

我们都知道在软件设计中有一个非常重要的原则就是要高内聚，低耦合。

怎样才能高内聚呢，这主要有两个层面：

1. 在函数级别，功能相关的函数要尽量放在一起或者要放在一个类中。
2. 在类型级别，功能相关的类型要尽量放在一个Project中，并且要在同一个大的命名空间下。

但是在实际开发中光是这样做还是不够的，这里主要有以下几个问题：

我们知道VSTS是由下面几个概念组成的：Solution，Project，File。

第一个问题，虽然同一个Project下面都是功能相关的类，但是这些类完成的功能还是会有所区别，如果采用扁平结构，所有文件都放在Project的根目录下面的话会导致难于表达和理解各个类之间的关系，但是如果能利用Folder机制对Project中的文件进行分区，把关系密切的类放在同一个Folder里面，并为这个Folder起一个有意义的名字，可以根据情况分成多个子级，这样就能很好的表达类之间的关系，方便开发者理解。

第二个问题，一般一个大型的项目在Solution下面会有很多的Project，这些Project有的是DAL层的，有的是BLL层的，有的还是UI层的，并且这些Project之间有的关系密切（比如契约与实现的关系）有的则没有任何联系。如果采用扁平结构，我们将很难表达和理解各个Project之间的关系（层次关系和契约与实现的关系），但是如果能利用Folder对Solution下面的Project进行分区，将关系密切的Project放在同一个Folder里面，并为这个Folder起一个有意义的名字，还可以根据情况分成多个子级，这样就可以很好的表达Project之间层次的关系以及契约和实现的关系。这样做还有一个非常积极的作用，那就是如果在Folder的命名中有分层的信息(层编号)，那么就可以在项目交付后的维护过程中实施项目设计之初就定下的分层规范：

- 1.为Solution添加Project时请为Project选择合适的层Folder，并在层Folder下再选择合适的Sub Folder.
- 2.为Project添加引用时，必须选择同层或下层的Project，底层的Project不能引用高层的Project.