

visual C++ 项目和解决方案的区别

项目：

项目是构成某个程序的全部组件的容器，该程序可能是控制台程序、基于窗口的程序或某种别的程序。程序通常由一个或多个包含用户代码的源文件，可能还要加上包含其它辅助数据的文件组成。某个项目的所有文件都存储在相应的项目文件夹中，**关于项目的详细信息存储在一个扩展名为 .vcproj 的 xml 文件中**，该文件同样存储在相应的项目文件夹中。项目文件夹还包括其它文件夹，它们用来存储编译及链接项目时所产生的输出。

即：项目是构成某个程序的全部组件的容器。

解决方案：

顾名思义，解决方案的含义是一种将所有程序和其它资源（它们是某个具体的数据处理问题的解决方案）聚集到一起的机制。例如，用于企业经营的分布式订单录入系统可能由若干个不同的程序组成，而各个程序是作为同一个解决方案内的项目开发的，因此，解决方案就是存储与一个或多个项目有关的所有信息的文件夹，这样就有一个或多个项目文件夹是解决方案文件夹的子文件夹。**与解决方案中项目有关的信息存储在扩展名为 .sln 和 .suo 的两个文件中**。当创建某个项目时，如果没有选择在现有的解决方案中添加该项目，那么系统将自动创建一个新的解决方案。

即：解决方案是一种将所有程序和其他资源聚集到一起的机制。一个或多个项目文件夹是解决方案文件夹的子文件夹。

当创建项目及解决方案时，可以在同一个解决方案中添加更多的项目。我们可以在现有的解决方案中添加任意种类的项目，但通常只添加与该解决方案内现有项目相关的项目。一般来说，各个项目都应该有自己的解决方案，除非我们有很好的理由不这样做。

VS2010 创建项目时，默认的项目文件夹的名称与项目名称相同，该文件夹还将容纳构成该项目定义的文件。如果不修改的话，解决方案文件夹具有与项目文件夹相同的名称，其中包含了项目文件夹、定义解决方案内容的文件。解决方案文件夹内包含如下四个文件：

- 扩展名为 .sln 的文件，记录着关于解决方案中项目的信息。
- 扩展名为 .suo 的文件，其中记录着应用于该解决方案的用户选项。
- 扩展名为 .sdf 的文件，记录着与解决方案的 Intellisense 有关的数据。Intellisense 是在 Editor 窗口中输入代码时提供自动完成和提示功能的工具。
- 扩展名为 .opensdf 的文件，记录着关于项目状态的信息，此文件只在项目处于打开状态时才有。

项目文件夹中的 Readme.txt 文件包含已经为该项目创建的所有文件的内容摘要。

构建项目时创建的文件：

构建项目完成后将看到解决方案文件夹中出现了一个新的子文件夹 Debug。该文件夹包含刚才构建项目时产生的输出，这个文件夹包含三个文件。

- .exe 文件：可执行文件
- .ilk 文件：在重新构建项目时被链接器使用，它使链接器能够根据修改的源代码生成的目标文件增量的链接到现有的 .exe 文件，从而避免每次修改程序时都重新链接所有文件。
- .pdb 文件：包含调试信息，在调试模式中执行程序要使用该调试信息。在调试模式中，可

以动态检查程序执行过程中所生成的信息。

在项目文件夹中也有一个 Debug 目录，它包含在构建过程中生成的很多文件，从 Windows Explorer 的 Type 描述中可以看到它们包含何种信息。

1. 解决方案相关文件

解决方案相关文件包括解决方案文件夹下的 .sdf 文件、.sln 文件、.suo 文件和 ipch 文件夹。

.sdf 文件和 ipch 目录一般占用空间比较大，几十兆甚至上百兆，与智能提示、错误提示、代码恢复、团队本地仓库等相关。如果你觉得不需要则可以设置不生成它们，方法是点击菜单栏 Tools->Options，弹出 Options 对话框，选择左侧面板中 Text Editor->C/C++->Advanced，右侧列表中第一项 Disable Database 由 False 改为 True 就可以了，最后关闭 VS2010 再删除 .sdf 文件和 ipch 目录以后就不会再产生了。但关闭此选项以后也会有很多不便，例如写程序时的智能提示没有了。

sln 文件和 .suo 文件为 MFC 自动生成的解决方案文件，它包含当前解决方案中的工程信息，存储解决方案的设置。

2. 工程相关文件

工程相关文件包括工程文件夹下的 .vcxproj 文件和 .vcxproj.filters 文件。

.vcxproj 文件是 MFC 生成的工程文件，它包含当前工程的设置和工程所包含的文件等信息。.vcxproj.filters 文件存放工程的虚拟目录信息，也就是在解决方案浏览器中的目录结构信息。

3. 应用程序头文件和源文件

应用程序向导会根据应用程序的类型（单文档、多文档或基于对话框的程序）自动生成一些头文件和源文件，这些文件是工程的主体部分，用于实现主框架、文档、视图等。

下面分别简单介绍下各个文件：<http://hovernote.com/menu/cpp/>

HelloWorld.h: 应用程序的主头文件。主要包含由 CWinAppEx 类派生的 CHelloWorldApp 类的声明，以及 CHelloWorldApp 类的全局对象 theApp 的声明。

HelloWorld.cpp: 应用程序的主源文件。主要包含 CHelloWorldApp 类的实现，CHelloWorldApp 类的全局对象 theApp 的定义等。

MainFrm.h 和 MainFrm.cpp: 通过这两个文件从 CFrameWndEx 类派生出 CMainFrame 类，用于创建主框架、菜单栏、工具栏和状态栏等。

HelloWorldDoc.h 和 HelloWorldDoc.cpp: 这两个文件从 CDocument 类派生出文档类 CHelloWorldDoc，包含一些用来初始化文档、串行化（保存和装入）文档和调试的成员函数。

HelloWorldView.h 和 HelloWorldView.cpp: 它们从 CView 类派生出名为 CHelloWorldView 的视图类，用来显示和打印文档数据，包含了一些绘图和用于调试的成员函数。

ClassView.h 和 ClassView.cpp: 由 CDockablePane 类派生出 CClassView 类，用于实现应用

程序界面左侧面板上的 Class View。

FileView.h 和 FileView.cpp: 由 CDockablePane 类派生出 CFileView 类, 用于实现应用程序界面左侧面板上的 File View。

OutputWnd.h 和 OutputWnd.cpp: 由 CDockablePane 类派生出 COutputWnd 类, 用于实现应用程序界面下侧面板 Output。

PropertiesWnd.h 和 PropertiesWnd.cpp: 由 CDockablePane 类派生出 CPropertiesWnd 类, 用于实现应用程序界面右侧面板 Properties。

ViewTree.h 和 ViewTree.cpp: 由 CTreeCtrl 类派生出 CViewTree 类, 用于实现出现在 ClassView 和 FileView 等中的树视图。

4. 资源文件

一般我们使用 MFC 生成窗口程序都会有对话框、图标、菜单等资源, 应用程序向导会生成资源相关文件: res 目录、HelloWorld.rc 文件和 Resource.h 文件。

res 目录: 工程文件夹下的 res 目录中含有应用程序默认图标、工具栏使用图标等图标文件。

HelloWorld.rc: 包含默认菜单定义、字符串表和加速键表, 指定了默认的 About 对话框和应用程序默认图标文件等。

Resource.h: 含有各种资源的 ID 定义。

5. 预编译头文件

几乎所有的 MFC 程序的文件都要包含 afxwin.h 等文件, 如果每次编译都要编译一次则会大大减慢编译速度。所以把常用的 MFC 头文件都放到了 stdafx.h 文件中, 然后由 stdafx.cpp 包含 stdafx.h 文件, 编译器对 stdafx.cpp 只编译一次, 并生成编译之后的预编译头 HelloWorld.pch, 大大提高了编译效率。

6. 编译链接生成文件

如果是 Debug 方式编译, 则会在解决方案文件夹和工程文件夹下都生成 Debug 子文件夹, 而如果是 Release 方式编译则生成 Release 子文件夹。

工程文件夹下的 Debug 或 Release 子文件夹中包含了编译链接时产生的中间文件, 解决方案文件夹下的 Debug 或 Release 子文件夹中主要包含有应用程序的可执行文件。

关于应用程序工程文件的组成结构就先讲到这了。其中包含了很多专有名词, 以后大家会慢慢熟悉的