

# Install Shield 安装程序制作图解

整理：松鼠

时间：2009-6-2

参考：<http://www.yesky.com/SoftChannel/72342371928702976/20040817/1843460.shtml>

**摘要：** 本文介绍了利用 Install Shield 制作应用程序安装软件的方法。在文中作者除了对常用的一些技术进行介绍外，还对安装过程位图的显示、标题和背景的定制等高级技术作了简要的阐述，本文所述方法能够满足大多数安装软件的制作需求。

**关键字：** 安装软件制作；Install Shield

阅读导航		
程序框架的建立	1、建立安装程序工程	
	2、工作区中的选项卡的功能	
	3、当前属性的修改	
	4、脚本代码分析	
必要的完善	1、添加程序文件	
	2、创建快捷方式	
	3、编写脚本代码	
安装程序的发布	1、发布媒介向导对话框	
	2、Builder Type 向导页	
	3、Tag File 向导页	
	4、Platform 向导页	
界面设计	1、修改启动画面	
	2、定制标题	
	3、定制背景颜色	
	4、显示位图	

## 前言

InstallShield 是一种非常成功的应用软件安装程序制作工具，以其功能强大、灵活性好、容易扩展和强大的网络支持而著称，并因此成为目前最为流行的安装程序专业制作工具之一。该软件不仅提供了灵活方便的向导支持，也允许用户通过其内建的脚本语言 InstallScript 来对整个安装过程在代码级上进行修改，可以象 VC 等高级语言一样对安装过程进行精确控制。InstallShield 也是 Visual C++ 附带的一个安装程序制作工具，在 VC 安装结束前将会询问用户是否安装 Install Shield 工具，如果当时没有安装，也可以在使用时单独从 VC 安装盘进行安装。本文将结合一个具体的实例来对 InstallShield 的使用做一个较为全面的介绍，使读者能够初步掌握使用 InstallShield 制作专业水准的安装程序。

# 程序框架的建立

运行 InstallShield 将会出现如图 1 所示的程序界面，与使用其他高级语言开发程序一样，首先要建立一个工程。

## 1. 建立安装程序工程

双击 Project Wizard 图标，将弹出图 2 所示的向导对话框，从中可以根据用户的需求创建安装工程并自动生成相应的脚本代码。首先出现的是 Welcome 向导页，要求输入安装的应用程序的名称、开发公司的名称、程序的开发环境、应用程序类型、版本号以及可执行文件等信息。这里输入的内容在向导建立了工程后也可以通过工作区窗口对其进行修改和添加。

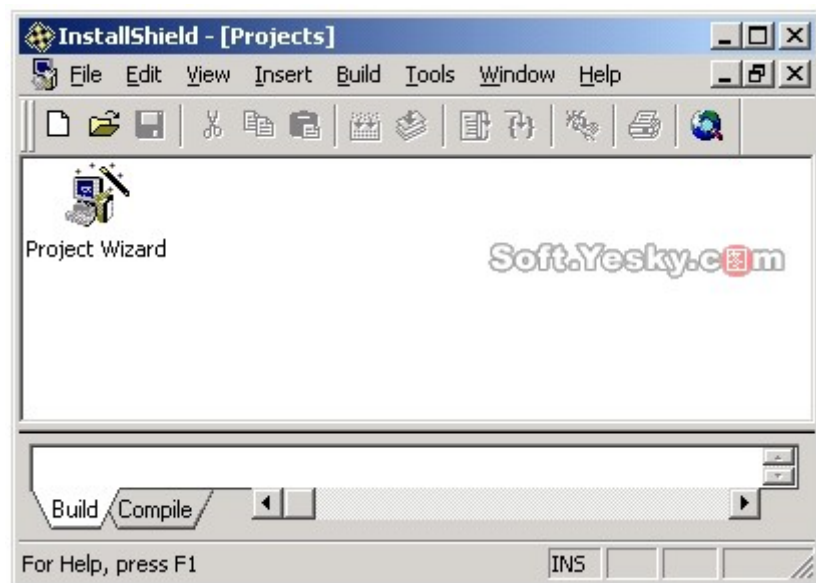


图 1 InstallShield 程序界面

输入完相关信息后点击下一步，将弹出图 3 所示的 Choose Dialogs 向导页。该向导页用来让用户从列表中选择将要在安装过程中出现的对话框。在选中某对话框列表项时，在向导页的左下角将会显示出经过缩小的对话框界面，单击 Preview 按钮将可以按正常比例预览该对话框。

## 2. 工作区中的选项卡的功能

创建了工程的 InstallShield 界面如图 10 所示，框架左侧为工程工作区，由 7 个选项卡构成。右侧显示的视图与工作区中当前选中的选项卡有关。下方则是信息输出区。下面主要对工作区中的选项卡进行介绍：



图 2 设置欢迎页面

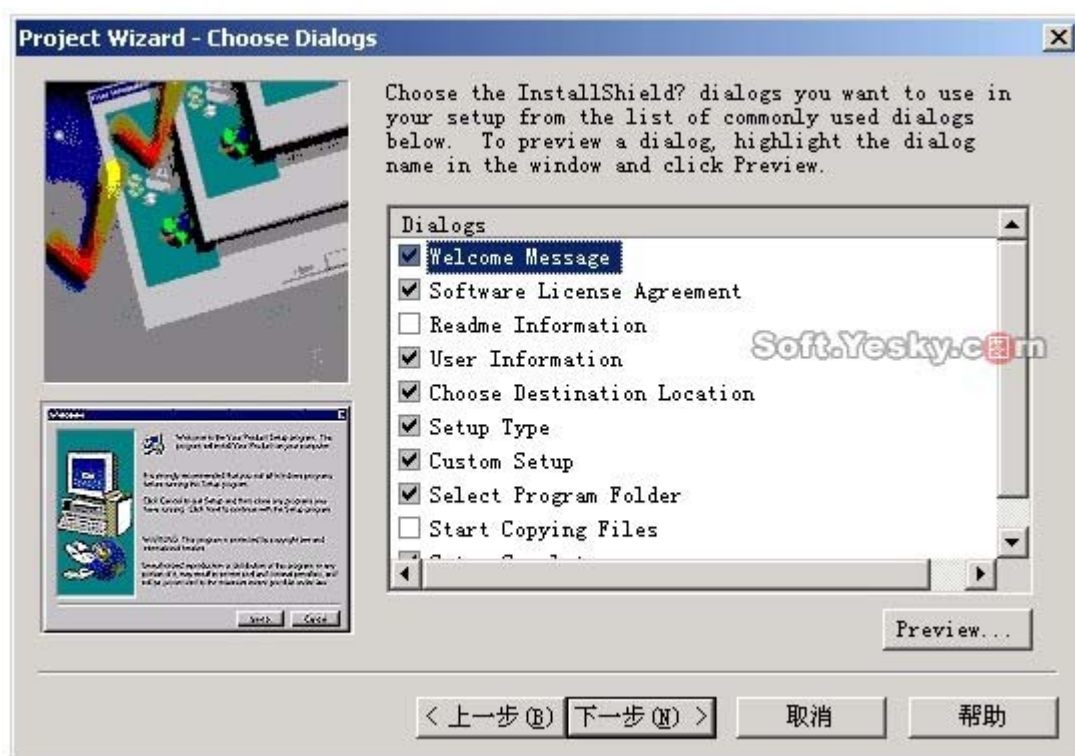


图 3 选择显示的对话框

通常使用缺省设置即可，在点击下一步后，将进入到图 4 所示的 Choose Target Platforms 向导页。该向导页让用户选择一个或多个操作系统平台，以决定将可以在哪些操作系统中进

行安装。

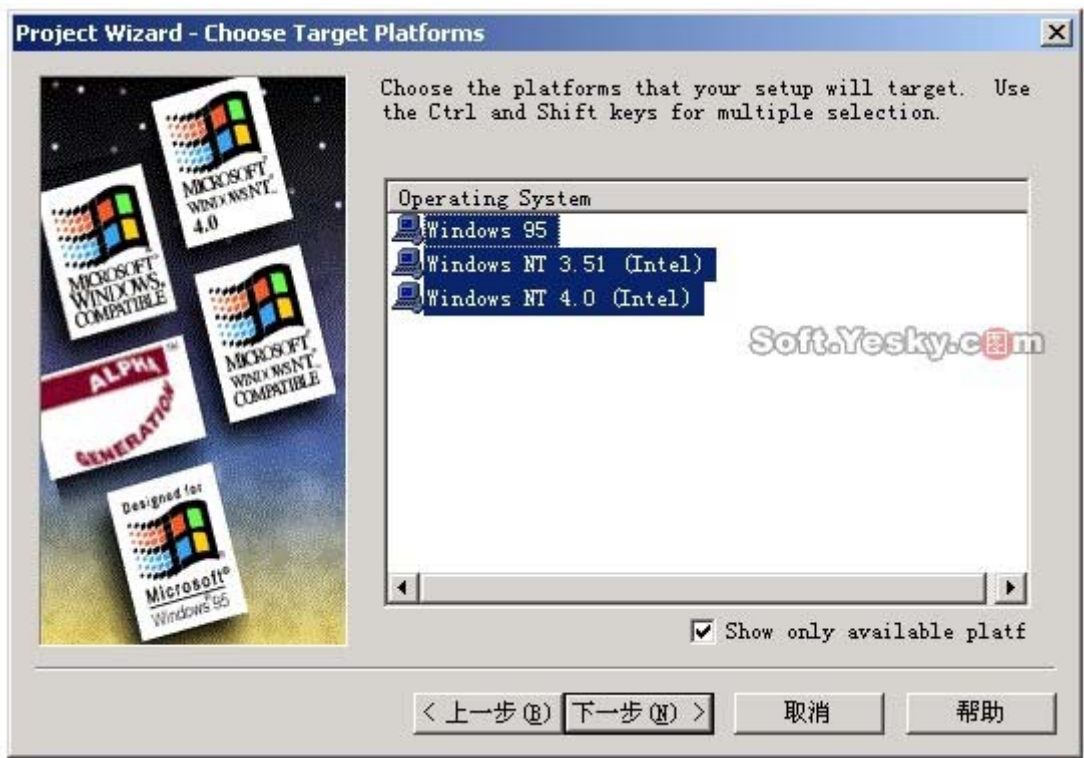


图 4 选择目标平台

向导的下一步是如图 5 所示的 Specify Languages 向导页。该向导页指出了安装程序所支持的语系。在这里只有 English 语系一种，可以在建立工程后通过工作区再添加对其他语系的支持。





图 5 指定语言

下一步将要在图 6 所示的 Specify Setup Types 向导页中设置一个或多个应用程序的安装类型，常用的有 Compact 紧凑型、Typical 典型和 Custom 自定义类型等几种安装类型。

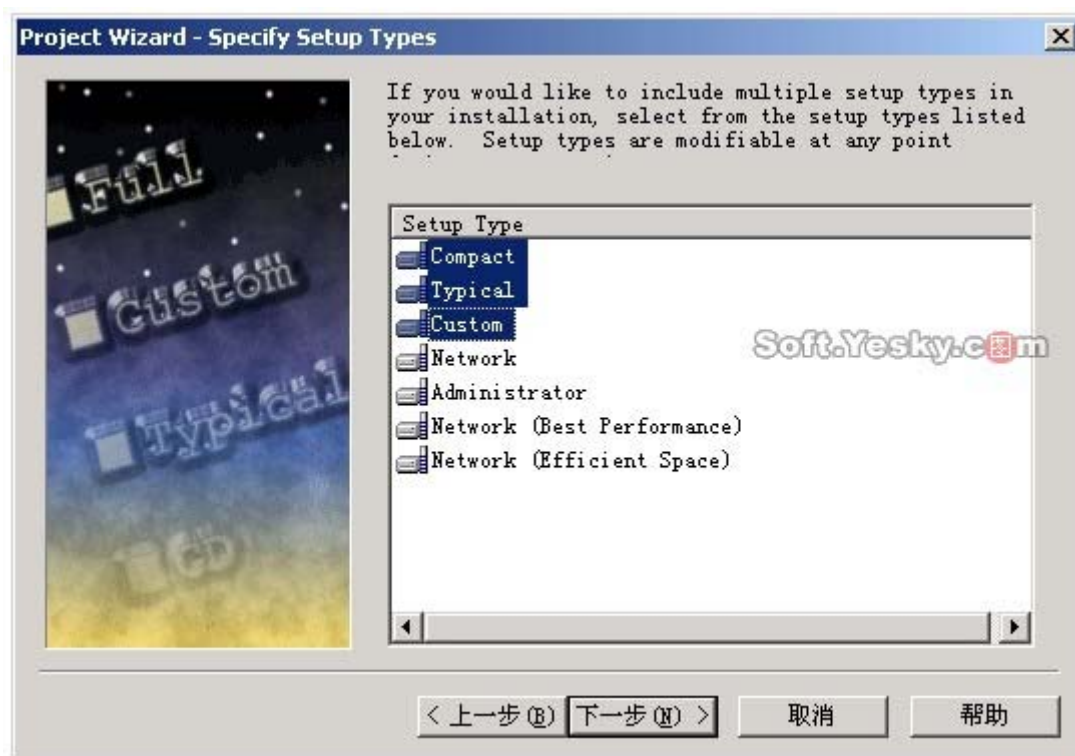


图 6 指定安装类型

接下来将要出现如图 7 所示的 Specify Components 向导页和如图 8 所示的 Specify File Groups 向导页，分别用来设置安装工程中的组件和文件组。可以通过 Add/Delete 按钮添加/删除组件或文件组。

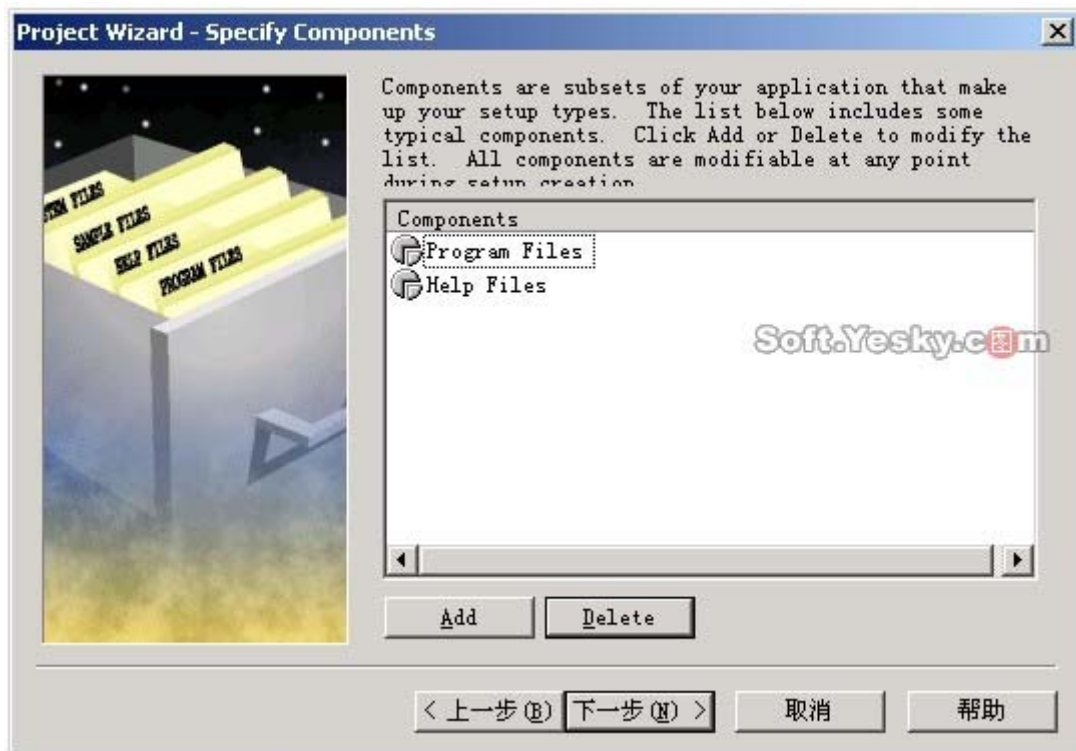


图 7 指定组件

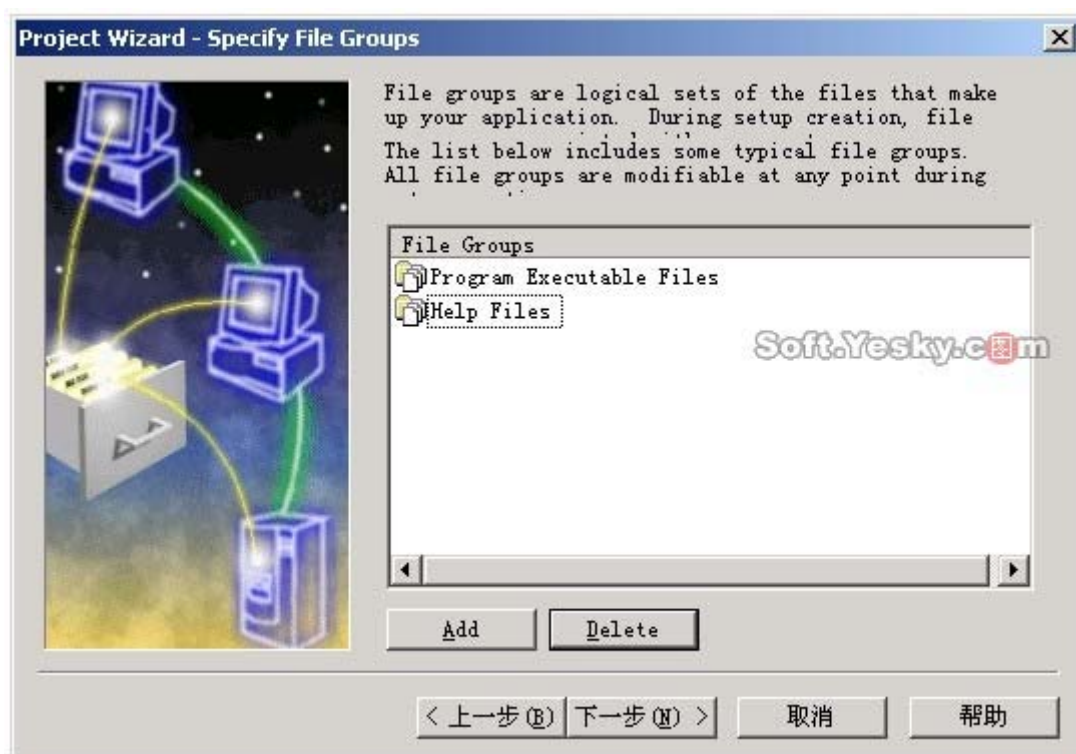


图 8 指定文件组

最后一步将出现如图 9 所示的 Summary 向导页, 该向导页显示了用户在前面的向导过程中所选定的信息摘要。如果在该列表中发现其中某项设置不合适, 可以通过上一步回到相应

的向导页对选项进行更改。最后点击完成将创建安装项目并根据用户的输入信息生成相应的脚本代码。向导以要安装的应用程序名作为安装工程名，创建的工程都被存放在缺省的 C:\My Installations\目录下，并用以工程名命名的文件夹管理该工程的所有相关文件。如果创建的多个工程具有相同的名称，向导将自动为其添加“-1”、“-2”等后缀以示区别。

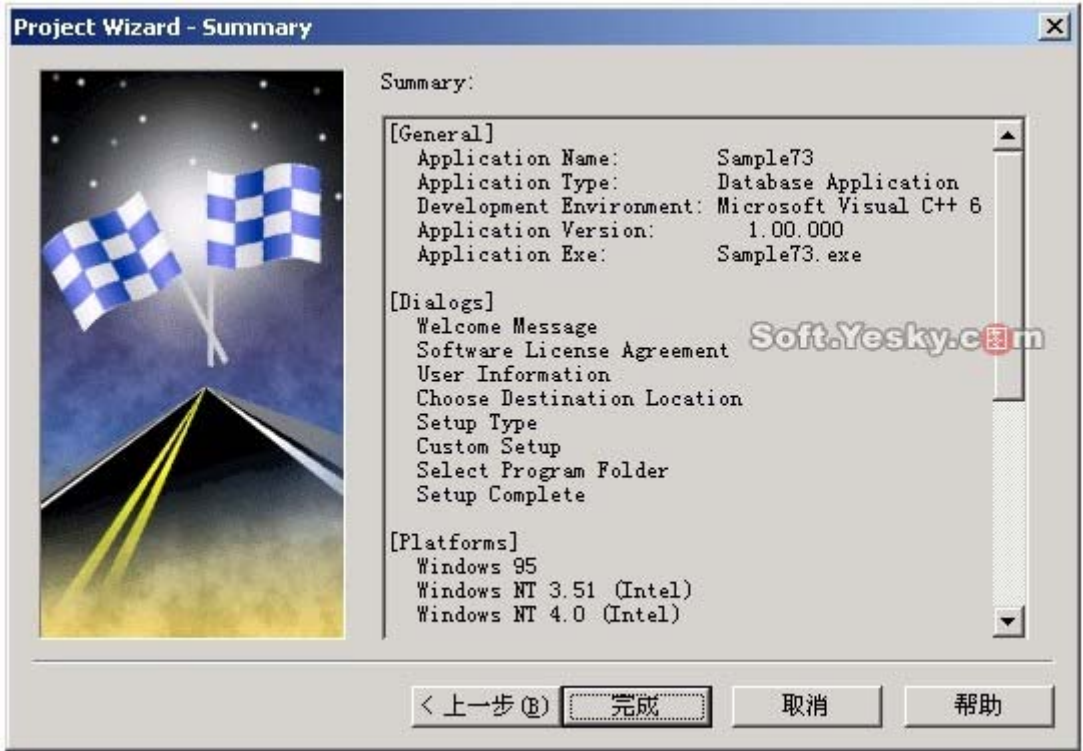


图 9 完成向导

### 3. 工作区中的选项卡的功能

创建了工程的 InstallShield 界面如图 10 所示，框架左侧为工程工作区，由 7 个选项卡构成。右侧显示的视图与工作区中当前选中的选项卡有关。下方则是信息输出区。下面主要对工作区中的选项卡进行介绍：

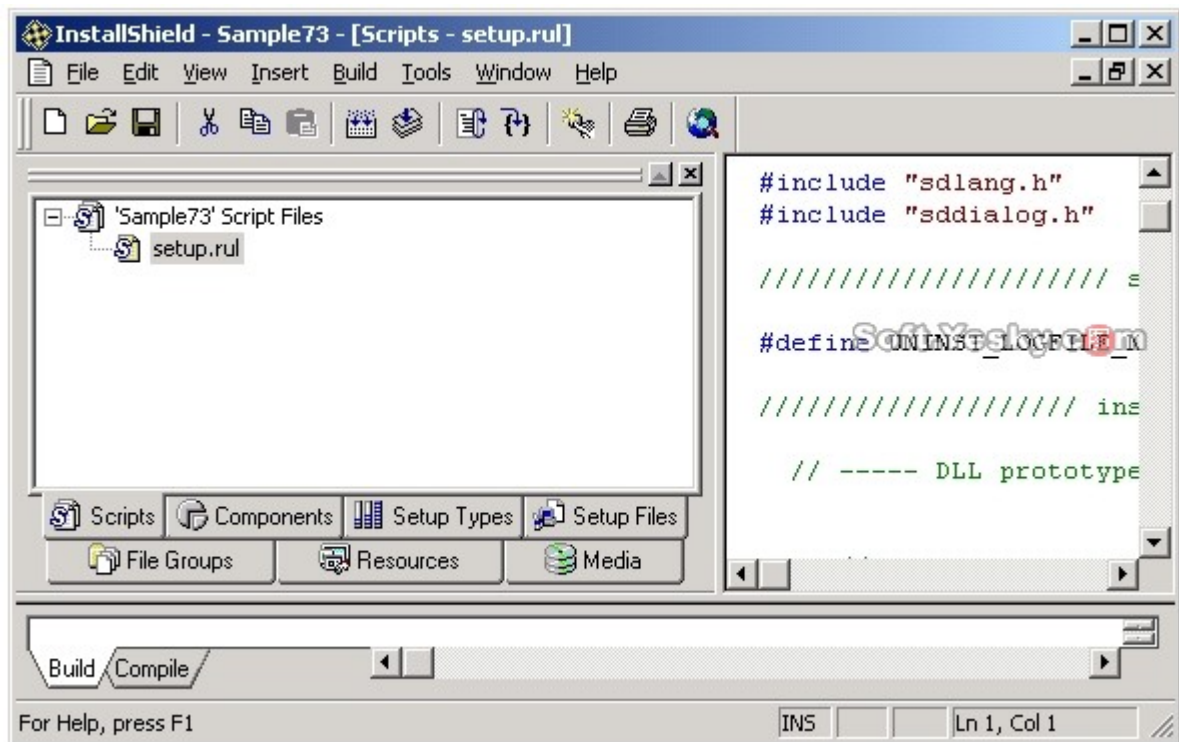


图 10 InstallShield 工程开发界面

**Scripts 页：**该选项卡页面用来管理安装程序的脚本文件，存在于该页面中的各项脚本文件均以树型结构进行组织显示。展开顶层目录项的各子项，双击以 .rul 为扩展名的脚本文件项，将在右侧文档窗口中显示出该脚本文件的内容。可以在该文档窗口中通过修改代码来控制安装程序在安装过程中的动作。

**Components 页：**管理安装程序的各组件项。缺省的组件项包括 Program Files（程序文件）、Example Files（示例文件）、Help Files（帮助文件）和 Shared DLLs（共享 DLL 文件）等。双击某组件的目录项，将在右侧窗口显示相关属性的完整列表并可对其进行编辑。

**Setup Types 页：**用来管理提供给用户的安装类型。缺省设置为 Typical（典型）、Compact（紧凑）和 Custom（自定义）等三种安装类型。在双击某安装类型后，右侧视图将显示与该安装类型相关的组件。可以选择或取消组件中文件组前面图标中的“√”图标来为不同类型组件指定相应的文件组。

**Setup Files 页：**该页面用来管理在安装过程中将要用到的安装文件。通常包含 Splash Screen、Language Independent、English 和 Advanced Files 等内容。Splash Screen 对启动画面进行管理，可以为不同的语系设置不同的启动位图，但如果将启动画面设置在 Language Independent 项，则不管操作系统是何种语系，均以此位图作为安装程序的启动画面。Language Independent 允许用户为各种操作系统指定相应的安装文件，不管是何种语系，安装程序都会根据相应的操作系统来选取相应的文件。而 English（或其他语系）项则要求安装程序根据相应的语系及语系下的操作系统来选取相应的文件。Advanced Files 将在使用具体的媒介发布时，允许用户放置一些非压缩文件。

**File Groups 页：**负责管理安装工程所需要的文件组。缺省设置为 Example Files（示例文件组）、Help Files（帮助文件组）、Program DLLs（应用程序所需的 DLL 文件组）、Program Executable Files（应用程序文件组）以及 Shared DLLs（共享 DLL 文件组）共五个文件组。双击某个文件组，将可在右侧视图对显示的文件组相关属性进行编辑修改。

**Resources 页：**用来管理安装工程所需要的安装资源。缺省设置为 String Table、



Registry Entries 和 Shell Objects 共三种资源。String Table 中定义了工程中用到的一些字符串常量。Registry Entries 设置了注册表入口。Shell Objects 资源则是经常用到的，通过该资源可以在指定位置（桌面、开始菜单等）为安装的应用程序创建文件夹或快捷方式。

**Media 页：**用来管理安装程序的发布媒介。该页面包含有 Media Build Wizard 向导、缺省的媒介以及用户新创建的媒介等项。可以通过 Media Build Wizard 向导创建新的媒介项。

#### 4. 当前属性的修改

点击 View 菜单下的 Projects 菜单项将显示包含有新创建工程的视图，在刚才创建的工程的图标上点击鼠标右键，并选中弹出菜单的 Properties 菜单项。这将弹出图 11 所示的对话框，通过其属性页可以对工程的当前属性进行更改。



图 11 工程属性对话框

由于在用向导创建工程时只支持英语，因此可以在这里增添其他语系的支持。通过 Add 按钮将弹出图 12 所示的语系添加对话框，在取消对 Show only available language 的选择后从列表框显示的语系中添加需要的即可。



图 12 添加语系

## 5. 脚本代码分析

下面对向导生成的脚本代码进行分析,在 Scripts 页中双击 Setup.rul 项即可在右侧编辑视图察看如下结构的脚本代码:

```
#include "sdlang.h"
#include "sddialog.h"
.....

// 函数声明
prototype ShowDialogs();
.....

// 全局变量声明
BOOL bWinNT, bIsShellExplorer, bInstallAborted, bIs32BitSetup;
.....

// 主程序
program
.....
endprogram

// 调用函数
.....

#include "sddialog.rul"
```

代码开始的这两个包含语句引入了预定义语言标识的头文件和 Sd 对话框的头文件。其中 Sd 对话框是一种脚本对话框 (Script Dialog), 是按 InstallScript 特定的机制来创建的, 调用 Sd 对话框的函数均以 Sd 开头。除了开始添加上述包含语句外, 还要在 endprogram

语句后添加#include "sddialog.rul"语句, 以将 Sd 对话框的实现脚本文件包含进来。随后由 prototype 关键字在 program 之前定义了函数的原型声明, 其函数实现则要在 endprogram 之后才能给出。按照 InstallScript 的规定, 在函数体内不允许声明变量, 只能在全局声明变量或是在函数名后、函数体前声明局部变量, 而且在声明变量时不允许同时为变量设置初始值。program 和 endprogram 这两个关键字定义了主程序的开始与结束, 其内代码即是对全部安装过程的描述:

```
program
Disable( BACKGROUND );
CheckRequirements();
SetupInstall();
SetupScreen();
if (ShowDialogs()<0) goto end_install;
if (ProcessBeforeDataMove()<0) goto end_install;
if (MoveFileData()<0) goto end_install;
if (ProcessAfterDataMove()<0) goto end_install;
if (SetupRegistry()<0) goto end_install;
if (SetupFolders()<0) goto end_install;
end_install:
CleanUpInstall();
// If an unrecoverable error occurred, clean up the partial installation.
// Otherwise, exit normally.
if (bInstallAborted) then
abort;
endif;
endprogram
```

上述代码首先通过 Disable() 函数指定安装过程不使用背景色, 然后调用 CheckRequirements() 检查当前系统是否能够满足被安装程序正常运行所需要的最低要求, 并通过 SetupInstall() 和 SetupScreen() 来建立安装界面的初始化。随后的这一系列 if 条件判断语句即是开始执行各个安装步骤, 如果某一安装步骤出现错误, 则通过 goto 语句跳转到 end\_install 标签处, 在通过 CleanUpInstall() 清除已经安装的文件后退出安装。其中, ShowDialogs() 函数负责显示诸如用户信息、选择安装路径等对话框, 由 ProcessBeforeDataMove()、MoveFileData() 和 ProcessAfterDataMove() 负责处理数据的复制事宜。最后通过 SetupRegistry() 完成注册, 通过 SetupFolders() 在桌面或开始菜单创建相应的快捷方式。这些函数的具体实现过程可以在 endprogram 后面的函数体实现中找到, 并可按自己的需要对向导生成的缺省代码进行修改。

对于此向导创建的程序框架, 可通过 Build 菜单下的 Compile 和 Run Setup 菜单项来编译和运行。由于这时创建的只是一个空的程序框架, 因此整个安装过程将不进行任何实质性工作, 即不进行数据文件的复制和对安装的注册。

## 必要的完善

为了使向导生成的框架工程能够真正安装应用程序,还需要通过 InstallShield 开发环境提供的各种向导对工程作进一步的设置,甚至还需要直接修改或添加部分脚本代码,以使安装程序能够完成数据文件的复制、安装的注册以及在桌面或开始菜单创建必要的快捷方式等功能。

### 1. 添加程序文件

首先向文件组添加要安装的程序文件。在 File Groups 选项卡中展开各文件组,并用鼠标右键点击各文件组下的 Links 项,选择弹出菜单的 Insert Files 菜单项以将待安装软件中的各文件添加到相应的文件组。对于没有用到的文件组,可以在该文件组上鼠标右键弹出菜单并通过 Delete 菜单项删除该文件组项。接下来切换到 Components 选项卡,在选中某组件项后,右侧列表视图将显示该组件项的相关属性,双击 Include File Groups 项将弹出图 13 所示的对话框,点击 Add 按钮并从弹出的对话框中添加与该组件项相关的文件组即可。在将所有的文件组添加到组件项后,将未用到的组件项删除。删除方法与删除文件组一样。



图 13 添加相关的文件组

另外,组件项的 Overwrite 属性也是比较关键的,它决定了在安装文件时的覆盖属性。缺省设置 ALWAYSOVERWRITE 将始终用安装文件去覆盖安装目录中已存在的文件。这样将有可能出现用老版本的文件去覆盖新版本文件的情况。通常需要更改此属性设置以对版本进行区



分。如果是 Shared DLLs 组件项，强烈建议更改该属性，否则将有可能影响系统其他软件的运行。双击该属性将弹出如图 14 所示的对话框，从组合框的下拉列表可以完成此设置。

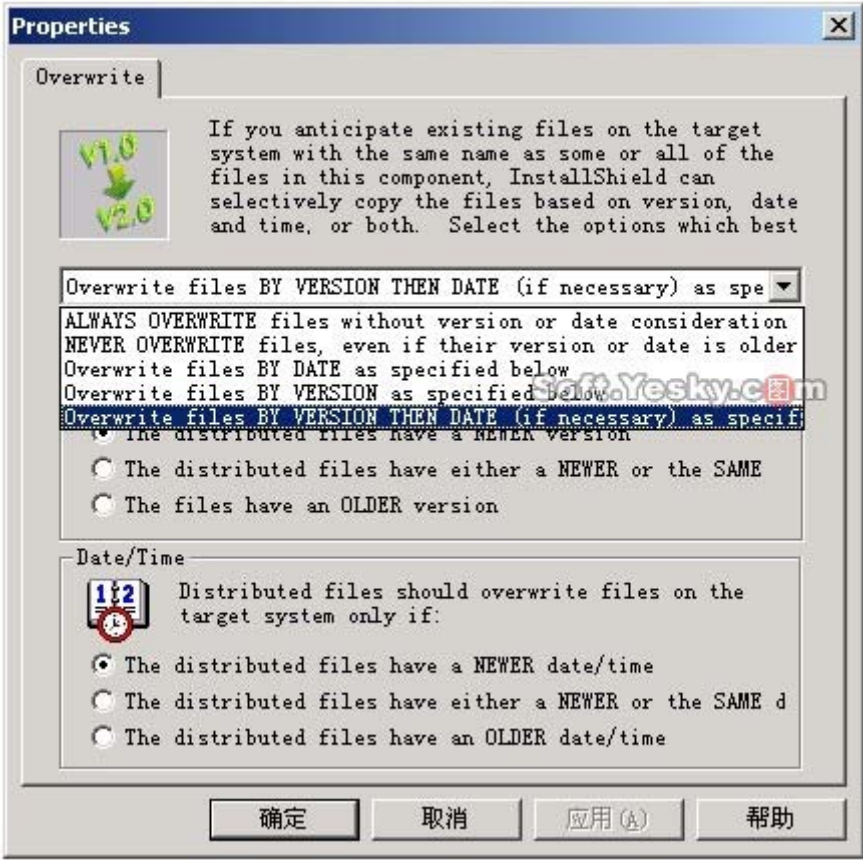


图 14 更改文件覆盖条件

## 2. 创建快捷方式

为了在桌面或开始菜单创建快捷方式，还应在 Resources 选项卡添加相关的 Shell 对象。在该选项卡中展开各项目（如图 15 所示）。

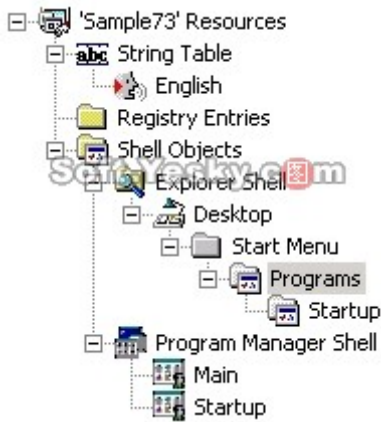


图 15 Resources 选项卡

其中的 Desktop、Start Menu、Programs 和 Startup 项分别对应于操作系统的桌面、开

始菜单、程序和启动。可以在相应的项目下添加相应的文件夹或快捷方式。添加方法为，在要添加文件夹或快捷方式的图标项上点击鼠标右键，从弹出菜单的 New 菜单项下选择 Folder 或 Shortcut 项即可。图 16 为添加了文件夹和相关快捷方式后的 Resource 选项卡设置。

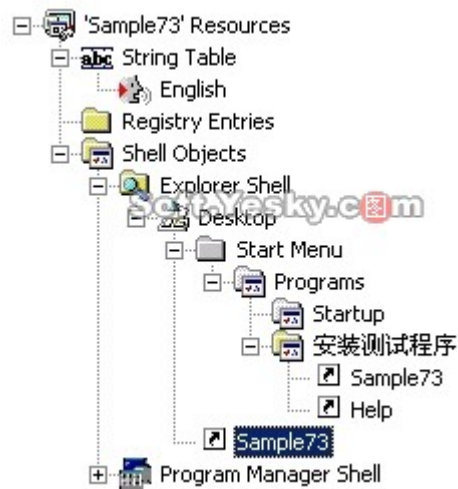


图 16 插入文件夹和快捷方式后的 Resource 选项卡

在点击添加的快捷方式，在右侧窗口将显示快捷方式的属性，双击属性列表将弹出图 17 所示的快捷方式属性对话框。在 Shortcut 栏输入快捷方式的文本字符或是通过右边的“>>”按钮弹出对话框并从中选用预定义的字符串资源。



图 A2-17 设置快捷方式属性

Target 栏为快捷方式的链接路径，<TARGETDIR>为用户在安装时所选择的路径，可以在此基础上指定具体的应用程序路径。至于字符串资源的添加，只需在 String Table（见图 16）中的相应语系所对应的字符串资源列表中点击鼠标右键，并从弹出的菜单中选择 New 菜单项，将弹出图 18 所示的对话框，从可以添加字符串的标识、内容以及注释：

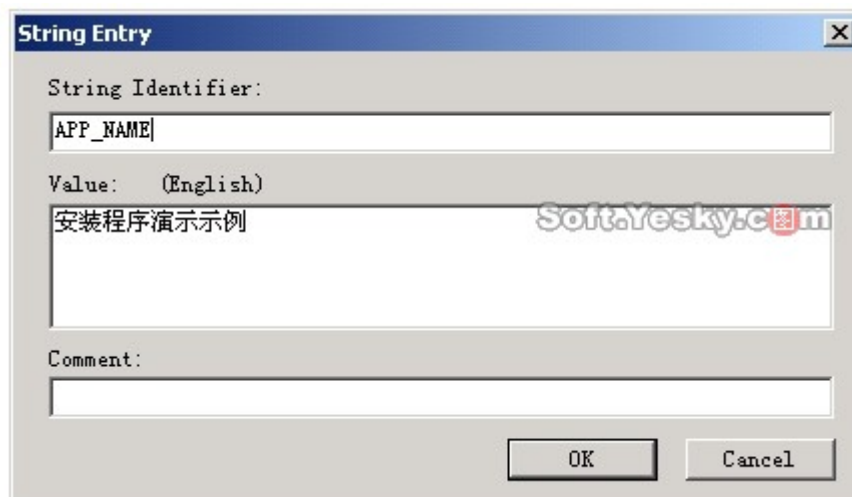


图 18 字符串资源的添加

### 3. 编写脚本代码

通过上述属性设置已经可以对程序进行安装,如果软件还需要使用序列号进行保护或是其他一些向导所没有提供的功能,也可以通过在 Scripts 页编写脚本代码来加以实现。例如,向导生成的脚本代码中虽然提供了用来处理用户在安装过程输入的序列号的 DialogShowSdRegisterUserEx() 函数,但并未对输入的序列号进行认证,下面给出添加了序列号认证的相关代码:

```
function DialogShowSdRegisterUserEx()
    NUMBER nResult;
    STRING szTitle, szMsg;
    begin
        svName = "";
        svCompany = "";
        szTitle = "";
        szMsg = "";
        Dlg_Register:
        // 输入注册序列号
        nResult = SdRegisterUserEx( szTitle, szMsg, svName, svCompany, svSerial );
        if (nResult = NEXT) then
            // 对序列号进行认证
            if (StrCompare( svSerial , "ABCDEFGH-IJKLMNOP-QRST-UVWXYZ") != 0) then
                MessageBox ( "输入的序列号"+svSerial+" 不正确! " , WARNING );
                // 如果序列号输入错误则重新输入序列号
                goto Dlg_Register;
            endif;
        endif;
        return nResult;
    end;
```

这时再运行安装程序,将真正把待安装的程序和数据进行复制,并在桌面和开始菜单创建快捷方式。如果添加了上述序列号认证代码,将只有输入正确序列号的用户才能够顺利安装程序。

## 安装程序的发布

在脚本代码没有错误,安装程序能够正常运行可情况下可以对其进行发布。这主要是通过 Media 属性页中的 Media Build Wizard 向导来完成的。



## 1. 发布媒介向导对话框

双击 Media Build Wizard 项，将弹出如图 19 所示的发布媒介向导对话框。Existing Media 列表框中列举了当前已经存在的发布媒介，可以选择缺省的 Default 媒介，也可以在 Media Name 中输入媒介名以建立一个新的媒介。



图 19 设置媒介名称

点击下一步将出现如图 20 所示的 Disk Type 向导页，列表框中给出了可以使用的发布介质，比较常用的是 3.5'' Diskette - 1.44 Mbytes 和 CD-ROM Default Size 650 Mbytes 这两项，分别为 1.44MB 容量的 3 存软盘发布和 650MB 容量的 CD-ROM 光盘发布。



图 20 设置存储介质类型

## 2. Builder Type向导页

接下来是如图 21 所示的 Build Type 向导页，Full Build 选项将全部创建所需要的文件数据，包括压缩应用程序的所有文件、创建 CAB 文件及 Setup.exe 程序等；Quick Build 选项则只是测试、检查安装程序能够按预期的方式进行。

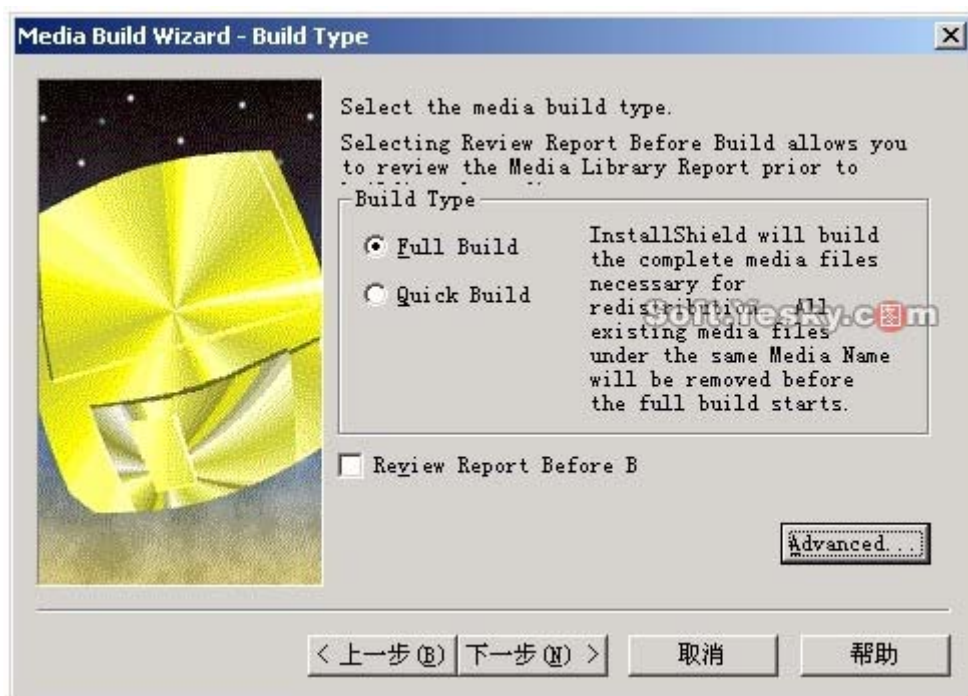


图 21 设置建立类型

Advanced 按钮将弹出图 22 所示的高级属性对话框，可以在其中设置口令、文件的时间/日期、媒介创建的目标位置等高级属性。



图 22 建立类型的属性页

### 3. Tag File向导页

在下一步的 Tag File 向导页（图 23）中将输入诸如公司名称以及其他一些与应用程序相关的信息，这些信息将被保存到 Data.tag 文件中。

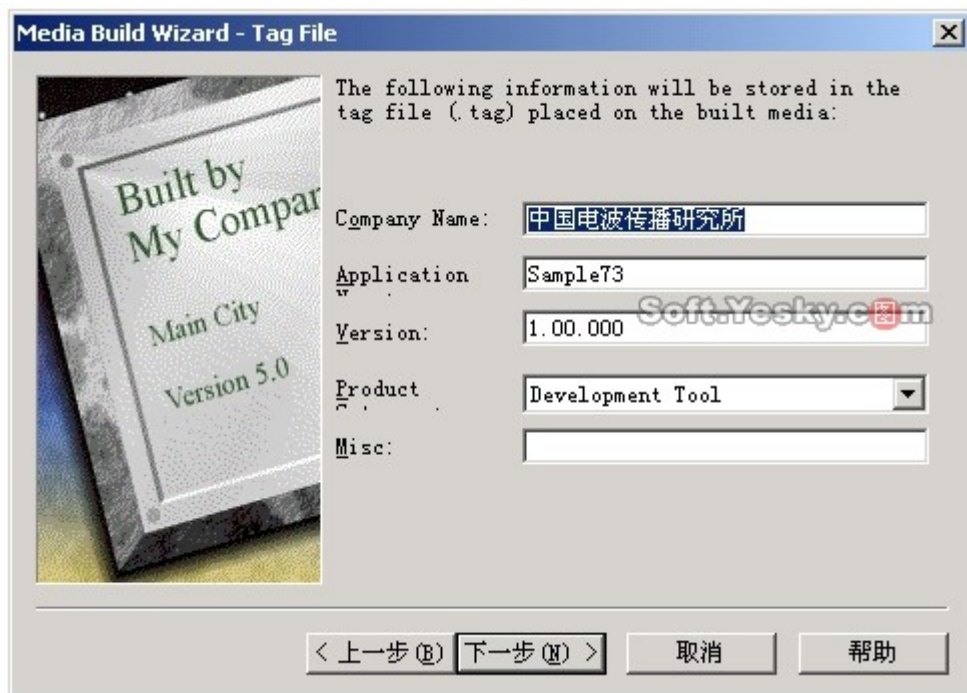


图 23 设置标签文件

#### 4. Platform向导页

继续向导，在图 24 所示的 Platform 向导页中选择可以使用的平台后，将完成对媒介发布向导的设置并出现如图 25 所示的简表。该简表对前面的向导设置进行了汇总，如果发现设置有误，可以通过上一步回到相应的向导页面对其进行更正。

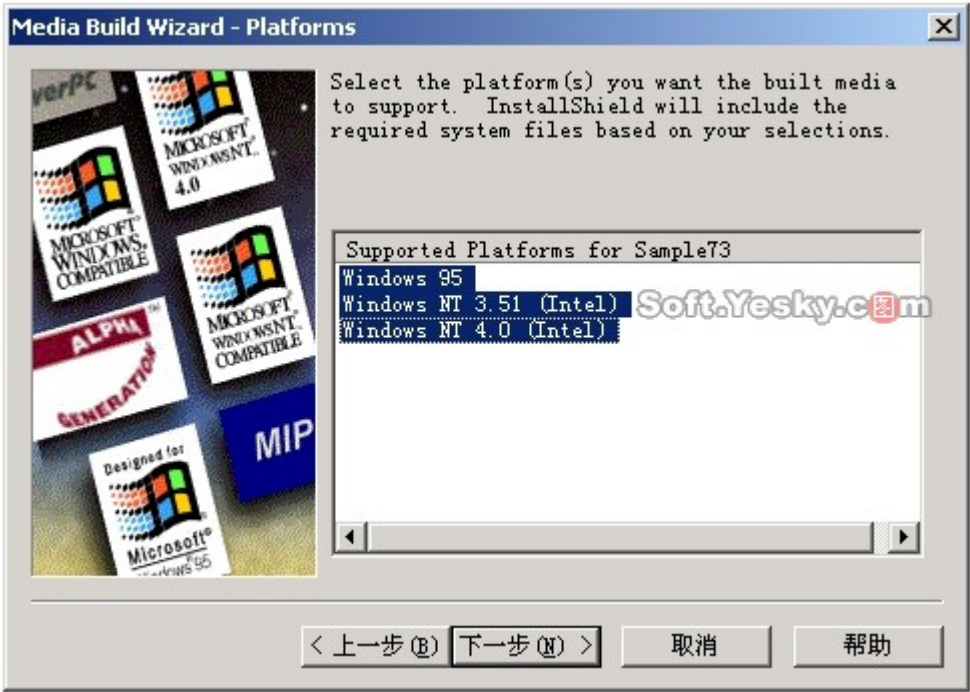


图 24 设置平台

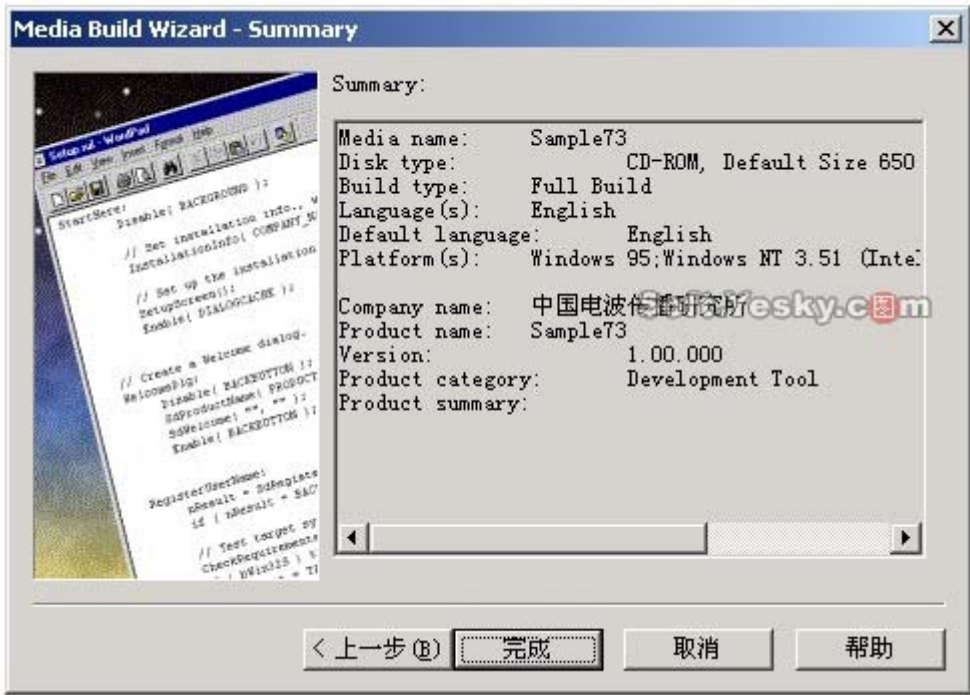


图 25 查看汇总信息

点击完成按钮将开始根据上述设置来创建发布媒介（见图 26）。





图 26 根据设置创建发布媒介

最后，在发布媒介创建完成后，点击图 26 所示对话框的 Finish 按钮完成发布媒介的创建。此时的 Media 选项卡将包含刚才创建的发布媒介（见图 27）。

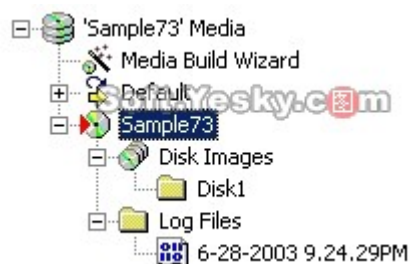


图 27 新创建的发布媒介

新创建的发布媒介中的 Disk Images 项给出了创建的发布媒介的磁盘映像，其下的 Disk1、Disk2 等项对应于发布该安装程序的每一张磁盘/光盘的映像，在选中某磁盘项目时，右侧列表视图将显示该磁盘映像项目所包含的安装文件列表。Log Files 项则是用来对发布媒介日志记录的管理。

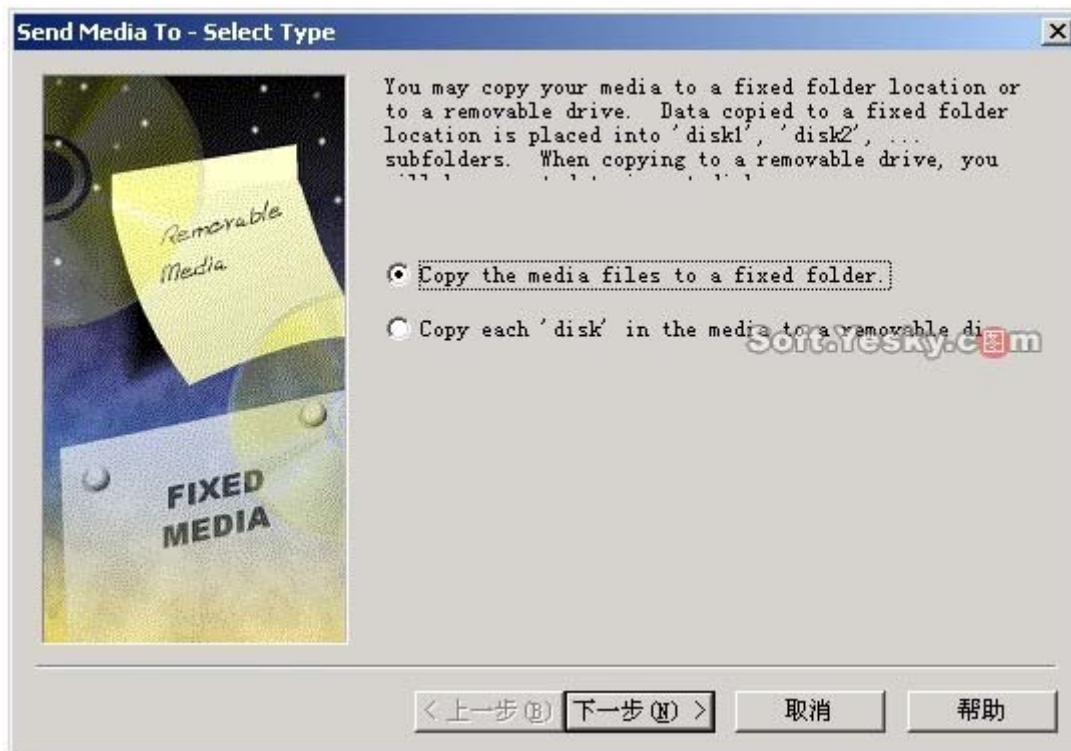


图 28 选择媒介文件的复制类型

最后,在完成创建的媒介文件到发布磁盘或相应的文件夹的复制后即完成了一个基本安装盘的制作。可以在新创建的媒介项目上点击鼠标右键,并从弹出菜单中选择 **Send Media To** 菜单项,将弹出如图 28 所示的发送媒介向导对话框,可以在该向导页中指定媒介文件是复制到硬盘中的某个文件夹还是直接复制到发布软盘。在随后的向导中,只需指定相应的路径并进行数据复制即可。运行发布媒介中的 **Setup.exe** 文件即可进行应用程序的安装。

## 界面的设计

前面介绍的方法固然能够制作出一个具有基本安装功能的程序,但由于没有对界面进行任何处理,而直接使用向导缺省设置下的安装界面。为了使制作的安装程序不落俗套,避免缺省设置下的单调界面,通常都需要根据被安装软件的性质而对界面进行重新设计。安装界面的设计通常主要针对标题、启动画面和安装程序背景等进行。

### 1. 修改启动画面

首先对安装程序的启动画面进行更改,此前应先准备好要使用的素材图象,画面不宜过大。如果是其他格式的图象应先将其保存为 **bmp** 格式的位图,并将文件名改为 **Setup.bmp**。安装程序将在开始运行时自动显示以该文件名命名的图象。在工作区的 **Setup Files** 页面,将可以看到 **Splash Screen** 项,在该项目下有 **Language Independent** 和各支持的语系项,点击各项目,将在右侧列表视图中显示当前使用的缺省的启动画面 **Setup.bmp**,选中后将其删除。接下来将要添加新的启动画面,如果对于各个语系均使用相同的启动画面,那么就在 **Language Independent** 项中添加,如果对于不同的语系有相应的启动画面,那么就在各个语

系项中添加与之相对应的启动画面。具体添加办法是在选中要添加的项目后，在右侧列表视图中点击鼠标右键，从中选取 **Insert File** 菜单项即可。需要注意的是，在完成上述设置后，编译运行安装程序并没有使用新的启动画面，应在 **Media** 选项页中通过向导重新建立发布媒介后，新更改的启动画面才能够生效。

## 2. 定制标题

接下来对安装程序的标题进行定制，在向导生成的 `SetupScreen()` 函数中通过 `SetTitle()` 函数来完成主标题的显示，缺省设置参数为 24 点大小的白色字符，使用缺省字体。可以在使用 `SetTitle()` 之前调用 `SetFont()` 来使用指定的字体。`SetFont()` 函数原型为：

```
SetFont (nItemID, nFontStyle, szFontName);
```

其参数 `nItemID` 为要设置的字体项目，目前只能设置为 `FONT_TITLE`，即只能对主标题进行设置。`nFontStyle` 为字体风格，允许的设置分别为 `STYLE_NORMAL`、`STYLE_BOLD`、`STYLE_ITALIC`、`STYLE_SHADOW` 和 `STYLE_UNDERLINE`，分别表示正常、加粗、斜体、阴影和下划线风格的字体，除了 `STYLE_NORMAL` 风格外，其他几种风格都允许组合使用。使用的字体由 `szFontName` 来指定，如果执行安装程序的系统中没有安装该字体，那么将仍使用缺省的字体。例如，下面这段代码将显示带阴影效果的、加粗的、使用“华文彩云”字体的 32 点大小的主标题：

```
// 设置字体
SetFont ( FONT_TITLE , STYLE_BOLD | STYLE_SHADOW , "华文彩云" );
// 使用点大小的字体
SetTitle( @TITLE_MAIN, 32, WHITE );
```

## 3. 定制背景颜色

目前，安装程序的背景界面为单一的暗青色，`InstallShield` 脚本程序允许用户设置一个预定义的或是自定义的背景颜色，而且还可以指定为颜色渐变。具体由 `SetColor()` 函数完成，该函数共包含两个参数，第一个参数通过 `BACKGROUND` 或 `STATUSBAR` 来指出是为背景指定颜色还是为进度条指定颜色。第二个参数即为设定的颜色值及颜色模式（是否渐变）。预定义的颜色值有：

- BK\_BLUE 渐变的蓝背景色
- BK\_GREEN 渐变的绿背景色
- BK\_MAGENTA 渐变的紫背景色
- BK\_RED 渐变的红背景色
- BK\_YELLOW 渐变的黄背景色
- BK\_SOLIDBLUE 蓝色（用于背景）
- BK\_SOLIDGREEN 绿色（用于背景）
- BK\_SOLIDMAGENTA 紫色（用于背景）
- BK\_SOLIDRED 红色（用于背景）
- BK\_SOLIDYELLOW 黄色（用于背景）
- BK\_SMOOTH 与自定义颜色组合为渐变的背景色。
- GREEN 绿色（用于进度条）
- RED 红色（用于进度条）

BLUE 蓝色（用于进度条）

MAGENTA 紫色（用于进度条）

YELLOW 黄色（用于进度条）

除了可以直接使用上述预定义的颜色外，也可以使用 RGB（）定义的自定义颜色，如果需要使用自定义的渐变色可以将其与 BK\_SMOOTH 共同使用即可。例如：

```
nColor = RGB(0, 0, 255);
```

```
SetColor (BACKGROUND, BK_SMOOTH | nColor);
```

和

```
SetColor (BACKGROUND, BK_BLUE)
```

都可以产生蓝色渐变的背景色。需要说明的是，在指定背景颜色时，应当保证背景是可以设置的，因此上述更改背景色的代码应当在 Enable (BACKGROUND) 语句后进行。

## 4. 显示位图

最后，再来介绍一下安装过程中对位图的显示。在安装过程中所能够显示的位图只有 BMP 和 WMF 这两种格式的位图。首先准备好素材位图并将其转换为可使用的位图格式，然后在 Setup Files 属性页中将这此文件添加到相应的项目中，这里所说的“相应”是指将要显示的图片与语系、操作系统相对应起来。例如，要添加的位图是用于英文 Windows NT 下的，那么就在 Setup Files 属性页中先找到语系“English”，然后再在该语系下找到相应的操作系统。具体的添加办法与添加启动位图一样。

在将位图添加到安装文件后，就可以通过 PlaceBitmap（）函数在指定的位置显示位图了。

PlaceBitmap（）函数原型如下：

```
PlaceBitmap (szName, nID_BITMAP, nDx, nDy, nDrawOp);
```

参数 szName 为要显示的位图，nID\_BITMAP 为位图的 ID 号，参数 nDrawOp 为图片在窗口中的放置位置，可以根据情况使用下述选项之一：

TILED 平铺位图

FULLSCREENSIZE 拉伸至全屏

CENTERED 居中显示

LOWER\_LEFT 在左下角显示位图

LOWER\_RIGHT 在右下角显示位图

UPPER\_LEFT 在左上角显示位图

UPPER\_RIGHT 在右上角显示位图

REMOVE 清除以前放置的位图

参数 nDx, 和 nDy 则表示在当前方式方式下的图象放置坐标

在同一窗口界面下将不能同时显示两幅位图。因此，在显示下一幅位图之前需要通过 REMOVE 操作符将第一幅位图清除，然后才能显示下一幅位图。另外，在设置图像文件路径时，由于安装程序是在其他用户的计算机中运行，因此不能使用位图在本地计算机的绝对路径，而应从刚才添加过位图的 Setup Files 属性页中获取。下面这段代码给出了陆续显示两幅位图的过程：

```
// 在新位置显示新位图
```

```
PlaceBitmap ( SUPPORTDIR ^"step4.bmp" , 4 , 10, 100, UPPER_LEFT );
```

```
.....
```

```
// 删除上一幅位图
```

```
PlaceBitmap ( "" , 4 , 0, 0, REMOVE );
```



// 在新位置显示新位图

```
PlaceBitmap ( SUPPORTDIR ~"step5.bmp" , 5 , 10, 10, LOWER_RIGHT );
```

这里在使用 REMOVE 操作之前，为 PlaceBitmap ( ) 的 nID\_BITMAP 参数指定了一个位图 ID 号，这样再用 PlaceBitmap ( ) 执行 REMOVE 操作时就不用再次指定相应的图像文件名了。

除了上述提到的一些用户界面函数外，还有许多其他用途的用户界面函数，InstallShield 为所有 API 函数的使用提供了一个向导（见图 29），其中就包含有用户界面函数。点击工具条上的 Function Wizard 按钮将可以启动该向导，左侧列表框给出了 API 函数的分类。在选中 User Interface 项后，右侧列表框将显示所有可用的用户界面函数，在选中某 API 函数后，将可以通过向导很简单的实现对参数的设置。

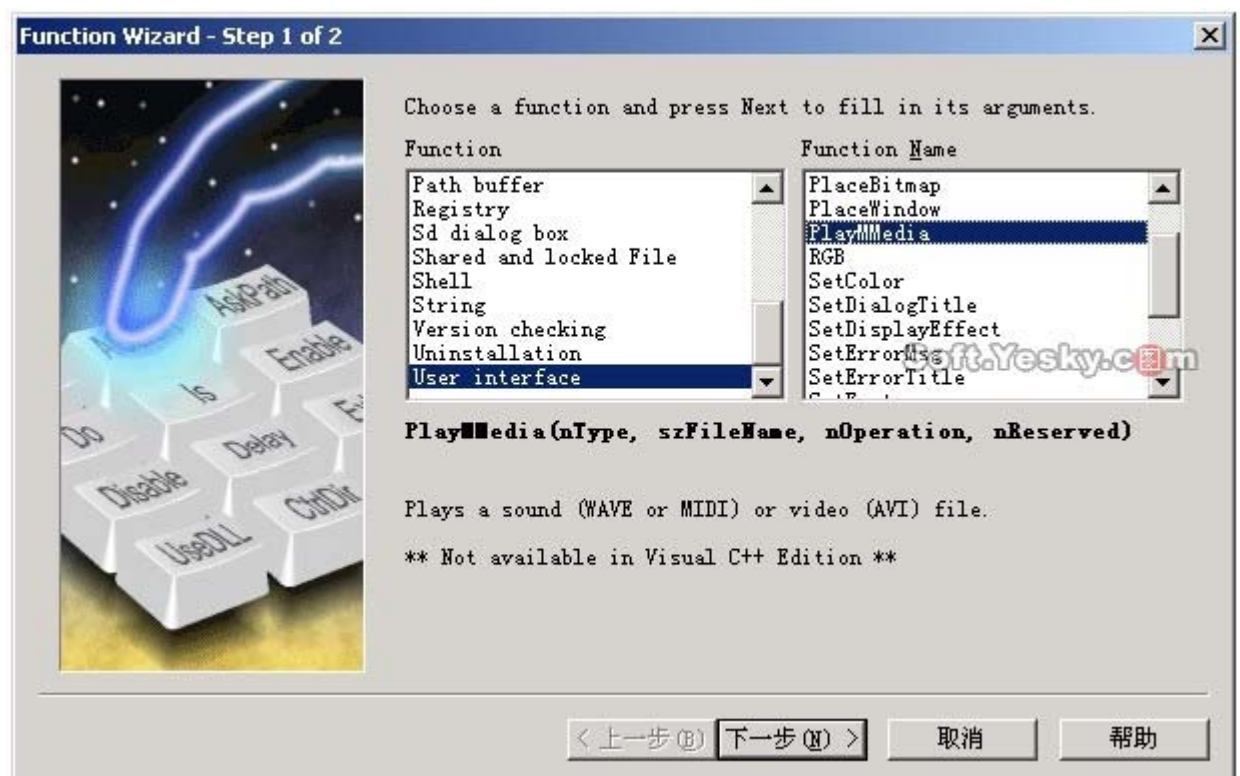


图 29 使用函数向导

## 5. 结论

本文以 Install Shield 6.0 为工具制作了一个可以安装待发布软件的安装程序。多用在商业或较正式软件的发布等场合，而这一类的软件通常都会有升级、更新的问题，使用 Install Shield 6.0 开发的安装工程可以很方便的解决这一问题，只需在 File Groups 页中以新版本的文件去替换旧版本的相应文件，并重新编译、发布即可，而不必为每一个版本的软件去开发一套与之相应的安装程序。