

商业开发 代码库系列

Visual C++.NET

案例开发集锦

马 煜 康祥顺 李海军 编著
陈海军 审校

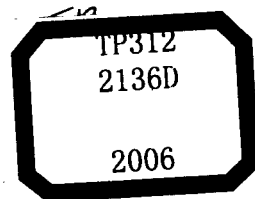


电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

商业开发代码库系列



Visual C++.NET案例开发集锦

马 煜 康祥顺 李海军 编著

陈海军 审校

电子工业出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书围绕Visual C++.NET提供的技术,详细探讨如何使用Visual C++.NET开发应用程序,每个技术要点均通过具体的案例来解析;从简单的控件应用到复杂的数据库函数,从本地应用到复杂的网络处理都详细地进行剖析,涵盖Visual C++.NET编程基础、图形图像处理、多媒体应用、系统文件处理、硬件的控制、数据库、网络处理等多个方面。读者在看完本书后,就能够结合实际,快速、高效、灵活地设计出专业级应用程序来,所有的案例基本上都可以直接嵌入到自己的应用中去。

本书适合对Visual C++感兴趣的大中专院校学生、软件开发人员以及Visual C++.NET产品爱好者阅读,尤其是希望能够精通Visual C++.NET编程的读者。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

Visual C++.NET案例开发集锦/马煜等编著. —北京:电子工业出版社, 2006.10

(商业开发代码库系列)

ISBN 7-121-03129-9

I. V… II. 马… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字(2006)第101204号

责任编辑:徐云鹏 李红玉

印 刷:北京天竺颖华印刷厂

装 订:三河市金马印装有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

北京市海淀区翠微东里甲2号 邮编:100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:32.75 字数:810千字

印 次:2006年10月第1次印刷

定 价:52.00元(含光碟1张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系电话:(010) 68279077。邮购电话:(010) 88254888。

质量投诉请发邮件至zlts@phei.com.cn。

服务热线:(010) 88258888。

前 言

什么是Visual C++

Visual C++是Microsoft公司的一款带有集成开发环境的C++编译器，它包含了能简化开发强大程序的工具以及必需的库文件，是近年来在国内外得到广泛应用的可视化、面向对象的编程语言。由Visual C++开发出来的应用程序具有与Windows操作系统结合紧密，可执行效率高等特点，一直以来是专业人员开发Windows应用程序的首选工具。

为了配合新的市场战略，微软四年磨一剑，推出最新的开发工具Visual Studio.NET。Visual C++.NET是Visual Studio.NET系列中重要的组成部分之一。Visual Studio.NET对程序库进行了许多更新和扩充，其中包括ATL（ActiveX Template Library）、ATL服务器、MFC、C的动态链接库、OLE DB模板，共享类、标准的C++库和原来的iostream库。另外，还增加了Web Service开发、新的控件和宏等。

本书面向的读者群

对于读者而言，阅读本书要注意两个重要的方面：

一是理解并建立起成熟的编程思想，因为程序是用来表达程序员思想的，所以掌握面向对象的分析方法以及对COM、ATL的深入理解是非常重要的。

二是熟练地使用一种语言和它的编程环境，比如本书介绍的Visual Studio.NET。

本书的读者对象是对Visual C++感兴趣的大中专院校学生、软件开发人员以及Visual C++.NET产品爱好者，尤其是希望能够精通Visual C++.NET编程的读者。

在阅读本书前读者朋友需要有一定的C++语言基础，本书是在假定读者已经有一定的C++语言基础上编写的。如果没有C++语言基础，在阅读本书前最好熟悉一下C++语言的基本语法和特点，否则阅读本书可能会有一定的困难。

本书章节安排

本书共分为9章。

第1章：VC++.NET编程基础。本章通过一些实用的案例介绍VC++.NET编程基础知识。通过本章的学习，读者朋友可以对VC++.NET的编程基础知识有一个整体的了解，为后续章节的学习打下一个良好的基础。

第2章：图形图像处理——GDI+编程。本章通过案例介绍GDI+编程的实际应用，读者通过本章的学习可以领略到GDI+的优势所在。

第3章：多媒体应用。在本章中，读者朋友会看到几个使用Visual C++.NET开发的多媒

体方面的案例。通过案例的讲解，读者可以了解MCI的原理和使用技巧，对于进一步开发功能更为完善的多媒体应用程序有较强的参考价值。

第4章：系统文件处理。这一章的13个案例都具有很好的实用价值，读者朋友完全可以在这些案例的基础上进行二次开发。

第5章：硬件的控制。本章总共有7个案例，通过本章的学习，读者可以了解诸如鼠标的控制、键盘的控制、系统设备的控制、内存的控制等知识。

第6章：数据库应用。通过本章的学习，读者可以对在VC++.NET下如何对数据源进行各种操作有一个整体的了解。本章中的所有案例都可以作为读者的应用参考。

第7章：网络应用。本章讲解了各种网络应用的案例，通过本章的学习，读者可以对Winsocket编程有个整体的理解，对今后在网络方面更加深入地进行研究起到一个良好的辅助作用。

第8章：控件操作。本章总共提供给读者5个案例，主要是让读者朋友对VC++.NET中的各个控件的应用有一个比较深入的理解。

第9章：综合案例。读者通过前面几个章节的学习，已经初步掌握了VC++.NET的编程原理，本章通过几个综合性的案例，使读者可以更为全面地巩固知识，为进一步的学习打下基础。

附带光碟

本书所有案例全部在Visual C++.NET中调试通过，并在随书附带的光碟中记录全部案例的源代码，可供读者在自行开发应用程序时查看和参考。

关于本书附带光碟的内容及其使用方法请参考光碟内说明文档。

致谢

本书的主要作者是马煜、康祥顺、李海军。参加本书编写工作的还有孟伟涛、陈海军、牛永杰、刘婷婷、张小李、王植、宋学义、张小恒、程娥、贺曼、张丽霞、鲁臻、张庆、张强、崔竞、李欣、保春艳。在本书的写作过程中，还得到了周围同行的大力支持，谨在此表示感谢。

由于时间紧迫，书中难免有错误之处，恳请读者不吝赐教！



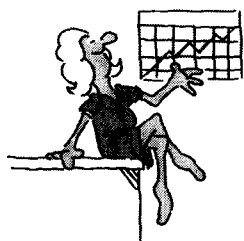
目 录

第1章 VC++.NET编程基础	1
案例1 第一个Windows程序	4
案例2 进制转换器	9
案例3 打印杨辉三角形	15
案例4 确定一个月的天数	18
案例5 石头剪子布猜拳游戏	24
案例6 创建动态菜单	28
案例7 创建停靠工具栏	32
案例8 递归与全局变量的案例——打靶程序	43
案例9 调用C++.NET中的数学函数	47
案例10 在列表框中加入图片	50
第2章 图形图像处理——GDI+编程	56
案例1 绘制风格迥异的直线	61
案例2 绘制Bezier曲线	63
案例3 不规则窗体的实现	65
案例4 如何在上传的图片中加上版权文字	68
案例5 在图片中绘制水印效果	72
案例6 动态多边形	75
案例7 绘制电子时钟	76
案例8 图片的淡入淡出效果	77
案例9 放大镜游戏	80
案例10 生成缩略图	84
案例11 透明的窗体	85
案例12 简单的画图程序	88
案例13 Windows涂鸦程序	92
案例14 抓取图像	93
案例15 动画范例——飞碟绑架地球人	101
第3章 多媒体应用	104
案例1 Flash动画播放器	108
案例2 OpenGL屏幕保护程序	114
案例3 显示器的分辨率调整	124

案例4	五子棋游戏	126
案例5	Windows媒体播放器	132
案例6	会跑的按钮	142
案例7	CD唱机	143
第4章	系统文件处理	150
案例1	建立一个具有查找功能的文本编辑器	150
案例2	建立文件浏览器	154
案例3	显示驱动器中的所有目录	162
案例4	实现目录遍历	166
案例5	读写文本文件和二进制文件	171
案例6	访问注册表中的硬件信息	174
案例7	显示系统的日期和时间	182
案例8	模拟Windows XP关闭系统	185
案例9	MDI窗体菜单的设计	191
案例10	登录对话框的实现	196
案例11	拖放操作的实现	199
案例12	创建自定义文件类型	201
案例13	监视系统文件的状态	205
第5章	硬件的控制	214
案例1	限制鼠标的活动区域	214
案例2	处理鼠标和键盘事件	216
案例3	打印及打印预览的实现	219
案例4	内存状态显示	222
案例5	显示系统设备	224
案例6	通过驱动程序控制系统外设	228
案例7	自动发送和接收传真	231
第6章	数据库应用	239
案例1	连接到数据源以及关闭数据库	243
案例2	参数化查询数据表	245
案例3	数据库的关联查询	248
案例4	编程实现记录的编辑	256
案例5	单记录更新数据源	260
案例6	动态创建数据库和基本表	266
案例7	大数据量的分页显示	268
案例8	图像在数据库中的存储	272



案例9	创建动态水晶数据报表	274
案例10	MSChart统计图表的使用	276
第7章	网络应用	280
案例1	获取DNS	281
案例2	获取自己的IP地址	286
案例3	在窗口中Ping一个IP地址	290
案例4	显示指定工作组内的所有计算机	303
案例5	电子邮件的发送	311
案例6	简易Web浏览器	320
案例7	FTP浏览器的制作	324
案例8	TCP点对点联机程序	333
案例9	UDP联机程序	346
案例10	C/S分布式QQ模型聊天室	359
第8章	控件操作	372
案例1	设置控件字体	372
案例2	动态填充组合框	374
案例3	列表控件的实现	382
案例4	动态添加控件	385
案例5	动画窗口的实现	397
第9章	综合案例	400
综合案例1	在OpenGL视图中绘制三维物体	400
综合案例2	使用DirectX制作粒子系统	415
综合案例3	中国象棋游戏	429
综合案例4	电子商务系统实现 (C/S)	489



第 1 章

VC++.NET编程基础

本章内容

- 案例1 第一个Windows程序
- 案例2 进制转换器
- 案例3 打印杨辉三角形
- 案例4 确定一个月的天数
- 案例5 石头剪子布猜拳游戏
- 案例6 创建动态菜单
- 案例7 创建停靠工具栏
- 案例8 递归与全局变量的案例——打靶程序
- 案例9 调用C++.NET中的数学函数
- 案例10 在列表框中加入图片

Visual Studio.NET是Microsoft公司推出的、目前使用极为广泛的、基于Windows平台的可视化编程环境，是计算机界公认的优秀的应用开发工具之一。使用Visual Studio.NET开发工具，能够进行可视化的设计，调试和配置端到端的解决方案，以用来解决所遇到的商业问题。通过开发服务器端的Web服务器组件，并且使这些组件通过XML和HTTP服务被其他开发者或者用户接受，就可以获得更多的商业机会。在非常完整的集成的IDE开发环境中，用户可以选择最适合的语言。

利用Microsoft的基本类库MFC（Microsoft Foundation Class，微软基础类）来开发Windows应用程序比以往任何时候都要容易，其最新版本增加了许多新的特性，如Web应用程序、托管C++扩展，在开发ATL、Web、DCOM、MFC、数据库应用程序等方面都有了很大的增强。这套开发软件为C++扩充提供一个便捷的方法，使得现有的C++代码可以在.NET平台上运行，特别是ATL服务器和COM组件两个方面有较大的扩充。

尽管如此，Visual C++在Windows桌面应用程序上的编程方法仍可分为两类：一类是非Window编程，如控制台程序（一种与早期DOS相兼容的程序，用来学习基本C++的编程思想），另一类是Window编程，这一类编程方法又可分为两种。一种是直接调用Windows提供的Win32 API（应用程序接口）函数开发Windows应用程序，像《Windows程序设计》（第五版）一书所讨论的就是这种方式；另一种是使用Visual C++所提供的MFC库中的“类”及其功能函数。由于MFC对Win32 API编程方式进行了极好的封装，使得我们不需要理解Windows内部编程机制就可以通过“应用程序向导”创建相应的应用程序框架，这些框架能满足绝大多数开发者的需求，并且使得我们不必添加太多的代码就能实现最常用的程序功能。因此，MFC是Visual C++中影响最深的一种编程方式，并且这种编程方式将会一



直影响下去。

Windows程序设计是一种完全不同于传统的DOS方式的程序设计方法。它是一种以消息为基础，以事件为驱动的程序设计模式，程序提供给用户的界面中有许多可操作的可视对象。用户从所有可能的操作中任意选择，而每种操作都对应于某个特定的事件。这些事件发生后的结果是向程序中的某些对象发出消息，然后这些对象通过MFC的消息处理程序来自动调用相应的消息处理函数以完成特定的操作。Windows应用程序最大的特点就是，程序没有固定的流程，而只是针对某个事件的处理有特定的子流程，Windows应用程序就是由这些子流程构成的。

Windows应用程序本质上是利用面向对象技术的，程序中的可视对象在程序内部一般也是一个对象，用户对某个对象的操作通过事件驱动模式触发相应的可用方法。程序的运行过程就是用户的外部操作不断产生事件，这些事件又调用相应的对象处理方法。标准Windows程序工作原理如图1-1所示。

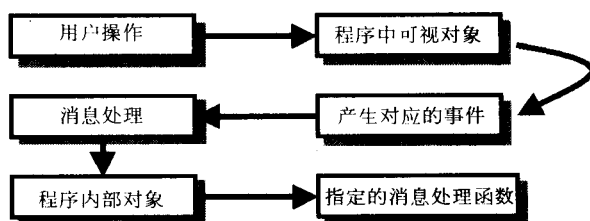


图1-1 Windows程序运行过程

Windows程序的运行是依靠外部事件来驱动的。简单来说就是程序利用一个While循环来不断地等待外部的输入，当检测到有新的输入后做出判断，并通过内部的分析处理程序转到相应的过程，然后做出适当的处理。

由此可见，消息在Windows程序中占据着举足轻重的作用，消息可以来自系统的各个方面，然后通过特定的消息翻译和分派程序，把消息发送到指定的处理工程，Windows程序都有一个如下所示的循环机制：

```

MSG msg;
While(GetMessage(&msg,NULL,NULL,NULL))
{
    TranslateMessage(&msg);    //翻译消息
    DispatchMessage(&msg);    //分派消息
}
    
```

消息实际上是一种类似于MSG的Windows内部结构，当系统中检测到WM_EXIT消息之后就会退出应用程序，这是Windows程序运行的核心过程。

刚刚使用Visual C++.NET开发程序的人多数被MFC中各式各样的宏所迷惑，因此在开始程序设计之前先熟悉MFC中的数据类型和各种宏是非常必要的。下面我们先向读者介绍一下MFC的基本内容。

MFC是微软公司推出的一套Application Frameworks，并且随着视窗操作系统的市场占有率的不断扩大，Visual C++集成编译环境的使用率也在不断增长，因此MFC也越来越多

地应用于Windows程序的开发过程中。

在早期，程序员想要开发Windows程序必须使用微软的SDK（Software Development Kit），通过直接调用数以千计的API（Application Programming Interface，应用程序接口），向系统提出各种请求来实现自己的程序要求。

而MFC则帮助我们把这些浩瀚的API封装起来，利用面向对象的技术组织起来，使它们具有面向对象的各种性质。MFC是一个非常巨大的程序类库，而整个MFC以一个Cobject类为基类，其他所有的类全部继承于它，这就给我们开发应用程序提供了非常方便的条件。

1. MFC中的宏

MFC中的宏非常多，而且不易理解，表1-1列出了常用的几个MFC宏的名称，供读者参考。读者可以通过阅读MSDN中相应的帮助内容来增加这方面的理解。

表1-1 MFC中的宏

宏名称	作用
DECLARE_DYNAMIC	执行时声明类信息
IMPLEMENT_ DYNAMIC	实现所声明的类
DECLARE_DYNCREATE	声明要动态创建的类
IMPLEMENT_ DYNCREATE	实现要动态创建的类
DECLARE_SERIAL	声明文档内容的读写
IMPLEMENT_ SERIAL	实现文档内容的读写
DECLARE_MESSAGE	声明消息
DECLARE_MESSAGE_MAP	声明消息循环图

2. MFC中的数据类型

表1-2所列的是MFC中的数据类型，经常出现在MFC的代码当中，希望读者能熟记这些数据类型，并在自己的应用程序中灵活使用，这对于提高程序效率、降低开发难度都有很大的帮助。

表1-2 MFC中的数据类型

数据类型	所表示的意义
BOOL	布尔型变量
BSTR	32bit的字符指针
BYTE	8bit的无符号整数
COLORREF	32bit数值，代表一个颜色值
DWORD	32bit无符号整数
LONG	32bit带符号整数
LPARAM	32bit数值，作为窗口函数或callback函数的指针
LPCSTR	32bit数值，用于指向一个常数字符串
LPSTR	32bit数值，用于指向一个字符串



(续表)

数据类型	所表示的意义
LPCTSTR	32bit数值, 用于指向一个常数字符串
LPTSTR	32bit数值, 用于指向一个字符串。此字符串可以移植到Unicode和DBCS
LPVOID	32bit数值, 指向一个未指定类型的数据
LRESULT	32bit数值, 作为窗口函数或callback函数的返回值
UINT	在Win16中是一个16bit的无符号整数, 在Win32中是一个32bit的无符号整数
WNDPROC	32bit指针, 用来指向一个窗口函数
WORD	16bit无符号整数
LPCRECT	32bit指针, 指向一个矩形结构
WPARAM	窗口函数或callback函数的一个参数, 在Win16中是16bit, 在Win32中是32bit
POSITION	用于MFC的集合类中, 用来描述一个位置结构

案例1 第一个Windows程序



案例演示与说明

如果读者是一个编程初学者, 且刚刚接触C语言的课程, 你可能会有点失望和怀疑: 这就是C语言吗? 靠它就能编出软件? 无法想像Windows桌面上一个普通的窗口是怎样出现在眼前的。

下面是学习C语言时的第一个程序: HelloWorld, 大家应该很熟悉了。

```
#include <stdio.h>
int main( )
{
    printf("Hello World\n");
    return 0;
}
```

如果是在Windows下该怎么做呢? 其实很简单, HelloWorld在Windows版本的等价程序如下:

```
#include <windows.h>
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, PSTR szCmdLine, int iCmdShow)
{
    MessageBox(NULL, "Hello Windows!", "HelloMsg", 0);
    return 0;
}
```

运行上面的程序后会随着“叮”的一声, 出现一个对话框。可以看到标题栏上有“HelloMsg”, 对话框中央出现“Hello Windows!”字样, 以及一个“确定”按钮。恭喜恭



喜！你的第一个Windows程序已经问世了！是不是很简单呢。虽然你对WinMain和MessageBox中的几个参数还感到迷惑，不过在以后的学习中你会慢慢掌握的，这里就不详细介绍了。当然，仅仅弹出一个消息窗口肯定不是我们想要的Windows程序，程序运行效果如图1-2所示。

在关闭窗口后，会弹出一个对话框，如图1-3所示。



图1-2 程序运行效果图

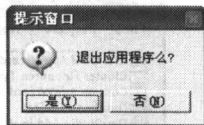


图1-3 弹出的对话框



知识要点

1. Windows应用程序的结构

Windows程序具有相对固定的结构，对开发者而言，不需要书写整个过程，大部分过程由系统完成。程序中只要按一定的格式填写系统留给开发者的那一小部分，所需要完成的程序有：窗口类的定义、窗口的建立、消息函数的书写、消息循环。

首先，我们要了解Windows的消息机制。消息，简单地说就是对Windows做出的动作。例如，用户每次单击键盘上的一个键、移动鼠标或单击控件（如滚动条），Windows都生成一条消息，其目的是告知程序已发生用户事件，并将该事件中的数据传递到程序中。同样，程序也可以生成消息，以允许该程序控制的各个窗口相互之间进行通信并分配任务。

一个消息由一个消息名称（UINT）和两个参数（WPARAM, LPARAM）组成，并由一个窗口接收。在窗口的过程（WNDPROC）中可以对消息进行分析，对自己感兴趣的消息进行处理。例如，如果希望在窗口中进行图形输出就必须对WM_PAINT进行处理。其中WM_PAINT就是Windows定义的标准消息之一。

其次，我们需要建立自己的窗口，建立后会得到系统返回的窗口句柄（HWND），后继的针对窗口的操作都针对句柄进行。

2. 注册窗口类

建立窗口前，需要制定好这个窗口的相关属性，最主要的就是将自己定义的消息处理函数与窗口关联，其他的属性还包括菜单、图标等。这个属性指定步骤是通过指定“窗口类”来完成的。对于自己建立的窗口，这个“窗口类”需要自己制定，也即自己填充一个WNDCLASS结构，然后向系统注册。对于一些特殊窗口，如按钮等控件，它们的行为是系统制定好了的，所以不需要自己注册，直接使用对应的“窗口类”名称就行了。

3. 建立窗口

建立窗口时，注册的“窗口类”名称作为参数传入。这样，当有针对该窗口的消息时，将调用“窗口类”中指定的消息处理函数，在其中得到处理。

最后，我们需要建立消息循环，系统会把针对这个消息依次放到程序的“消息队列”中，由程序自己依次取出消息，再分发到对应的窗口中去。因此，建立窗口后，将进





入一个循环。在循环中，取出消息、派发消息，循环往复，直到取得的消息是退出消息。循环退出后，程序即结束。



实现步骤

(1) 先建立一个空的Win32 Application项目，如图1-4所示。

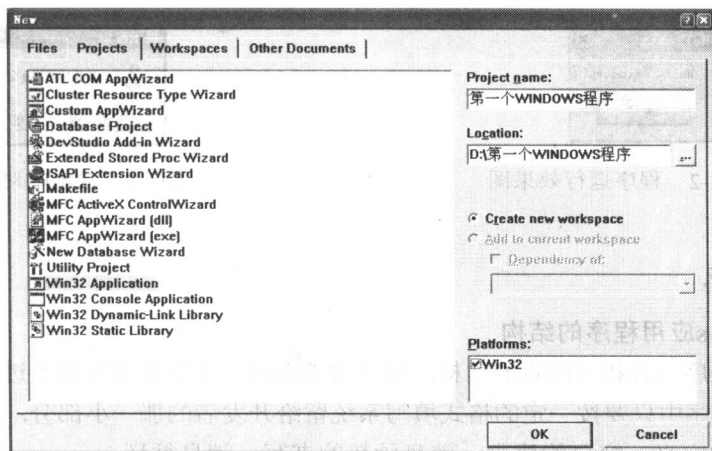


图1-4 建立项目对话框

(2) 选择建立一个空项目，如图1-5所示。

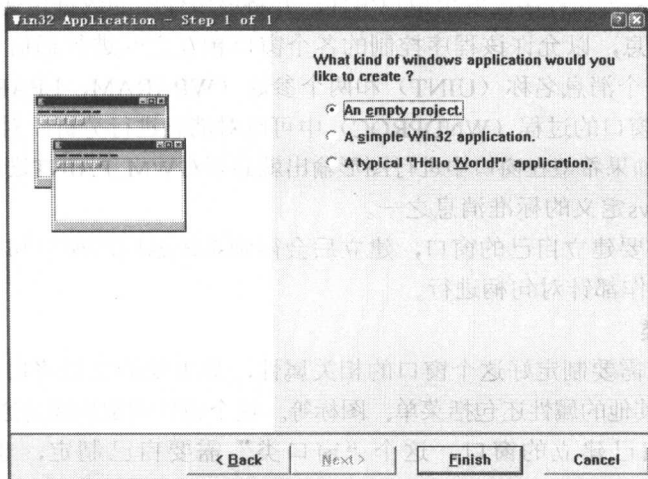


图1-5 建立项目步骤对话框

(3) 新建一个C++程序 (C++ Source File)，如图1-6所示。



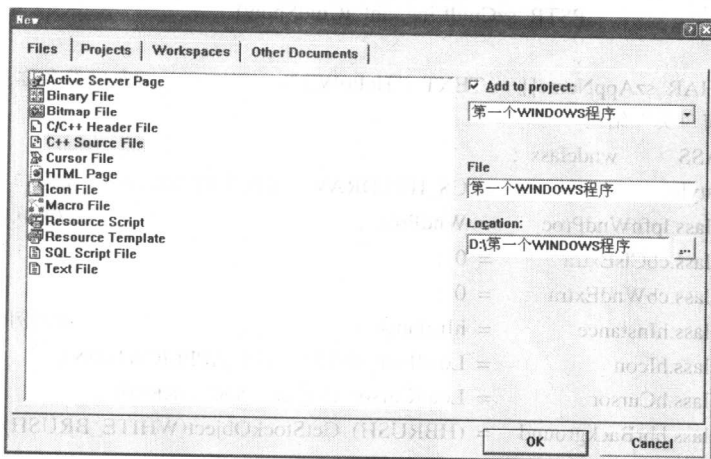


图1-6 建立项目步骤对话框



程序代码与解释

(1) 消息处理函数

```
#include <windows.h>
//参数: 窗口句柄, 消息, 消息参数, 消息参数
LRESULT CALLBACK WndProc(HWND hwnd, UINT message, WPARAM wParam, LPARAM
lParam)
{
    //处理感兴趣的消息
    switch (message)
    case WM_CLOSE:
        //关闭窗口消息
        if (MessageBox (NULL, TEXT("退出应用程序么?"), TEXT("提示窗口"),
MB_YESNO | MB_ICONQUESTION) == IDNO)
        {
            //选择“否”, 返回0, 停止关闭窗口
            return 0;
        }
        case WM_DESTROY:
            //窗口销毁消息, 程序需结束, 发退出消息, 以退出消息循环
            PostQuitMessage (0);
            return 0;
        }
    //其他消息交给由系统提供的默认处理函数
    return ::DefWindowProc (hwnd, message, wParam, lParam);
}
```

(2) 应用程序主函数

```
//参数: 实例句柄, 前一个实例的句柄, 命令行参数, 窗口显示方式
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
```





```

        PSTR szCmdLine, int iCmdShow)
{
    static TCHAR szAppName[] = TEXT ("HelloWin") ;           //窗口类名称
    //定制“窗口类”结构
    WNDCLASS    wndclass ;
    wndclass.style      = CS_HREDRAW | CS_VREDRAW;           //窗口类型, 风格
    wndclass.lpfWndProc  = WndProc ;                         //关联消息处理函数
    wndclass.cbClsExtra   = 0 ;
    wndclass.cbWndExtra   = 0 ;
    wndclass.hInstance   = hInstance ;                       //实例句柄
    wndclass.hIcon        = LoadIcon (NULL, IDI_APPLICATION); //图标
    wndclass.hCursor      = LoadCursor (NULL, IDC_ARROW);    //光标
    wndclass.hbrBackground = (HBRUSH) GetStockObject(WHITE_BRUSH); //画刷
    wndclass.lpszMenuName = NULL ;
    wndclass.lpszClassName = szAppName;                       //类名称
    //注册窗口
    if (!RegisterClass (&wndclass))
    {
        MessageBox (NULL, TEXT ("注册程序失败!"), szAppName, MB_ICONERROR) ;
        return 0 ;
    }
    //建立窗口
    HWND hwnd ;
    hwnd = CreateWindow (szAppName,           //窗口类名称
        TEXT ("Hello, Windows"),             //窗口标题
        WS_OVERLAPPEDWINDOW,                 //窗口风格
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        CW_USEDEFAULT,
        NULL,
        NULL,
        hInstance,                             //实例句柄
        NULL);
    ShowWindow (hwnd, iCmdShow) ;
    UpdateWindow (hwnd) ;
    //消息循环
    MSG        msg ;
    while (GetMessage (&msg, NULL, 0, 0))    //从消息队列中取消息
    {
        TranslateMessage (&msg) ;           //转换消息
        DispatchMessage (&msg) ;           //派发消息
    }
    return msg.wParam ;
}

```



案例2 进制转换器



案例演示与说明

在本案例中，我们将要解决一个不同进制的相互转换问题。一般来说，对于任意大于1的整数 n ，存在 n 进制，其特点是基数为 n ，逢 n 进一。其中最常用的是二进制、八进制和十六进制。

任意进制的数字对应的十进制数值为：

$$K_n \times B^n + K_{n-1} \times B^{n-1} + \cdots + K_1 \times B^1 + K_0 \times B^0 + K_{-1} \times B^{-1} + K_{-2} \times B^{-2} + \cdots + K_{-m} \times B^{-m}$$

上式中， B 称为数字系统的基数， B^n 至 B^0 称为数字 K_n 至 K_0 的权值。

案例运行效果如图1-7所示。

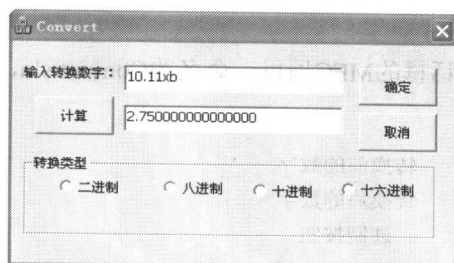


图1-7 程序运行效果图



知识要点

1. 数的进制与基数

计数的进制不同，则它们的基数也不同，常用的几种进制如表1-3所示。

表1-3 常见的进制

进制	基数	特点
二进制	0, 1	逢二进一
八进制	0, 1, 2, 3, 4, 5, 6, 7	逢八进一
十六进制	0, 1, 2, ..., 9, A, B, C, D, E, F	逢十六进一

2. 数的权

不同进制的数，基数不同，每位上代表的值的大小（权）也不相同。

如：

$$(219)_{10} = 2 \times 10^2 + 1 \times 10^1 + 9 \times 10^0$$

$$(11010)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$(273)_8 = 2 \times 8^2 + 7 \times 8^1 + 3 \times 8^0$$

$$(27AF)_{16} = 2 \times 16^3 + 7 \times 16^2 + 10 \times 16^1 + 15 \times 16^0$$

