

特版精品系列

Visual C++6开发 使用手册

(美) Kate Gregory 著

前导工作室 译

机 械 工 业 出 版 社
西蒙与舒斯特国际出版公司

这是一本介绍如何使用Visual C++6.0的书, 它将详细地告诉你如何使用Visual C++建立32位应用程序, 包括数据库应用程序、Internet应用程序以及ActiveX应用程序。

本书包括第1章到第28章以及附录部分。每一章都将介绍一个重要的编程任务, 或两个紧密相关的任务(如创建任务栏, 或为应用程序增加Help功能)。在每一章中, 详细的指导会告诉你如何创建一个或几个可运行的应用程序。本书的重点内容包括对话框与控件、消息和命令、视图/文档模型、在屏幕上绘图、打印、持久性和文件I/O、ActiveX编程、Internet编程、数据库访问、高级教材等。

本书适合于Visual C++6.0的开发人员, 尤其对具有Visual C++编程经验的程序员来说具有非常好的参考价值。

Kate Gregory: Special Edition Using Visual C++6.

Authorized translation from the English language edition published by Que Corporation.

Copyright 1998 by Que Corporation.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版, 未经出版者书面许可, 本书的任何部分不得以任何方式复制或抄袭。

本书封底贴有Prentice Hall防伪标签, 无标签者不得销售。

版权所有, 翻印必究。

本书版权登记号: 图字: 01-98-2510

图书在版编目(CIP)数据

Visual C++6开发使用手册/(美)格里高瑞 (Gregory, K.) 著; 前导工作室译. —北京: 机械工业出版社, 1999

(特版精品系列)

书名原文: Special Edition Using Visual C++ 6

ISBN 7-111-07007-0

I. V… II. ①格…②前… III. C语言—程序设计 IV. TP312

中国版本图书馆CIP数据核字 (98) 第40581号

出版人: 马九荣 (北京市百万庄大街22号 邮政编码100037)

责任编辑: 李云静 陈剑瓿

北京第二外国语学院印刷厂印刷 · 新华书店北京发行所发行

1999年2月第1版第1次印刷

787mm × 1092mm 1/16 · 40印张

印数: 0 0001- 10000册

定价: 62.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

译者序

在大家的期盼下，Visual C++ 6.0终于呈现在众人面前。相信大家最关心的一定是它究竟有哪些新特性和重大的改进。本书全面系统地介绍了Visual C++ 6.0的最新特性，通过丰富的示例和有力的论述，使你能够快速掌握这些新特性。

当然，在满足老用户需求的同时，本书也没有忘记那些初学Visual C++的读者。本书从创建最简单的Windows编程概念入手，详细地介绍了对话框和控件、消息和命令、文档/视图模型等内容，继而进一步深入地讨论了怎样编写32位的应用程序(包括数据库应用程序、Internet应用程序以及ActiveX应用程序)。同时，还介绍了一些高级编程知识，例如内存泄漏、异常处理、代码重用。本书最后的附录对于初学Visual C++的读者非常有用，因为它介绍了C++编程的基本概念、Visual C++ 6.0集成环境的使用、调试程序的方法等内容，这些都是在编写C++程序时必须掌握的。

本书的原文内容丰富，论述充分，非常适合初学Visual C++和希望对Visual C++有全面了解的读者。在翻译过程中，译者忠实于原著的内容和特点，付出了辛勤的劳动。相信本书一定能给读者带来很大的收获。

本书由前导工作室的由军、谭力和刘明辉组织翻译，前导工作室的所有成员共同完成了本书的翻译审校和录排工作。

在全书的翻译过程中，译者力求准确地反映原著的内容，但由于水平有限，加上时间紧迫，错误之处在所难免，望广大读者批评指正。

译者

1998年11月

前 言

Visual C++是在Windows 95和Windows NT上建立32位应用程序的强大、复杂的工具。与16位Windows应用程序或者不使用图形界面的老程序相比，这些应用程序要大得多，而且更加复杂。然而，随着程序的规模和复杂性的增加，程序员所做的工作实际上减少了，至少对于使用了正确工具的程序员来说是如此。

Visual C++就是正确的工具之一，借助于其生成代码的向导，它能在数秒内生成可运行的Windows应用程序的外壳。Visual C++附带的类库，即Microsoft Foundation Classes(MFC)，已成为许多C++编译器进行Windows软件开发的工业标准。可视化编辑工具使菜单和对话框的布局非常简单。学习使用这个产品所花的时间和精力在你完成第一个编程项目时将得到回报。

本书使用对象

本书将介绍如何使用Visual C++建立32位应用程序，包括数据库应用程序、Internet应用程序，以及使用ActiveX技术能力的应用程序。为了在本书有限的篇幅中包含众多的内容，必须省去一些内容，因此本书不包含下列内容：

- C++编程语言：你应当已经熟悉了C++语言。附录A“C++回顾和面向对象的概念”是为那些C++技巧需要提高的人准备的。

- 如何使用应用程序：你应当是个熟练的Windows用户，例如，能够缩放和移动窗口、双击，以及能识别常见的工具栏按钮。

- 怎样使用Visual C++作为C编译器：如果你已经使用了C语言，那么可以用Visual C++作为编译器。但是，新的开发人员应当使用C++。

- 不采用MFC的Windows编程：这种编程技术对于熟悉它的人来说也是可行的，但既然有了MFC，就不必去学了。

- ActiveX编程的本质：在关于ActiveX的章中有介绍，不过它只告诉你如何使用它工作。

如果符合下列情况，那么你就应该阅读本书：

- 了解一些C++和Windows的编程技术，但不熟悉Visual C++。如果刚开始编写程序，那么就可以更快速地学会使用本产品。

- 使用过Visual C++的老版本。你经常以一种方式学习编程，而注意不到新产品功能。

- 使用过一段时间的Visual C++ 6.0，并且开始觉得使用起来有困难了。也许你就是这样。

- 已经正式使用Visual C++ 6.0，并且需要在自己的产品中增加一个功能。对于像Help、打印和线程这样的任务，你将会找到一个很好的起点。

在开始阅读之前

你需要有Visual C++6.0的一个拷贝并已经安装完了。安装过程是很简单的，所以本书不包含这部分内容。

在购买Visual C++6.0之前，需要32位操作系统：Windows95、Windows NT Server或

Workstation。这意味着计算机的功能必须相当强大：比如说486或更好的处理器，至少16MB的内存以及500MB的硬盘空间，以及能以800 × 600像素或更高分辨率进行显示的显示器。本书中的插图均是以800 × 600的分辨率制作的，而且，正如你将看到的那样，有时也有些拥挤。书中的示例代码可以从Web获得，如果你拥有调制解调器并且可以访问Web的话，那么学习起来就会更简单了。

最后，必须保证在阅读本书过程中经常上机试验。如果不想输入所有代码，则不必输入，因为这些代码全部在Web网站上。不过在阅读时，你应随时准备打开文件查看代码。

本书包含的内容

利用Visual C++进行Windows编程这样的主题所包含的内容很广。本书分成学习章节(第1章~第28章)和附录(附录A~附录F)。你应该现在就浏览一下附录的标题，而且每当需要时，还应该再回过头来仔细阅读它们，这些附录是：

- 附录A “C++回顾和面向对象的概念”，帮助你回忆C++语言的基本概念和原理，以及使用面向对象编程的好处。
- 附录B “Windows编程回顾以及CWnd内幕”，讨论了Windows编程的具体细节，而这些细节现在已被CWnd这样的MFC类隐含了。
- 附录C “Developer Studio的界面、菜单和工具栏”，介绍了屏幕上所有的菜单、工具栏、编辑区，以及快捷方式等，它们组成了用户和Visual Studio之间高度复杂而且功能强大的界面。
- 附录D “调试”，介绍了与调试运行的应用程序有关的附加菜单、窗口、工具栏和命令。
- 附录E “MFC宏和全局变量”总结了Developer Studio向导所生成代码中的许多预处理宏、全局变量和函数。
- 附录F “有用的类”，介绍了本书所使用的操作日期、字符串和对象集合的类。

根据你的背景知识水平以及是否喜欢利用联机帮助，你可以决定略读这些附录，或者仔细地阅读。尽管这些内容并没有提供很多的示例程序，但是它们的确可以教会你许多东西。

本书的主要内容是从第1章到第28章。每一章都会介绍一个重要的编程任务，或两个紧密相关的任务，如创建任务栏，或为应用程序增加Help功能。在每一章中，详细的指导会告诉你如何创建一个或几个可运行的应用程序。

前9章讨论了几乎每个Windows应用程序都会用到的概念，其余章节的内容则不一定很常用。下面是本书所介绍的内容的快速浏览。

对话框与控件

哪个Windows程序会没有对话框？编辑框？按钮？对话框和控件对Windows用户界面来说是至关重要的，所有这些，甚至是简单的按钮或静态文本都是窗口。常用的控件使你可以充分地利用你的时间。例如像其他人一样使用相同的File Open对话框以及相同的分级树型控件等等。这样你就可以不必编写这部分的详细程序，可以把省下的时间用来学习其他程序。同样开发人员也可以采用控件，省下时间来编写操作系统。要想更多地了解控件，请参见第2章“对话框和控件”和第10章“常用控件”。

消息和命令

消息形成了Windows编程的核心。每当某个事件作用于Windows计算机时，例如单击鼠标

或按下某个键，就会触发一个消息并发送到一个或多个窗口，然后执行相应的动作。Visual C++使你易于编写捕捉这些消息并按消息进行动作的代码。第3章“消息和命令”介绍了消息的概念，以及MFC和Visual C++的其他方面是怎样处理这些消息的。

视图/文档(View/Document)模型

范例(Paradigm)是一个模型，是一种查看对象的方式。MFC的设计者假设每个程序均有一些内容想要存储在一个文件中，因此选择设计了一种框架。这种信息的集合就被称为“文档”，“视图”是查看文档的一种方式。把“文档”和“视图”分开有许多优点，这一点将在第4章“文档和视图”中进一步解释。MFC提供了继承文档类和视图类的类，因此像实现滚动条这样的普通编程任务将不再是什么问题了。

在屏幕上绘图

不管Windows程序多么聪明，如果不能在屏幕上显示文字或图形来告诉你程序正在运行什么，那么，没有人会知道程序已经做了什么。大量的工作已由视图类自动完成(这是采用文档/视图范例模型的优点之一)，但有时你不得不需要自己绘图。在第5章“在屏幕上绘图”中，你将学到关于设备环境、滚动以及更多的内容。

打印

为程序增加打印功能是非常容易的，只要重新利用一下绘制屏幕的代码即可。但如果包含一页以上的信息，事情就变得复杂了。第6章“打印和打印预览”介绍了所有这些内容，以及映射模式、页眉、页脚和其他内容。

持久性和文件I/O

有些美好的事物只是暂时的，如日历显示或联机闲谈窗口。但大多数程序能把文档保存到一个文件里，打开并加载该文件可重新创建某个已存储的文档，通过使用档案文件以及扩展I/O流运算符>>和<<的使用，MFC使这一功能非常容易完成。你会在第7章学到关于读写文件的所有内容。

ActiveX编程

ActiveX是OLE的后继产品，正是这一技术促进了应用程序在对象的层次上进行通信，允许在Excel电子表格中嵌入Word文档，或在任意的ActiveX应用程序中嵌入几百种对象中的任意一个。ActiveX的内容将在第13章“ActiveX的概念”、第14章“创建ActiveX容器应用程序”、第15章“创建ActiveX服务器应用程序”、第16章“创建自动服务器”和第17章“创建ActiveX控件”中介绍。

Internet

Microsoft公司意识到分布计算(由两台或多台不同的计算机共同承担任务)正变得越来越普及。程序需要相互通信，人们需要通过局域网(LAN)或Internet发送消息，而MFC正好具有支持这些通信方式的类。本书中有四章关于Internet的内容，它们是：第18章“Sockets、MAPI

和Internet”、第19章“使用WinInet类进行Internet编程”、第20章“创建一个Internet ActiveX控件”和第21章“活动模板库”。

数据库访问

数据库编程变得越来越容易。ODBC(Microsoft公司的开放数据库连接软件包)允许你的代码调用函数,来访问多种数据库文件(Oracle、DBase、Excel电子表格、普通文本文件以及传统的使用SQL的主机系统等)。你可以调用标准的命名函数,以及由数据库销售商或第三方提供的API来处理转换的问题。详细内容参看第22章“数据库访问”、第23章“最新的企业版”。

高级教材

本书还为已经掌握了基本内容的开发者专门提供了介绍高级内容的章节以提高编程的技巧。第24章“改进应用程序性能”介绍的技术用于防止内存泄漏、查找瓶颈以及发现代码中的错误。

重用是目前软件开发中非常流行的概念,尤其对于那些寻求机会降低开发预算的管理人员来说更是这样。如果想编写可重用的代码和组件,参见第25章“使用Gallery和自己的AppWizard实现重用”。

C++程序员都忙于学习使程序正确运行的基础知识,而忽略了真正使C++强大的特性。在第26章“异常和模板”中,你将学到如何有效地捕获错误,以及在许多不同的情况下怎样使用这些代码。

用户对高性能软件的要求的不断增长,开发人员必须学习所有的新技术,以便编写出能够提供快速响应的强大的应用程序。对于许多开发人员来说,编写多线程的应用程序是一种关键技术。关于这个内容,请参见第27章“Windows线程的多任务处理”。

第28章“未来的开发方法”介绍了几个肯定不用向初学者讲解的主题,包括讲述如何生成控制台应用程序,以及使用和创建你自己的DLL和使用Unicode。

使用本书的约定

本书附有很多代码。有时你需要看的只是一两行代码而已,因此代码和文本是混在一起的,例如:

```
int SomeFunction( int x, int y);
{
    return x+y;
}
```

你可以通过使用的字体来区别代码和正文。有时,你会看到一段代码太长而不能与正文混在一起的,则采用程序清单的方式,如:

清单0-1 清单例子

```
CHostDialog dialog(m_pMainWnd);
if (dialog.DoModal() == IDOK)
{
    AppSocket = new CSocket();
    if (AppSocket->Connect(dialog.m_hostname,119))
```

```

{
    while (AppSocket->GetStatus() == CONNECTING)
    {
        YieldControl();
    }
    if (AppSocket->GetStatus() == CONNECTED)
    {
        CString response = AppSocket->GetLine();
        SocketAvailable = TRUE;
    }
}
}
if (!SocketAvailable)
{
    AfxMessageBox("Can't connect to server. Please
    ➡ quit.", MB_OK | MB_ICONSTOP);
}

```

倒数第2行的字符(➡)称为代码继续符。它表示：为了适应页面，一行代码被分割了，但实际上这一行代码并不在这个位置分割。如果输入了该段代码，则不要在这个位置处分割代码，只要接着输入即可。如果从Web站下载该段代码，则会发现该行代码并没有在那个位置被分割，对此请不要感到困惑。

记住，代码就在书中以便于你理解，而不是供你输入的。所有代码也都在相应的Web站上。有时，你会在开发一个应用程序时看到一段代码的几个版本，而最终的版本在Web站点上。通过访问www.mcp.com/info或www.gregcons.com/uv6.htm就可以访问这个站点。

提示：提示的内容是你可能想知道的捷径或感兴趣的功能。

注意：它介绍了微妙但重要的特性。即使忽略了提示，也不要忽略注意。

警告：发出一个警告，并且是很严重的警告。它警告你：错误步骤会造成的可怕后果，因此应保证阅读遇到的警告。

开始入门学习

上面就是本书内容的简介。你已了解到了入门所必须的一切。现在，立即着手来学习用MFC编写Windows应用程序，接着就可以开发自己的应用程序了！祝你好运！

与原作者及出版商的联系方式：

原书书名与书号：Special Edition Using Visual C++6 ISBN:0-7897-1539-2

原作者Kate Gregory: WWW.gregcons.com

Executive Editor of Macmillan: adv_prog@mcp.com.

本书站点：www.mcp.com/info 或 www.gregcons.com/uv6.htm.

目 录

译者序

前言

第一部分 开始学习

第1章 创建第一个Windows应用程序	1
1.1 生成一个Windows应用程序	1
1.1.1 选择文档界面风格	2
1.1.2 数据库	3
1.1.3 复合文档支持	4
1.1.4 外观和其他选项	5
1.1.5 其他选项	7
1.1.6 文件名和类名	7
1.1.7 生成应用程序	8
1.1.8 试验应用程序	8
1.2 生成一个基于对话框的应用程序	10
1.3 生成DLL、控制台应用程序及其他	13
1.3.1 ATL COM AppWizard	13
1.3.2 Custom AppWizard	13
1.3.3 Database Project	13
1.3.4 DevStudio Add-In Wizard	13
1.3.5 ISAPI Extension Wizard	14
1.3.6 Makefile	14
1.3.7 MFC ActiveX ControlWizard	14
1.3.8 MFC AppWizard(DLL)	14
1.3.9 Win32 Application	14
1.3.10 Win32 Console Application	14
1.3.11 Win32 Dynamic Link Library	14
1.3.12 Win32 Static Library	15
1.4 在应用程序向导中改变选择项目	15
1.5 理解应用程序向导所生成的代码	15
1.5.1 单文档界面应用程序	15
1.5.2 其他文件	20
1.5.3 理解多文档界面应用程序代码	20
1.5.4 理解基于对话框的应用程序代码	23

1.6 回顾AppWizard的选择项和本章

小结 25

第2章 对话框和控件 27

2.1 理解对话框 27

2.2 创建对话框资源 27

2.2.1 定义对话框和控件ID 29

2.2.2 创建示例对话框 29

2.3 编写对话框类 30

2.4 使用对话框类 32

2.4.1 显示对话框 32

2.4.2 屏幕之后 34

2.4.3 使用列表框控件 35

2.4.4 使用单选按钮 38

第3章 消息和命令 40

3.1 理解消息机制 40

3.2 理解消息循环 41

3.3 消息映射 43

3.3.1 消息映射宏 43

3.3.2 消息映射如何工作? 44

3.3.3 MFC代码捕获的消息 45

3.4 怎样利用ClassWizard捕获消息 46

3.4.1 ClassWizard对话框 46

3.4.2 添加Windows消息处理对话框 47

3.4.3 捕获消息的类 48

3.5 识别消息 48

3.6 理解命令 49

3.7 理解命令更新 50

3.8 怎样利用ClassWizard捕获命令与命令更新 51

第二部分 从应用程序得到信息

第4章 文档和视图 53

4.1 理解文档类 53

4.2 理解视图类 55

4.3 创建Rectangles应用程序	57
4.4 其他视图类	61
4.5 文档模板、视图和框架窗口	62
第5章 在屏幕上绘图	64
5.1 理解设备环境	64
5.2 介绍Paintl应用程序	65
5.3 创建Paintl应用程序	66
5.3.1 在MFC程序中画图	66
5.3.2 切换显示	67
5.3.3 使用字体	69
5.3.4 改变窗口的大小和位置	72
5.3.5 使用画笔	73
5.3.6 使用画刷	74
5.4 滚动窗口	77
5.4.1 创建Scroll应用程序	77
5.4.2 添加代码增加行	80
5.4.3 添加代码减少行	80
第6章 打印和打印预览	83
6.1 理解MFC的基本打印和打印预览	83
6.2 缩放	85
6.3 打印多页	86
6.3.1 设置原点	91
6.3.2 MFC和打印	92
第7章 持久性和文件I/O	95
7.1 对象的持久性	95
7.2 File Demo应用程序	95
7.2.1 文档类的回顾	95
7.2.2 创建File Demo应用程序	96
7.3 创建一个持久类	99
7.3.1 File Demo 2 应用程序	100
7.3.2 CMessages类一览	100
7.3.3 在程序中使用CMessages类	103
7.4 直接读写文件	104
7.5 创建自己的CArchive对象	106
7.6 理解注册	106
7.6.1 注册是如何设置的	107
7.6.2 预定义的键	108
7.6.3 在MFC应用程序中使用“注册”	108
7.6.4 简单的应用程序	109

第8章 创建一个完整的应用程序:

ShowString	110
8.1 建立显示字符串的应用程序	110
8.1.1 用AppWizard创建一个程序的 空壳	110
8.1.2 显示字符串	111
8.2 建立ShowString菜单	113
8.3 创建ShowString对话框	116
8.3.1 ShowString的About对话框	116
8.3.2 ShowString的Options对话框	117
8.4 使菜单工作	118
8.4.1 对话框类	118
8.4.2 捕获消息	121
8.5 使对话框工作	122
8.6 向Options对话框添加外观选项	123
8.6.1 改变Options对话框	123
8.6.2 在对话框类中增加成员变量	124
8.6.3 在文档中增加成员变量	125
8.6.4 改变OnToolsOptions()	126
8.6.5 改变OnDraw()	127

第三部分 改善用户界面

第9章 状态栏和工具栏	129
9.1 使用工具栏	129
9.1.1 删除工具栏按钮	129
9.1.2 添加按钮到工具栏	131
9.1.3 CToolBar类的成员函数	134
9.2 使用状态栏	135
9.2.1 创建新的命令ID	136
9.2.2 创建缺省字符串	137
9.2.3 添加ID到指示器数组	138
9.2.4 创建窗格的命令更新处理程序	138
9.2.5 设置状态栏的外观	140
9.3 使用Rebars	142
第10章 常用控件	145
10.1 进度条控件	147
10.1.1 创建进度条	147
10.1.2 初始化进度条	148
10.1.3 操作进度条	148

10.2 滑块控件	149	11.7.3 更改“How to Modify Text”主题	192
10.2.1 创建滑块	149	11.8 Contents的调整	193
10.2.2 初始化滑块	150	第12章 属性页和属性表	195
10.2.3 操作滑块	150	12.1 属性表介绍	195
10.3 上下控件	151	12.2 创建属性表演示应用程序	196
10.4 图像列表控件	152	12.2.1 创建基本文件	196
10.4.1 创建图像列表	153	12.2.2 编辑资源	196
10.4.2 初始化图像列表	153	12.2.3 添加新资源	199
10.5 列表视图控件	154	12.2.4 建立资源和类的关联	200
10.5.1 创建列表视图	156	12.2.5 创建属性表类	201
10.5.2 创建列表视图的列	157	12.3 运行属性表演示应用程序	204
10.5.3 创建列表视图的项目	158	12.4 在应用程序中添加属性表	205
10.5.4 操作列表视图	159	12.5 把属性表变成Wizard(向导)	205
10.6 树形视图控件	163	12.5.1 运行Wizard Demo应用程序	206
10.6.1 创建树形视图	164	12.5.2 创建Wizard页	207
10.6.2 创建树形视图的项目	165	12.5.3 显示Wizard	207
10.6.3 操作树形视图	167	12.5.4 设置Wizard的按钮	208
10.7 Rich Edit控件	168	12.5.5 响应Wizard的按钮	208
10.7.1 创建Rich Edit控件	168		
10.7.2 初始化Rich Edit控件	169	第四部分 ActiveX应用程序与ActiveX控件	
10.7.3 操作Rich Edit控件	169	第13章 ActiveX概念	211
10.8 IP地址控件	172	13.1 ActiveX的目的	211
10.9 日期提取(Date Picker)控件	172	13.2 对象链接	213
10.10 月历(Month Calendar)控件	174	13.3 对象嵌入	214
10.11 滚动视图	174	13.4 容器和服务	215
第11章 帮助	176	13.5 面向更加直观的用户界面	216
11.1 不同类型的Help	176	13.6 组件对象模型	217
11.1.1 获得Help	176	13.7 自动化	219
11.1.2 显示Help	177	13.8 ActiveX控件	220
11.1.3 使用Help	178	第14章 创建ActiveX容器应用程序	221
11.1.4 编写Help程序	179	14.1 更改ShowString	221
11.2 帮助系统的组件	179	14.1.1 AppWizard生成ActiveX容器 代码	221
11.3 AppWizard支持的Help	180	14.1.2 恢复ShowString的功能	235
11.4 设计你自己的Help的方法	181	14.2 移动、改变大小和跟踪	236
11.5 命令Help的编程	182	14.3 处理多个对象和对对象选择	239
11.6 上下文Help的编程	183	14.3.1 击中测试	239
11.7 编写Help文本	186	14.3.2 画多个项目	240
11.7.1 更改占位符字符串	187	14.3.3 处理单击	241
11.7.2 添加主题	189		

14.3.4 处理双击	243
14.4 实现拖放	244
14.4.1 实现拖动源	244
14.4.2 实现释放目标	245
14.4.3 将视图作为释放目标注册	246
14.4.4 建立函数框架并添加成员函数	246
14.4.5 OnDragEnter()	247
14.4.6 OnDragOver()	249
14.4.7 OnDragLeave()	251
14.4.8 OnDragDrop()	251
14.4.9 测试和拖动目标	253
14.5 删除对象	253
第15章 创建ActiveX服务器应用程序	255
15.1 给ShowString增加服务器功能	255
15.1.1 AppWizard的服务器示例程序	255
15.1.2 再次显示字符串	269
15.2 既是服务器又是容器的应用程序	273
15.2.1 创建另一个版本的ShowString	273
15.2.2 嵌套和递归问题	273
15.3 ActiveX文档	274
15.3.1 ActiveX 文档做些什么	275
15.3.2 使ShowString成为ActiveX文档 服务器	276
第16章 创建自动服务器	279
16.1 再次设计ShowString	279
16.1.1 AppWizard的自动模板	279
16.1.2 揭示属性	282
16.1.3 OnDraw()函数	286
16.1.4 显示窗口	288
16.2 在Visual Basic中创建控制器应用 程序	290
16.3 类型库和ActiveX实质	292
第17章 创建ActiveX控件	295
17.1 创建一个骰子滚动控件	295
17.1.1 创建控件外壳	295
17.1.2 AppWizard的代码	297
17.1.3 设计控件	299
17.2 显示当前值	299
17.2.1 添加属性	299

17.2.2 编写绘图代码	301
17.3 响应鼠标单击和滚动骰子	303
17.3.1 通知容器	303
17.3.2 滚动骰子	305
17.4 创建更好的用户界面	306
17.4.1 位图图标	306
17.4.2 显示圆点	307
17.5 生成属性表	310
17.5.1 数字与圆点	310
17.5.2 用户选定的颜色	314
17.6 按要求滚动	319
17.7 将来的改进	319
17.7.1 允许与禁止滚动	319
17.7.2 具有与众不同的面数的骰子	320
17.7.3 骰子数组	320

第五部分 Internet编程

第18章 Sockets、MAPI和Internet	321
18.1 使用Windows Sockets	321
18.2 使用消息收发API(MAPI)	324
18.2.1 何为MAPI	324
18.2.2 Windows 95标识要求	325
18.2.3 MAPI的高级应用	326
18.3 使用WinInet类	328
18.4 使用Internet服务器API(ISAPI)类	329
第19章 用WinInet类进行Internet编程	333
19.1 设计Internet查询程序	333
19.2 创建查询对话框	334
19.3 查询HTTP站点	336
19.4 查询FTP站点	341
19.5 查询Gopher站点	342
19.6 使用Gopher发送Finger查询	345
19.7 使用Gopher发送一个Whois查询	347
19.8 未来的工作	349
第20章 创建一个Internet ActiveX控件	351
20.1 在Microsoft Internet Explorer Web页中 嵌入ActiveX控件	351
20.2 在Netscape Navigator Web页中嵌入 ActiveX控件	354

第六部分 高级编程技术

20.3 为了改编和初始化安全而注册	355
20.4 ActiveX控件与Java Applets的选取	358
20.5 使用AppWizard创建更快的ActiveX 控件	359
20.6 利用异步属性提高控件的加载速度	363
20.6.1 属性	363
20.6.2 使用BLOB	364
20.6.3 修改Dieroll	365
20.6.4 测试和调试Dieroll	370
第21章 活动模板库	372
21.1 为什么使用ATL	372
21.2 使用APPWizard入门	373
21.3 使用Object Wizard	373
21.3.1 在项目中添加一个控件	374
21.3.2 命名控件	374
21.3.3 设置控件属性	374
21.3.4 支持Stock属性	376
21.4 给控件添加属性	377
21.4.1 由Object Wizard生成的代码	377
21.4.2 添加ReadyState Stock属性	379
21.4.3 增加定制属性	379
21.4.4 初始化属性	381
21.4.5 添加异步属性	382
21.5 绘制控件	385
21.6 持久性和属性页	389
21.6.1 添加属性页	390
21.6.2 连接属性页到CDieRoll	393
21.6.3 在属性包中的持久性	393
21.7 在Control Pad中使用控件	393
21.8 增加事件	394
21.8.1 给事件接口添加方法	394
21.8.2 实现IConnectionPoint接口	395
21.8.3 触发Click事件	395
21.8.4 触发ReadyStateChange事件	396
21.9 暴露DoRoll()函数	397
21.10 Init Safe和Script Safe的注册	398
21.11 准备在设计模式中使用的控件	398
21.12 最小化可执行文件大小	399
21.13 在Web页中使用控件	401

第22章 数据库访问	403
22.1 理解数据库的概念	403
22.1.1 使用平面数据库模型	403
22.1.2 使用关系数据库模型	404
22.1.3 数据库访问	405
22.1.4 Visual C++ ODBC类	405
22.2 创建ODBC数据库应用程序	406
22.2.1 注册数据库	406
22.2.2 创建基本的Employee应用程序	408
22.2.3 创建数据库显示	411
22.2.4 添加和删除记录	413
22.2.5 OnRecordAdd函数分析	418
22.2.6 OnMove函数分析	418
22.2.7 OnRecordDelete()函数分析	419
22.2.8 排序与过滤	419
22.2.9 OnSortDept()函数分析	424
22.2.10 DoFilter()函数分析	424
22.3 在ODBC与DAO之间作出选择	424
22.4 OLE DB	426
第23章 最新的企业版	427
23.1 什么是企业版	427
23.2 理解SQL	427
23.3 在C++中使用SQL操作数据库	428
23.4 开发发布应用程序	429
23.4.1 建立Data Source	429
23.4.2 创建应用程序外壳	432
23.4.3 建立数据连接	434
23.4.4 使用查询设计器	436
23.4.5 存储过程	437
23.4.6 编写新的存储过程	439
23.4.7 存储过程与C++代码的连接	440
23.5 使用数据库	443
23.5.1 数据库设计器	443
23.5.2 数据库图表	444
23.6 理解Microsoft事务处理服务器	445
23.7 使用Visual SourceSafe	446
第24章 改进应用程序性能	448

24.1	使用ASSERT和TRACE防止错误	448
24.1.1	ASSERT: 检测逻辑错误	448
24.1.2	TRACE: 隔离程序中存在问题 的区域	449
24.2	增加只调试特性	450
24.3	防止内存泄漏	451
24.3.1	内存泄漏的常见原因	451
24.3.2	调试使用new和delete的代码	453
24.3.3	自动指针	454
24.4	通过优化使代码效率更高	455
24.5	利用配置文件找到瓶颈	456
第25章	使用Galiery和自己的AppWizard 实现重用	457
25.1	编写可重用代码的优点	457
25.2	使用Component Gallery	457
25.2.1	向Gallery中添加一个组件	458
25.2.2	在用户项目中使用Gallery组件	459
25.2.3	开发Gallery	460
25.3	定制AppWizard简介	460
25.4	小结	463
第26章	异常和模板	464
26.1	异常	464
26.1.1	简单的异常处理	465
26.1.2	异常对象	466
26.1.3	放置catch块	467
26.1.4	处理多种类型的异常	471
26.1.5	旧的异常机制	473
26.2	使用模板	473
26.2.1	模板介绍	473
26.2.2	创建函数模板	473
26.2.3	创建类模板	475
26.3	标准模板库	479
26.3.1	管理指针模板: auto_ptr	479
26.3.2	其他有用的STL模板	481
26.4	名字空间	481

26.4.1	定义一个名字空间	481
26.4.2	名字空间范围确定	482
26.4.3	未命名的名字空间	483
26.4.4	名字空间别名	484
第27章	Windows线程的多任务处理	485
27.1	简单线程	485
27.2	线程通信	488
27.2.1	利用全局变量通信	489
27.2.2	利用用户定义的消息通信	490
27.2.3	利用事件对象通信	492
27.3	使用线程同步	494
27.3.1	使用临界区	494
27.3.2	使用互斥	499
27.3.3	使用信号量	500
第28章	未来的开发方法	504
28.1	创建控制台应用程序	504
28.1.1	创建控制台可执行程序	504
28.1.2	编写一个面向对象的控制台 应用程序	505
28.1.3	搭架起离散的算法	507
28.2	创建并使用一个32位的动态 链接库	507
28.2.1	创建32位DLL	512
28.2.2	使用32位DLL	514
28.3	发送消息和命令	515
28.4	软件的国际化发展趋势	517

第七部分 附 录

A	C++回顾和面向对象的概念	519
B	Windows编程回顾以及CWnd内幕	533
C	Developer Studio的界面、菜单和 工具栏	542
D	调试	585
E	MFC宏和全局变量	598
F	有用的类	605

第一部分 开始学习

第1章 创建第一个Windows应用程序

本章提要:

- 生成一个Windows应用程序
- 生成一个基于对话框的应用程序
- 生成DLL、控制程序及其他
- 在应用程序向导中改变选项
- 理解应用程序向导所生成的代码
- 理解多文档界面应用程序
- 理解基于对话框的应用程序的几个要素
- 回顾应用程序向导的选项和本章的其他内容

1.1 生成一个Windows应用程序

Visual C++不仅能够编译代码, 并且还可以生成代码。使用应用程序向导(AppWizard)工具, 可以在很短的时间内创建一个Windows应用程序。本章将介绍如何利用应用程序向导来生成一个应用程序的原型, 它包含了所需的全部Windows样板文件代码。毕竟, 你并不是第一个想要应用程序具有如下特点的程序员: 其窗口界面的大小是可以调整的并含有最小和最大化窗口按钮, 它的“File”菜单中包含有“Open”、“Close”、“Print Setup”、“Print”和“Exit”等选项。

应用程序向导可以生成许多类型的应用程序, 但是对于初学者来说, 他最需要的是可执行程序(.exe)。绝大多数人也需要应用程序向导产生样板文件代码, 比如每一个程序都不可缺的类、对象和函数。要创建像这样的程序, 应选择“File | New”, 在弹出的New对话框中单击Projects选项卡, 如图1-1所示。

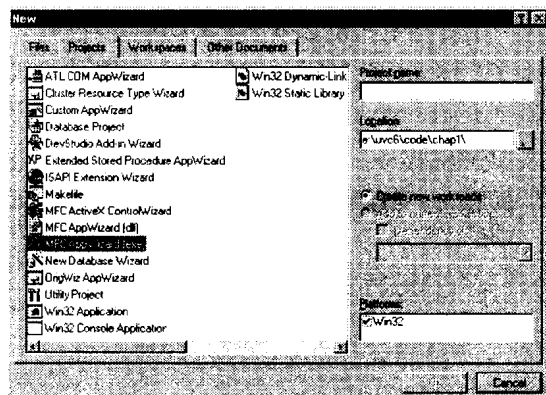


图1-1 New对话框的Project选项卡含有所要应用程序类型的选项

在左边的列表框中选择“MFC AppWizard[exe]”，并单击“OK”按钮，应用程序向导就会以一定的步骤进行工作。在每一步中，按照所要生成的应用程序的类型做出选择，然后单击“Next”按钮。在此过程中的任何时候，可以单击“Back”按钮以退回到上一步的设置，单击“Cancel”按钮退出整个生成过程，单击“Help”按钮获取更加详细的帮助信息，或者单击“Finish”按钮直接跳到向导的最后一步生成应用程序，而不必再应答任何问题(除非是最后一步，否则最好不要这样做)。以下部分将对每一步做出说明。

注意 MFC应用程序使用的是微软基本类库(Microsoft Foundation Classes)。在本书中将会介绍更多的有关MFC的知识。

1.1.1 选择文档界面风格

应用程序向导第一步会询问所要生成的应用程序其界面是否基于MDI、SDI或者对话框，它会根据选定的类型生成不同的代码和类，如图1-2所示。

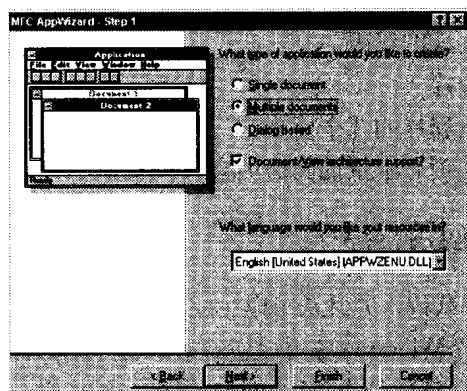


图1-2 用AppWizard创建典型应用程序的第一步是选择界面风格

有三种界面类型的应用程序可以选择：

- 单文档界面(SDI)应用程序，如“记事本”，一次只能打开一个文档。在SDI程序中，当打开一个新文件时，当前文件就会被关闭。
- 多文档界面(MDI)应用程序，如“Excel”或“Word”，可以同时打开多个典型文档文件。在程序界面上，有一个“Window”菜单和一个含有“Close”选项的“File”菜单。如果想把多个视图放在一个文档上，也必须创建MDI应用程序。
- 基于对话框的应用程序，例如图1-3所示的Windows附带的字符映射表程序(要启动这个程序可以单击任务栏“开始|程序|附件”，在“附件”中也许会有；如果没有，需要通过控制面板的“添加/删除程序”安装)，它根本不含文档，也没有菜单。

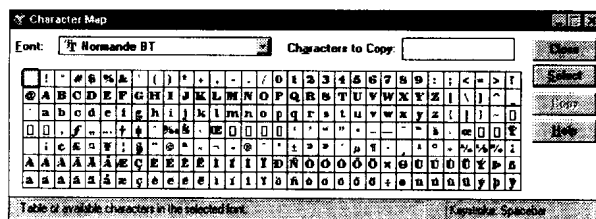


图1-3 基于对话框界面的字符映射表应用程序

当改变单选按钮的选择时，所选界面类型的示意图就会显示在屏幕的左部。

注意 基于对话框界面的应用程序与MDI或SDI应用程序有很大的不同。应用程序向导在生成基于对话框界面的应用程序时，其对话框也是不同的。这在下一节“生成一个基于对话框的应用程序”中会详细讲解。

在这些单选按钮的下方有一个复选框，它指示所生成的应用程序是否支持文档/视图(document/view)结构。应用程序的这种框架结构将在第4章“文档和视图”中加以解释。有经验的Visual C++程序开发者，特别是那些要从其他的开发系统中移植应用程序的开发人员，或许不会选择这种支持。作为本书的读者还是要选择这个选项。

在屏幕的下方有一个下拉框，它列出了支持应用程序资源的语言。如果设置的语言不是系统缺省设定的English[United States]，请确保在此设置的应用程序资源支持语言同系统语言保持一致。如果不是这样，将会在以后的类生成向导(Class Wizard)中遇到不可预料的错误(当然，如果最终用户将自行设定语言为美国英语，就不必选择这个选项。在那种情况下，可从控制面板改变系统语言)。完成以上选择后，单击“Next”按钮进入后续步骤。

1.1.2 数据库

使用AppWizard生成可执行Windows应用程序的第二步是选择数据库支持的层次，如图1-4所示。

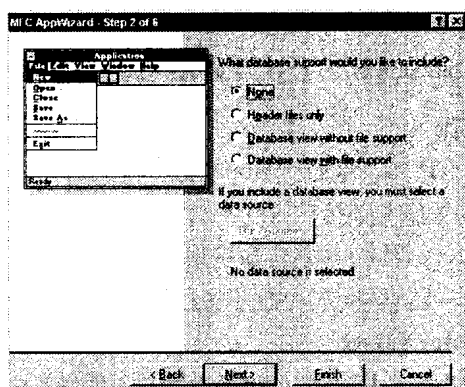


图1-4 利用AppWizard创建典型应用程序的第二步是设置所要使用的数据库选项

有四个数据库支持的选项：

- 如果不编写数据库应用程序，选择“None”。
- 如果想要访问数据库，但不想应用程序从类CFormView派生视图，或者其界面不含有记录(Record)菜单，则选择“Header Files Only”。
- 如果想要应用程序从类CFormView派生视图，并且其界面含有记录(Record)菜单，但又不需要连载文档，选择“Database View Without File Support”。可以使用CRecordset更新数据库记录，关于这个MFC类将在第22章“数据库访问”中详加讨论。
- 如果还需要将文档存盘(也许一些用户会这样做)，选择“Database View With File Support”。

第22章阐明了这些选项的意义，并举例说明使用MFC进行数据库编程的方法。如果选择