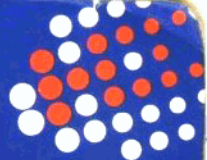


软件开发技术丛书



Windows 98

开发人员指南

刘 鎏 等著



机械工业出版社
China Machine Press

00133491

软件开发技术丛书

Windows 98开发人员指南

刘 璠 等著

沈 刚 审



机械工业出版社
China Machine Press

本书是一本关于Windows 98编程的书籍，主要讲述了如何使用Visual C++ 6.0编写Windows 98应用程序。全书共分为三个部分：第一部分为“Windows 编程入门”，主要讲述了如何编写一些常见的用户界面元素——菜单、工具栏、状态栏、对话框。第二部分为“编程进阶”，主要讲述了切分窗口与多视图、文档与视图、内存管理、线程和DLL方面的知识。第三部分为“ActiveX”，主要针对当今比较热门的ActiveX技术，为读者了解这项技术并进一步提高打下基础。

版权所有，翻印必究。

图书在版编目(CIP)数据

Windows 98开发人员指南 / 刘鉴等著.-北京:机械工业出版社, 1999.6

(软件开发技术丛书)

ISBN 7-111-07242-1

I. W… II. 刘… III. 窗口软件, Windows 98-程序设计 IV. TP 316

中国版本图书馆CIP数据核字(1999)第17255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: 陈剑珏

北京昌平第二印刷厂印刷·新华书店北京发行所发行

1999年6月第1版第1次印刷

787mm × 1092mm 1/16 · 28.75印张

印数: 0 001-6000册

定价: 63.00元(附光盘)

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

前 言

Windows 98是当今世界上最流行的操作系统之一，能在其上运行的应用程序千姿百态，数以千万计。之所以有众多的人使用Windows系统是因为它是图形操作系统，具有友好的用户界面、操作简单，用户易学、易用。而Windows编程则是利用了Windows自身所提供的API与Windows系统交互，从而实现Windows应用程序。

本书就是要讲述如何实现Windows编程。我们并不是直接使用Windows API函数，而是在Visual C++中，利用MFC——MFC封装了API函数——进行Windows编程。本书共分为三个部分：

第一部分 Windows编程入门

“Windows编程入门”这一部分主要向读者介绍编写Windows应用程序的一些基础知识。它包括第1章~第8章的内容。

通过学习这个部分，读者便能够编写管理应用程序中的一些常用用户界面元素的例程，可以通过菜单、工具栏、状态栏、对话框在应用程序与用户之间进行交互，可以利用图形设备接口向用户反映文档中的数据。此外，在该部分中还向用户简单介绍了异常处理。

第二部分 编程进阶

要成为一个高级程序员，单纯地学习一些关于界面的基础知识还远远不够。为了充分地利用有限的空间让尽可能多的信息同屏显示，我们需要了解切分窗口与多视图。为了理清应用程序中文档与视图的关系，我们需要学习文档与视图的关系。此外，为了充分发掘Windows操作系统的潜在威力，我们需要具备一些内存管理、多任务编程以及动态链接库方面的知识。

这个部分分别在第9章、第10章中讲述了切分窗口与多视图的关系、文档与视图的关系。在第11章~第13章中分别叙述了内存管理多任务编程以及动态链接库。通过学习这些知识，读者可以提升对应用程序、甚至操作系统的高度上来理解编程，而不再像以前一样仅从某一功能考虑，从而形成整体、系统的编程意识。

第三部分 ActiveX

ActiveX是一个专项技术，它是随着网络的蓬勃发展而开始流行的。ActiveX是由早期的OLE技术发展而来的。最初提出OLE技术仅是为了降低系统的复杂程度，提高程序的通用性、一致性。而如今ActiveX技术已成为网上畅游所不可缺少的技术。本部分主要是讲述如何用Visual C++实现容器应用程序、服务器应用程序、自动化应用程序以及ActiveX控件，从而为读者进一步学习ActiveX技术打下基础。

本书本着易学、实用原则出发，以大量的练习一步步引导读者轻松地实现Windows编程，特别适用于大中专院校学生及编程爱好者。相信读者在读完本书后一定会有所收益。

本书是集体智慧的结晶，参与本书编写工作的人员有刘饴、刘小明、李宏军、张世扬、赵天亮、吴齐、关汗羽、李治、李海涛、苏震、卢雪阳、张士华、陈辉、郑明峰、姚正思、沈毅明、汪宝川、高朴真、王立、顾仁、刘航、方东、许胜利、冯卫国、黄丽云和张梦天。由于时间短促，再加之作者的水平有限，书中的错误以及疏漏处在所难免，望广大专家、同行批评指正。

作者

1999年2月

本书附带光盘说明

在本书所附光盘中含有书中所有示例的源代码。这些文件按照章节进行组织，例如，第3章的示例程序均在chapter3子目录下。各章的示例程序均按照“Ex+章号+示例+ _数字”格式表示。例如，Ex13A_1则代表第13章第1个示例程序第一步完成的代码，Ex13B为第13章第2个示例程序。凡无“A”、“B”字样表示的则说明本章仅有此示例，凡没有“_1”或“_2”字样的则说明这些示例程序均一步完成。在书中完成示例处均以CD-ROM形式列出了源程序所在位置，读者可在阅读本书时运行光盘中的程序进行学习。

目 录

前言

第一部分 Windows编程入门

第1章 Windows 98与编程基础1

- 1.1 Windows 98的新特性1
- 1.2 Visual C++ 6.0简介1
- 1.3 Visual C++的系统工具3
 - 1.3.1 向导3
 - 1.3.2 资源编辑器3
- 1.4 编写Hello World普通窗口程序5
- 1.5 Hello World Win32控制台程序12
- 1.6 Hello World Java应用程序14
- 1.7 Hello World Visual Basic应用程序21
- 1.8 本章小结23

第2章 菜单24

- 2.1 菜单资源与菜单资源编辑器24
 - 2.1.1 菜单资源24
 - 2.1.2 菜单资源编辑器25
- 2.2 CMenu类27
- 2.3 菜单消息响应28
- 2.4 菜单的UI机制和CCmdUI类30
 - 2.4.1 菜单的UI机制30
 - 2.4.2 CCmdUI类31
- 2.5 普通下拉菜单32
- 2.6 动态地改变菜单38
 - 2.6.1 得到当前的CMenu对象38
 - 2.6.2 改变已存在的弹出式菜单39
 - 2.6.3 动态地改变菜单示例42
- 2.7 上下文菜单44
- 2.8 本章小结53

第3章 图形设备接口54

- 3.1 图形设备接口简介54
- 3.2 设备环境55
 - 3.2.1 获取设备环境55
 - 3.2.2 释放设备环境56

- 3.2.3 使用设备环境56
- 3.3 基本图形的绘制57

- 3.3.1 画点58
- 3.3.2 画直线58
- 3.3.3 画折线58
- 3.3.4 画曲线59
- 3.3.5 画封闭曲线60
- 3.3.6 绘制基本图形示例程序62

3.4 文本输出64

- 3.4.1 显示文本65
- 3.4.2 设置文本的前景颜色65
- 3.4.3 设置文本的背景颜色65
- 3.4.4 设置文本的背景模式66
- 3.4.5 设置字符间距66
- 3.4.6 设置文本对齐66
- 3.4.7 得到文本度量67

3.5 GDI对象68

- 3.5.1 画笔——CPen类68
- 3.5.2 画刷——CBrush类73
- 3.5.3 字体——CFont类78

3.6 本章小结81

第4章 鼠标、键盘与计时器82

4.1 鼠标82

- 4.1.1 鼠标消息82
- 4.1.2 鼠标消息处理84
- 4.1.3 捕捉鼠标85
- 4.1.4 更改鼠标的显示模式86
- 4.1.5 鼠标程序示例86

4.2 键盘101

- 4.2.1 输入焦点101
- 4.2.2 系统键与非系统键101
- 4.2.3 虚拟键代码102
- 4.2.4 键盘消息及其映射函数103
- 4.2.5 字符消息及其映射函数104
- 4.2.6 使用插字符105

| | | | |
|------------------------------|-----|----------------------------------|-----|
| 4.2.7 键盘输入示例程序 | 106 | 第7章 对话框和控件 | 179 |
| 4.3 计时器 | 113 | 7.1 对话框的种类 | 179 |
| 4.3.1 启动和关闭计时器 | 113 | 7.2 对话框资源 | 179 |
| 4.3.2 响应计时器消息 | 114 | 7.3 Windows标准控件 | 180 |
| 4.3.3 计时器程序示例 | 115 | 7.3.1 静态控件 | 181 |
| 4.4 本章小结 | 124 | 7.3.2 编辑控件 | 181 |
| 第5章 工具栏与状态栏 | 125 | 7.3.3 按钮控件 | 181 |
| 5.1 工具栏 | 125 | 7.3.4 列表框控件 | 182 |
| 5.1.1 工具栏资源 | 125 | 7.4 对话框的创建和显示 | 183 |
| 5.1.2 工具栏按钮及其状态 | 127 | 7.5 CDialog对话框类 | 183 |
| 5.1.3 工具栏与命令消息 | 127 | 7.6 对话数据交换/对话数据验证 | 185 |
| 5.1.4 CToolBar类 | 127 | 7.6.1 DDX机制 | 185 |
| 5.1.5 工具栏的创建和处理 | 129 | 7.6.2 DDV机制 | 186 |
| 5.1.6 工具栏的停靠和漂浮 | 130 | 7.7 使用MFC的标准对话框 | 186 |
| 5.1.7 工具提示 | 131 | 7.8 使用模式对话框 | 190 |
| 5.1.8 工具栏程序示例 | 132 | 7.9 使用无模式对话框 | 202 |
| 5.2 状态栏 | 157 | 7.10 本章小结 | 216 |
| 5.2.1 CStatusBar类 | 157 | 第8章 属性页和属性单 | 217 |
| 5.2.2 状态栏的创建 | 158 | 8.1 概述 | 217 |
| 5.2.3 状态栏的提示信息与状态指示 | 159 | 8.1.1 创建属性页的模板资源 | 217 |
| 5.2.4 状态栏程序示例 | 159 | 8.1.2 创建CPropertyPage派生类 | 218 |
| 5.3 本章小结 | 170 | 8.1.3 创建并显示模式CPropertySheet | 218 |
| 第6章 MFC异常处理 | 171 | 8.1.4 创建并显示非模式CPropertySheet | 218 |
| 6.1 Visual C++的异常处理机制 | 171 | 8.2 CPropertyPage与CPropertySheet | 219 |
| 6.2 设置自己的terminate()函数处理器 | 172 | 8.2.1 CPropertyPage | 219 |
| 6.3 MFC中的异常类 | 173 | 8.2.2 CPropertySheet | 220 |
| 6.3.1 CException | 173 | 8.3 属性对话框示框 | 222 |
| 6.3.2 CMemoryException | 173 | 8.4 本章小结 | 232 |
| 6.3.3 CNotSupportedException | 174 | 第二部分 编程进阶 | |
| 6.3.4 CFileException | 174 | 第9章 切分窗口与多视图 | 233 |
| 6.3.5 CArchiveException | 175 | 9.1 切分窗口 | 233 |
| 6.3.6 CResourceException | 175 | 9.2 CSplitterWnd类 | 233 |
| 6.3.7 COleException | 175 | 9.3 多视图选择 | 236 |
| 6.3.8 CUserException | 176 | 9.4 动态与静态切分窗口 | 236 |
| 6.3.9 CDBException | 177 | 9.5 滚动显示文本 | 237 |
| 6.3.10 CDaoException | 177 | 9.6 单视图动态切分窗口 | 239 |
| 6.3.11 CInternetException | 177 | 9.7 多视图静态切分窗口 | 246 |
| 6.4 Win32系统级异常 | 177 | 9.8 本章小结 | 251 |
| 6.5 本章小结 | 178 | | |

| | | | |
|----------------------------|-----|------------------------|-----|
| 15.3.1 COleTemplateServer类 | 370 | 16.5.1 创建框架自动化客户应用程序 | 414 |
| 15.3.2 COleServerDoc类 | 371 | 16.5.2 从类型库中创建类 | 415 |
| 15.3.3 COleServerItem类 | 372 | 16.5.3 编辑应用程序的资源 | 416 |
| 15.4 定制应用程序的资源 | 373 | 16.5.4 完善视图类 | 417 |
| 15.5 完善视图类 | 377 | 16.6 从客户程序控制服务器程序 | 424 |
| 15.6 完善应用程序的文档类 | 381 | 16.7 本章小结 | 425 |
| 15.7 完善服务器项的类 | 386 | 第17章 ActiveX控件 | 426 |
| 15.8 运行服务器应用程序 | 396 | 17.1 ActiveX/OLE控件 | 426 |
| 15.9 本章小结 | 401 | 17.2 ActiveX控件的属性集 | 426 |
| 第16章 自动化 | 402 | 17.2.1 属性 | 426 |
| 16.1 自动化概述 | 402 | 17.2.2 事件 | 427 |
| 16.2 自动化服务器应用程序 | 403 | 17.2.3 方法 | 428 |
| 16.3 创建自动化服务器应用程序 | 403 | 17.3 创建框架ActiveX控件 | 428 |
| 16.3.1 创建框架自动化服务器应用程序 | 403 | 17.4 完善应用程序的CEx17Ctrl类 | 431 |
| 16.3.2 定制应用程序的资源 | 404 | 17.5 增加库存事件和定制事件 | 433 |
| 16.3.3 完善视图类 | 406 | 17.5.1 增加库存事件 | 433 |
| 16.3.4 完善应用程序的文档类 | 407 | 17.5.2 增加定制事件 | 435 |
| 16.3.5 定义服务器的属性和方法 | 409 | 17.6 增加属性 | 436 |
| 16.4 自动化客户应用程序 | 413 | 17.7 增加控件方法 | 438 |
| 16.5 创建自动化客户应用程序 | 414 | 17.8 本章小结 | 449 |

第一部分 Windows编程入门

第1章 Windows 98与编程基础

随着计算机技术的飞速发展,软件编程概念已经发生了根本性的转变,应运而生的可视化编程技术获得了广泛应用。Microsoft公司推出的Visual Studio即是可视化编写Windows 98程序的重要工具,而Visual C++ 6.0是Windows 98编程的最强有力工具之一,是集创建、编辑、编译、链接、运行调试为一体的软件集成开发环境。

1.1 Windows 98的新特性

Windows 98不像Windows 95仅仅是一个优于DOS的32位操作系统,Windows 98是一个完全集成的32位保护模式的操作系统。对额外MS-DOS备份的需求已经不存在了。

程序支持多线程的能力是Windows 95具有的新特性,这个特性在Windows 98中得到了更好的支持。Windows 98是用其执行抢占多任务的功能来管理这些线程的。这些功能使系统具有更好的响应能力,而且理顺了后台处理。

Windows不再受旧的DOS FAT文件系统的束缚。那些12位和16位的FAT索引将不再存在。现在,Windows文件系统是由32位索引进行控制。用户不再担心磁盘文件簇的大小或是将批处理文件限制在8KB和32KB的磁盘空间上。除了节省小文件所占有的磁盘空间外,增强的文件系统使得长文件名更易于管理。在经过了许多年来文件名受8.3命名约定的限制后,这才是一个真正的好处。

设备驱动程序现在可以用32位代码编写。只要所编写的代码利用了32位编程模块的优点,这些驱动程序将具有高性能和智能化的内存使用。

此外,与Windows 95不同,Windows 98的内核完全是32位的,这意味着内存管理、进程调度和进程管理将更为有效。

1.2 Visual C++6.0简介

Visual C++ 6.0开发环境Developer Studio是在Windows 98/NT环境下运行的一套集成工具,由文本编辑、资源编辑器、项目建立工具、优化编译器、增量连接器、源代码浏览器、集成调试器等组成。

使用Developer Studio,不仅可以创建由Visual C++ 6.0使用的源文件和其他文档,而且可以创建、查看和编写任何与ActiveX部件有关的文档(ActiveX文档)。

在Developer Studio中,可以在项目工作区中组织文件(file)、项目(project)和子项目。可以使用工作区窗口来查看和访问项目中的各种元素。项目工作区可以含有多个项目,每个项目要么是顶层项目,要么是其他项目的子项目。

与Visual C++5.0相比较,Visual C++ 6.0还增加了许多新特征。这些新特征在编辑器、

编译器、调试器中都有所反映。下面我们将介绍一些在编程过程中经常用到的新特征：

1. 编辑器的语句自动完成特征

这种特征是由Intellisense系统实现的。当用户编辑代码时，Intellisense会在光标或鼠标位置附近显示类的成员函数、变量以及全局变量，如图1-1所示。用户可以在成员列表中选择成员来插入代码中。



图1-1 Intellisense自动显示类的成员函数、变量

使用Intellisense，用户还可以察看代码注释、函数声明和变量类型等信息，如图1-2所示。

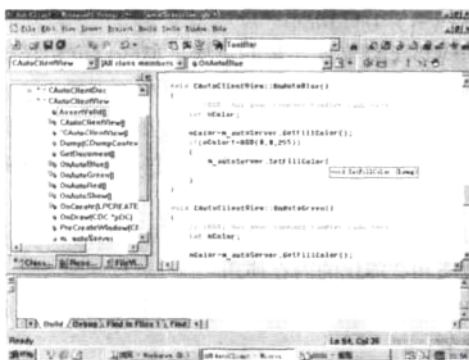


图1-2 Intellisense自动显示函数原型、标识符定义等

Intellisense的存在使我们工作得更轻松编写的代码更不容易出错。

2. AppWizard的增强

在Visual C++ 6.0中，当用户使用MFC AppWizard来创建应用程序时，AppWizard有了更多的选项。Visual C++ 6.0的AppWizard可以支持不基于文档/视图结构的单文档界面或多文档界面应用程序。此外，AppWizard还可以创建具备Windows Explorer风格、Web浏览器风格的应用程序。

3. 在线帮助与集成编程环境相分离

当打开Visual C++ 6.0帮助时，我们可以发现它与Visual C++ 5.0之间的不同。首先，Visual C++ 5.0把帮助文件集成到Developer Studio中。而在Visual C++ 6.0中，帮助是一个独

立的应用程序，可以单独运行。用户在集成开发环境中调用帮助时，系统会自动打开这个应用程序，如图1-3所示。

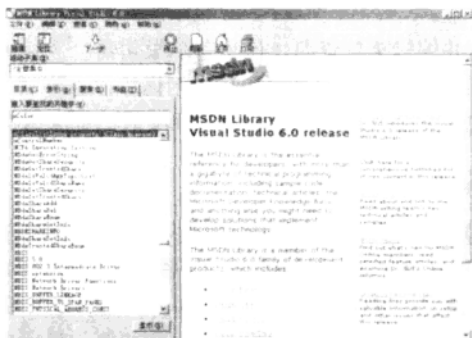


图1-3 独立运行的在线帮助

另外，读者可以从图1-3中看到，Visual C++ 6.0的帮助是使用HTML语言编写的，这使得访问相关帮助时更简单方便。

1.3 Visual C++的系统工具

Visual C++为我们顺利编程提供了两个向导工具——AppWizard、ClassWizard，以及若干个资源编辑工具。

1.3.1 向导

在Visual C++ 6.0中，可以使用向导(Wizard)、MFC类库和活动模板库(ATL)来开发Windows应用程序。

向导用于帮助用户生成各种不同类型应用程序的基本框架。例如，可以方便地使用MFC AppWizard来生成完整的从模板文件出发的基于MFC类库的源文件和资源文件；可以使用MFC ActiveX Control Wizard生成创建ActiveX控件所需要的全部模板文件(如源文件、头文件、资源文件、模块定义文件、项目文件、对象描述语言文件等)；可以使用ISAPI Extension Wizard生成创建Internet服务器(Server)或过滤器(Filter)所需要的全部文件；使用ATL COM AppWizard来创建ATL应用程序；使用Custom AppWizard来创建自定义的项目类型，并将其添加到创建项目时的可用项目类型列表中。

在创建应用程序的基本框架后，可以使用ClassWizard来创建新类，定义消息处理函数，覆盖虚拟函数，从对话框、表单视图或者记录视图的控件中获取数据并验证数据的合法性，在自动化对象中添加属性、事件和方法。此外，还可以使用WizardBar来定义消息处理函数、重载虚函数并浏览实现文件(.CPP)。

关于AppWizard和ClassWizard向导的使用方法，我们会在编程过程中进一步讲述。

1.3.2 资源编辑器

为支持可视化编程，Visual C++ 6.0除了提供集成的编译器、调试器以外，它还提供了大量用于可视化编程的资源编辑工具。这些资源编辑工具主要有：菜单编辑器、加速键编辑器、

资源符号编辑器、图形编辑器、工具栏编辑器、对话框编辑器等编辑器。这些编辑器对于可视化编程具有重要作用。

1. 菜单编辑器

菜单编辑器用于创建并编辑菜单资源。使用菜单编辑器，我们可以创建标准菜单和菜单项，可以为菜单或菜单项定义热键、加速键和状态栏提示；也可以创建上下文菜单，以使用鼠标右键来执行要频繁使用的命令。建立菜单或菜单项后，可以用ClassWizard为菜单选项编写要执行的代码。

2. 加速键编辑器

加速键表是一种Windows资源，它包含应用程序用到的所有加速键及相应的命令标识符。Visual C++ 6.0允许应用程序包含多个加速键表。加速键通常是菜单或工具栏上所用程序命令的键盘快捷键。定义加速键后，可以使用ClassWizard为加速键命令编写要执行的代码。

使用加速键编辑器，我们可以添加、删除、更改和浏览项目所用到的加速键，可以查看和更改与加速键表中每个条目有关的资源标识符(资源标识符用于在程序代码中引用加速键表中的每个条目)，还可以为某个菜单选项定义加速键。

3. 资源符号

资源符号由映射到整数值上的文本串组成，它用于在源代码或资源编辑器中引用资源或对象。在创建新的资源或对象时，系统自动为其提供缺省符号名(如IDD-ABOUTBOX)和符号值。缺省时，符号名和符号值自动保存在系统生成的资源文件resource.h中。

4. 图形编辑器

图形编辑器由一套功能强大的绘图工具组成，它可以用于绘制位图、图标和光标。

5. 工具栏编辑器

工具栏编辑器用于创建工具栏资源并可以将已有位图转换为工具栏资源。工具栏编辑器以图形方式显示要处理的工具栏和正被选择的工具栏按钮图像。

6. 对话框编辑器

对话框编辑器用于创建或者编辑对话框资源或对话框模板。使用对话框编辑器，可以做以下工作：

- 添加、排列或编辑控件。
- 改变控件的制表顺序(Tab Order)或助记键(Mnemonic Key)。
- 调整对话框布局。
- 添加或编辑ActiveX控件。
- 创建用户自定义控件。
- 导入Visual Basic表单到对话框资源中。
- 测试对话框。

7. 串编辑器

串表是一种Windows资源，它包含应用程序用到的所有串ID号、值和标题。例如，状态栏提示可以放在串表中。每个应用程序只能有一个串表。在串表中，串以16个为一组构成段或块。某一串属于哪一段取决于该串的标识符值。例如，标识符值为0-15的串放在第一段，为16-32的串放在第二段，等等。要将串从某一段移到另一段，必须改变其标识符值。通常，每个串只在使用时才调入内存。

串编辑器用于编辑应用程序的串表资源。使用串编辑器,可以在串表中浏览串,向串表添加新的串,从串表删除某一串,将串从某一段移到另一段,将串从某一资源文件移到另一资源文件,修改串及其标识符等等。

关于这些资源编辑器的具体用法,我们会在各章中第一次使用时具体介绍。

1.4 编写Hello World普通窗口程序

上一节曾提示, Visual C ++提供一种被称为AppWizard的工具。利用该工具,用户可以方便地按照自己的需要创建符合需要的应用程序模板。所谓应用程序模板,就是Visual C ++提供的,可以被其他程序采用的基本程序,即应用程序框架。用户可以在此基础上进一步编写自己的程序。下面,我们就来学习如何生成MfcApp应用程序。事实上, MfcApp应用程序仅对框架应用程序做了很少一点改动。

为了生成MfcApp应用程序的框架,启动Visual C ++,从File菜单中选择New菜单项。Visual C ++弹出New对话框,切换到Projects选项卡,项目的类型选择MFC AppWizard(exe), 在Project name: 编辑框中输入MfcApp, 在Location: 编辑框中选择想放置项目的位置, 如图1-4所示。一般而言, 最后一级文件夹的名称应当与项目的名称相同。然后单击OK按钮。

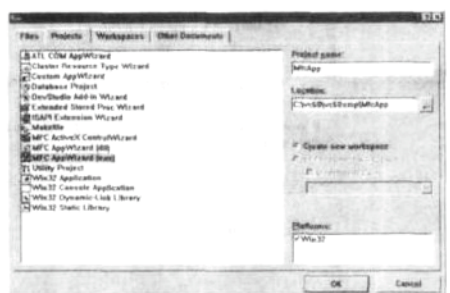


图1-4 Visual C ++的New对话框

在单击OK后,AppWizard开始询问关于应用程序特性的问题。这些问题包含六个“步骤”, 每一步都集中于MFC框架的一个特定部分。正如在图1-5中看到的, 每一步都有一个Next/Back按钮。这意味着如果走到了第4步时,发现在第2步犯了一个错误,那么可以回到前面的步骤。这样做不会丢失已经输入的任何信息。图1-5显示的是第1步的画面。

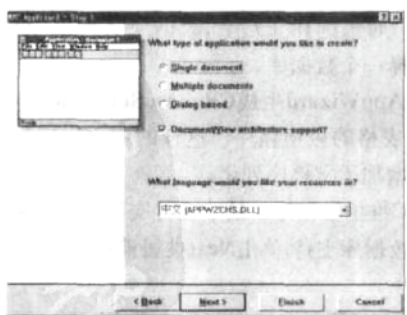


图1-5 AppWizard的第1步,从中选择应用程序的类型

1. 第1步——应用程序的类型和语言的类型

在第1步里,系统提问建立什么类型的应用文件。读者可以选择单文档(主窗口中只有一个窗口)、多文档(如Microsoft Word或Sysedit,可以在主窗口内开多个子窗口)或基于对话框(主窗口只不过是一个对话框,如CD Player应用程序)。

系统的缺省选项是多文档(Multiple document)。MfcApp使用单文档(Single document)。

读者也可以为资源选择语言。AppWizard为应用程序生成了大量的基于文本的资源(菜单提示、对话框和工具帮助)。

在What type of application would you like to create?下选择Single Document单选按钮,再单击Next按钮就到了第2步。

2. 第2步——数据库支持

第2步用于决定应用程序是否打算用ODBC类去存取一个数据库。图1-6显示了第2步的提问。

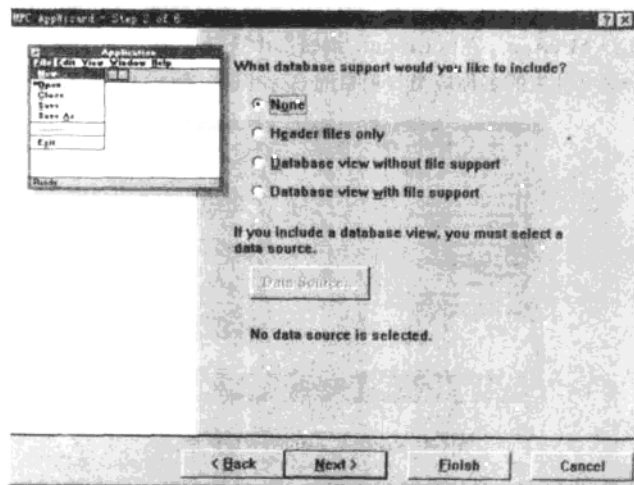


图1-6 AppWizard的第2步,从中选择是否要用ODBC支持

如图1-6所示,存在四种选择。缺省时表示不包括数据库支持。MfcApp使用这个选择,因为它不需要ODBC支持。如果想使用ODBC支持,只需让AppWizard包含ODBC头文件即可,或者为应用程序指定一个数据库。如果选择了后者,数据库必须已经存在。在继续下面的内容之前,AppWizard就要求用户选择一个数据库。

指定一个数据库会导致AppWizard生成CRecordSet和CRecordView的派生类来支持数据库。这些选择生成一个基于表格的应用程序。这样的应用程序易于查看和更新数据库。文件支持的扩充部分为应用程序增加了文档序列化。

这两个选项允许用ODBC维护数据库。缺少文件支持并不影响对数据库的读写。

由于该应用程序不使用数据库支持,单击Next按钮就可以到下一步。

3. 第3步——ActiveX支持

第3步为ActiveX支持,如图1-7所示对话框即为该步的画面。

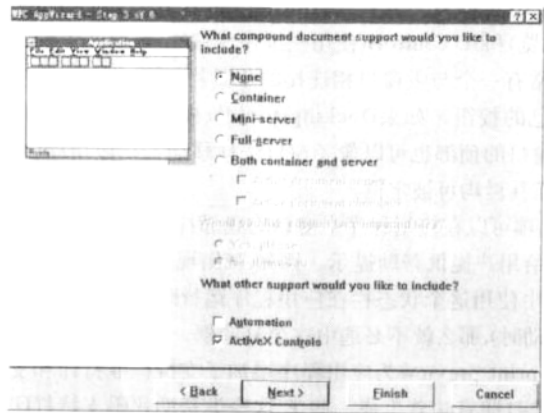


图1-7 AppWizard的第3步,从中选择是否要用ActiveX支持

缺省情况下,代码中没有对ActiveX的支持,MfcApp不会是ActiveX-aware,所以需要使用缺省值。

应用程序可以是一个ActiveX容器或服务器,或二者兼具。如果一个应用程序就是一个ActiveX容器,它可以包含链接和嵌入对象。容器不能为其他的ActiveX程序提供支持,它只能维护嵌入对象。另一方面,服务器创建复合文档对象,使其嵌入ActiveX容器。如果选择了Mini-Server,应用程序就不能独立运行,而是被调用使之作为其他程序建立ActiveX对象。全服务器能够独立运行,并能够为其他应用程序建立ActiveX对象。从图1-7中可以看到,一个应用程序可以同时作为容器和服务器。

如果想使应用程序面向ActiveX自动化服务,则选择“*Yes, please*”。否则,就不能被其他使用ActiveX自动服务的程序所访问。不管在什么情况下,单击一下Next按钮就进入第4步。

关于ActiveX更多的知识将在第三部分中详细讨论。

4. 第4步——应用程序的特性和高级选项

第4步是最复杂的一步,因为它有许多选择。图1-8显示了第4步的画面。

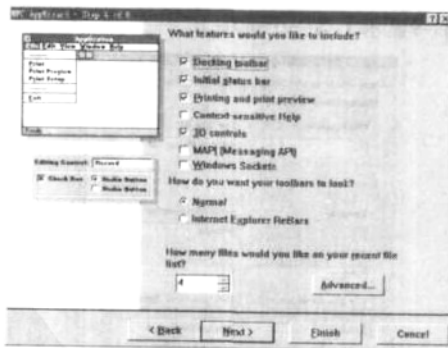


图1-8 AppWizard的第4步,从中选择应用程序的其他功能

读者想让自己的应用程序具有哪些特性呢?第4步中的问题是与这些有联系的。在缺省的

情况下,Docking toolbar(可停靠工具栏)、Initial status bar(初始状态栏)、Printing and print preview(打印和打印预览)和3D controls(使用三维控件)等设置都处于打开状态。

MFC应用程序通常有一个与主窗口相连接的工具栏。AppWizard提供了一组缺省位图,读者可以往其中加入自己的按钮。如果Docking toolbar被选中,那么应用程序就得到它的支持。这时工具栏既可以在窗口的顶部也可以像子窗口一样移动。在应用程序主窗口的上、下、左、右各个位置上停靠的工具栏均可被定位。

Initial status bar选项可以在应用程序主窗口的底部打开一个状态栏。当鼠标在界面上移动时,应用程序用状态栏给用户提供帮助提示。例如,在出现一个串表时(用在第1步选择的语言编写的),就可以在代码中使用这个状态栏在应用程序运行时为用户提供信息。如果最初不使用状态栏(在应用程序启动时),那么就不必选中这个复选框。

选择Printing and print preview为应用程序增加了支持标准打印和支持在屏幕上进行打印预览的程序代码。标准打印意味着生成一些类,这些类帮助代码支持打印而不需要编写打印程序。同时它也提供了支持打印预览(每次1页或2页的打印预览、预览的缩放等等)的功能。然而,打印预览代码是非常基本的。打印预览的缺省代码没有提供页眉、页脚或特殊格式化的打印功能。为了使预览效果更好,大部分的应用程序都可能需要修改这些代码。

如果选择了Context-sensitive Help(上下文相关帮助),那么AppWizard将生成一系列RTF(Rich Text Format)文件以及Windows帮助编译程序使用的支持文件。同时AppWizard还为生成的应用程序菜单栏增加帮助菜单。RTF文件包含应用程序的信息占位。为了能够使用帮助信息,还需要对该RTF文件进行修改。虽然可以用任何一个编辑器来编辑RTF文件,但最好使用一个能理解RTF文件的程序(如Microsoft Word)来修改帮助文件。在缺省状态下不选择上下文相关帮助。MfcApp不使用联机帮助,所以不选择这个选项。

最后,如果3D Controls复选框被选中,那么在应用程序界面上将有一个“倾斜的”外观式样。

另外一个区域与最近使用文件的列表有关,MFC用列表维护应用程序。缺省值是4。MFC为应用程序维护一个INI文件,这个文件包含它最近用到文件的列表清单。这个列表清单将显示在应用程序File菜单的底部。

Advanced按钮产生另外一个对话框,这个对话框包含两个选项卡。一个选项卡用于查看应用程序窗口风格的选项。图1-9显示了另一个选项卡。

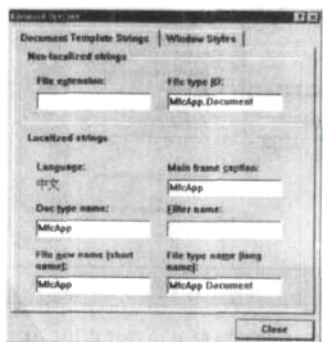


图1-9 Advanced Options对话框的Document Template Strings选项卡