

第 1 章 ObjectCRX 编程基础

初学 ObjectCRX 编程，需要了解其开发和调试环境的构建，本章首先帮助读者建立起开发环境，进而介绍加载和运行 ObjectCRX 应用程序的方法，最后仍然以 Hello,World 来作为进入 ObjectCRX 开发的第一个程序。

1.1 ObjectCRX 开发环境

1.1.1 说明

使用 ObjectCRX，首先要确定你的目标平台，获得适当的开发环境。如果你是在电子图板 2011 平台上开发，你就需要具备下面的工具和软件：

- 电子图板 2011 中文版或英文版。
- Visual Studio 2010 英文版。
- ObjectCRX 2011 开发包。

本节将要介绍 ObjectCRX 开发环境的构建，包括开发包的获得、开发包的组成部分、ObjectCRX 向导的安装，以及自动提示和显示 ObjectCRX 关键字的方法。

1.1.2 思路

本章的内容不涉及具体的编程，此部分从略。

1.1.3 步骤

在开始本书的旅行之前，首先要指出，本书所有的程序都是基于电子图板 2011 中文版，因此构建的开发环境为“Visual Studio 2010 英文版 + 电子图板 2011 中文版 + ObjectCRX2011”。按照

下面的步骤，一步一步构建开发环境：

(1) 安装电子图板 2011 中文版和 Visual Studio 2010 英文版。软件的安装可以参考相关软件的说明文件，不再介绍。

注意：请选择正确的 Visual Studio 版本，否则可能无法与电子图板 2011 集成开发。

(2) 获得 ObjectCRX 开发包。

安装完《CAXA 电子图板 2011 - 机械版》后会在其安装目录下获得 ObjectCRX 开发包，比如将《CAXA 电子图板 2011 - 机械版》安装在 C 盘下后，ObjectCRX 开发包在 C:\Program Files\CAXA\CAXA DRAFT MECHANICAL\2011\CRX 目录下。在上述目录下

得到下面几个文件夹：

- docs：包含所有的联机帮助文件。
- docsamps：包含在《ObjectCRX 开发者向导》（在 docs 文件夹中，为英文的资料）中所提到的源代码和说明文件。
- inc：包含 ObjectCRX 的头文件。
- lib：包含 ObjectCRX 的库文件。
- Wizard：Visual Studio 新建 crx 项目的向导工具程序。

（3）安装 ObjectCRX 开发向导。按照开发包中的路径 Wizard 目录下，找到一个名称为 CRXWizard.exe 安装文件。点击安装后，系统弹出如图 1.1 所示的对话框，单击【Install】按钮开始安装向导。



图 1.1 安装 ObjectCRX 开发向导

经过一段时间，系统弹出如图 1.2 所示的对话框，单击【Finish】按钮完成 ObjectCRX 向导的安装。

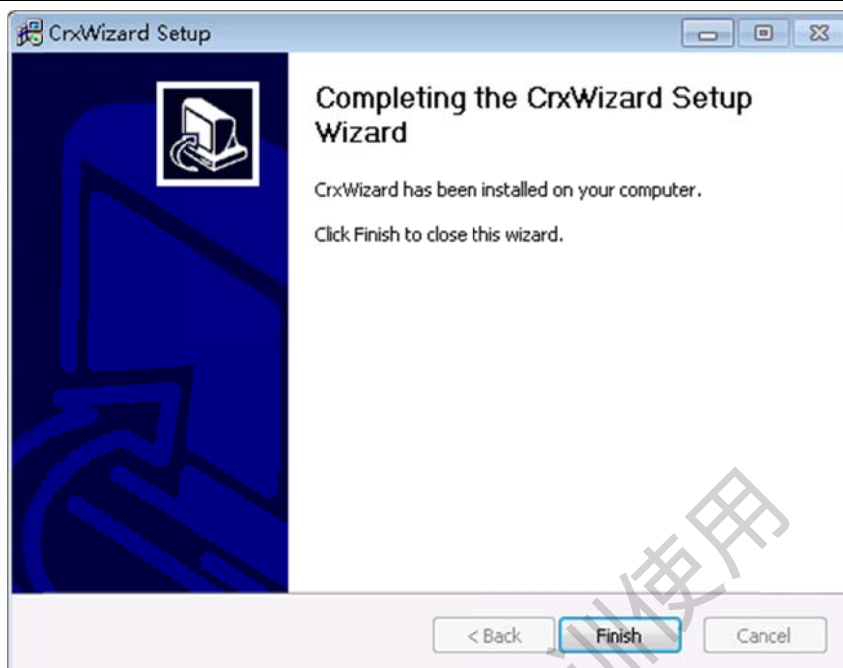


图 1.2 向导安装完成

此时启动 Visual Studio 2010, 选择【File/New/Project】菜单项, 系统会弹出【New/Project】对话框, 其中的项目列表中已经包含了 ObjectCRX 2011 AppWizard, 如图 1.3 所示。

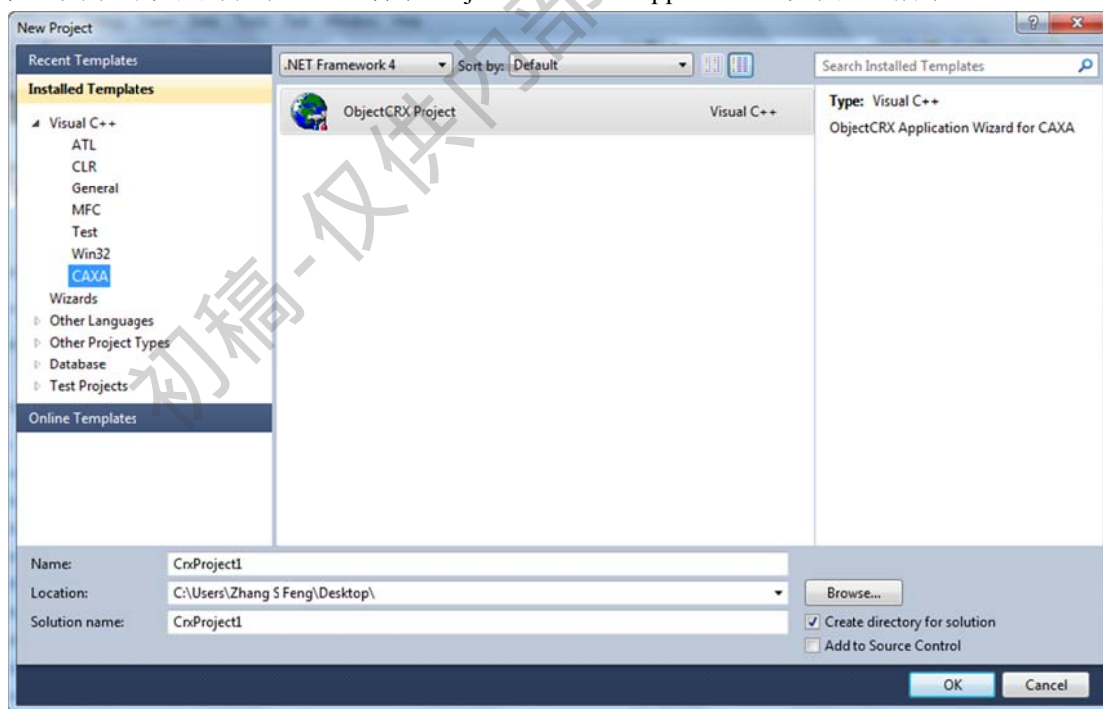


图 1.3 ObjectCRX 开发向导被添加到列表中

1.1.4 效果

本节不需要实例。

1.1.5 小结

虽然本书介绍的是 ObjectCRX2011，电子图版 2011 。

1.2 ObjectCRX 应用程序的加载和运行

1.2.1 说明

本节介绍在开发 ObjectCRX 程序阶段加载和运行应用程序的方法，这是调试程序的基础，也是程序员所使用的方法。在创建打包程序的部分，还要介绍一些自动加载 ObjectCRX 应用程序的方法，那些则是用户使用的方法。

1.2.2 思路

加载 ObjectCRX应用程序可以通过多种方法：

- 使用 MANAGE 命令。

执行 ObjectCRX应用程序仅可在命令行输入程序中注册的命令。

1.2.3 步骤

下面的步骤演示加载和运行 ObjectCRX应用程序的方法：

- (1)使用MANAGE命令加载程序。在电子图板2010中，选择【工具/模块管理器】ribbon bar 菜单项，系统会弹出如图1.4所示的【模块管理器】对话框。

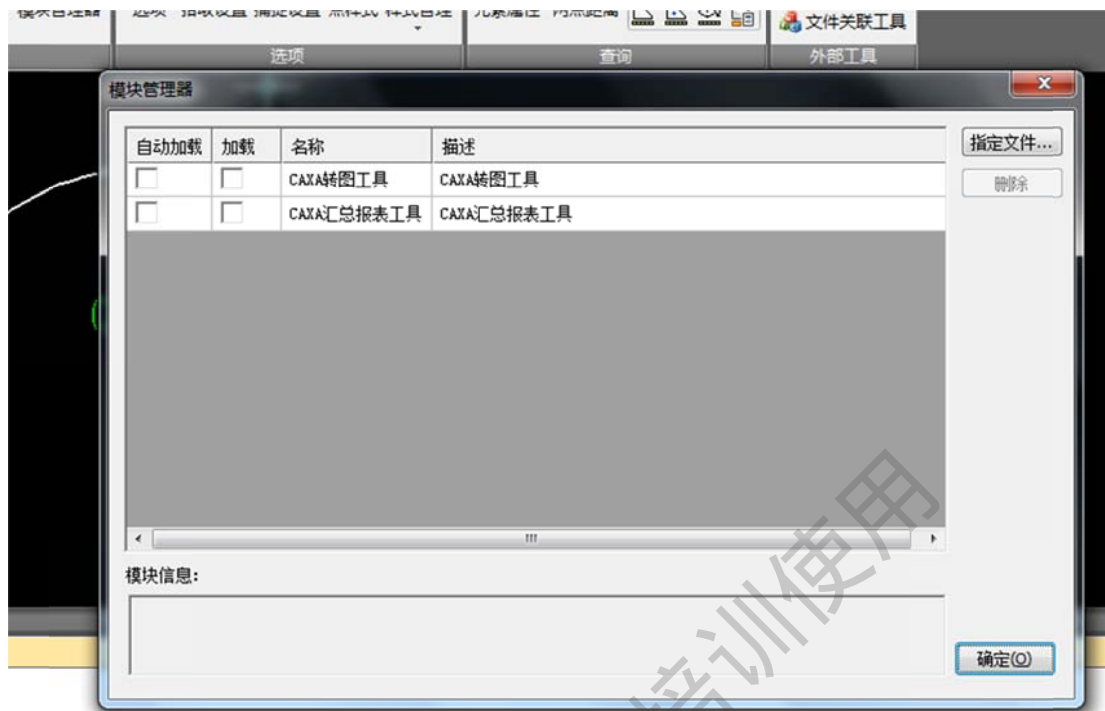


图 1.4 加载应用程序

从文件列表中选择所要加载的程序，勾选【加载】复选框，就可以将选择的程序加载到电子图板 2010 中。需要注意的是，所有二次开发的程序必须加载后才能使用。

如果要经常使用某个程序，就可以让电子图板 2004 在启动时自动加载该程序。在【模块管理器】对话框中单击【指定文件】按钮，系统会弹出如图 1.5 所示的【选择需要加载的 CRX 应用程序】对话框。单击【打开】按钮，系统会弹出【将文件添加到启动组中】对话框，在此对话框中选择所要添加到启动组中的程序。

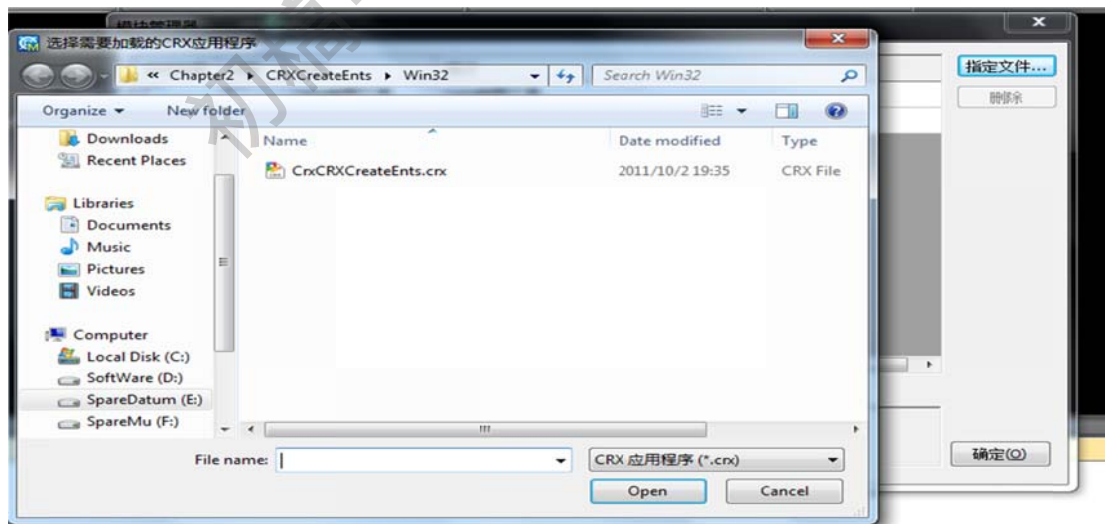


图 1.5 选择需要加载的 CRX 应用程序

注意：将程序添加到启动组，使其在电子图板启动时自动加载，对于 ObjectCRX 程序的调试很有帮助。

1.2.4 效果

参见上一节的内容。

1.2.5 小结

学习本节内容之后，读者应该掌握使用 MANAGE 命令加载 ObjectCRX 应用程序的方法，以及运行 CRX 程序的方法。

1.3 用向导创建 Hello,World 程序

1.3.1 说明

本节使用 ObjectCRX 向导创建一个 Hello,World 程序，了解向导创建的工程的基本结构，并且学习使用 ObjectCRX 内嵌工具栏注册命令的方法。程序执行的结果与上一节的程序一样，在 电子图板命令行显示 “Hello, World!” 字符串。

1.3.2 思路

使用向导创建 ObjectCRX 应用程序，能够大大简化操作的步骤，向导自动生成了大量的代码。同学习 MFC 一样，如果了解了向导工作的内容，那么使用向导能够大大节省工作量；如果不了解，则可能越用越糊涂。

1.3.3 步骤

(1) 在 Visual Studio 2010 集成开发环境中，选择【File/New/Project】菜单项，系统会弹出【Project】对话框。在【Projects】选项卡的工程类型列表中选择【Visual C++/CAXA】，然后在其工程列表中选择【ObjectCRX Projects】，在下面【name】文本框中输入工程名称 Hello2，在【Location】文本框中输入适当的工程位置，如图 1.6 所示。完成设置后，单击【OK】按钮。

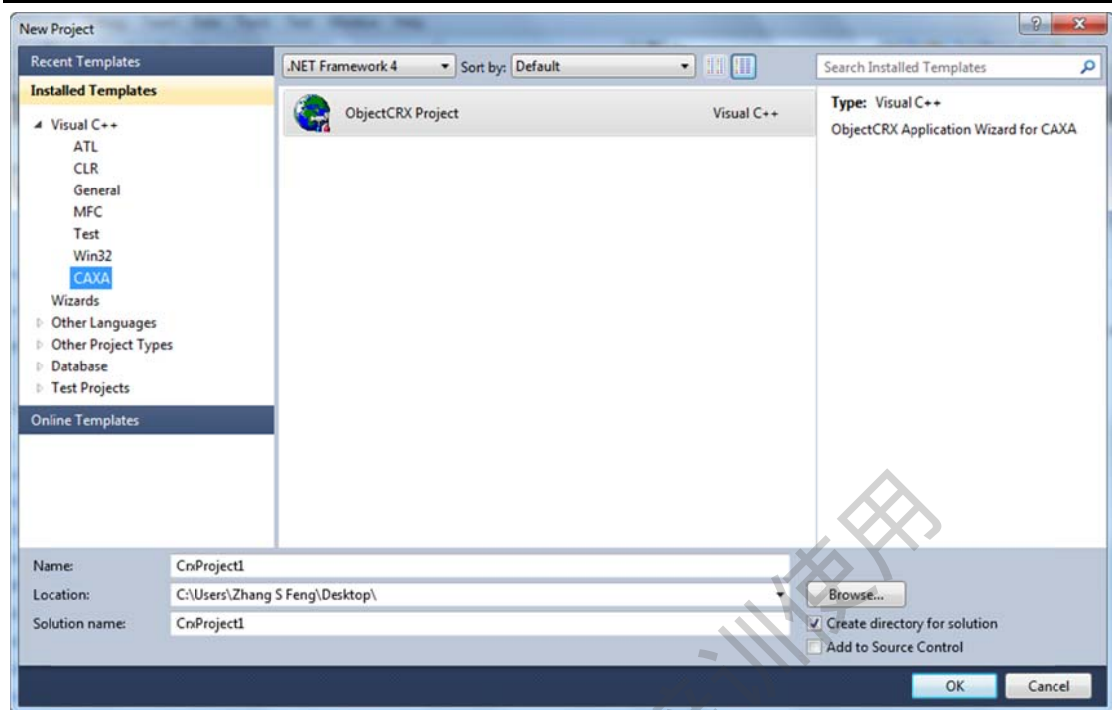


图 1.6 创建 crx 项目

(2) 在弹出的对话框中, 选择 Create a extension DLL(MFC shared)(Q) 选项, 如图 1.7 所示。设置完毕之后, 单击【Finish】按钮。

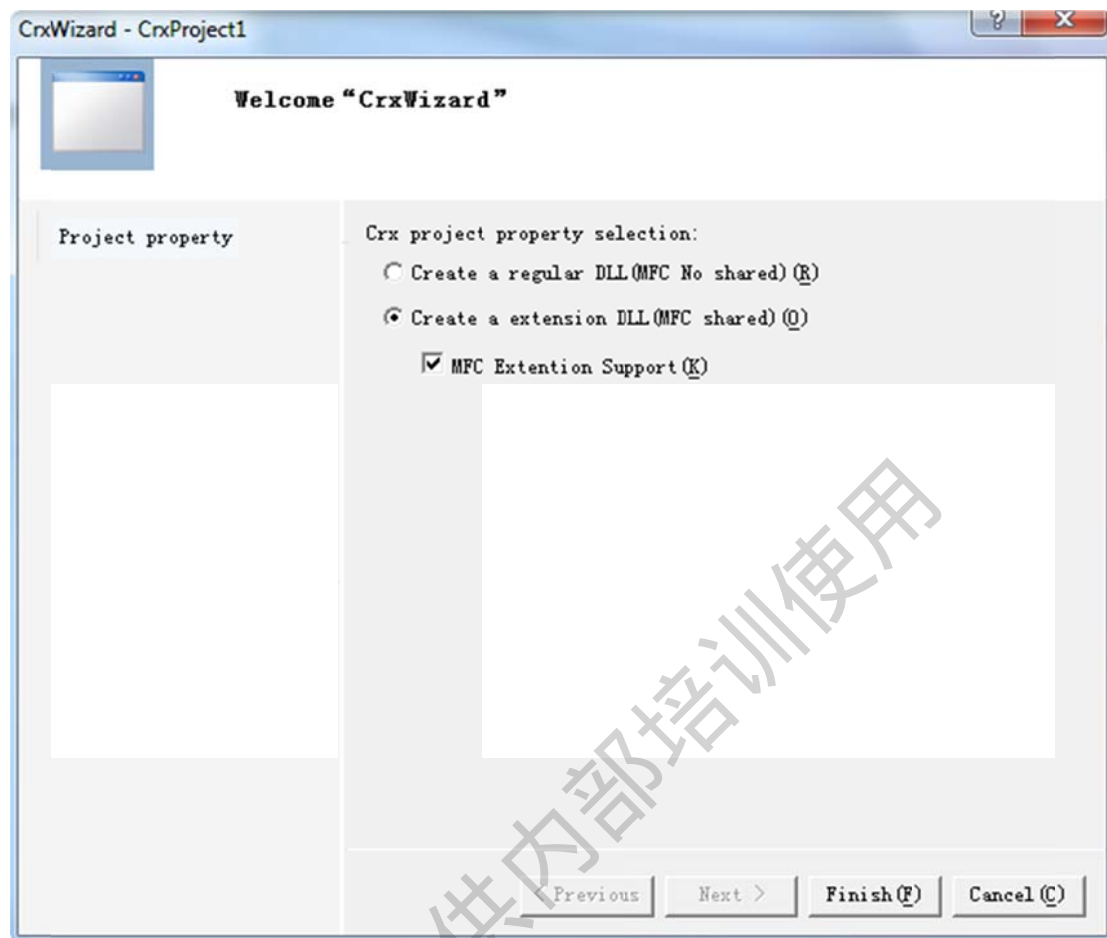


图1.7 创建空的动态链接库

(3) 向导所创建的动态链接库的基本信息，如图 1.8 所示

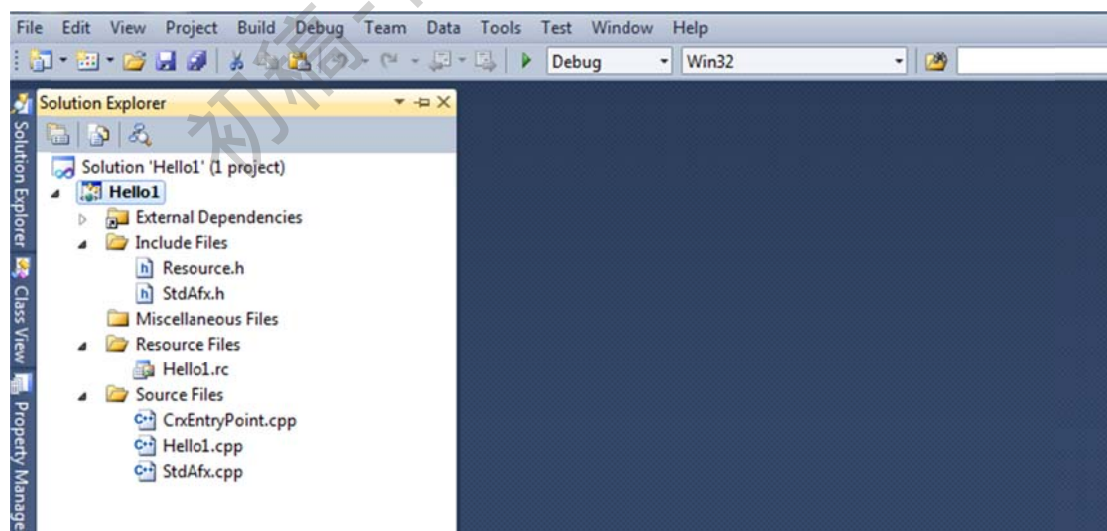


图 1.8 显示向导生成的 CRX 工程信息

创建的各个文件及其作用：

- StdAfx.h: 预编译头文件的创建。
- Resource.h: 资源标记的声明。
- Resource Files\Hello1.rc2: 工程的第二个资源文件。
- StdAfx.cpp: 预编译文件
- CrxEntryPoint.cpp 程序入口类所在的文件。其中包含两个函数On_kInitAppMsg和On_kUnloadAppMsg。On_kInitAppMsg实际上这里它只做一件事情, 就是使用电子图板的命令机制注册一个新命令。这个命令同电子图板的内部命令一样, 可以直接在命令行中执行。实际上, 这就是运行CRX 程序的方法。On_kUnloadAppMsg函数则在程序被卸载时所调用。此文件中包含addCommand 函数, 其封装在AddCommand 函数, 用于向电子图板注册命令。
- Hello1.cpp: 应用程序入口点。其中包含_hdllInstance 变量和DllMain 函数。_hdllInstance 变量是在DllMain 函数中使用, 一般不用注意。 DllMain 函数是动态链接库 (DLL) 的入口点函数, 一般来说不用注意。

(4)编译环境设置

项目要准确编译需要附加项目所需要的头文件和库文件。

在vs2010的Solution Explorer窗口选择Hello1项目, 右键选择“属性”, 弹出属性窗口, 在属性窗口中选择【C/C++ / General】, 然后在Additional Include Directories项中, 附加ObjectCRX 开发包下头文件所在的目录

C:\Program Files\CAXA\CAXA DRAFTMECHANICAL\2011\CRX\Inc,如下图1.9

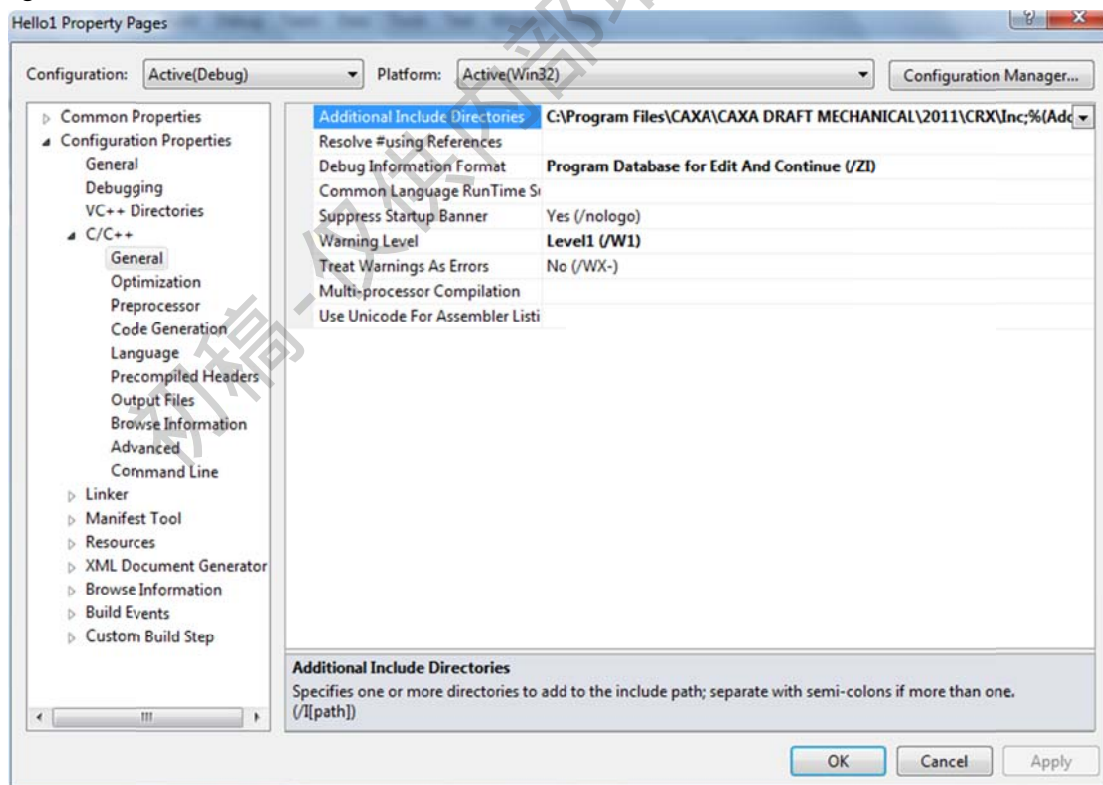
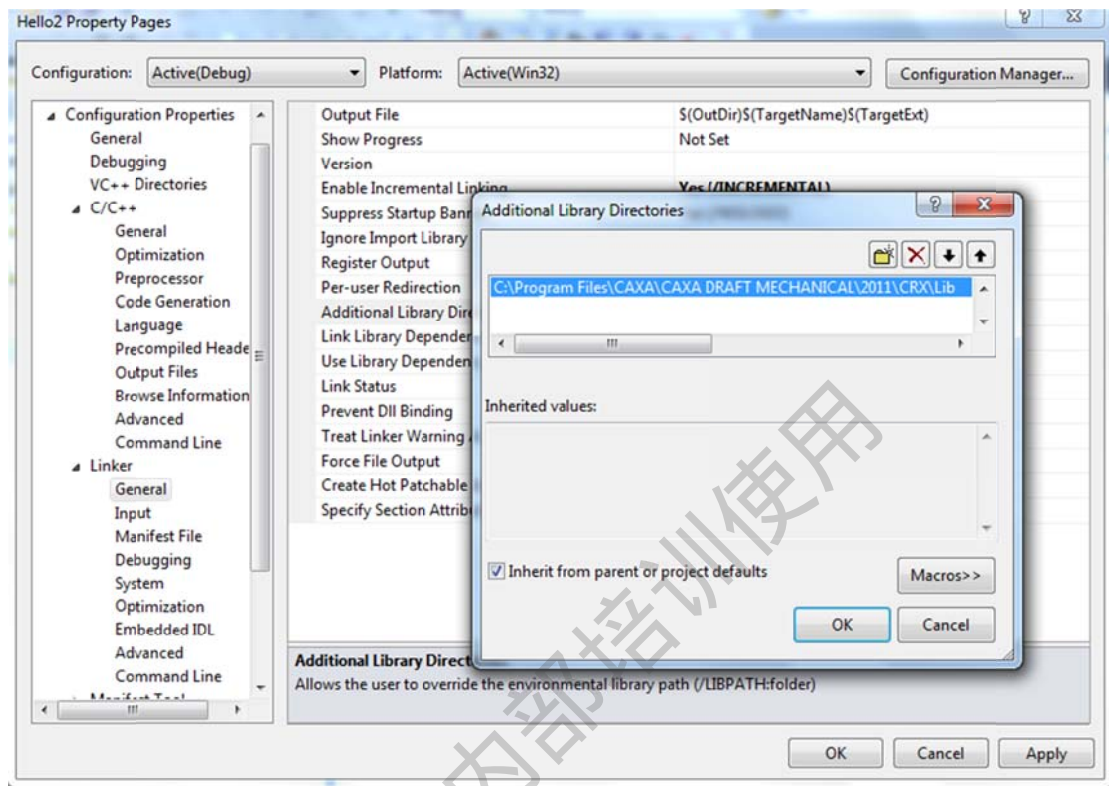


图 1.9 附加头文件目录

然后再属性窗口上选择【Linker\General】, 然后在 Additional Library Directories 项中, 附件库文件所在目录

C:\Program Files\CAXA\CAXA DRAFTMECHANICAL\2011\CRX\Lib,如下图1.10



1.10 附件库文件目录

注意：将 Configuration properties/General/Target Extension 参数设置为 .crx 能够避免默认工程编译出现警告。

(5) helloworld 函数如下：

```
void cmdHello()
{
    crxutPrintf(_T("\nHello World!"));
}
```

需要附件头文件如下：

```
#include "crxutcds.h"
```

(6) 注册 hello 命令。

需要在项目的 CrxEntryPoint.cpp 文件中，添加在电子图板中注册命令和卸载命令的代码，详细代码如下：

```
class CHello1App:public AcRxArxApp
{
public:
    CHello1App():AcRxArxApp()
    {
```

```

    }

    virtual AcRx::AppRetCode On_kInitAppMsg(void *pkt)
    {
        // TODO: Load dependencies here

        // You *must* call On_kInitAppMsg here
        AcRx::AppRetCode retCode = AcRxArxApp::On_kInitAppMsg(pkt);

        // TODO: Add your initialization code here
        //注册命令
        crxedRegCmds->addCommand(_T("HelloApp"),_T("GHello"),_T("Hello"),ACRX_CMD_MOD
        AL,&cmdHello);

        return (retCode);
    }

    virtual AcRx::AppRetCode On_kUnloadAppMsg(void *pkt)
    {
        // TODO: Add your code here

        // You *must* call On_kUnloadAppMsg here
        AcRx::AppRetCode retCode = AcRxArxApp::On_kUnloadAppMsg(pkt);

        // TODO: Unload dependencies here
        //卸载命令
        crxedRegCmds->removeGroup(_T("HelloApp"));
        return (retCode);
    }

    virtual void RegisterServerComponents()
    {
    }

};

```

1.3.4 效果

在 Visual Studio 2010 中，按下快捷键 F5 对程序进行调试，系统会自动启动电子图板 2011。使用 MANAGE 命令加载生成的 CRX 文件，在命令行执行 Hello 命令，能够得到如图 1.9 所示的结果。



图 1.9 程序运行的结果

1.3.5 小结

后面的程序全部使用向导来创建，规模也会更大，这里用向导生成简单的 Hello,World 程序，目的就是让读者经过分析掌握向导的辅助作用，而不至于在大量的辅助代码中迷失。从现在开始，我们要专注于 ObjectCRX 的精髓内容。