#### CrossApp的MVC框架

##### MVC模式简介

关于MVC模式，并没有一个非常明确的概念，在不同的开发环境中，设计可能有所区别。

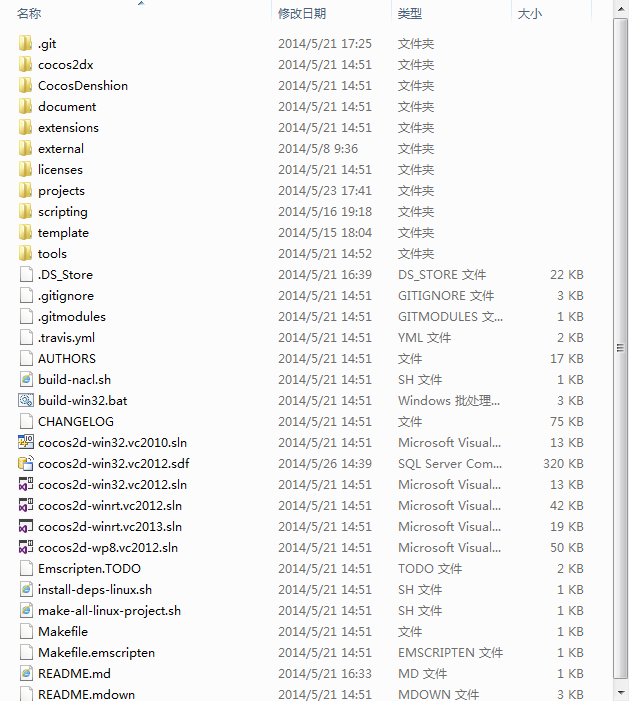
面向对象的目标就是设计出低耦合、高聚合的程序，MVC模式提供了一种很好的解决方案。CrossApp是基于MVC模式的，这里的MVC即模型层(Model)、视图层(View)、控制层(Controller)，三者各司其职，共同维持一个完整的应用程序。

在CrossApp中的Model层也就是我们各种数据原型、业务逻辑和算法，View层顾名思义就是展现给用户的UI界面，而Controller层的职责就是把Model和View两个相互关联不大的层衔接起来。CrossApp通过CAView和丰富的UI控件来实现View层的功能，通过CAViewController及其子类来共同完成Controller层，实现Model和View之间的通信(注：CrossApp可以使用Xcode和Visual Studio来进行开发，以后所有的示例和详解都基于Visual Studio 2013来进行展示)。

##### 如何使用CrossApp开发应用程序

使用CrossApp开发移动应用程序必要的准备工作可参考以下步骤：

1. **从CrossApp官网**[www.9miao.com下载CrossApp](http://www.9miao.com下载CrossApp)**的源码，解压至本地计算机。以下是CrossApp解压的目录结构**



1. **开发前的环境配置，在CrossApp的官网有环境配置的详细说明，参考说明进行开发前的环境配置，以便开发完成导出各个平台的安装包。**
2. **以上两步完成后，现在就可以开始我们的App开发，首先需要新建一个自己的CrossApp工程，同样在环境配置文档中也说明了如何使用命令行新建一个CrossApp工程，运行proj.win32下的.sln文件就可以运行我们新建的工程，新建工程的目录结构如下：**





1. 参考CrossApp的API文档就可以进行我们跨平台应用程序的开发了，以下是我们新建一个CrossApp工程运行后的目录结构，也就是默认的HelloWorld示例，以后我们自己的编码通常就是从AppDelegate类开始，具体的使用范例可以参考CrossApp官网的Demo示例。

