

21世纪大学计算机基础课程教材

Visual C++.Net

程序设计案例教程

杨琦 编



西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

21世纪大学计算机基础课程教材

Visual C++.Net

程序设计案例教程

杨琦 编

西安交通大学出版社
· 西安 ·

内 容 提 要

本书详尽介绍了使用 Visual C++ .Net 设计 Windows 应用程序的各种技术,并通过丰富的例题帮助读者迅速掌握这些方法。全书共分 12 章,内容主要包含 MFC 应用程序框架、消息传递机制、设备环境、资源、文档/视图结构、对话框和控件等编程基础,以及多媒体、COM 和 ActiveX 控件、数据库、软件安装和部署等专题。

本书适用于具有 C/C++ 知识的读者学习,可作为高等学校计算机专业或相关专业的教材或参考书,也可供应用开发人员学习参考。

图书在版编目(CIP)数据

Visual C++ .Net 程序设计案例教程/杨琦编. —西安:
西安交通大学出版社,2007.5
ISBN 978-7-5605-2431-3

I. V… II. 杨… III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 013883 号

书 名: Visual C++ .Net 程序设计案例教程
编 著: 杨琦
责任编辑: 屈晓燕 贺峰涛
出版发行: 西安交通大学出版社
地 址: 西安市兴庆南路 10 号 (邮编:710049)
网 址: <http://unit.xjtu.edu.cn/unit/jtupress>
电 话: (029)82668357 82667874(发行部)
(029)82668315 82669096(总编办)
电子邮箱: eibooks@163.com
印 刷: 西安市新城区兴庆印刷厂
版 次: 2007 年 5 月第 1 版 2007 年 5 月第 1 次印刷
开 本: 787 mm×1 092 mm 1/16
印 张: 16.75
字 数: 409 千字
书 号: ISBN 978-7-5605-2431-3/TP·492
定 价: 22.00 元

版权所有 侵权必究

前 言

Visual C++ .Net 是目前最新和最先进的软件开发工具之一,它汇集了 Microsoft 公司的技术精华。较原来的 Visual C++ 6.0 版本而言,在很多方面,如 ATL、DCOM、MFC、数据库等领域都做了改进,尤其在开发环境界面上变化很大,采用了全新的平面化操作界面,较其他同类产品具有明显的优势。

VC++ 程序设计对于理工科大学生是一门非常重要的课程。2006 年教育部规定理工类非计算机专业大学生计算机课程为 1+X,即大学计算机基础、程序设计、多媒体技术、网络和数据库等。如果不学习 VC++ 程序设计就很难掌握多媒体技术、网络和数据库的实现手段,也就难于理解这些课程真正的原理概念,因此关键是如何使学生容易学习 VC++。

作者从 1999 年开始教授 VC++,经过多年的教学摸索,总结了学好本课程的 3 个重点,即:(1)掌握添加消息处理函数的方法;(2)掌握将控件映射为对象的方法;(3)掌握两个工具(对象浏览器和 ActiveX 控件测试容器)的使用。只要学生能够掌握这些技术就会发现编写 VC++ 程序并不比编写 VB 程序困难多少,也可以理解为什么 VC 在功能上比 VB 强大。

本书是 Visual C++ 入门教科书,定位于非计算机专业学生程序设计能力的培养。为了使基础不高的初学者也能很快地掌握程序设计方法,本书在内容的选取与组织上做到精益求精,对于非主流技术的内容坚决删除,力求做到概念简单,示例典型,容易理解,易于学习,学有所用。具体特点如下。

(1)在设计本教程内容时,以面向对象的和结构化的程序设计方法的思想贯穿全书。

(2)重点介绍几个而不是全部具体的 MFC 类库,过渡平缓,对初学者来说入门容易。

(3)内容完整,难度适中,每道例题的代码并不很多,容易理解,对于组件和 ActiveX 也做了详细的介绍。

(4)举例恰当,方法适用,技巧很有代表性。

本书共分 12 章,内容主要包含消息传递机制、MFC 应用程序框架、设备环境、资源、文档/视图结构、对话框和控件等编程基础,以及多媒体、COM 和 ActiveX 控件、数据库、软件安装与部署等编程专题。

为了保证教学效果,在条件许可的情况下最好采用直接在计算机房进行的联机电化教学。在这种情况下,每个教学单元(即每章)可使用连续的 4 课时,先由教师讲解授课部分并对自学部分和调试技术等内容进行简短的指导(共 2 学时),然后学生便可在教师指导下上机练习(2 学时)。除此而外,如果能够一定数量的课外机时(如 20~30 小时)则更好。

在本书编写过程中得到冯博琴教授、刘路放教授和罗建军副教授的指导和帮助,并与吕军、赵英良、顾刚、朱丹军、崔舒宁、仇国巍、李波、卫颜俊、薛涛和张伟等同事进行了多次交流,

受益匪浅。以上同志还提供不少有用资料,在此对他们表示感谢。

另外,本书部分程序的源代码已放在网站:Http:202.117.35.163/vccourse/资源下载,需要的读者请自行下载。由于水平所限,不足之处在所难免,肯请广大读者批评指正。

作者的 E-mail:qyang@ctec.xjtu.edu.cn。

编 者

2006 年 12 月

目 录

第 1 章 用 MFC 开发 Windows 应用程序	(1)	4.4 快捷键和字符串表	(74)
1.1 Visual C++ .Net 简介	(1)	4.5 工具栏	(75)
1.2 Visual C++ 集成开发环境	(2)	4.6 状态栏	(76)
1.3 利用应用程序向导生成一个单文档应用程序	(3)	4.7 典型案例	(77)
1.4 MFC 应用程序框架剖析	(7)	习题	(83)
1.5 MFC 的简单数据类型	(10)	第 5 章 文档视图结构	(84)
1.6 在窗口的客户区输出文字和图形	(14)	5.1 文档/视图概念	(84)
1.7 匈牙利命名法	(17)	5.2 应用程序类和文档模板类	(85)
1.8 基本范例	(17)	5.3 框架类	(88)
习题	(20)	5.4 视图类	(89)
第 2 章 窗口类和消息处理机制	(21)	5.5 文档类	(90)
2.1 事件驱动与消息循环机制	(21)	5.6 文档/视图结构中各类对象之间的协作关系	(93)
2.2 标准的 Windows 消息	(24)	5.7 序列化	(94)
2.3 MFC 基本类	(26)	5.8 多文档应用程序	(96)
2.4 MFC 预定义宏	(30)	5.9 其他视图类	(98)
2.5 基本范例	(34)	5.10 典型案例	(104)
习题	(39)	习题	(109)
第 3 章 图形设备接口	(40)	第 6 章 对话框	(110)
3.1 图形设备接口的基本概念	(40)	6.1 模态对话框	(110)
3.2 绘图工具类	(43)	6.2 对话框的数据交换和数据检验	(114)
3.3 画笔与画刷	(44)	6.3 通用对话框	(117)
3.4 位图	(47)	6.4 非模态对话框	(120)
3.5 字体	(51)	6.5 典型案例	(121)
3.6 绘图模式	(55)	习题	(127)
3.7 库存图形对象	(56)	第 7 章 标准控件	(128)
3.8 典型案例	(58)	7.1 控件概述	(128)
习题	(63)	7.2 基于对话框的应用程序	(130)
第 4 章 资源	(64)	7.3 静态文本和编辑框控件	(130)
4.1 Windows 应用程序资源	(64)	7.4 按钮控件	(134)
4.2 位图和图标资源及其应用	(65)	7.5 列表框和组合框	(137)
4.3 菜单资源及其应用	(68)	7.6 滚动条控件	(141)

7.7 动画控件	(145)	10.3 使用 MFC 创建 ActiveX 控件	(202)
7.8 标签控件	(146)	10.4 ActiveX 控件测试容器	(207)
7.9 树形控件	(149)	10.5 在 Web 站点客户端使用 ActiveX 控件	(208)
7.10 典型案例	(151)	10.6 在 ActiveX 控件中引入对话框技术	(210)
习题	(153)	10.7 使用 ATL 创建 ActiveX 控件	(212)
第 8 章 多媒体应用程序设计	(154)	10.8 Visual Studio. Net 和 Active X 控件	(214)
8.1 多媒体基础	(154)	10.9 典型案例	(215)
8.2 媒体控制接口 MCI	(155)	习题	(217)
8.3 MCIWnd 窗口类	(158)	第 11 章 数据库应用程序开发	(218)
8.4 使用 Windows Media Player 控件	(161)	11.1 数据库概念及术语	(218)
8.5 利用 shockwave Flash 控件浏览 Flash 动画	(163)	11.2 结构化查询语言	(220)
8.6 使用 Microsoft Web 浏览器控件	(165)	11.3 Visual C++ 数据库开发技术简介	(221)
8.7 处理各类图像	(166)	11.4 使用 DAO 访问数据库	(222)
8.8 典型案例	(168)	11.5 使用 ODBC 访问数据库	(232)
习题	(176)	11.6 使用 ADO 访问数据库	(236)
第 9 章 创建 COM 组件	(177)	11.7 典型案例	(239)
9.1 COM/COM+ 与 .Net	(177)	习题	(247)
9.2 COM 的基本概念	(178)	第 12 章 软件的安装和部署	(248)
9.3 用 MFC 开发 COM 组件	(180)	12.1 打包和分发简介	(248)
9.4 用 ATL 开发 COM 组件	(183)	12.2 创建安装项目	(249)
9.5 组件的安装和注册	(186)	习题	(252)
9.6 使用 OLE/COM 对象浏览工具	(187)	附录一 安装 Visual C++ .NET	(253)
9.7 COM 组件的调用	(187)	附录二 MFC 类库	(256)
9.8 典型案例	(192)	参考文献	(262)
习题	(199)		
第 10 章 创建 ActiveX 控件	(200)		
10.1 ActiveX 组件技术基本概念	(200)		
10.2 开发 ActiveX 控件的方法	(201)		

学习目标

- 熟悉 Visual Studio .Net 集成开发环境
- 熟悉 Visual C++ 程序的建立、编译与运行
- 掌握在窗口的客户区输出文字和图形的函数
- 学会使用 Visual Studio .Net 帮助系统的使用

1.1 Visual C++ .Net 简介

2000 年 6 月 22 日微软向全球发布了 .Net, 引起全球的轰动, .Net 使程序员的编程方式产生了很大的变化。

.Net 平台在以 Internet 为基础的计算和通信激增的背景下, 通过先进的软件技术, 利用网络协议和格式协调众多的智能设备与网站, 从而提供更简单、更为个性化且更有效的 Internet 服务。

微软为了配合其 .Net 战略, 发布了 Visual Studio. Net。Visual Studio. Net 的主要目标是帮助企业更有效地适应快速变化且具有竞争性的市场环境。通过提供端对端的 Web 开发能力和可升级、可重用的服务器端组件, Visual Studio. Net 可以容易、快速地创建出强大、可靠的企业级应用程序解决方案。

Visual Studio. Net 主要包括 3 套产品: Visual C#. Net、Visual Basic. Net 和 Visual C++. Net。Visual C#. Net 和 Visual Basic. Net 主要用于开发基于微软公共语言运行库的应用程序和 XML Web 服务程序。Visual C++. Net 不仅可以利用 C++ 托管展开开发面向公共语言运行库程序, 而且还可以开发本机运行的应用程序, 直接对操作系统进行写操作, 因此可以得到最大的控制权。

Visual C++. Net 是使用 C++ 开发语言创建基于 Microsoft Windows 和 .Net 的应用程序、动态 Web 应用程序和 XML Web 服务的综合开发语言平台。Visual C++. Net 的开发环境包括活动模板库 ATL (Active Template Library)、微软基础类库 MFC

(Microsoft Foundation Class)、高级语言扩展以及允许开发人员有效编辑和调试代码的强大集成开发环境 IDE(integrated development environment)。Visual C++ .Net 可以开发 MFC 应用程序、托管扩展、ATL Server 等等。

Visual C++ .Net 为开发人员创建强大的应用程序提供了快捷的面向对象的开发语言。通过高级模板功能、低级平台访问和优秀的优化编译器,为开发功能强大的应用程序和组件提供了非常出色的功能。Visual C++ .Net 能给开发人员提供广泛的解决方案,能够建 Web 应用程序、基于智能客户端的 Windows 应用程序,以及用于瘦客户端和智能客户端移动设备的解决方案。C++ 是全球最流行的语言,Visual C++ .Net 则为开发人员提供了创建软件的世界级工具。

1.2 Visual C++ 集成开发环境

Visual C++ .Net 集成开发环境(IDE, Integrated Development Environment)提供程序的编辑、编译、链接、执行和调试。另外也提供 Icon、对话框(Dialog Box)、菜单(Menu)与工具栏(Toolbar)等资源的编辑,以及强大的在线帮助。通过该集成环境,程序员可以观察和控制整个开发进程。

图 1-1 显示了一个典型的 Developer Studio 主窗口。可以看出,主窗口可以分为几个部分:窗口顶部是菜单和工具栏,左面的一个子窗口是程序与资源编辑区,工作区的右面是项目工作区窗口。最下面是输出子窗口。

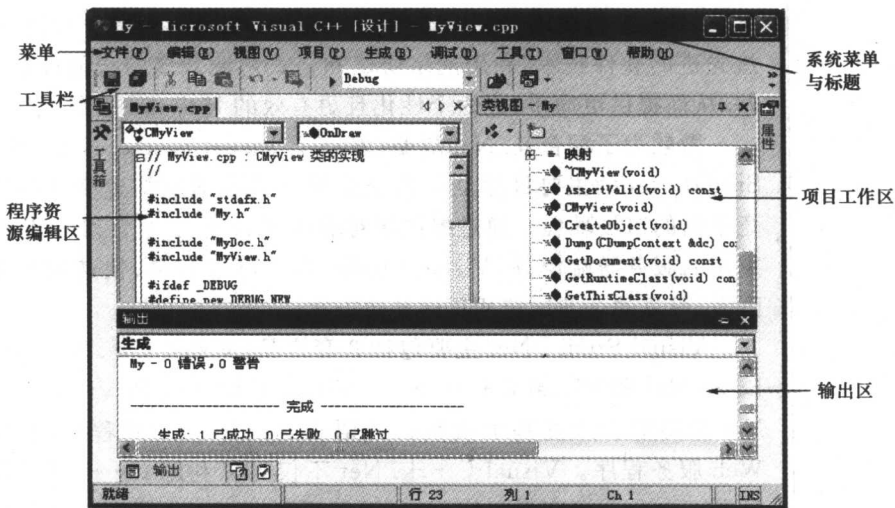


图 1-1 Developer Studio 主窗口

项目工作区包括解决方案资源管理器、类视图和资源视图。解决方案资源管理器显示项目中用到的程序文件,在某一个程序文件以鼠标左键双击,会在“程序与资源编辑区”出现这个程序文件的内容。

类视图显示项目中所包括的各个类的成员变量与成员函数,还显示全局变量与函数。以鼠标左键双击某一个显示项,就会在“程序与资源编辑区”出现这个显示项的程序位置。如果

以鼠标右键单击一个类,则会出现一个弹出菜单,允许加入这个类的成员变量、成员函数等。

资源视图显示项目中使用到的所有资源,在某一个资源项以鼠标左键双击,会在“程序与资源编辑区”出现这个资源的编辑页。以鼠标右键单击的话,则会出现一个弹出菜单,可以插入新的资源或进入这个资源的属性页等。

程序与资源编辑区用于编辑程序与对话框等资源。资源是用户界面组件的集合,它包括图标、菜单、工具栏、对话框等。

输出区显示程序编译、链接的结果,也可以显示程序调试执行时打印输出的结果,或者显示从文件搜索一段文字的结果。

1.3 利用应用程序向导生成一个单文档应用程序

所谓项目,指的是为了达成某个预期目标、在计划之下的所有作为。Visual C++项目的建立也是为此目的;它包括项目内所有文件的清单、使用的类、准备的资源,以及这个项目在 Visual C++ 的集成开发环境的设置状态等。只要取出一个项目文件,所有的内容与设置都会立刻载入集成开发环境。以下就是利用应用程序向导来创建一个项目的过程。

选择菜单选项“文件”|“新建”,在弹出的对话框中选择项目类型为“Visual C++项目”|“MFC”,在模板框中选择“MFC 应用程序”。输入项目名称 My。应用程序向导用项目名来构造项目中的文件和 MFC 派生类的类名。项目一旦建立后,就没有办法改变项目名了。如果想使用其他目录,可在“位置”文本框中另外指定一个路径。做好以上工作后,单击 OK 按钮时,正式启动应用程序向导创建一个应用项目,该向导共分 7 个步骤。如图 1-2 所示。

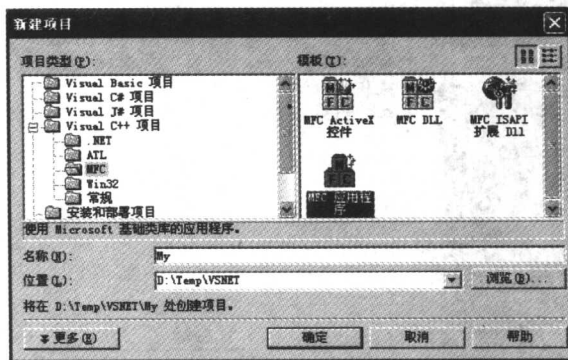


图 1-2 使用应用程序向导建立项目

步骤 1: 选择应用程序类型

在“MFC 应用程序向导”对话框的“应用程序类型”选项卡中对“应用程序类型”选择“单文档”,如图 1-3 所示。

步骤 2: 选择复合文档支持类型

从图 1-4 所示的“复合文档支持”选项卡中可以看到,这里有几种不同的复合文档选项,这里保留默认选项。

步骤 3: 填写文档模板字符串

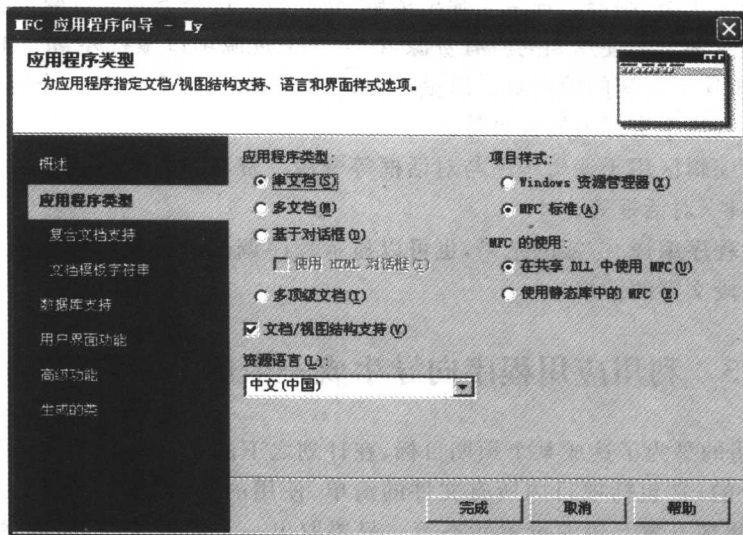


图 1-3 确定应用程序类型

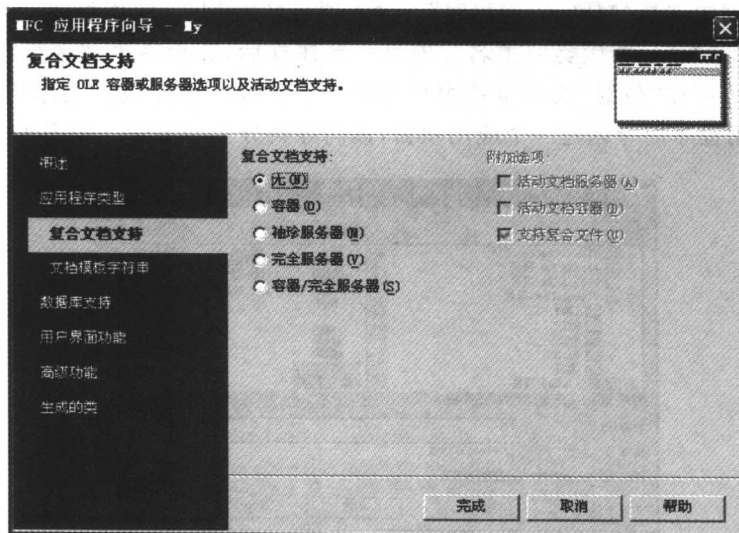


图 1-4 选择复合文档支持类型

在“文件扩展名”文本框中填上所需要的文档扩展名。如果希望文档最后保存成扩展名为 dic 的文件格式,就在这里填上 dic。

步骤 4: 确定数据库支持选项

如果应用程序不使用数据库,单击 Next 按钮跳过这一步,如图 1-5 所示。

该步骤的对话框右部的单选按钮决定了应用程序向导添加到项目中的数据库支持程度:

- 无:在生成项目时,把数据库支持的库文件排除在外。
- 仅支持头文件:只生成数据库包含头文件和库文件,但不为数据库类生成源代码。适

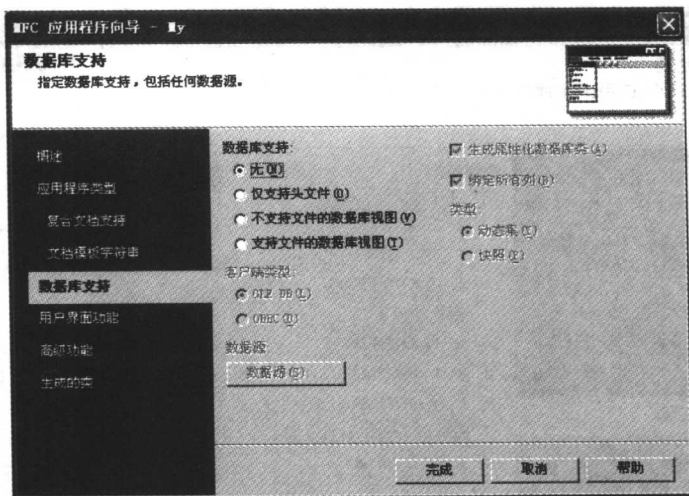


图 1-5 确定数据库支持选项

用于那些开始时不使用数据库,但打算以后添加数据库支持的项目。

- 不支持文件的数据视图:包含数据库头文件和库文件,同时也创建一个记录视图和记录集。生成的应用程序支持文档,但不支持序列化。

- 支持文件的数据视图:生成的应用程序框架既支持数据库文档,也支持序列化操作。

步骤 5:确定用户界面特征

用于设置主框架窗口的一些属性,包括框架窗口是否使用最大化按钮、最小化按钮,窗口启动时是否最大化或最小化等。如果希望应用程序一执行就占据整个屏幕,那么就选中“最大化”复选框;如果不希望让应用程序的右上角具有最小化按钮,可以取消选中“最小化框”复选框。

步骤 6:确定高级功能

这里涉及了一些特殊的选项,如图 1-6 所示,比如“打印和打印预览”在默认时被选中,也可以不被选中它。“区分上下文的帮助”在默认情况下是没有选中的,如果希望文档有“帮助”的话,请选中这个复选框。其他几个选项与 ActiveX 及组件对象模型有关。这里接受默认值。

步骤 7:查看向导即将创建的类的情况

向导一共创建 4 个类(不包括 About 类,它与应用类 CMyApp 类放在一个文件中,用于显示应用的“关于对话框”),它们分别是应用类 CMyApp、框架类 CMainFrame、文档类 CMyDoc 与视图类 CMyView。下面先看一下程序的运行结果。

【例 1-1】 利用 MFC 应用程序向导创建一个单文档应用程序。

解题步骤:

(1) Visual C++ .Net, 在 Windows 的任务栏中通过菜单“开始”|“程序”| Microsoft Visual Studio .Net, 调用 Microsoft Visual Studio .Net 的开发环境。

(2) 在 Microsoft Visual Studio .Net 的开发环境中,通过菜单“文件”|“新建”|“项目”,调用“新建项目”窗口,新建 Visual C++ 项目,选用 MFC 应用程序模板,项目名称为“My”。单击“确定”按钮,进入 MFC 应用程序向导界面。

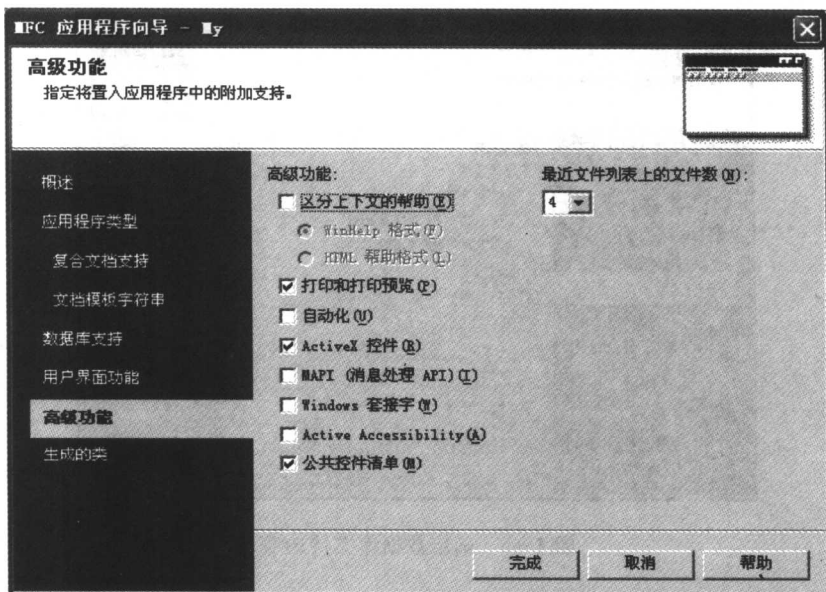


图 1-6 确定高级功能

(3)在 MFC 应用程序向导中的“应用程序类型”选项中,应用程序类型选择“单文档”,其余选项采用默认设置。

(4)在“用户界面功能”选项中,可以选择应用程序的界面特性,这里采用默认值。

(5)在“生成的类”选项中,可以看到应用程序向导即将创建的类、头文件和实现文件的名称。

(6)其他选项均采用默认属性,单击“完成”按钮,向导将为应用程序生成必要的框架文件并在工作区窗口中打开生成的项目。

(7)至此,利用 MFC 向导已完成了应用程序框架的全部创建工作,使用“生成”菜单的“生成 My”项,编译源程序,编译成功后按下 Ctrl+F5 运行程序,即可见到程序的运行效果,如图 1-7 所示。

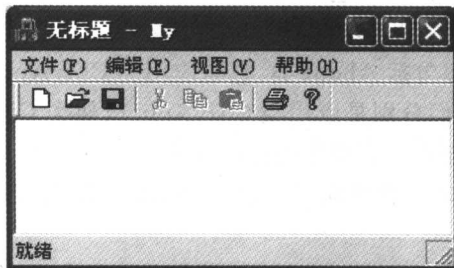


图 1-7 第一个 SDI 程序运行界面

(8)修改工程的目录名“My”为“01-01”,为编写下一个程序做准备。

分析:在项目文件夹中找到 readme.txt 文件,此文件包含组成 My 应用程序的每个文件

的内容摘要。

My.vcproj:这是使用“应用程序向导”生成的 VC++ 项目的主项目文件。它包含有关生成文件的 Visual C++ 版本的信息,以及有关用“应用程序向导”所选择的平台、配置和项目功能的信息。

My.h:这是应用程序的主头文件。它包含其他项目特定的头文件(包括 Resource.h),并声明 CMyApp 应用程序类。

My.cpp:这是包含应用程序类 CMyApp 的主应用程序源文件。

My.rc:这是程序使用的所有 Microsoft Windows 资源的列表。它包含存储在 RES 子目录中的图标、位图和光标。可直接在 Microsoft Visual C++ 中编辑此文件。项目资源包含在 2052 中。

res\My.ico:这是一个图标文件,用作应用程序的图标。此图标包含在主资源文件 My.rc 中。

res\My.rc2:此文件包含不由 Microsoft Visual C++ 编辑的资源。应将所有不能由资源编辑器编辑的资源放在此文件中。

MainFrm.h, MainFrm.cpp:这些文件包含框架类 CMainFrame,该类派生自 CFrameWnd 并控制所有的 SDI 框架功能。

res\Toolbar.bmp:此位图文件用于创建工具栏的平铺图像。初始工具栏和状态栏在 CMainFrame 类中构造。使用资源编辑器编辑此工具栏位图,并更新 My.rc 中的 IDR_MAINFRAME TOOLBAR 数组以添加工具栏按钮。

MyDoc.h、MyDoc.cpp 文档:这些文件包含 CMyDoc 类。编辑这些文件以添加特殊的文档数据并实现文件的保存和加载(通过 CMyDoc::Serialize)。

MyView.h、MyView.cpp — 文档视图:这些文件包含 CMyView 类。CMyView 对象用于查看 CMyDoc 对象。

StdAfx.h、StdAfx.cpp:这些文件用于生成名为 My.pch 的预编译头文件(PCH)和名为 StdAfx.obj 的预编译类型文件。

Resource.h:这是标准头文件,它定义新资源 ID。Microsoft Visual C++ 将读取并更新此文件。

1.4 MFC 应用程序框架剖析

1.4.1 MFC 的程序进入点

要追踪一般 C++ 语言的应用程序,当然要先找到它的 main 函数,再从它开始下手。而 Windows 的程序,是由 WinMain 函数开始的。在 Windows API(Application Programming Interface)中编写窗口程序的基本结构大致如图 1-8 所示。

当然,实际上 WinMain 函数与 WinProc 函数的交互关系并没有如此的简单,例如关闭窗口动作其实是由一连串的消息驱动的。在 MFC,情况也大致相同,主程序(WinMain)在建立窗口的同时,也建立其他用户界面,并初始化应用程序。建立完成之后,剩下的动作几乎都由消息激活。所以一个窗口响应函数(WinProc)必须配合着使用。

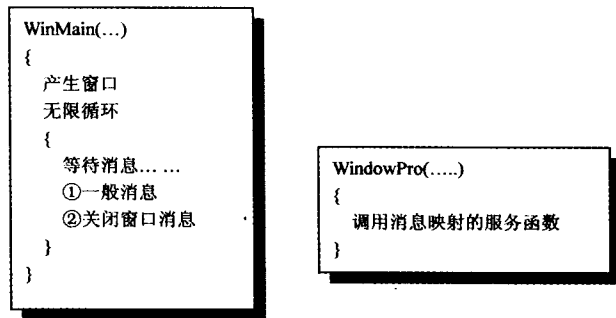


图 1-8 Windows API 窗口程序的基本结构

WinProc 函数处理的消息,大部分是 MFC 默认的系统消息。这些消息或者是在 MFC 的机制下生成,或者依应用程序的设置或要求而发出。此外,程序员也可以自行定义新的信息及其消息响应函数。

WinMain 与 WinProc 函数等基本的窗口框架都被 MFC 隐藏封装起来,程序员不必撰写基本的窗口程序框架,就可以开发应用程序。在应用程序中,几乎都只是为了反应用户通过用户界面的一个动作要求。所以,只要专注于设计用户界面,与其映射的消息相应函数所必须执行的任务就可以了。

1.4.2 MFC 库简介

Microsoft 提供了一个基础类库 MFC(Microsoft Foundation Class),其中包含用来开发 C++ 应用程序和 Windows 应用程序的一组类。使用 MFC 可以大大简化 Windows 编程工作。它将各种类结合起来构成了一个应用程序框架,其目的就是让程序员在此基础上建立 Windows 应用程序。

MFC 框架定义了应用程序的轮廓,并提供了用户接口的标准实现方法,程序员所要做的仅仅就是通过预定义的接口把具体应用程序所特有的东西填入这个框架内。而且 Visual C++ 也提供了相应的工具来完成这个工作:应用程序向导用来生成初步的框架文件;资源编辑器用于帮助直观地设计用户接口;类视图的属性用来协助添加代码;类库实现了应用程序特定的功能等。

目前的 MFC 版本中包含了 100 多个类,不同的类实现不同的功能。MFC 中的类根据是否从 CObject 类派生可分为两种:CObject 类的派生类及非 CObject 派生类。非 CObject 派生类数量不多,但大多很常用,如 CTime,CTimeSpan,CString 和 CFile 等。CObject 是大多数 MFC 类的根类或基类,它具有很多有用的特性:支持序列化、运行时类信息访问、对象诊断输出和与集合类兼容等。

MFC 库中的类根据层次关系划分可分为如下若干类:

- (1) 根类:CObject;
- (2) 应用程序体系结构类,包括 CCmdTarget、CWinThread、CWinApp、CDocument 等;
- (3) 窗口、对话框和控件类,如 CWnd、CView、CDialog 等;
- (4) 绘图和打印类,如 CDC、CPen、CBrush 等;

- (5)简单数据类型类,如 CTime,CTimeSpan,CString 等;
- (6)数组、列表和映射类如 CArray,CList,CMap 等;
- (7)文件和数据库类,如 CFile,CDatabase,CRecordset 等;
- (8)Internet 和网络工作类,如 CSocket,CAsyncSocket 等;
- (9)OLE 类,如 COleDocument,COleClientDoc,COleServer 等;
- (10)调试和异常类,如 CException 等。

1.4.3 最简单的 MFC 应用程序

【例 1-2】 创建最简单的 MFC 应用程序,如图 1-9 所示。

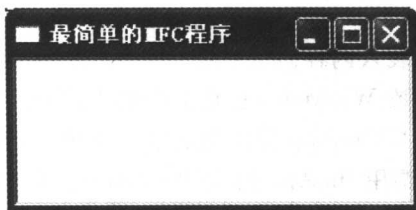


图 1-9 最简单的 MFC 应用程序

解题步骤:

(1)Visual C++.Net 在 Windows 的任务栏中,通过菜单“开始”|“程序”|Microsoft Visual Studio .Net,进入 Microsoft Visual Studio .Net 的开发环境。

(2)在 Microsoft Visual Studio .Net 的开发环境中,通过菜单“文件”|“新建”|“项目”,调用“新建项目”窗口,新建 Visual C++项目,选用 Win32 应用程序模板,下项目名称为“My”。单击“确定”按钮,进入 win32 应用程序向导界面。

(3)选择应用程序设置,在附加选项中选择“空工程”。

(4)选择窗口右边解决方案,右击源程序文件夹,在弹出的菜单选择“添加”|“新项目”,在文件名中输入 a1,添加如下代码:

```
#include <afxwin.h>
class CMainFrame:public CFrameWnd
{public:
    CMainFrame()
    {
        Create(NULL,"最简单的 MFC 程序");
    }
};
class CMyApp:public CWinApp
{
    virtual BOOL InitInstance();
};
CMyApp theApp;
```

```
BOOL CMyApp::InitInstance()
{
    m_pMainWnd=new CMainFrame();
    m_pMainWnd->ShowWindow(SW_SHOW);
    m_pMainWnd->UpdateWindow();
    return TRUE;
}
```

(5)设置应用程序属性,选择“工程”|“win32 属性”,再“使用 MFC”选择“共享 MFC”。

(6)编译链接并运行程序,运行结果如图 1-9 所示。

分析:Microsoft Windows MFC 库应用的具体启动步骤:

(1)Windows 将用户程序装入内存。

(2)Windows 调用全程函数 WinMain,它是类库的一部分。

(3)WinMain 会自动搜索 CWinApp 派生类的唯一实例 theApp。

(4)找到后,WinMain 会调用 theApp 的 InitInstance 函数,它在派生类中被重新设置了。

(5)IniInstance 函数装入文档对象、主框架对象、视图对象。

(6)WinMain 调用 theApp 的 Run 函数,它启动窗口消息和命令消息的传递处理过程。

(7)当消息循环获得 WM_QUIT 消息后,消息循环结束,Run 函数调用 ExitInstance 函数。

(8)ExitInstance 函数负责完成应用程序结束之前的清楚工作。

1.5 MFC 的简单数据类型

在 MFC 定义了一些常用的简单数据类型,包括 CPoint 类、CSize 类、CRect 类、CString 类、CTime 类,分别代表坐标点、尺寸大小、矩形区域、字符串、时间和日期。

1. CPoint 类

CPoint 类封装了老的 POINT 结构,并重写了一些操作,使得对象使用起来更简单。POINT 是一个很简单的结构:

```
typedef struct tagPOINT
{
    LONG x;
    LONG y;
}POINT;
```

该结构描述了一个点,通常这个点是指屏幕或着在设备上下文环境中输出设备上的点。C++ 中 POINT 和 CPoint 是可以互换使用的,如

```
CPoint p1(24,36);
POINT p2={18,30};
```