



ThinkPHP Framework 1.5

App autoBuild & Compire

ThinkPHP 1.5

项目自动生成和编译

编写：ThinkPHP 文档组

最后更新：2008-12-16

目录

1	概述.....	3
2	生成目录.....	3
3	安全文件.....	4
4	项目编译.....	5
5	注意事项.....	6

1 概述

我们知道，使用框架开发项目的时候，都需要遵守框架的一些目录规范，所以我们经常需要在每次项目的开始来——创建这些目录结构，虽然 Tp 已经有很多第三方的目录生成工具，他们能够最大程度减轻你手动创建的工作量。但是即使你手边没有自动生成工具，ThinkPHP 框架已经内置了项目目录的自动生成功能。而 ThinkPHP 独特的项目编译机制更是可以让应用在没有服务器缓存支持的情况下获得更好的运行性能。

2 生成目录

ThinkPHP 具有项目目录自动创建功能，你只需要定义好项目的入口文件，第一次执行入口文件的时候，系统会自动创建项目的相关目录结构，如果是 linux 环境下面需要给项目入口文件里面指定的路径设置可写权限。

例如，我们在 WEB 的根目录下面创建一个 App1 目录，并且在下面定义了一个项目入口文件 index.php 如下：

```
// 定义 ThinkPHP 框架路径
define('THINK_PATH', '../ThinkPHP/');

//定义项目名称和路径
define('APP_NAME', 'App1');
define('APP_PATH', '.');

// 加载框架入口文件
require(THINK_PATH."/ThinkPHP.php");

//实例化一个网站应用实例
$app = new App();
```

```
//应用程序初始化
```

```
$App->run();
```

然后，我们什么都不用做，仅仅是在浏览器里面输入

```
http://localhost/App1/
```

系统在第一次执行的时候会自动判断项目的目录结构是否存在，如果不存在的话就会完成整个项目的目录结构的自动创建，并自动创建了一个 Hello,world 应用，你可以修改 Lib\Action\IndexAction.class.php 文件里面的 index 方法来开始你的第一个 ThinkPHP 应用。

3 安全文件

在 1.0.3 版本以上，如果在入口文件里面设置了 BUILD_DIR_SECURE 参数为 True，还会自动给项目目录生成目录安全文件（在相关的目录下面生成空白的 htm 文件），并且可以自定义安全文件的文件名 DIR_SECURE_FILENAME，默认是 index.html，如果你想给你们的安全文件定义为 default.html 可以使用

```
define('DIR_SECURE_FILENAME', 'default.html');
```

还可以支持多个安全文件写入，例如你想同时写入 index.html 和 index.htm 两个文件，以满足不同的服务器部署环境，可以这样定义：

```
define('DIR_SECURE_FILENAME', 'index.html,index.htm');
```

默认的安全文件只是写入一个空白字符串，如果需要写入其他内容，可以通过 DIR_SECURE_CONTENT 参数来指定，例如：

```
define('DIR_SECURE_CONTENT', 'deney Access!');
```

下面是一个完整的使用目录安全写入的例子

```
define('BUILD_DIR_SECURE', true);
```

```
define('DIR_SECURE_FILENAME', 'default.html');
```

```
define('DIR_SECURE_CONTENT', 'deney Access!');
```

4 项目编译

ThinkPHP 正式版本开始引入了新的项目编译机制，而在 1.0.3 版本中又进行了一些优化设置。

所谓的项目编译机制是指系统第一次运行的时候会自动生成核心缓存文件~runtime.php 和项目编译缓存文件~app.php，第二次就会直接载入编译过的缓存文件，从而省去很多 IO 开销，加快执行速度，并且编译缓存文件默认是去掉空白和注释的，因此存在一个预编译的过程。项目编译机制对运行没有任何影响，预编译操作和其他的目录检测机制只会执行一次，因此无论在预编译过程中做了多少复杂的操作，对后面的执行没有任何效率的缺失。

核心编译缓存文件~runtime.php 包含了下面的一些文件：

1. 系统定义文件
2. 系统函数库
3. 系统基类 Base
4. 异常基类 ThinkException
5. 日志类 Log
6. 应用类 App
7. 控制器基类 Action
8. 模型基类 Model
9. 视图类 View

项目编译缓存文件~app.php 包含了下面的一些文件：

1. 项目配置文件（由惯例配置、项目配置合并而成）

2. 项目公共函数文件 common.php

默认的编译缓存目录为 TEMP_PATH (项目的临时文件目录)。

还可以在入口文件里面设置 RUNTIME_PATH，例如：

```
defined('RUNTIME_PATH','./MyApp/runtime/');
```

系统会自动把编译缓存文件放到该目录下面生成，注意这样的话在 Linux 环境下面需要对

RUNTIME_PATH 目录设置可写权限。

注意在调试模式下面不会生成项目编译缓存，但是依然会生成核心缓存。

如果不希望生成核心缓存文件的话，可以在项目入口文件里面设置 CACHE_RUNTIME

例如：

```
define('CACHE_RUNTIME',false);
```

如果在非调试模式下面，修改了配置文件或者项目公共文件，需要删除项目编译文件。

1.0.4 版本开始，修改配置文件无需删除编译缓存文件。

1.0.3 版本以上，还可以设置对编译缓存的内容是否进行去空白和注释，例如：

```
define('STRIP_RUNTIME_SPACE',false);
```

则生成的编译缓存文件是没有经过去注释和空白的，仅仅是把文件合并到一起，这样的好处是便于调试的错误定位，建议部署模式的时候把上面的设置为 True 或者删除该定义。

5 注意事项

下面的一些问题需要引起注意，便于在开发过程中更快的解决问题：

- 1、编译缓存目录必须设置为可写
- 2、如果有修改核心文件，可能需要删除核心编译缓存~runtime.php
- 3、如果在非调试模式下有修改配置文件或者项目的 common 文件，需要删除项目编译缓存文件~app.php
- 4、如果定义的项目目录不存在会自己进行创建，如果连项目名称没有定义的话，系统会自动生成一个系统路径 MD5 后的字符串作为项目名称，并且会以这个名称自动创建目录。
- 5、系统是根据核心编译缓存文件来判断是否需要检测目录结构的，如果你删除了核心缓存文件的话，这个过程会重新检查，但是不会覆盖已有的文件和目录。