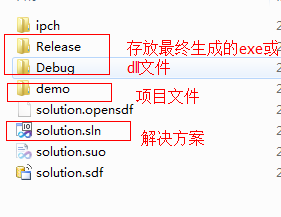
vs2010使用方案管理项目，一个解决方案下可包含多个项目。

默认情况下，项目属性的设置的目录起点为项目配置文件所在的位置，实际上就是项目头文件和源文件所在的位置。

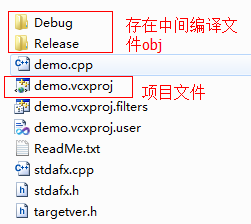
vs2010中默认建立C++项目，则解决方案总目录下包含一个sln和一个项目文件夹，在vs2010编译器中生成debug和release解决方案后，总目录下还会生成对应的debug和release目录，存放最终生成的exe或dll文件，同时也会在项目文件夹下生成debug和release目录（存放的是中间编译文件obj）。

下面结合例子讲一下：solution为创建的解决方案，demo为创建的项目

解决方案solution总目录如下：



项目demo目录如下：



项目配置及系统变量关系：

在vs2010的项目属性页，会有一些系统变量，如下：

SolutionDir：解决方案目录

Configuration：指debug或release

ProjectName：项目名字

OutDir：在 常规--输出目录 中定义的值，如$(SolutionDir)$(Configuration)\，表示XXX\lolution\debug\目录值

IntDir：中间目录

TargetDir：生成exe或dll文件所在位置，如链接器-常规-输出文件为$(OutDir)$(TargetName)$(TargetExt)(即定义了exe输出位置，也就决定了TargetDit的值)，此时TargetDir表示在XXX\lolution\debug\

TargetName：目标输出名，不包括扩展名

TargetPath：目标输出文件的全路径名

ProjectDir：表示项目目录值，一般在“调试-工作目录”中设置该值

TargetExt：扩展名

PlatformToolsetVersion：

ConfigurationName：配置名字，通常是Debug或者Release

默认情况下“输出目录”和“输出文件”对应的目录值是一样的。

配置属性

             常规

                      输出目录：$(SolutionDir)$(Configuration)\

                      中间目录：$(Configuration)\

                      目标文件名：$(ProjectName)

                      目标文件扩展名：.exe

                      生成日志文件：$(IntDir)\$(MSBuildProjectName).log

            调试

                     命令：$(TargetPath)，表示调试器要启动的exe全名，TargetPath就表示目标输出文件的全路径名，所以一般情况下它代表的值就等于“输出文件”属性代表的值

                     工作目录：$(ProjectDir)

C/C++

          预编译头

                    预编译头输出文件：$(IntDir)$(TargetName).pch

          输出文件

                    ASM列表位置：$(IntDir)

                    对象文件名：$(IntDir)

                    程序数据库文件名：$(IntDir)vc$(PlatformToolsetVersion).pdb

          浏览信息

                    浏览信息文件：$(IntDir)

链接器

          常规

                    输出文件：$(OutDir)$(TargetName)$(TargetExt)

         清单文件

                   清单文件：$(IntDir)$(TargetName)$(TargetExt).intermediate.manifest

         调试

                  生成程序数据库文件：$(TargetDir)$(TargetName).pdb

         优化：

                  按配置优化数据库：$(TargetDir)$(TargetName).pgd