

# Bo-Blog V2.1.0 插件手冊

---

## 目 錄

I. 插件工作原理 .....	2
II. 插件存放的位置 .....	2
III. 檔案構成 .....	2
IV. setup.php的檔格式 .....	3
V. 介面說明 .....	4
VI. 實例演示 .....	8
VII. 兩個特殊函數 .....	9
VIII. 安全提示 .....	10

修正作者：Mesak

文件作者：Bob Shen

最後更新於 2007/9/23

最後適用版本：V2.1.0 rc 1

## I. 插件工作原理

插件是嵌入程式工作流程的一部分。Bo-Blog 主程序預留了一些介面 ( 鉤子 )，當程式執行到預留介面的地方時，會檢查此介面是否已安裝了插件，若已安裝插件，則請求插件的程式檔，將其載入並執行，然後繼續執行主程序其餘的部分。如果不存在插件，則直接跳過該介面，繼續執行其餘的部分。

## II. 插件存放的位置

所有插件都存放於 **plugin** 檔夾下，並單獨建立自己的文件夾。比如 **plugin/mynewplugin**。

## III. 檔案構成

檔案名	作用
<b>setup.php</b>	必需，插件識別資訊，如名稱、描述、製作者等
<b>install.php</b>	非必需，插件安裝時調用的檔，可用於新建資料表，建立檔，初始化資料等
<b>uninstall.php</b>	非必需，插件移除時調用的檔，可用於刪除插件產生的暫存檔案等
<b>admin.php</b>	非必需，插件後台調用文件
<b>&lt;介面名&gt;.php</b>	非必需，插件執行時調用的檔 ( 解釋見後 )
<b>*.*</b>	其他必要的檔，如圖片

所有檔如有多位元組文字，需以 **UTF-8** 編碼保存。

可參閱第 VI 部分——實例演示，瞭解如何創立一個完整的插件。

## IV. setup.php的檔格式

該檔是整個插件的初始化內容，以巴巴變相冊插件為例。

```
<?php
$info['name']="bababian"; //插件識別名，必需是拉丁字母或數位，必需與插件所在檔夾同名，大小寫
需一致

$info['author']="Bob"; //插件作者

$info['intro']="巴巴變相冊"; //插件作用描述

$info['version']="1.0"; //插件自身的版本

$info['authorurl']="http://www.bo-blog.com"; //插件開發者網站

$info['blogversion']="2.0.1"; //插件可正常執行的最低 blog 程式版本

$info['register']="page"; //插件調用的介面名，多個介面之間用半形逗號分割
?>
```

## V. 介面說明

以下將列出介面的類型及使用方法。

### V.1. 介面類型

Bo-Blog 的介面分為 4 種類型，我們把它們稱為“walk 型”、“get 型”、“run 型”和獨立型。這三種介面的區別在於：

#### V.1.1 Get 型：

這類插件的作用原理是“直接添加內容到介面所在的位置”。

當主程序執行到介面處時，會載入相應的介面檔（以該介面名為主檔案名的檔。例如，一個名叫 `example` 的插件，需要在名為 `entrycontentend` 的介面處直接添加內容，則對應的介面檔應該是 `plugin/example/entrycontentend.php`。以下提到的介面檔均遵循此命名規律）。

在介面檔中，插件可以執行任何 PHP 運算，但必須要有一個“回傳值”。“回傳值”是 html 代碼，將被直接添加到介面所在位置。

給出“回傳值”的方法是：

```
$plugin_return="需要添加在當前介面位置的 html 代碼";
```

可參閱第 VI 部分——實例演示，瞭解如何運用 Get 型介面。

當前已支援的 Get 型介面有：

`entrycontentend` , `entryend` , `custompagebegin` , `custompageend` , `entrybegin` ,  
`commentbegin` , `afterpagebar`

#### V.1.2 Walk 型：

這類插件的作用原理是“接受傳入的值，返回經過運算後的值，程式將返回的值替換掉原先傳入的值”。

當主程序執行到介面處時，會載入相應的介面檔。同時，會在介面檔中尋找一個以特殊命名規則命名的函數，並將當前該介面的值傳入這個函數，再用該函數返回的值替換掉原先的值。

函數的命名規則是：

```
plugin_插件識別名_run()
```

函數的輸入和輸出內容：

string **plugin\_example\_run** ( string subject )

可參閱第 VI 部分——實例演示，瞭解如何運用 Walk 型介面。

當前已支援的 Walk 型介面有：

**firstheader , storecontent , ubbanalyse , ubbanalyseadvance**

### V.1.3 Run 型：

這類插件的作用原理是“打斷當前執行的主程序，執行介面檔中的內容，完畢後再繼續執行主程序”。

當主程序執行到介面處時，會載入相應的介面檔。在介面檔中，插件可以執行任何 PHP 運算，且無需返回

任何值。當介面檔中的 PHP 代碼被執行完畢後，主程序會繼續執行。

可參閱第 VI 部分——實例演示，瞭解如何運用 Run 型介面。

當前已支援的 Run 型介面有：

**plannedpublish , ubbeditor , sidebarcategory , sidebarstatistics , sidebararchive , sidebarmisc , sidebarentries , sidebarreplies**

### V.1.4 獨立型：

獨立型只包括兩個介面：**page** 和“管理介面”。

**page** 用於產生一個可單獨打開的頁面。**page** 只能經由 **index.php** 訪問，訪問位址為：

**index.php?act=插件識別名** 。

當用戶請求 **index.php?act=example** 這個位址時，程式會尋找介面檔（即

**plugin/example/page.php**）。

載入這個檔後，把這個檔的“回傳值”作為頁面的主體內容輸出。（頁面的 header 和 footer 仍沿用當前前台範本的。）

**plugin/example/page.php** 輸出“回傳值”的方法是：

```
$plugin_return="需要顯示在頁面上的 html 代碼";
```

“管理介面”是一個特殊的介面，不需要在 `setup.php` 裏定義任何介面名。在一個插件安裝時，程式會尋找這個插件檔夾下是否包含 `admin.php` 這個檔。如果存在，那麼今後就可以從後台的插件管理欄目中點擊進入這個插件的管理介面。

當（管理員）用戶請求 `admin.php?act=example` 這個位址時，程式會尋找介面檔（即 `plugin/example/admin.php`）。載入這個檔後，把這個檔的“回傳值”作為頁面的主體內容輸出。（頁面的 header 和 footer 仍沿用當前後台範本的。）

`plugin/example/admin.php` 輸出“回傳值”的方法是：

```
$plugin_return="需要顯示在頁面上的 html 代碼";
```

“管理介面”介面允許通過 `::` 這個字串，分割出多個分頁。當（管理員）用戶請求

`admin.php?act=example::home` 這個位址時，程式會尋找介面檔

`plugin/example/admin_home.php`，並載入這個檔；當（管理員）用戶請求

`admin.php?act=example::saveme` 這個位址時，程式會尋找介面檔

`plugin/example/admin_saveme.php`，並載入這個檔。注意：如果

`plugin/example/admin_saveme.php` 不存在，則仍將載入 `plugin/example/admin.php`。

## V.2 介面列表

以下將列出所有現在支援的介面，以及它們各自的位置、類型，並注出了可參考的示範插件。

介面名	類型	位置/作用	可參考的示範插件
<b>entrycontentend</b>	Get	日誌正文下方	相關日誌插件、 txtinclude 插件
<b>entryend</b>	Get	日誌管理工具條下方	Bookmarkit 插件、 txtinclude 插件
<b>custompagebegin</b>	Get	自定義頁面開始處	N/A
<b>entrybegin</b>	Get	日誌開頭的位置	txtinclude 插件
<b>custompageend</b>	Get	自定義頁面結束處	N/A
<b>commentbegin</b>	Get	評論（或評論框）開始處	txtinclude 插件
<b>afterpagebar</b>	Get	分頁條下方	N/A
<b>firstheader</b>	Walk	傳入值：頁面中，從<head>到</head>區域的內容 作用：可用戶載入<head>區域需要的 javascript、css 檔等	飯否插件
<b>storecontent</b>	Walk	傳入值：新增/修改日誌後，當前發表的日誌正文的內容 作用：可以更改寫入資料庫的正文內容；也可以完成一些在日誌發佈時需要執行的動作，如 ping 搜索引擎等	飯否插件，Google Ping 插件
<b>ubbanalyse</b>	Walk	傳入值：(日誌正文、評論正文等需要進行 UBB 解析) 文字 作用：自定義 UBB 標籤的解析	DPHighlighter 插件、 Fmplayer 插件
<b>ubbanalyseadvance</b>	Walk	同 <b>ubbanalyse</b> ，但是這裏的解析只對日誌正文發揮作用，不解析評論、留言	CC 視頻聯盟插件
<b>plannedpublish</b>	Run	在“定期發佈日誌”發佈時執行	N/A
<b>ubbeditor</b>	Run	在 UBB 編輯器產生時執行（請參考第 VI 部分的示例，瞭解如何在 UBB 編輯器上添加按鈕和功能）	CC 視頻聯盟插件、 Fmplayer 插件
<b>sidebarcategory</b>	Run	在產生側邊欄“分類”時執行	N/A
<b>sidebarstatistics</b>	Run	在產生側邊欄“統計”時執行	N/A
<b>sidebararchive</b>	Run	在產生側邊欄“歸檔”時執行	N/A
<b>sidebarmisc</b>	Run	在產生側邊欄“雜項”時執行	N/A
<b>sidebarentries</b>	Run	在產生側邊欄“最新日誌”時執行	N/A
<b>sidebarreplies</b>	Run	在產生側邊欄“最新評論”時執行	DetailedReplies 插件
<b>page</b>	獨立	建立空白頁面	CC 視頻插件
無	獨立	建立管理介面	Fmplayer 插件

## VI. 實例演示

例一：新建一個獨立頁面，進入該頁面時顯示“Hello, world”字樣。

A. 在插件所在文件夾建立 **setup.php**，在這個檔中，指出調用的介面名為 **page**。即有這樣一行：

```
$info['register']="page";
```

B. 在插件所在文件夾建立 **page.php**，內容大致如下：

```
<?PHP
$plugin_return='Hello, world';
?>
```

如果想在頁面打開時自動遮罩側邊欄，可在 **page.php** 中加入：

```
$plugin_closesidebar=1;
```

例二：在日誌正文下方插入一個文字框，注明“版權所有，請勿轉載”。

A. 在插件所在文件夾建立 **setup.php**，設置

```
$info['register']="entrycontentend";
```

B. 在插件所在文件夾建立文件 **entrycontentend.php**。該檔的內容大致如下：

```
<?php
$plugin_return="<div style=\"border: 1px solid #000; width: 90%;\">版權所有，請勿轉載</div>";
?>
```

例三：自定義 UBB 代碼[marquee]。該代碼的作用是使文字橫向滾動。

A. 設該插件的識別名為 **domarquee**。

B. 建立 **plugin/domarquee/setup.php**，設置

```
$info['register']="ubbanalyse";
```

C. 建立文件 **plugin/domarquee/ubbanalyse.php**。該檔的內容大致如下：

```
<?php
function plugin_domarquee_run ($strInput) {
    $strInput=preg_replace("/\[marquee\](.+?)\[\/marquee\]/i", "<marquee>\\1</marquee>",
    $strInput);
    return $strInput;
}
?>
```



例四：在例三的基礎上，增加 UBB 編輯器上的一個按鈕，點擊可直接插入 `marquee` 代碼。

A. 修改 `plugin/domarquee/setup.php`，設置

```
$info['register']="ubbanalyse,ubbeditor";
```

B. 建立文件 `plugin/domarquee/ubbeditor.php`。該檔中，需包含兩個變數：

變數：`$plugin_ubbeditor_buttons`

作用：增添在編輯器尾部的 html 內容，比如添加一個按鈕

變數：`$plugin_ubbeditor_functions`

作用：javascript 代碼（不包含 `<script></script>` 部分），比如按下這個按鈕後執行的 javascript 動作

注意：為了避免此處的自定義內容覆蓋其他的自定義內容，以上兩個變數的賦值請使用疊加方式；並在文件開頭寫上

```
global $plugin_ubbeditor_functions, $plugin_ubbeditor_buttons ;
```

編輯器相關 javascript 動作：

```
AddText( STRING );
```

將指定的字串插入到編輯框內。

## VII. 兩個特殊函數

在這兩個檔中，您可以使用函數 `add_module()` 和 `remove_module()` 來建立/移除一個模組專案。

示例：在安裝該插件的同時，安裝一個名為 `example` 的模組專案。

A. 建立該模組專案的安裝檔 `example.blog`。

B. 在 `install.php` 中寫上

```
add_module ('minivote.blog'); ;
```

在 `uninstall.php` 中寫上

```
remove_module ('minivote.blog');
```

## VIII. 安全提示

請注意，為了避免插件文件被獨立執行可能帶來的問題，請在各個 PHP 的頭部加入必要的條件判斷。

在前台，可通過判斷常數 `VALIDREQUEST` 是否定義過來判斷是否為正常調用。

在後台，可通過判斷常數 `VALIDADMIN` 是否定義過來判斷是否為正常調用。

示例代碼如：

```
if (!defined('VALIDREQUEST')) die ('Access Denied.');
```