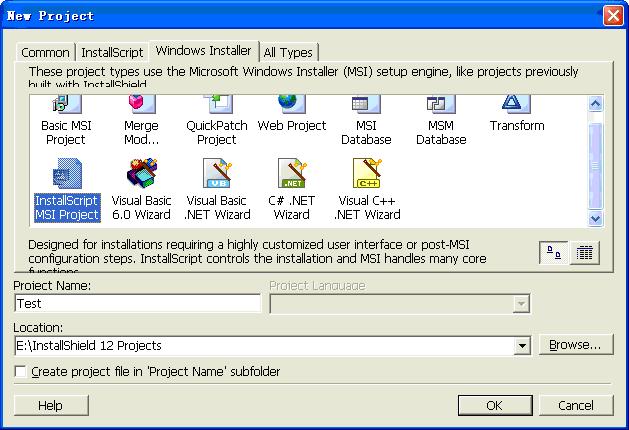
下面我们一步一步来建立一个基本的工程，并且使用脚本来完善和丰富所需功能

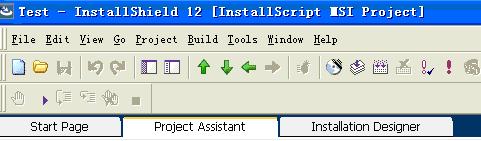
第一部分：基本设置

1.         打开Installshield 2010 Premier Edition，新建一个Installscript MSI Project，这种被称之为半脚本程序，因为兼具Basic Project基本类型和Installscript Project全脚本类型两者的优点，我比较喜欢用。像我这样需求的，既要用到Wizard的便利，又想写一点脚本来实现一点自定义操作的，就比较适合用这种类型啦。

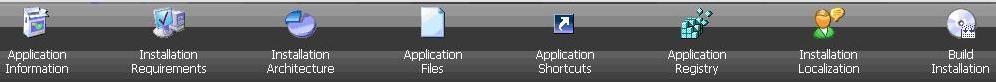
选择类型为Windows Installer | InstallScript MSI Projcet，输入工程名，指定工程所在的文件夹。



2.         界面会切换到Project Assistant，我们先从这里开始把工程的基本组件和基本文件建立好。



3.      在Project Assistant界面的底部，会有一个引导动作条，在建立该工程的基本结构和文件时，我们都将在此界面进行操作，下文都将以“引导条”来指代这个引导动作条。



4.         点击引导条上的Application Information

这里输入：

公司名，公司名将会出现在Setup.exe的注解中

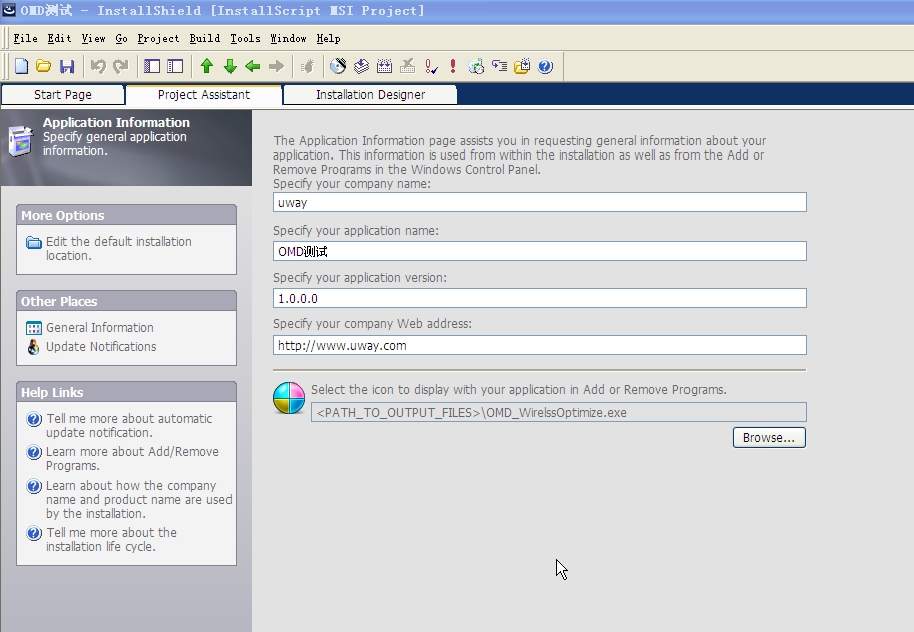
软件名，将会出现在安装过程的左上角标题栏上

版本号，没看到在哪，不过自己比较方便地知道自己在编译哪个版本的软件

公司网址，没看到在哪，而且如果该公司没有网址呢？而且这里有点bug，好像默认的值总是会报一个String\_ID1为空的错误，自己输入一个网址就不会报错。

是否在你创建了更新时自动通知最终用户，没用过，我都选了No。

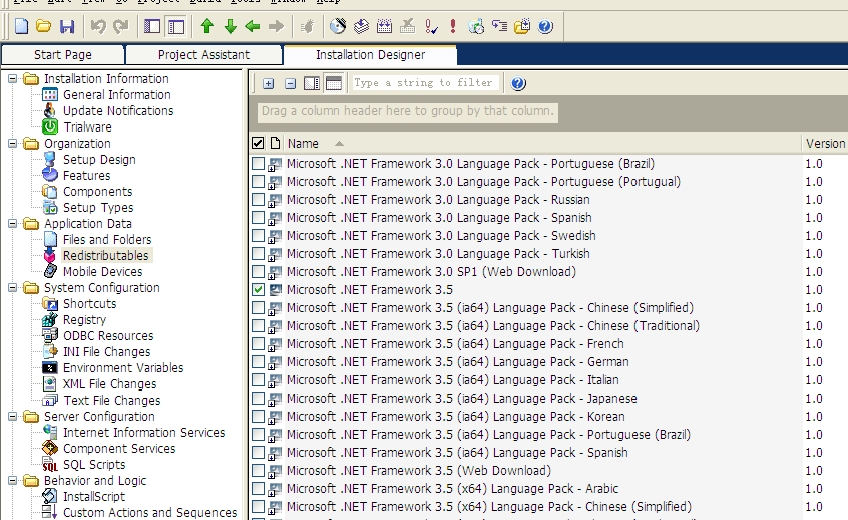
选择一个图标，这个图标会出现在“添加或删除程序”里，我一般用默认的，当然你可以替换成自己想要的图标。

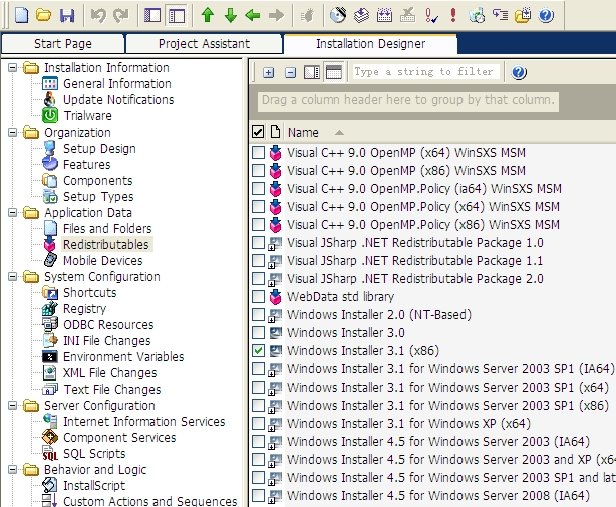


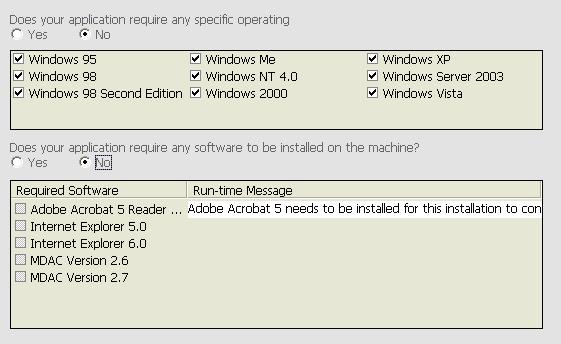
5.         点击引导条上的Installation Requirement

这里选择对操作系统和一些软件的需求。根据自己需要来选择是否要求操作系统的版本，已经是否要求安装了某些软件。

需要选中Windows Installer 3.1(x86) 和Mirosoft .NET Framework 3.5:







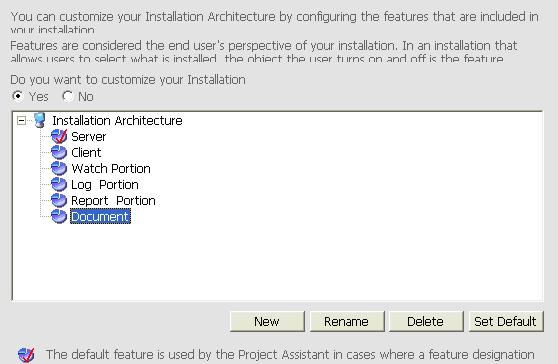
6.         点击引导条上的Installation Architecture

这是个十分有用的设置，对于本文所用的分布式软件来说非常合适，分布式软件的每个组件可以设置为一个Feature，用户可以自由选择安装某些功能。

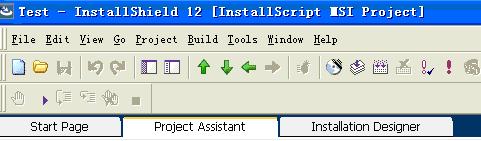
将选项Do you want to customize your Installation选择为Yes。

点击选中根节点Installation Architecture，点击New创建新的Feature，可以为每个Feature指定新名称。

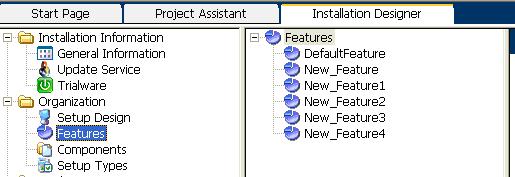
还可以在Feature下创建子Feature，比如如果文档Feature下包括软件本身文档，和软件所需的运行环境的文档，那么可以创建两个子Feature，分别包含两种文档，用户在安装时就可以选择安装部分或者全部文档了。这里我们没有用到子Feature，用途和普通Feature一样。



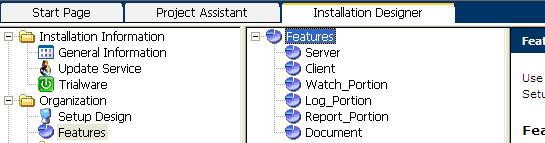
这里，建立好所有Feature后，我们将切换到Installation Designer做一个设置



找到Installation Designer页面上左边导航树Organization | Features分支，你会看到这里Features都显示为原始的名称，而非我们改过的名字，因为Feature有Name和Display Name两种名称，我们刚才改的不过是Display Name，为了便于查看和使用，我们在这里把Name也改一下



注意Name不可以有空格，可以使用下划线



7. 点击引导条上的Application Files

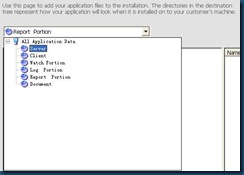
我们将在这里对安装路径进行微调，并且为每个Feature指定需要安装的文件

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image002_2.jpg)

这里我不想使用Program Files | Company Name | Product Name这个路径，我想使用Program Files | Product Name，我直接点击选中My Product Name[INSTALLDIR]拖动到ProgramFileFolder下，还可以直接将My Product Name 改成自己想要的文件夹名字

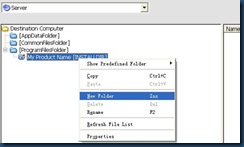
[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image004_2.jpg)

接下来，为每个Feature指定要安装的文件。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image006_2.jpg)

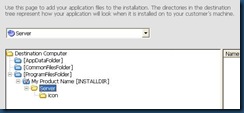
打开这个下拉列表，所有的Feature都在这里，按顺序来给每一个Feature建立文件夹，并且导入所需的文件。

选择第一个Feature， 即Server，点击My Product Name[INSTALLDIR]节点，右键点击，在菜单上选择New Folder来创建一个文件夹。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image008_2.jpg)

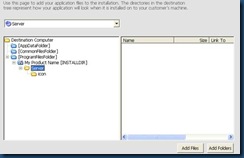
创建一个Server文件夹，这个文件夹将用来存放该组件需要的一些文件。

再在Server文件夹下创建一个icon文件夹，存放该组件所用的图标。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image010_2.jpg)

然后为该Feature添加安装时该Feature要安装的文件。

这里我们建立的icon文件夹是用来存放这个feature在后面要建立快捷方式时使用的图标的。为这个icon文件夹添加相应的图标文件，并且记住图标文件的来源文件夹，后面设置快捷方式的时候要用。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image012_2.jpg)

点击选中要添加文件的文件夹，然后点击右下角的Add Files，然后添加文件

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image014_2.jpg)

接下来我们为Feature添加文件夹，如果这个文件夹中的全部文件都为这个Feature所需。添加文件夹的好处在于只要文件夹位置和名称不变，那么文件夹里面的文件都是动态加载的，有多少加载多少，不用考虑文件名的改动带来的影响。

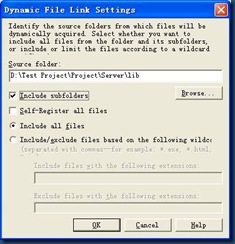
点击选中要添加文件夹的文件夹，然后点击右下角的Add Folders，然后添加文件夹。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image016_2.jpg)

选中文件夹，点击确定。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image018_2.jpg)

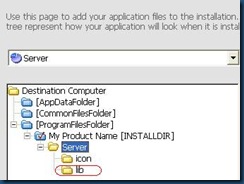
会询问你是否要使用动态文件链接，我都选择确定，好处就在于我刚才上面所述。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image020_2.jpg)

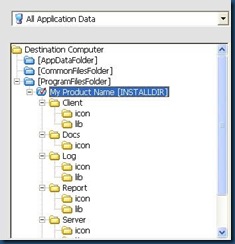
显示了源文件夹，如果这个文件夹下有子文件夹，并且也需要一并添加进来的话，务必钩选Include subfolders选项。

这里还允许做一些简单设置来包含或者排除一些特定文件，支持通配符。

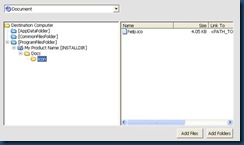
点击OK确定加入文件夹。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image022_2.jpg)

如法炮制为每个Feature建立文件夹，并且添加文件，最后效果如图所示

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image024_2.jpg)

Document这个Feature，除了文件所用的图标外，什么都不要添加，后面我们将用安装时实时拷贝的方式来拷贝文档进来。

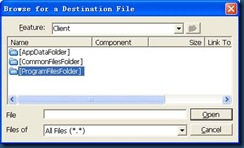
[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image026_2.jpg)

8. 接下来我们为可执行文件创建快捷方式。

点击引导条上的Application Shortcuts

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image028_2.jpg)

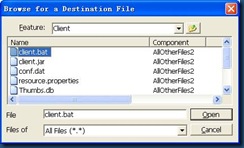
点击New新建一个快捷方式

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image030_2.jpg)

选择一个要建立快捷方式的Feature。

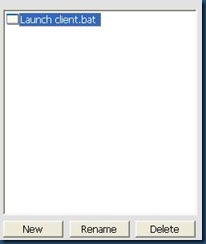
如果要建立快捷方式的程序为非.exe形式，请把Files Of选择选为All Files(\*.\*)格式。

我们的程序安装目标路径设置在Program Files下，因此双击[ProgramFilesFolder]打开，层层点击进入。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image032_2.jpg)

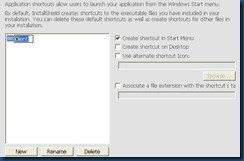
我们这里要为client.bat建立一个快捷方式，因为这个是启动用的批处理文件。

Installshield可以自动监测到.exe文件的存在，自动生成快捷方式，用户只需要做一些适当修改即可。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image034_2.jpg)

新建的快捷方式将出现在这里，名字不好听，样子也不好看，我们将为它改一个名字，并且换一个图标。

选中快捷方式，点击Rename，并且为这个快捷方式改一个适当的名字。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image036_2.jpg)

注意右边的几个选项。

Create shortcut in Start Menu，将在开始菜单里创建一个快捷方式。

Create shortcut on Desktop，将在桌面上创建一个快捷方式。

Use alternate shortcut Icon，替换快捷方式的图标

Associate a file extension with the shortcut’s target，没用过，不知道什么意思。

我们在这里将只创建开始菜单的快捷方式，因此钩选第一项。

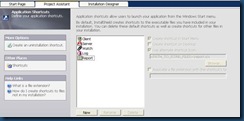
钩选第三项，并且点击Browse来浏览图标。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image038_2.jpg)

请回想刚才在为Feature添加文件的时候，每个feature都添加了对应的icon。这里，请把浏览的文件夹设定为刚才添加icons所用的文件夹，通俗的说，就是你刚才从哪儿添加一个图标进feature的，现在还是从哪儿添加的这个图标。

其实这一点我是一直很费解的，当初不知道要这么选择图标，随便从外面一个任意文件夹里添加了一个图标，以至于打包后死活找不到图标，后来经过试验才知道这个被选中的图标文件要拷贝进来，打包进安装文件才可以。这一点上不能不提一下visual studio，这个工具做安装程序虽然功能一般，但是思想还是不错的，当它的组件指定拷贝了图标文件后，在建立快捷方式时，快捷方式使用的图标是指向虚拟的安装目标路径下的图标文件的，而不是指定到这个实实在在的源文件夹。这一点差别就体现出了思想上的差异。

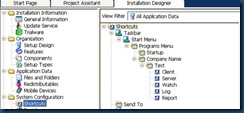
如法炮制为每个Feature指定快捷方式，Document除外，因为我们在这个feature里除了图标文件外什么都没有添加。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image040_2.jpg)

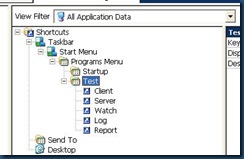
至此我们为每个可执行程序添加了开始菜单下的快捷方式。

我们再切换去Installation Designer，找到System Configuration | Shortcuts。

看到快捷方式在开始菜单中是以　公司名　｜　软件名　｜　快捷方式　　　这种形式存在的。事实上我是不喜欢这种形式了，想想点开一层还有一层，不如直接了当来得干脆，因此做一些修改。

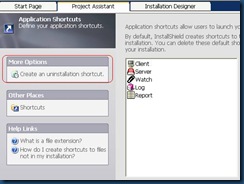
[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image042_2.jpg)

这里我改成了如下设置

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image044_2.jpg)

不要告诉我你不会改，直接拖动Test文件夹往Program Menu（即开始菜单下的那个“所有程序”）下一塞即可，然后删除掉多余的Company Name文件夹。

9. 可能刚才在Project Assistant界面有人已经注意到了左边栏上More Options下Create an uninstallation shortcut这个诱人的字样了。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image046_2.jpg)

可是我要告诉你，如果你选择了这种方式建立卸载快捷方式的话，你会很沮丧地发现：

a) 似乎只有在安装某个feature的时候这个卸载快捷方式才会出现（当然，就是那个default feature，这种要命的feature形式决定了每个文件或者快捷方式都必须明确地归属到某个feature下），因此，当你的客户只选择了其他feature安装时，这个卸载方式不会出现，而他必须去“添加或卸载程序”里面去卸载

b) 如果你写脚本使得安装时会拷贝一些外部文件进来，那么这些文件在这种卸载方式下是删除不掉的。（如果你确实想保存这些文件，你可以在脚本里设置它们属性为permanent，这个属性可以保证什么卸载方式都不能删除你的这些文件）。

所以这里我们忽视这个卸载快捷方式的存在，而将在后面采用脚本形式实现完美卸载。

10. 点击引导条上的Application Registry

向注册表写键和键值，由于本工程不需要，忽略之。有需要的朋友可以查阅相关资料，不难。

11. 点击引导条上的Installation Location

这个是用来设置安装包的语言的，选择了多个语言后，用户可以在安装界面开始的时候选择安装时所用的语言；不过作为一个公司产品来说，这么偷懒，客户的印象是要打折扣的，所以还是选个单语言吧，该什么语言的安装包就什么语言的安装包，各归各。

不过你又会沮丧地发现，如果要选择一种其他语言作为Default Language，好像又报错了。

这个问题当时折腾了我一个星期（当然那时候才接触Installshield，还一窍不通），最后问了技术支持才得以解决。

切换去Installation Designer，找到Installation Information | General Information，看到String Tables下面是什么？对，所有你选的语言都列出来了，选中你要的语言，右键，选择Make Default，OK，再切换回Project Assistant去把所有不要的语言统统去掉钩选即可。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image048_2.jpg)

看到此处，已经变成了English为默认语言了。

[](http://images.cnblogs.com/cnblogs_com/Cindy_weiwei/WindowsLiveWriter/e68d4c8b8326_D397/clip_image050_2.jpg)

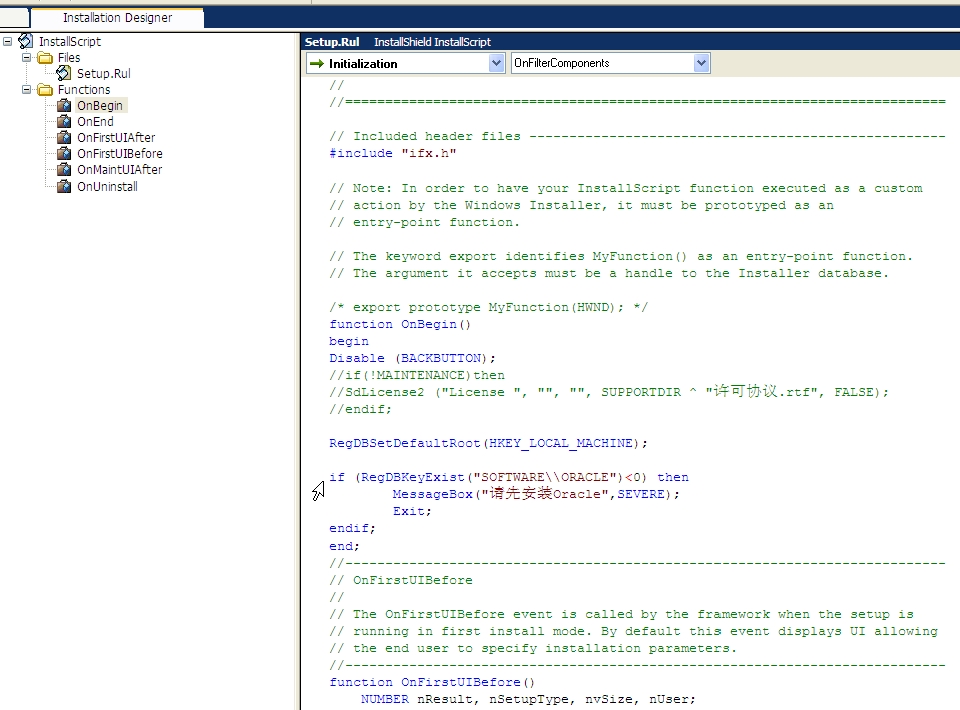
12. 点击引导条上的Build Installation。

打包安装盘的设置，本人从来不用这个选项，都用工具条上的Release Wizard。

至此，第一部分基本完成。如果是一些没有特别要求的安装包，这部分讲解的内容足够可以做一个基本的安装包了

**第二部分：脚本编程**

**下面就是我的脚本的主要实现：**

****

**脚步主要包括六个部分，即六个事件处理过程：OnBegin,OnEnd,OnFirstUIAfter,OnFirstUIBefore,OnMainUIAfter,OnUninstall这六个事件都是自动生成的，需要那些特殊的处理就这写事件里面写脚步就可以了。onEnd和OnUninstall两个事件我没有用到，很奇怪的，卸载代码写在OnUninstall里面没有启作用，没有运行，所以我不得不写在OnMainUIAfter事件里面。另外需要提示的是unistall shield自带的卸载快捷方式不好用，卸载不完全，所以必须写脚步才能卸载完全。**

**下面就是我做的具体的脚本：**

**1：开始部分：**

**//===========================================================================**

**//**

**// File Name: Setup.rul**

**//**

**// Description: Blank setup main script file**

**//**

**// Comments: Blank setup is an empty setup project. If you want to**

**// create a new project via. step-by step instructions use the**

**// Project Assistant.**

**//**

**//===========================================================================**

**// Included header files ----------------------------------------------------**

**#include "ifx.h"**

**// Note: In order to have your InstallScript function executed as a custom**

**// action by the Windows Installer, it must be prototyped as an**

**// entry-point function.**

**// The keyword export identifies MyFunction() as an entry-point function.**

**// The argument it accepts must be a handle to the Installer database.**

**/\* export prototype MyFunction(HWND); \*/**

**function OnBegin()**

**begin**

**Disable (BACKBUTTON);**

**//if(!MAINTENANCE)then**

**//SdLicense2 ("License ", "", "", SUPPORTDIR ^ "许可协议.rtf", FALSE);**

**//endif;**

**RegDBSetDefaultRoot(HKEY\_LOCAL\_MACHINE);**

**if (RegDBKeyExist("SOFTWARE\\ORACLE")<0) then**

**MessageBox("请先安装Oracle",SEVERE);**

**Exit;**

**endif;**

**end;**

**2OnFirstUIBefore事件部分：**

**//---------------------------------------------------------------------------**

**// OnFirstUIBefore**

**//**

**// The OnFirstUIBefore event is called by the framework when the setup is**

**// running in first install mode. By default this event displays UI allowing**

**// the end user to specify installation parameters.**

**//---------------------------------------------------------------------------**

**function OnFirstUIBefore()**

**NUMBER nResult, nSetupType, nvSize, nUser;**

**STRING szTitle, szMsg, szQuestion, svName, svCompany, szFile;**

**STRING szLicenseFile;**

**LIST list, listStartCopy;**

**BOOL bCustom, bIgnore1, bIgnore2;**

**STRING szDir,szTemp;**

**begin**

**// TO DO: if you want to enable background, window title, and caption bar title**

**// SetTitle( @PRODUCT\_NAME, 24, WHITE );**

**// SetTitle( @PRODUCT\_NAME, 0, BACKGROUNDCAPTION );**

**// Enable( FULLWINDOWMODE );**

**// Enable( BACKGROUND );**

**// SetColor(BACKGROUND,RGB (0, 128, 128));**

**// Added in InstallShield 15 - Show an appropriate error message if**

**// -removeonly is specified and the product is not installed.**

**if( REMOVEONLY ) then**

**Disable( DIALOGCACHE );**

**szMsg = SdLoadString( IDS\_IFX\_ERROR\_PRODUCT\_NOT\_INSTALLED\_UNINST );**

**SdSubstituteProductInfo( szMsg );**

**MessageBox( szMsg, SEVERE );**

**abort;**

**endif;**

**nSetupType = TYPICAL;**

**Dlg\_SdWelcome:**

**szTitle = "欢迎使用UWAY-OMD GB优化系统";**

**szMsg = "安装软件将在您的计算机中安装 'UWAY-OMD GB优化系统' 若要继续，请单击下一步";**

**nResult = SdWelcome(szTitle, szMsg);**

**if (nResult = BACK) goto Dlg\_SdWelcome;**

**szTitle = "";**

**svName = "";**

**svCompany = "";**

**Dlg\_SdRegisterUser:**

**szMsg = "请您填写用户信息";**

**szTitle = "欢迎您\b\n使用UWAY-OMD GB优化系统";**

**nResult = SdRegisterUser( szTitle, szMsg, svName, svCompany );**

**if (nResult = BACK) goto Dlg\_SdWelcome;**

**Dlg\_SetupType:**

**szTitle = "安装类型\b\n请选择安装类型";**

**szMsg = "请选择安装";**

**nResult = SetupType2(szTitle, szMsg, "", nSetupType, 0);**

**if (nResult = BACK) then**

**goto Dlg\_SdRegisterUser;**

**else**

**nSetupType = nResult;**

**if (nSetupType != CUSTOM) then**

**nvSize = 0;**

**FeatureCompareSizeRequired(MEDIA, INSTALLDIR, nvSize);**

**if (nvSize != 0) then**

**MessageBox(szSdStr\_NotEnoughSpace, WARNING);**

**goto Dlg\_SetupType;**

**endif;**

**bCustom = FALSE;**

**goto Dlg\_SQL;**

**else**

**bCustom = TRUE;**

**endif;**

**endif;**

**Dlg\_SdAskDestPath:**

**nResult = SdAskDestPath(szTitle, szMsg, INSTALLDIR, 0);**

**if (nResult = BACK) goto Dlg\_SetupType;**

**Dlg\_SdFeatureTree:**

**szTitle = "";**

**szMsg = "";**

**if (nSetupType = CUSTOM) then**

**nResult = SdFeatureTree(szTitle, szMsg, INSTALLDIR, "", 2);**

**if (nResult = BACK) goto Dlg\_SdAskDestPath;**

**endif;**

**Dlg\_SQL:**

**nResult = OnSQLLogin( nResult );**

**if( nResult = BACK ) then**

**if (!bCustom) then**

**goto Dlg\_SetupType;**

**else**

**goto Dlg\_SdFeatureTree;**

**endif;**

**endif;**

**Dlg\_SdStartCopy:**

**szTitle = "可以安装UWAY-OMD GB优化系统了\b\n安装准备工作已就绪，可以开始安装了。";**

**szMsg ="";// "单击"安装"以开始安装。\b\n如果要检查或更改任何安装设置，请单击"上一步"。单击"取消"退出安装向导。";**

**nResult = SdStartCopy2( szTitle, szMsg );**

**if (nResult = BACK) then**

**goto Dlg\_SQL;;**

**endif;**

**listStartCopy = ListCreate( STRINGLIST );**

**switch (nSetupType)**

**case TYPICAL : ListAddString(listStartCopy,"Setup Type: 典型",AFTER);**

**case COMPACT: ListAddString(listStartCopy,"Setup Type: 压缩",AFTER);**

**case CUSTOM: ListAddString(listStartCopy,"Setup Type: 自定义",AFTER);**

**endswitch;**

**ListAddString(listStartCopy," ",AFTER);**

**ListDestroy(listStartCopy);**

**// Added in IS 2009 - Set appropriate StatusEx static text.**

**SetStatusExStaticText( SdLoadString( IDS\_IFX\_STATUSEX\_STATICTEXT\_FIRSTUI ) );**

**// setup default status**

**Enable(STATUSEX);**

**return 0;**

**end;**

**3： OnFirstUIAfter事件部分**

**//---------------------------------------------------------------------------**

**// OnFirstUIAfter**

**//**

**// The OnFirstUIAfter event called by the framework after the file transfer**

**// of the setup when the setup is running in first install mode. By default**

**// this event displays UI that informs the end user that the setup has been**

**// completed successfully.**

**//---------------------------------------------------------------------------**

**function OnFirstUIAfter()**

**STRING szTitle, szMsg1, szMsg2, szOpt1, szOpt2;**

**BOOL bOpt1, bOpt2;**

**STRING szSrcFile1;**

**STRING szSrcFile2;**

**STRING szTarFolder1;**

**STRING szTarFolder2;**

**STRING szOption1;**

**STRING szOption2;**

**STRING szExeFile;**

**STRING szDocFile;**

**NUMBER nResult;**

**STRING szfilename,szFolder ,szmsg1,szmsg2;**

**NUMBER nresult;**

**STRING svStartDir;**

**NUMBER nSize,nRet;**

**STRING szFeatureName1 ;**

**STRING szFeatureName2 ;**

**STRING szFeatureName3 ;**

**STRING szFeatureName4 ;**

**STRING szDir1;**

**STRING szDir2;**

**begin**

**Disable(STATUSEX);**

**szFeatureName1 ="DB\_TS";**

**szFeatureName2 ="DB\_TD";**

**szFeatureName3 ="MAP";**

**szFeatureName4 ="vcredist\_x86";**

**nSize=MAX\_PATH;**

**MsiGetProperty ( ISMSI\_HANDLE , "StartDir" , svStartDir , nSize);**

**szDir1 = TARGETDIR;**

**if (0 = ExistsDir(szDir1)) then**

**StrSub(szDir2,szDir1,0,StrLength(szDir1)-1);**

**endif ;**

**if (FeatureIsItemSelected(MEDIA, szFeatureName1)=1) then**

**XCopyFile (svStartDir^"ConfigFiles\\CreateDbTableSpace.bat" , szDir2 , COMP\_NORMAL );**

**XCopyFile (svStartDir^"ConfigFiles\\db\_TableSpace.sql" , szDir2, COMP\_NORMAL );**

**LaunchAppAndWait ( TARGETDIR^"CreateDbTableSpace.bat" , "" ,LAAW\_OPTION\_WAIT );**

**DeleteFile (TARGETDIR^"CreateDbTableSpace.bat");**

**DeleteFile (TARGETDIR^"db\_TableSpace.sql");**

**endif;**

**if (FeatureIsItemSelected(MEDIA, szFeatureName2)=1) then**

**XCopyFile (svStartDir^"ConfigFiles\\DmpTableData.bat" , szDir2 , COMP\_NORMAL );**

**XCopyFile (svStartDir^"ConfigFiles\\omd.dmp" , szDir2 , COMP\_NORMAL );**

**XCopyFile (svStartDir^"ConfigFiles\\dbImpUser.bat" , szDir2 , COMP\_NORMAL );**

**LaunchAppAndWait ( TARGETDIR^"DmpTableData.bat" , "" ,LAAW\_OPTION\_WAIT );**

**DeleteFile (TARGETDIR^"DmpTableData.bat");**

**DeleteFile (TARGETDIR^"omd.dmp");**

**DeleteFile (TARGETDIR^"dbImpUser.bat");**

**endif;**

**if (FeatureIsItemSelected(MEDIA, szFeatureName3)=1) then**

**if (RegDBKeyExist ("SOFTWARE\\MapInfo\\MapXtreme\\6.8.0") < 0) then**

**LaunchAppAndWait (svStartDir^"ConfigFiles\\MapInfo\\mapxtremeTRIAL.exe" , "" ,LAAW\_OPTION\_WAIT );**

**XCopyFile (svStartDir^"ConfigFiles\\MapInfo\\mapxtreme680.lic" , "C:\\Program Files\\Common Files\\MapInfo\\MapXtreme\\6.8.0" , COMP\_NORMAL );**

**DeleteFile ( INSTALLDIR^"ConfigFiles\\MapInfo\\mapxtreme680.lic" );**

**endif;**

**endif;**

**if (FeatureIsItemSelected(MEDIA, szFeatureName4)=1) then**

**LaunchAppAndWait ( svStartDir^"ConfigFiles\\vcredist\_x86.exe" , "" ,LAAW\_OPTION\_WAIT );**

**endif;**

**szfilename = UNINSTALL\_STRING +" /UNINSTALL";**

**nresult = StrFind(szfilename,".exe");**

**if nresult >=0 then**

**StrSub(szmsg1,szfilename,0,nresult + 4);**

**StrSub(szmsg2,szfilename,nresult + 4,200);**

**LongPathToQuote(szmsg1, FALSE );**

**LongPathToQuote(szmsg2, FALSE );**

**szfilename = "\"" + szmsg1 + "\"" +szmsg2;**

**endif;**

**AddFolderIcon(FOLDER\_PROGRAMS^"uway\\OMD","Uninstall",szfilename,WINDIR,"",0,"",REPLACE);**

**bOpt1 = FALSE;**

**bOpt2 = FALSE;**

**if ( BATCH\_INSTALL ) then**

**SdFinishReboot ( szTitle , szMsg1 , SYS\_BOOTMACHINE , szMsg2 , 0 );**

**else**

**SdFinish ( szTitle , szMsg1 , szMsg2 , szOpt1 , szOpt2 , bOpt1 , bOpt2 );**

**endif;**

**end;**

**4: OnMaintUIAfter()事件部分：**

**function OnMaintUIAfter()**

**STRING szTitle, szMsg1, szMsg2, szOpt1, szOpt2,szDirectory1,szDirectory2;**

**NUMBER bOpt1, bOpt2;**

**begin**

**Disable(STATUSEX);**

**if( REMOVEALLMODE ) then**

**szTitle = SdLoadString(IFX\_SDFINISH\_REMOVE\_TITLE);**

**szMsg1 = SdLoadString(IFX\_SDFINISH\_REMOVE\_MSG1);**

**szTitle="卸载完毕";**

**szMsg1="已完全卸载UWAY-OMD GB优化系统";**

**//删除该程序文件**

**/\* szDirectory1 = TARGETDIR;**

**if (0 = ExistsDir(szDirectory1)) then**

**StrSub(szDirectory2,szDirectory1,0,StrLength(szDirectory1)-1);**

**DeleteDir (szDirectory2,ALLCONTENTS);**

**endif ; \*/**

**szDirectory1 = TARGETDIR;**

**if (0 = ExistsDir(szDirectory1)) then**

**StrSub(szDirectory2,szDirectory1,0,StrLength(szDirectory1)-1);**

**DeleteDir (TARGETDIR^"CardAlarm",ALLCONTENTS);**

**DeleteDir (TARGETDIR^"GPP",ALLCONTENTS);**

**DeleteDir (TARGETDIR^"images",ALLCONTENTS);**

**DeleteDir (TARGETDIR^"interfaceassoc",ALLCONTENTS);**

**DeleteDir (TARGETDIR^"me\_main",ALLCONTENTS);**

**DeleteDir (TARGETDIR^"zh-chs",ALLCONTENTS);**

**DeleteFile (TARGETDIR^"DevExpress.Data.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.Utils.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraGrid.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraLayout.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraEditors.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraBars.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraSpellChecker.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraSpellChecker.v9.1.Core.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraScheduler.v9.1.Extensions.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraScheduler.v9.1.dll");**

**DeleteFile (TARGETDIR^"DevExpress.XtraScheduler.v9.1.Core.dll");**

**DeleteFile (TARGETDIR^"OMD\_WirelssOptimize.exe");**

**DeleteFile (TARGETDIR^"OMD\_WirelssOptimize.exe.config");**

**DeleteFile (TARGETDIR^"System.Core.dll");**

**DeleteFile (TARGETDIR^"VerifyDll.dll");**

**DeleteFile (TARGETDIR^"zccfg.ini");**

**DeleteFile (TARGETDIR^"omd.log");**

**DeleteFile (TARGETDIR^"License.dat");**

**endif ;**

**// 在桌面上删除快捷方式**

**DeleteFolderIcon (FOLDER\_DESKTOP, "OMD");**

**//删除程序中的快捷方式**

**DeleteFolderIcon(FOLDER\_PROGRAMS^"uway\\OMD","OMD");**

**else**

**szTitle = SdLoadString(IFX\_SDFINISH\_MAINT\_TITLE);**

**szMsg1 = SdLoadString(IFX\_SDFINISH\_MAINT\_MSG1);**

**endif;**

**bOpt1 = FALSE;**

**bOpt2 = FALSE;**

**szMsg2 ="谢谢您使用UWAY-OMD GB优化系统。";**

**if ( BATCH\_INSTALL ) then**

**SdFinishReboot ( szTitle , szMsg1 , SYS\_BOOTMACHINE , szMsg2 , 0 );**

**else**

**SdFinish ( szTitle , szMsg1 , szMsg2 , szOpt1 , szOpt2 , bOpt1 , bOpt2 );**

**endif;**

**end;**