应的表
7. 模板部分,分别在前后台建立个 templates\_sub 文件夹,子站从这里读取模板,引入  
\$xml = new JSimpleXML()对象,方便读取模板的 xml 元素  
模板部分的代码修改参见/com\_installer/views/install/tmpl/default\_form.php,因为在  
安装完模板后还要修改/com\_installer/models/templates.php,根据域名取得模板文件

夹下的模板名字

8. 关于模板修改部分集中在 `com_templates/controller.php` 和 `admin.templates.html.php` 文件

**Controller.php: function editTemplate()**

主要修改:根据域名判断对`$tBaseDir` 做出路径修改

**Preview** 操作的修改:

因为预览模板涉及到模板的执行过程所以要涉及修改到 `includes/application.php`,

**Function render()**:渲染模板的时候注意判断加载模板路径

**EditHTMLsource** 操作的修改:

在 `com_templates/controller.php` 文件中修改,function `editTemplatesource()`,

**EditCSS** 操作的修改:

在 `com_templates/controller.php` 文件中修改,function `chooseTemplateCSS()`,  
function `editTemplateCss()`

- 9.

F.