[VS2010 C/S模式winForm打包部署.详细!](http://blog.csdn.net/xhf55555/article/details/7702212)

分类： [.net开发](http://blog.csdn.net/xhf55555/article/category/1055022)2012-06-29 14:14 27621人阅读 [评论](http://blog.csdn.net/xhf55555/article/details/7702212#comments)(30) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/xhf55555/article/details/7702212#report)

[winform](http://www.csdn.net/tag/winform)[2010](http://www.csdn.net/tag/2010)[installer](http://www.csdn.net/tag/installer)[windows](http://www.csdn.net/tag/windows)[microsoft](http://www.csdn.net/tag/microsoft)[web服务](http://www.csdn.net/tag/web%e6%9c%8d%e5%8a%a1)

先说说什么是打包部署和发布,对于c/s模式打包就是把我们的程序打成压缩包,打包将行李入包,发布是将打好的包展开.  这样的好处一是保护版权和安装方面,二是打包会把cs文件编译成为dll文件,第一次运行的速度快,同时保护代码.三是打包发布可以节省控件,基本解决了安全性的问题,使得程序员的原codes不被泄漏.

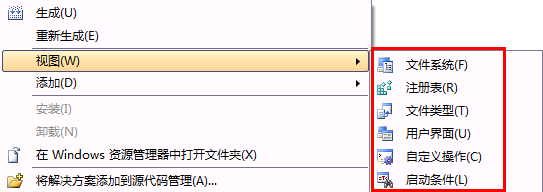
说说打包和发布的区别,发布是没有安装文件的,要求服务器上必须安装上一个frontpage server extension,发布是把页面文件aspx,资源文件放到网站里,代码编译到程序集中放到bin中.

打包部署,就是打成一个msi文件,运行后将在服务器上自动新建一个虚拟的目录,把文件的内容拷贝进去,拷贝的内容是你在打包中指定的.

下面就详细讲述VS C/S模式winForm打包部署.

**基础.**

安装项目的六个子项:文件系统注册表文件类型用户界面 自定义操作启动条件.



我们最常用的是**文件系统**了.

也就是**“文件系统编辑器”**

**“文件系统编辑器”**中的文件夹可以包含文件、项目输出和程序集。 项目输出表示解决方案中另一个项目所包含的项，并且可以包含以下内容：主要的生成输出（如可执行文件）、本地化资源、符号调试信息、内容文件（如 HTML 页）和项目源文件。其中的每个输出都称为项目输出组，项目输出组包含主要输出（也称为键输出）和任何附加的输出与依赖项。还支持快捷方式的创建，这样，您可以将文件放置在一个文件夹中，然后从桌面或另一文件夹中的快捷方式指向这些文件。

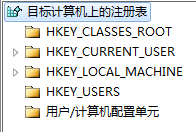
**文件系统**下有三个文件夹,"应用程序文件夹","用户的'程序'菜单","用户桌面".

**“桌面文件夹”**相当于“桌面”系统文件夹。 Windows会跟踪系统文件夹的位置，因此无论文件夹的位置或名称如何，放置在 **“桌面文件夹”**中的文件最后始终位于“桌面”系统文件夹中。

"**应用程序文件夹**":表示要安装的程序需要添加的文件;可以在 **“应用程序文件夹”**下创建“应用程序数据”文件夹，这样不论 **“应用程序文件夹”**位于目标计算机上的什么位置，放置在“应用程序数据”文件夹中的文件都始终安装在同一相对位置。

"**用户的'程序'菜单**":表示应用程序安装完毕,用户的"开始菜单"中显示的内容.

**注册表.**



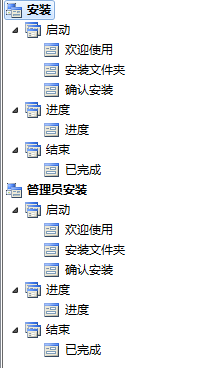
可以向注册表中插入键,在安装应用程序的同时创建注册表项.

**文件类型:**

右击可"添加文件类型"

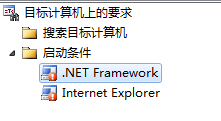
**“文件类型编辑器”**包含一个窗格，其中显示了文件类型和操作的分层列表。 默认情况下， **“文件类型编辑器”**包含一个 **“目标计算机上的文件类型”**节点；新文件类型可以添加到该节点的下面。

**用户界面:**



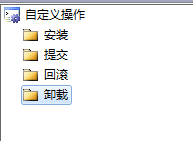
系统自动提供一组在安装期间显示的默认的对话框.根据正在构建的项目的类型,所提供的这组对话树中包含两部分,安装与管理员安装.安装部分包括终端用户运行安装程序时显示的对话框,而管理员安装部分包含系统管理员将安装程序上传到某个网络位置时显示的对话框.如果想要构建一个没有用户界面的安装,只需要在对话框树上删除安装部分中的所有对话框.或者从一组可用模版添加到安装项目中.

**启动条件:**



使用启动条件编辑器来定义成功运行安装所必须的条件.例如,特地你个版本的操作系统,配置启动条件来搜索目标计算机上的某个特定文件\注册表\或组件,在安装时进行评估并按照启动条件编辑器中所示的顺序执行.

**自定义操作:**

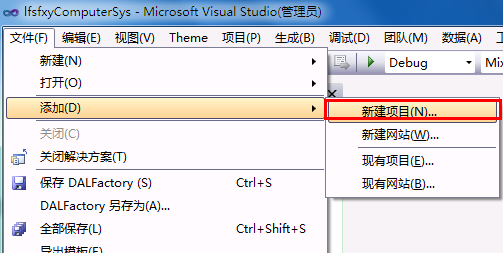


"**自定义操作**"提供了一种在安装结束时执行任务的机制。例如，您可能希望在安装应用程序之前先配置好数据库，此时您可以用自定义操作来完成数据库的创建。自定义操作构建为单独的  .dll  或   .exe  文件，然后使用自定义操作编辑器添加到项目中，其中这些操作与不同的安装阶段有关。自定义操作编辑器包括几个单独的文件夹，用于放置执行   Install、Commit、Rollback或Uninstall 任务的二进制文件。

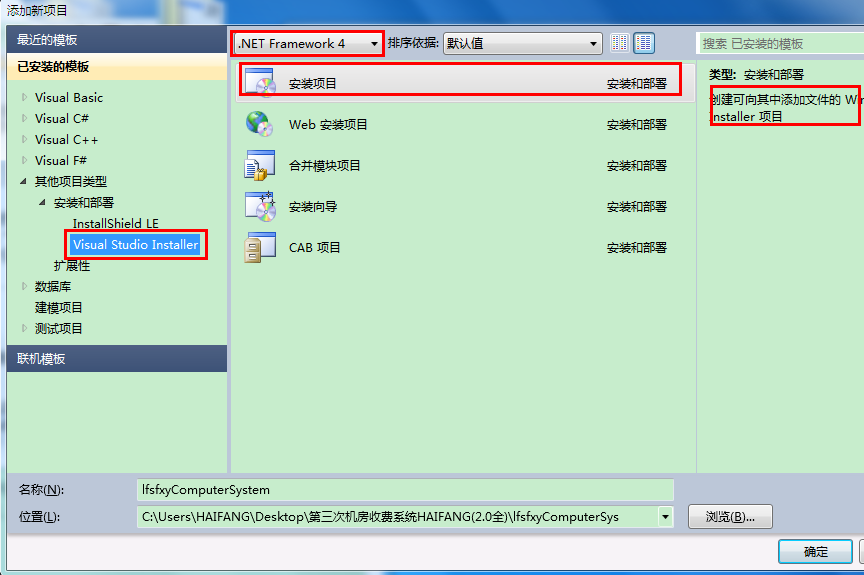
**操作演练.**

**创建新部署项目.**

1. 在"文件"菜单上指向"添加",然后单击"新建项目".



1. 我们随即出现的"添加新建项目"的对话框中,打开"项目类型"窗格中的"其他项目类型"节点,打开"安装和部署项目",在选择"Visual Studio Installer "



3.在"模版"窗格中,选择要创建的部署项目的类型.vs为我们提供了四种类型的部署项目提供了模版:合并模版项目安装项目 web安装项目和CAB项目.

这四种模版,

* **安装项目** 主要是通过使用InstallShield 2010 Limited Edition来创建、生成并打包windows应用程序和服务.
* **合并模块项目安装**,可以利用"合并模块"项目将文件或组建打包到单个模块中,以便共享此模块.产生的.msm文件可以包含任何其他部署项目中.
* **web安装**,为web应用程序生成安装程序,web安装项目的文件安装在web服务器上的virtual Root 目录中.
* **CAB项目**,创建cab文件,使用CAB项目可以从Web服务器下载到Web浏览器的ActiveX组建进行打包.

在这里我们使用的是"安装项目"模版,最终生成的是.msi文件,包含应用程序、任何依赖文件以及有关程序的信息,如注册表项和安装说明等.当msi文件在另一个台计算机上分发运行时,安装失败,计算机则会返回到安装前的状态.

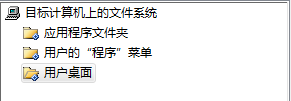
**向解决方案中添加现有的部署项目**

在"文件"菜单上指向"添加",然后单击"现有项目".

添加了解决方案sln的文件.

**将windows的应用程序添加到安装程序中.**

在我们建立一个"安装项目"后,你会看到这样的三个文件夹.(在文章的开始时已做讲述.)



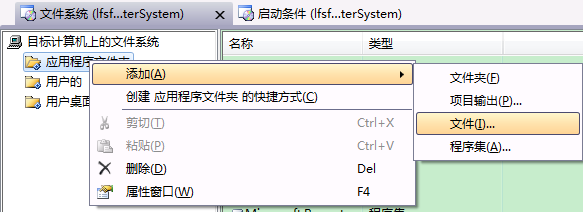
这三个文件夹依次是:

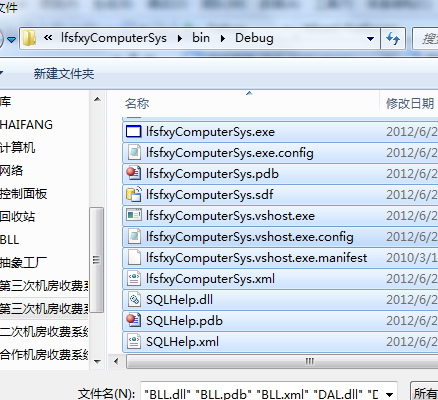
"应用程序文件夹":表示要安装的程序需要添加的文件;

"用户的'程序'菜单":表示应用程序安装完毕,用户的"开始菜单"中显示的内容,一般这个文件夹中,需要创建一个文件用来存放:应用程序.exe和卸载.exe.

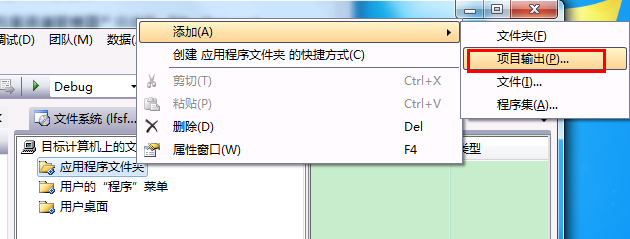
"用户桌面":表示这个应用程序安装完,用户的桌面上的创建.exe的快捷方式.

在应用程序文件夹节点右击,指向"文件",添加要打包的文件,添加的文件一般是程序的Debug目录下的文件;

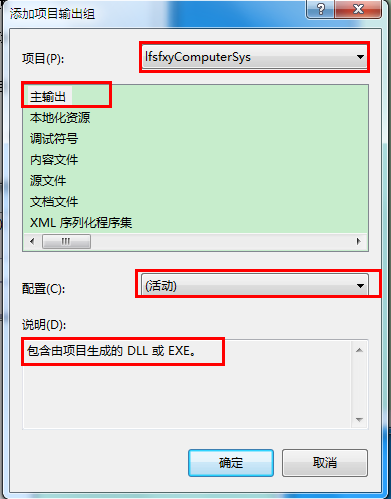


****

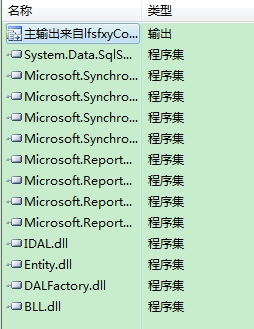
在应用程序文件夹节点右击,指向"添加",然后单击"项目输出".



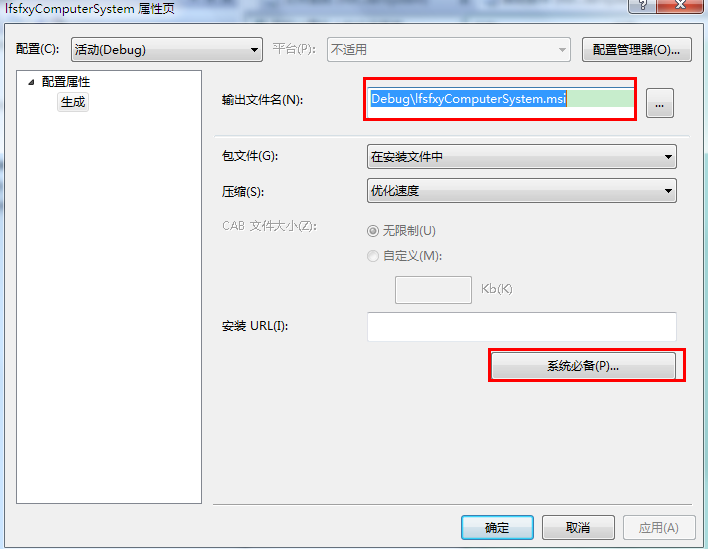
在列表中选择"主输出"组.在"配置"框中选择"(活动)".单击确定.

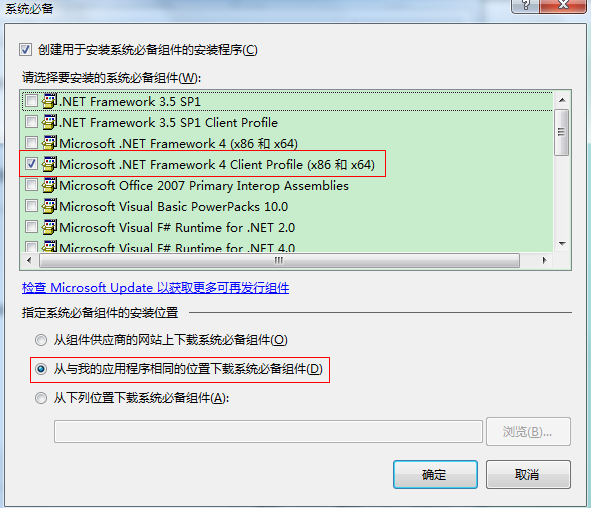


我们可以看到自动加载的依赖项.



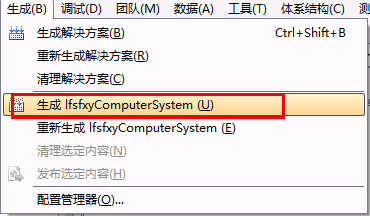
为启动条件设置初值.右击我们的安装文件,选择"属性"





这样安装包就会打包.NET FrameWork，在安装时不会从网上下载.NET  FrameWork组件；但是安装包会比较大。

在"生成"菜单上单击生成.

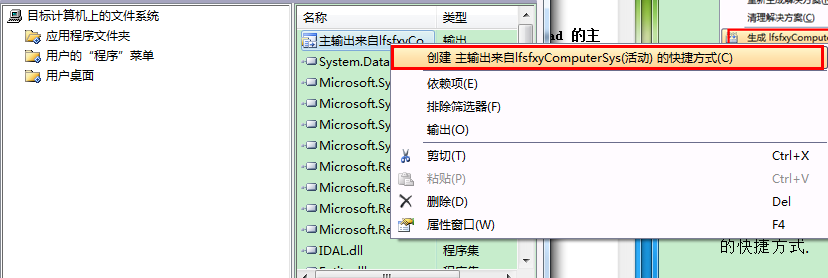


以上便是完成了基本的打包.

**可选的部署功能.**

为基于windows的应用程序创建快捷方式.

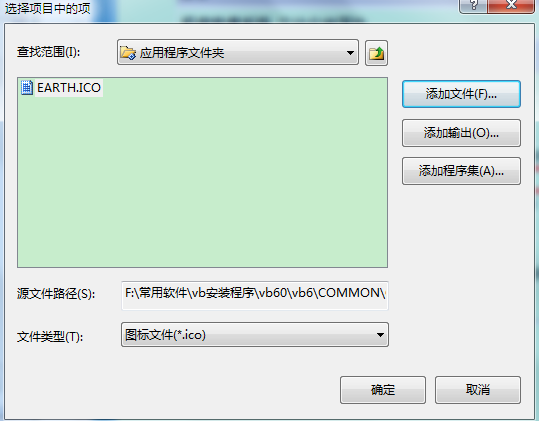
我们在"应用程序文件夹"节点下选择应用程序的主输出,右击,选择创建主输出的快捷方式.



右击属性窗口,或F4显示该快捷方式的属性菜单栏.可以为启动系统的快捷方式选择恰当的图表.

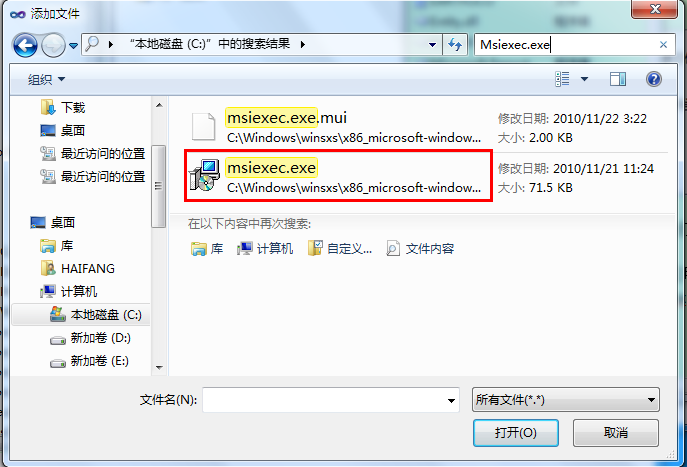


双击进入添加图标的窗口.



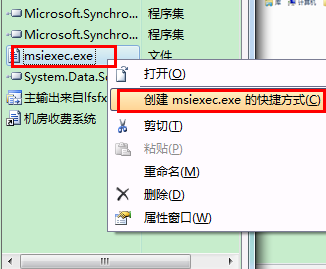
为我们的.NET程序创建一个卸载程序,并在桌面上显示卸载程序的快捷方式.

在"应用程序文件夹"中添加文件,文件位置:"C：Windows\System32\Msiexec.exe"或直接搜索该文件名.

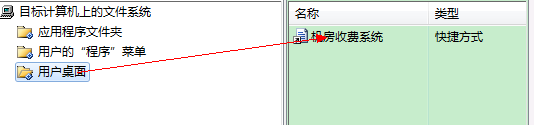


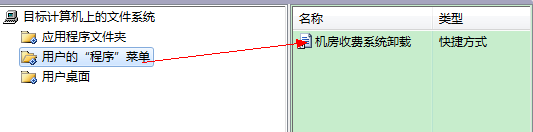
简要说说[msiexec.exe](http://baike.baidu.com/view/449791.htm)，系统进程，是WindowsInstaller的一部分。用于安装Windows Installer安装包（MSI），对系统的正常运行是非常重要的，一般在运行Microsoft Update安装更新或安装部分软件的时候出现，占用内存比较大。

同样我们要创建该系统进程exe的快捷方式,

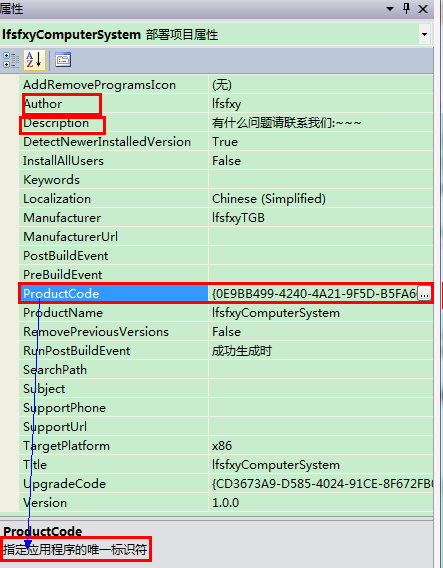


我们的快捷方式如果是在桌面显示的就拖动到"用户桌面"文件夹.如果在开始菜单中显示的,就拖动到"用户的'程序'菜单"文件夹中.



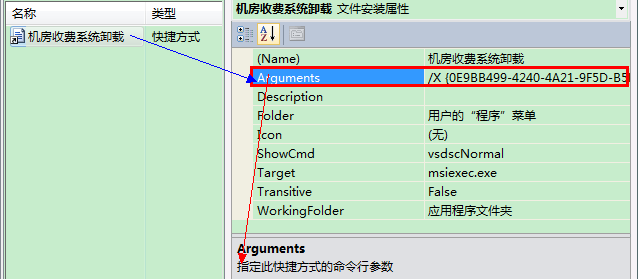


点击安装项目,F4显示安装项目的属性



vs中会给出提示该属性的作用.找到productCode属性,指定应用程序的唯一标识符.复制这个标识符到黏贴板.

粘贴卸载快捷方式属性栏的命令行参数,在前加上/X空格.



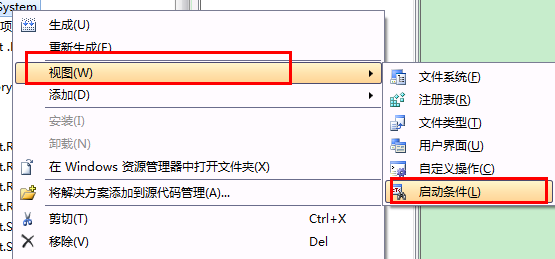
这里还在要说说给我们的安装程序添加启动条件.

此次步骤是检查目标计算机上是否安装了,Internet Explorer 5.0或更高版本,如果未安装所需要的文件,将停止安装.

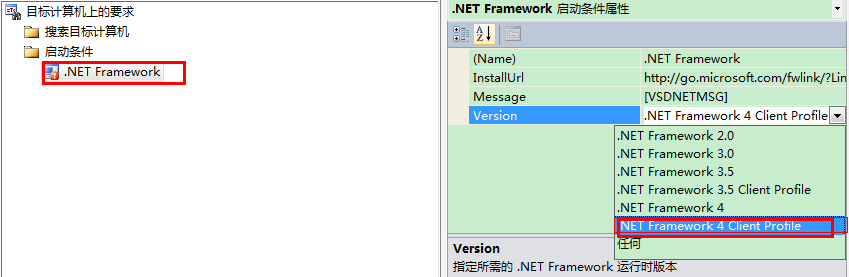
**添加启动条件来检查Internet Explorer版本(可选)**

在"解决方案资源管理器"中选择我们的安装程序lfsfxyComputerSystem

在"视图"菜单上指向"编辑器"然后单击"启动条件"

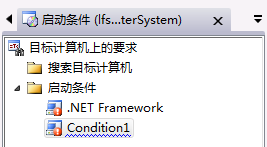


默认的依赖项NET Framework,且无法删除,我们在这里选择如下.

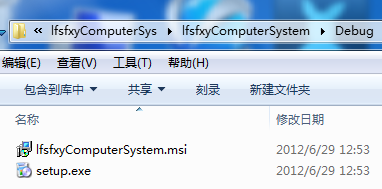


我们可自行添加启动条件.

右击"启动条件"文件夹,"添加启动条件",可以自定义添加了.



再次生成解决方案.生成解决方案后,我们可以看到,debug文件夹中就是我们的安装包.

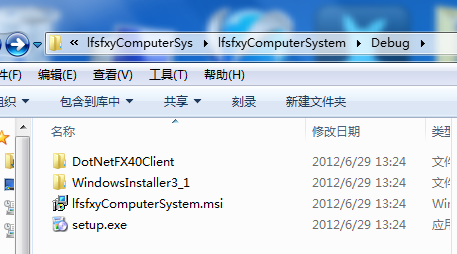


在这里还是要说说这里面msi和exe两个安装程序的区别.关于具体的[msi.](http://blog.csdn.net/femalelover/article/details/568876)    <http://hi.baidu.com/yinhorse/blog/item/5bd65387652f2727c65cc3d6.html>

msi是Windows installer开发出来的程序安装文件,它可以让你安装,修改,卸载你所安装的程序.说白了msi就是Windows installer的数据包,把所有和安装文件相关的内容封装在一个包里了.

setup.exe也允许你安装程序,但程序员在开发setup.exe的时候要比开发setup.msi困难的多,需要人工编写和安装,修改,卸载相关的很多内容.而msi把这写功能都集成化了,易于开发WINDWOS程序安装包.

如果我们选择了安装程序,"属性"-->"系统必备", 添加了安装系统必备,就会把系统必备的安装包也会打包在debug文件夹中.在我们的debug中就有一个.NET FrameWork的安装包.



在开始文件夹中.会看到我们的卸载.

在桌面有我们要显示的程序启动的快捷方式.



vs环境中自带报表使用过程详解:<http://blog.csdn.net/xhf55555/article/details/7285325>

vs环境中生成的文件夹类型详解:<http://blog.csdn.net/xhf55555/article/details/7653009>

vs环境中快捷键高效使用详解:请您继续关注!

参考:http://www.cnblogs.com/souso/archive/2010/01/12/164571.html

更多探索见vs帮助:http://127.0.0.1:47873/help/1-4924/ms.help?method=page&id=02CBCE50-6FC2-4B89-BC32-3720A7651087&product=VS&productVersion=100&topicVersion=100&locale=ZH-CN&topicLocale=ZH-CN