

万水计算机技术实用大全系列

Windows 98 编程实用大全

[美] Richard C. Leinecker, Tom Archer 著

齐舒创作室 译

毅 弘 审校

中国水利水电出版社

内 容 简 介

本书主要介绍 Windows 98 编程技术。本书共分成五个部分,分别介绍了基本原理、程序结构、MFC 编程高级话题、数据库编程、扩展应用程序、ActiveX 编程、Internet 编程、使用 Visual Studio 工具等内容。书中丰富的例子以及翔实的操作步骤对读者的学习及知识掌握提供了很大的灵活性。本书结构清晰、内容新颖、由浅入深,可作为编写 Windows 98 应用程序的参考手册和开发工具。

“Copyright © 1998 by China WaterPower Press. Original English language edition copyright © 1998 by IDG Books Worldwide, Inc. All right reserved including the right of reproduction in whole or in part in any form. This edition published by arrangement with the original publisher, IDG Books Worldwide, Inc., Foster City, California, USA.”
“Windows is a trademark of Microsoft Corporation. - Studio Secrets, and the IDG Books Worldwide logos are trademarks under exclusive license to IDG Books Worldwide, Inc., from International Data Group, Inc. Used by permission.”

北京市版权局著作权合同登记号:图字 01-98-1300

图书在版编目(CIP)数据

Windows 98 编程实用大全/(美)莱内克(Leinecker, R. C.), (美)阿彻(Archer, T.)著;齐舒创作室译.-北京:中国水利水电出版社,1998.12
(万水计算机技术实用大全系列)

书名原文:Windows 98 Programming Bible

ISBN 7-80124-899-6

I. W… II. ①莱… ②阿… ③齐… III. 窗口软件, Windows 98
IV. TP316

中国版本图书馆 CIP 数据核字(98)第 29611 号

书 名	Windows 98 编程实用大全
作 者	[美]Richard C. Leinecker, Tom Archer
译 者	齐舒创作室
审 校	毅弘
出版、发行	中国水利水电出版社(北京市三里河路 6 号 100044) 网址:www.waterpub.com.cn E-mail:sale@waterpub.com.cn 电话:(010)63202266(总机)、68331835(发行部)
经 售	全国各地新华书店
排 版	北京万水电子信息有限公司
印 刷	北京天竺颖华印刷厂
规 格	787×1092 毫米 16 开本 54.75 印张 1231 千字
版 次	1999 年 1 月第一版 1999 年 1 月北京第一次印刷
印 数	0001—5000 册
定 价	95.00 元(含光盘)

凡购买我社图书,如有缺页、倒页、脱页者,本社发行部负责调换
版权所有·侵权必究

译 者 序

Windows 98 是 Microsoft 自 Windows 95 以来所开创的令人称奇的作品延续。Windows 98 的抢占多任务的特性是当前 windows 软件的高品质鲁棒性。Windows 98 的 32 位特性使得应用程序运行的更快更好。

本书将带领用户从诸如菜单和鼠标输入等基础内容进入到 Internet 编程及数据库连接等高级话题。由于强大开发工具的出现,编程技术的更新,使得编写类似于 Windows 98 这样的图形用户界面不再是不可能的事情。用户也可以非常容易地创建出像菜单栏、工具栏、按钮、对话框、窗口等通用的图形元素。可以充分体验编程的乐趣,将自己的研究成果以专业的水准提供给别人。本书还涉及了 MFC 编程高级话题、数据库编程、扩展应用程序、ActiveX 编程和 Internet 编程等高级编程部分,从而使读者也能掌握 Window 98 高级编程技巧。

全书内容涉及相当全面,既大致介绍了 Windows 98 操作系统,又对应用程序编程的技术进行了概括性的叙述。本书在涉及到具体的编程工作时,利用大量的例子和详实的操作步骤来说明解决问题的方法。本书的侧重点是理论与实践相结合,循序渐进,由浅入深地给出了大量的实例,从而使读者达到学以致用效果。本书取材新颖,内容丰富,是一本不可多得的学习 Windows 98 编程的好书。

本书是齐舒创作室集体劳动的结晶,参加本书翻译的有郭志鹏、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、张梦天、程浩、高雪松、余一鸣、丁易伟、汪平、柳飞驰、林旭、刘文燕、扬仲贤、陆春生、肖君、韩志学、张露、徐子望、李铁山、刘健、赵志刚、周理明、汪晓梅、杨立芳、李安、郝连军、王卫林、黄琼、户东凯、田壮、张为新、刘凯、苏明、卢宏芳、乔然、樊勇、叶子奇、项东、卢建山、刘京生和蒋才国等。

因译者水平有限,错误之处在所难免,敬请读者和同行批评指正。

译 者
1998 年 10 月

前 言

能够写一本关于 Windows 98 应用程序开发这样一个可畏的主题的书,可以说是一种特权。Windows 98 是 Microsoft 自 Windows 95 以来所开创的令人称奇的作品延续。消费者的 PC 机现在已能执行抢占多任务;这种特性有利于当前 Windows 软件的高品质与健壮性。

第二个优越之处是 Windows 98 是一个 32 位的操作系统,软件开发人员不必担心分段的内存结构及由此所带来的麻烦,32 位的应用程序将会工作得更快更好。

本书讲了些什么

本书将教会用户如何编写绝妙的 Windows 98 应用程序,它将带领用户从诸如菜单和鼠标输入等基础内容进入到 Internet 编程及数据库连接等高级话题。每一章都讲述一个主题,教会用户如何使用所讲述的技术,而且随后提供至少一个示例程序。

本书还附带一张 CD-ROM,用户将会发现它物有所值。对于书中所讲述的技术,用户不需输入任何内容进行尝试,只需打开 CD-ROM 上的文件,查看代码,并进行任何所需的更改即可;同时该盘上还包含了许多第三方提供的演示程序和库。用户还有机会尝试可用于 Windows 98 程序开发的一些最佳的附件。

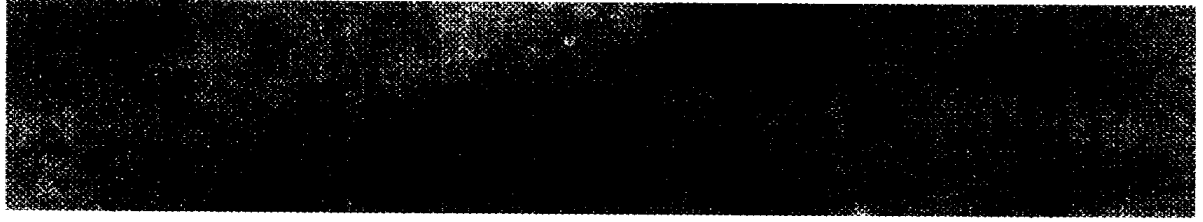
本书适合于什么人

本书适合于任何想学习 Windows 98 编程的读者。如果您是使用 Visual C++ 的新手甚至是 Windows 编程的新手,那么您可以立刻学习到 Windows 编程中基本和主要的基础内容。

一些程序员,甚至是一些经验丰富的老手,想要拥有覆盖各个方面的关于 Windows 98 编程综合手册。用户可能不会找到其他像本书所覆盖的主题范围这样广的编程书籍了。本书在 Internet 编程及数据库连接等有价值的主题上花费了相当的功夫——而这些课题往往是其他书籍所回避或以非常肤浅的方式来处理的内容。

其他读者可能会找到一些他们需要帮助的指定章节。例如,用户可能在工作中负责编写一个与 FTP 服务器之间进行发送和接收文件工作的程序,如果用户不明白如何做这项工作,只需翻至第二十九章并学习相关内容。用户无需为了这个课题花费数天时间在手册中查找,或是查阅联机帮助,或是在 Internet 上搜索。本书将在一个小时内教会用户如何对 FTP 服务器进行上述操作。

简而言之,几乎所有人都可以从本书中受益,这是一本关于如何编写 Windows 98 应用程序的综合指导书,相信您是不会错过它的。



本部分要点：

本书前十一章将教会用户 Windows 98 编程的基本知识。学习这些基本章节后,用户将能轻松地编写 Windows 程序。

第一部分的主体范围覆盖了从菜单和鼠标及键盘输入到使用 GDI 及位图在屏幕上绘图。用户将学习如何创建和使用对话框以及如何通过捕捉异常来处理错误。

前十一章为本书剩余部分将要介绍的高级话题提供了一个坚实的基础。

第一章 入门

本章要点:

- 讲述 Windows 98 中有什么新特性
- 测试 Visual Studio 6 有什么新特性
- 探讨 Visual C++ 程序
- 查看 Visual J++ 程序
- 研究 Visual Basic 程序

许多动机演讲者都曾说过“只要头脑中能构思和相信的事物,就一定能实现。”若对 Windows 98 编程来套用这句话,则实现用户的梦想将比用户所想的要简单。因为 Windows 98 是一个令人称奇的操作系统,它给开发者提供力量,而且为用户提供了很多绝妙的特性和强大的生命力。

把 Microsoft 的 Visual Studio 6 增加到 Windows 98 中,用户可以一举两得。这种可能性是无限的,我仅仅是想想它们就觉得兴奋。

1.1 Windows 98 中有什么新特性

Windows 98 不像 Windows 95 一样仅仅是一个优于 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 备份的需求取消了。

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 用其执行抢占多任务的功能来管理这些线程。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。现在是由 32 位索引进行控制。用户不用再担心磁盘文件簇的大小或是将批处理文件限制在 8K 到 32K 的磁盘空间上。除了节省小文件所占用的磁盘空间外,增强的文件系统使得长文件名更易于管理。在渡过了多年的文件名受 8.3 命名约定的限制后,这才是一个真正的改进。

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。

该系统将更加稳定更加强壮。我想起曾多次与朋友间的谈话,我告诉他们相对 Windows 3.1 而言我是多么喜欢 Windows 95,他们却告诉我他们希望使用 Windows 3.1,因为

Windows 95 经常出问题(Windows 95 中的抢占多任务导致了 Windows 95 的许多问题)。

该系统在检测硬件及管理系统资源上更为出色,这在用户安装新硬件时非常有用。

也许 Windows 98 最具有争议的新特性是与 Internet Explorer 的集成。这使得 Internet 成为用户计算机的延伸,而不是网络空间中计算机的抽象集合。

1.2 对用户而言这意味着什么

对用户而言,这其中的某些特性并不意味着什么。这并不是说他们不能从这个优点众多的操作系统中受益。对大多数用户而言,他们不会注意到新系统不像 Windows 95 那样容易出问题。谁又会在一切都平稳进行的过程时注意到这些呢?只有在事物进展不顺利时,人们才会注意到这一点。

增强的检测和管理资源的能力对用户而言大为有益。但是需再次指出,用户可能不会注意到 Windows 98 正像他们设想的那样工作,而 Windows 95 更多地是即插(插插头接通电源)即祈祷,而不是即插即用。

用户可能真正会注意到的一件事是 Windows 98 和 Internet Explorer 的集成。司法部一定已经注意到了这点,而且已经在这个问题上给 Microsoft 带来了很多人头痛的事。抛开法律上的问题,我认为大多数用户都会转向使用这个集成的操作系统,因为它具有很多优点。

1.3 对开发人员而言这意味着什么

Windows 98 像 Windows 95 一样是送给开发人员的一份厚礼,这主要是因为 Windows 98 给开发人员提供了利用占先的多任务的机会。用户不用象计时器调度的一部分工作那样去执行针对处理器的函数,而只需启动一个线程后就让其自己运行。操作系统只考虑如何在过程之间分配时间,以避免任何一个过程“控制”操作系统。

Internet Explorer 和 Windows 98 的集成对开发者而言是另一件好事。我们中那些在应用程序中使用了 Internet Browser 控件的用户总是要确保客户机上安装了 Internet Explorer。如果没有安装,用户必须安装它,因为所需的 DLL 不在系统目录中,这会导致 Browser 控件的失败(顺便说一下,Microsoft 不允许对 DLL 进行重新发布,而只允许对整个 Internet Explorer 进行发布)。

Internet Explorer 控件甚至被一些 Microsoft Foundation Classes (MFC)所包含。现在使用 Browser 控件比以往任何时候都简单。用户可以在第三十一章查阅更多的关于新的 CHtmlView 类的内容。

1.4 Visual Studio 6 有何新特性

Visual Studio 6 在某些方面领先于上一版本,而在其他方面与之大体相同。Visual C++ 像以往一样存在于相同的集成开发环境(IDE)中。实际上,在花费了大量时间学习 Visual C

++ IDE 后,我很欣慰它保持了原样。Visual C++ 的 MFC 库也没有太多的新特性。CHtmlView 类是好东西,还有一些新控件。当然,如果我真的想要的话,我想我可以列出一份很长的新特性清单,但是大多数都不太重要。

Visual Basic 像 Visual C++ 一样,也使用与旧版本类似的 IDE。Visual Basic 具有几个新特性,且更具强健的生命力。不过除非用户对诸如 DHTML 或 IDE 之类的扩展感到兴奋,否则它就没有多少新内容。

然而,Visual J++ 却相当不同,我必须花费大量的时间去学习这个新环境,Microsoft 在努力使所有的 Visual Studio 工具都存在于相同的 IDE 中。现在 Visual J++ 所在的 IDE 是各种集成环境发展的共同方向,但是在 Visual Studio 版本 6 发布之前没有足够的时间将其硬塞入 Visual C++ 和 Visual Basic 中。Visual C++ 开发组希望到下一个版本(代码命名为“Ranier”)发布前,所有 Visual Studio 工具都能使用相同的 IDE。希望如此!

Visual InterDev 位于 Visual J++ 所处的相同的 IDE 中。我只在工作时 DBA(数据库管理员)皱着眉严厉地告诉我“你可以自己利用 Visual InterDev 来完成工作”时,才使用这个工具。这时,我会极不情愿进入 Visual InterDev 的领地。从我所了解到的方面来看,新版本使该任务更为简单。

在本章的剩余内容中,我将通过创建一些应用程序,一个控件和一个小应用程序来带领用户在 Visual Studio 6 中漫游。在每一个项目中我都会谈论所使用的工具并注明这些工具的要点。这种浏览将为用户学习本书的其余部分作准备。它使我们走进 Windows 98 的 Visual Studio 6 编程的实质。

1.5 Visual C++ HelloWorld1 程序

为这一章创建的第一个程序是传统的所有 C 和 C++ 程序开发人员刚开始编程所接触的“HelloWorld”程序的变体。该程序是由 MFC AppWizard 创建的。在整个 AppWizard 的进程中对缺省设置所作的唯一修改是使该应用程序成为一个单文档用户界面的程序。版本 5 中的大多数项目类型,例如 AppWizard、ATL Wizard 和 MFC DLL,都被包括在 Visual C++ 6 中。图 1-1 显示出了可用的选择。

Visual C++ 6 中的项目类型基本与 Visual C++ 5 中的一样。需要着重指出的是用户现在可以用 AppWizard 创建不含文档/视图结构的程序。对我而言,这是一个非常有用的选项。我几乎不能数出有多少次我创建了文档/视图结构的程序,却仅仅是用来显示图形和图像,而完全忽略文档所有的特性。每当这个时候,我都察看项目中的文件,并希望删除文档的源代码文件(还好我没有这么做,否则项目将无法编译通过)。

我第一次使用 Visual C++ 6,并试着向视图类中的 OnDraw() 函数中添加代码时,就被 IntelliSense 特性震惊了。它接管了我的工作并使其变得简单。IntelliSense 是一个系统,编辑器在其中根据用户键入的上下文为用户列出了函数和参数的清单。例如,如果我输入 TextO,则显示包括 TextOut 的列表。如果我按下 Enter (或 Tab) 键,则将完成该函数名,这样我需要输入的内容就少了。IntelliSense 还给用户提供该函数所需参数的清单。这大大减少

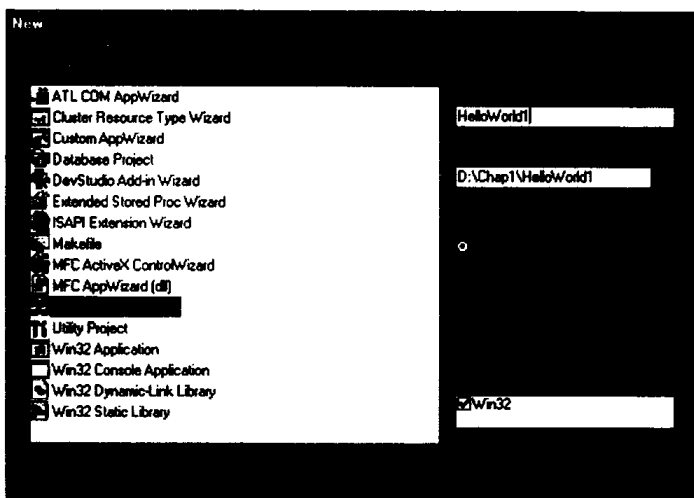


图 1-1 Visual C++ 版本 6 中的项目类型几乎与 Visual C++ 版本 5 中的一致

了用户在联机帮助中进行查找的次数。图 1-2 显示了在我创建 HelloWorld1 程序时 IntelliSense 特性所显示的样子。

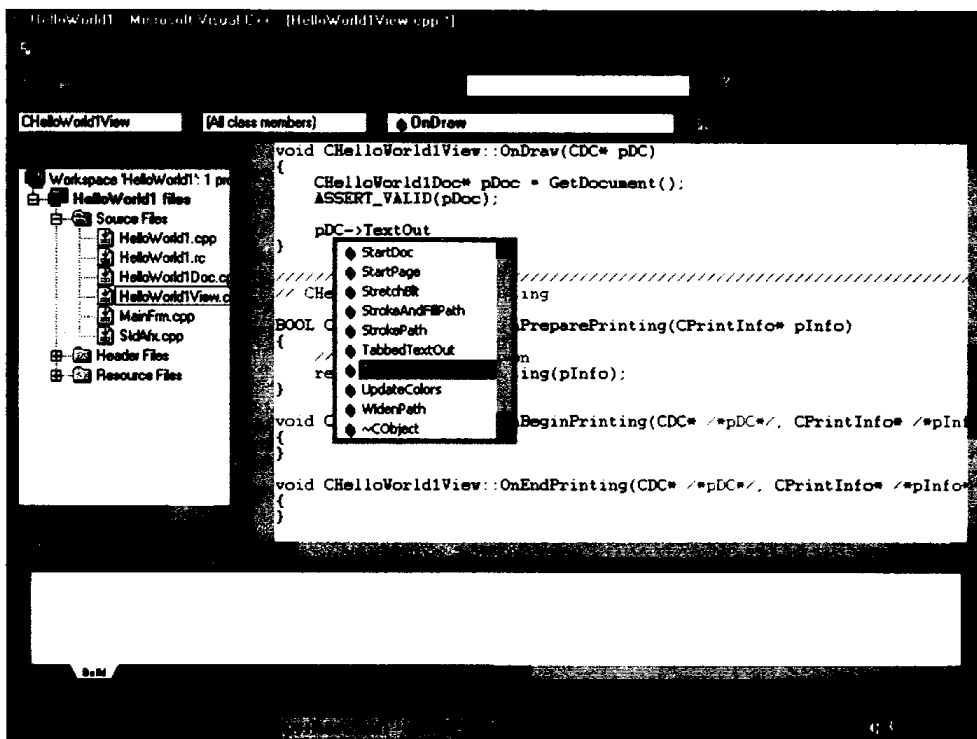


图 1-2 IntelliSense 和语句完成功能使得编辑文本变得容易，但更重要的是它帮助用户记住类成员函数

AppWizard 的另一个新特性是创建 Explorer 样式程序的能力,这种程序有一个树状视图控件来显示对象。图 1-3 显示了使用 Explorer 样式的程序。

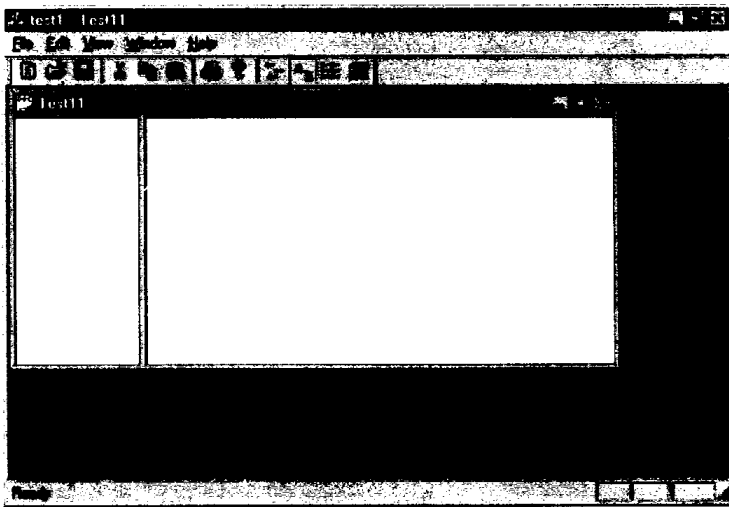


图 1-3 用户可容易地创建 Explorer 样式的应用程序

程序清单 1-1 显示了 HelloWorld1 程序的视图类的重要部分。



HelloWorld1

CD-ROM 位置: **Chap1\HelloWorld1**

程序名称: **HelloWorld 1.exe**

文本形式的源代码模块: **HelloWorld1View.cpp**

```
// HelloWorld1View.cpp : implementation of the
// CHelloWorld1View class
//

#include "stdafx.h"
#include "HelloWorld1.h"
#include "HelloWorld1Doc.h"
#include "HelloWorld1View.h"
////////////////////////////////////
// CHelloWorld1View

IMPLEMENT_DYNCREATE(CHelloWorld1View, CView)
BEGIN_MESSAGE_MAP(CHelloWorld1View, CView)
    //{{AFX_MSG_MAP(CHelloWorld1View)
    //}}AFX_MSG_MAP
    // Standard printing commands
    ON_COMMAND(ID_FILE_PRINT, CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_DIRECT, CView::OnFilePrint)
    ON_COMMAND(ID_FILE_PRINT_PREVIEW, CView::OnFilePrintPreview)
END_MESSAGE_MAP()
```

```
////////////////////////////////////  
// CHelloWorld1View construction/destruction  
  
CHelloWorld1View::CHelloWorld1View()  
{  
}  
  
CHelloWorld1View::~~CHelloWorld1View()  
{  
}  
  
BOOL CHelloWorld1View::PreCreateWindow(CREATESTRUCT& cs)  
{  
    return CView::PreCreateWindow(cs);  
}  
  
////////////////////////////////////  
// CHelloWorld1View drawing  
  
void CHelloWorld1View::OnDraw(CDC * pDC)  
{  
    CHelloWorld1Doc * pDoc = GetDocument();  
    ASSERT_VALID(pDoc);  
  
    pDC->TextOut( 20,20, "Hello Visual C++ World!" );  
}  
  
////////////////////////////////////  
// CHelloWorld1View printing  
  
BOOL CHelloWorld1View::OnPreparePrinting(  
    CPrintInfo * pInfo)  
{  
    // default preparation  
    return DoPreparePrinting(pInfo);  
}  
  
void CHelloWorld1View::OnBeginPrinting(CDC * /* pDC */,  
    CPrintInfo * /* pInfo */)  
{  
}  
  
void CHelloWorld1View::OnEndPrinting(CDC * /* pDC */,  
    CPrintInfo * /* pInfo */)  
{  
}
```

当 HelloWorld1 程序运行时,用户会看到熟悉的 SDI 应用程序。文本串“Hello Visual C++ World!”在视图窗口内显示,如图 1-4 所示。

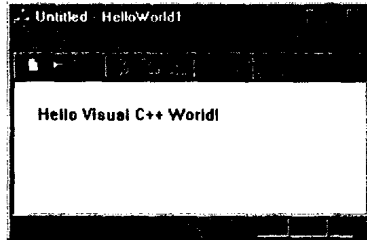


图 1-4 AppWizard 创建用户所熟悉的 SDI 应用程序

1.6 Visual C++ Win32 控制台应用程序

我原来一直认为 Win32 控制台应用程序没有自己的一席之地,直到发生了下面两件事才改变了我的看法。第一件事是,我在 Rockingham Community College 开始教授 C++ 时,通过创建一个基于 AppWizard 的 Windows 应用程序来开始我的教学,但仅仅 15 分钟后,我就意识到大家是多么迷惑不解,于是我迅速学习了 Win32 控制台应用程序。通过这种程序我能够在教授 C++ 概念时不使学生们在神秘的项目和 MFC 类中迷失;第二件事是,在使用 Win32 控制台应用程序时,我发现我能快速和方便地使执行实用程序函数的程序运行。

一旦有了这些闪光的发现,我就开始意识到,Win32 控制台应用程序对那些不需要用户界面的实用工具而言是多么便利。几乎不费什么力气,我就能创建能执行有用任务的程序,如移动和重命名文件,将二进制文件转换为源代码,以及其他像我一样的软件开发人员所需的工具程序。我不怀念过去能在五分钟之内建立起自动执行某些编程任务的 DOS 应用程序的那些日子。甚至(通常)不用担心分段指针或内存耗尽等问题。

第二个演示程序是一个 Win32 控制台应用程序,它显示信息“Hello Visual C++ Console World!”。当用户创建 Win32 控制台应用程序时,不论用户相信与否,其中一个选择就是一个“Hello World”应用程序(参见程序清单 1-2)。



演示程序

HelloWorld2

CD-ROM 位置: Chap1\HelloWorld2

程序名称: HelloWorld2.exe

文本形式的源代码模块: HelloWorld2.cpp

```
// HelloWorld2.cpp : Defines the entry point
// for the console application.
//
#include "stdafx.h"
```

```
int main(int argc, char * argv[])
{
    printf("Hello Visual C++ Console World! \n");
    return 0;
}
```

当该程序运行时,用户应能看见一个控制台窗口,在该控制台窗口中,将显示信息“Hello Visual C++ Console Application World!”,如图 1-5 所示。

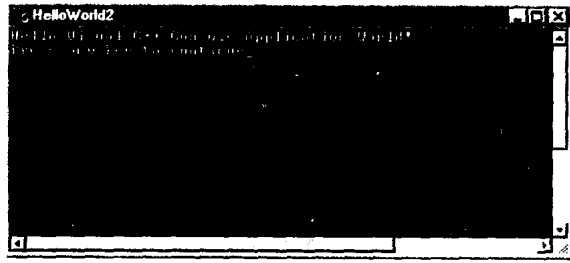


图 1-5 用户可能会发现自己也需要 Win32 控制台应用程序

1.7 HelloWorld3 ATL 控件

如果用户尚未接触过 ActiveX 控件或未使用过 Active Template Library (ATL),那么用户需要创建 ActiveX 控件的日子也为时不远了,尤其是这个大方向是在 Microsoft 的推动之下。在大多数情况下,使用它的最好的方法是利用 Active Template Library Wizard。它创建 ATL 项目所需的启动代码,并将节省用户亲自编写所要花费的时间和精力(参见程序清单 1-3)。

除了用户可以很容易地创建 ATL 控件外,它所需的空间也是相对较少的。像 HelloWorld3 控件那样的,添加了最小代码的完整控件大约占用 37K 空间。ATL 控件有许多选项,其中之一是可以使用 MFC。如果用户想使用 MFC,则需要预备增加另外 80K 到 100K 的空间。



HelloWorld3

CD-ROM 位置: Chap1\HelloWorld3

演示程序

程序名称: HelloWorld3.dll

文本形式的源代码模块: Full.h

```
// Full.h : Declaration of the CFull
```

```
#ifndef __FULL_H__
```

```
#define __FULL_H__
```

```

# include "resource.h" // main symbols
# include <atlctl.h>

////////////////////////////////////

// CFull
class ATL_NO_VTABLE CFull :
public CComObjectRootEx < CComSingleThreadModel > ,
public IDispatchImpl < IFull, &IID_ IFull,
    &LIBID_ HELLOWORLD3Lib > ,
public CComControl < CFull > ,
public IPersistStreamInitImpl < CFull > ,
public IOleControlImpl < CFull > ,
public IOleObjectImpl < CFull > ,
public IOleInPlaceActiveObjectImpl < CFull > ,
public IViewObjectExImpl < CFull > ,
public IOleInPlaceObjectWindowlessImpl < CFull > ,
public IPersistStorageImpl < CFull > ,
public ISpecifyPropertyPagesImpl < CFull > ,
public IQuickActivateImpl < CFull > ,
public IDataObjectImpl < CFull > ,
public IProvideClassInfo2Impl < &CLSID_ Full, NULL,
    &LIBID_ HELLOWORLD3Lib > ,
public CComCoClass < CFull, &CLSID_ Full >
{
public:
    CFull()
    {
    }
};

DECLARE_REGISTRY_RESOURCEID(IDR_ FULL)

BEGIN_ COM_ MAP(CFull)
    COM_ INTERFACE_ ENTRY(IFull)
    COM_ INTERFACE_ ENTRY(IDispatch)
    COM_ INTERFACE_ ENTRY(IViewObjectEx)
    COM_ INTERFACE_ ENTRY(IViewObject2)
    COM_ INTERFACE_ ENTRY(IViewObject)
    COM_ INTERFACE_ ENTRY(IOleInPlaceObjectWindowless)
    COM_ INTERFACE_ ENTRY(IOleInPlaceObject)
    COM_ INTERFACE_ ENTRY2(IOleWindow,
        IOleInPlaceObjectWindowless)
    COM_ INTERFACE_ ENTRY(IOleInPlaceActiveObject)
    COM_ INTERFACE_ ENTRY(IOleControl)
    COM_ INTERFACE_ ENTRY(IOleObject)
    COM_ INTERFACE_ ENTRY(IPersistStreamInit)
    COM_ INTERFACE_ ENTRY2(IPersist, IPersistStreamInit)
    COM_ INTERFACE_ ENTRY(ISpecifyPropertyPages)
    COM_ INTERFACE_ ENTRY(IQuickActivate)

```

```

COM_INTERFACE_ENTRY(IPersistStorage)
COM_INTERFACE_ENTRY(IDataObject)
COM_INTERFACE_ENTRY(IProvideClassInfo)
COM_INTERFACE_ENTRY(IProvideClassInfo2)
END_COM_MAP()

BEGIN_PROP_MAP(CFull)
    PROP_DATA_ENTRY("_ cx", m_sizeExtent.cx, VT_UI4)
    PROP_DATA_ENTRY("_ cy", m_sizeExtent.cy, VT_UI4)
    // Example entries
    // PROP_ENTRY("Property Description", dispid, clsid)
    // PROP_PAGE(CLSID_StockColorPage)
END_PROP_MAP()

BEGIN_MSG_MAP(CFull)
    CHAIN_MSG_MAP(CComControl < CFull > )
    DEFAULT_REFLECTION_HANDLER()
END_MSG_MAP()

// Handler prototypes:
// HRESULT MessageHandler(UINT uMsg, WPARAM wParam,
// LPARAM lParam, BOOL& bHandled);
// HRESULT CommandHandler(WORD wNotifyCode, WORD wID,
// HWND hWndCtl, BOOL& bHandled);
// HRESULT NotifyHandler(int idCtrl, LPNMHDR pnmh,
// BOOL& bHandled);

// IViewObjectEx
DECLARE_VIEW_STATUS(VIEWSTATUS_SOLIDBKGND | VIEWSTATUS_OPAQUE)

// IFull
public:

    HRESULT OnDraw(ATL_DRAWINFO& di)
    {
        RECT& rc = *(RECT*)di.prcBounds;
        Rectangle(di.hdcDraw, rc.left, rc.top,
            rc.right, rc.bottom);

        SetTextAlign(di.hdcDraw, TA_CENTER | TA_BASELINE);
        LPCTSTR pszText = _T("Hello Visual C++ ATL World!");
        TextOut(di.hdcDraw,
            (rc.left + rc.right) / 2,
            (rc.top + rc.bottom) / 2,
            pszText,
            lstrlen(pszText));
        return S_OK;
    }
};

```

```
# endif // __FULL_H_
```

用户必须意识到的关于该项目的第一个问题是用户还不能执行它。用户需要某些容器来装载它。在此处,容器是指某些能加载和执行控件的程序。在把这些控件加载至容器中前,用户需要注册控件。要注册这些控件,以便应用程序能使用它,就要用注册的控件为参数运行 Regsvr32(在 Windows \ System 目录下)——例如,C:\Windows\System\Regsvr32 HelloWorld3.dll。

如果用户编译该项目,Visual C++ 将为用户注册该控件。这使得用户在开发控件时易于对它们进行测试。一旦控件被注册,一般总是假设它位于被注册的那个目录中。如果用户将其移至其他目录,用户必须对其进行注册。

Visual C++ 向用户提供了可以加载控件的容器程序。这节省了用户只是为测试该控件而编写自己的容器对象所花的时间。要运行测试容器程序,只需从 Tools 菜单中选择 ActiveX Control Test Container。图 1-6 显示了 HelloWorld3 控件加载到测试容器中。

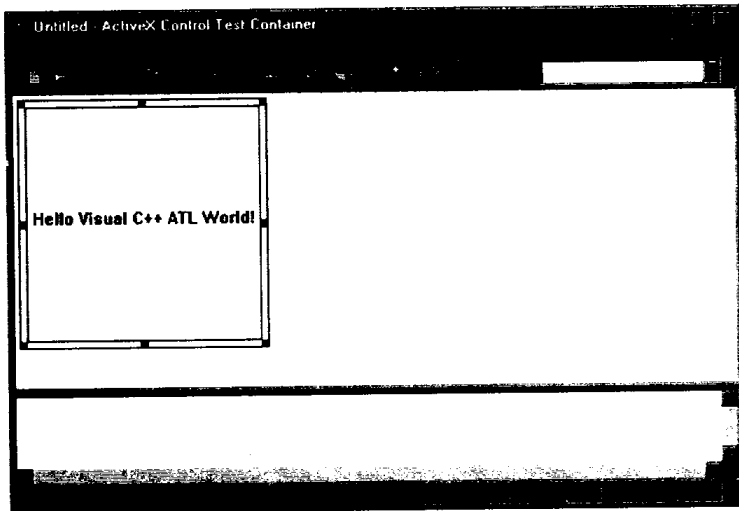


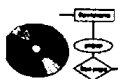
图 1-6 HelloWorld3 控件必须被加载至一个控件容器中。这里它被加载至 Visual C++ 所附带的 ActiveX Control Test Container 工具中

1.8 HelloWorld4 Java 小应用程序

一直以来,每个人都创建了几个 Java 小应用程序和/或应用程序。某些人继续加强掌握编程语言并把 J++ 作为一种开发工具加以广泛使用。我最初接触 Java 是因为它易于使用图像,而且我乐于为我的 Web 站点增添一些趣味。最终,我学习了 Java 的其他方面内容,并继承了我在工作中获得的几个来自商业的小应用程序(这是我在开发 Java 过程中得到的多余的收获)。

抛开一切无关紧要的,用户需要了解一点 Java 以跟上当今的发展需要。而 Visual Studio 向用户提供了 J++, 一个强大的、易于使用的 Java 开发工具。

程序清单 1-4 所列出的 HelloWorld4 小应用程序是在大约两分钟之内写完的。Java 相当易学,并且 J++ 创建了足够的启动代码以避免用户自找麻烦。



HelloWorld4

CD - ROM 位置: Chap1\HelloWorld4

演示程序

程序名称: HelloWorld4.class

文本形式的源代码模块: HelloWorld4.java

```
// HelloWorld4.java

import java.awt.*;
import java.applet.*;

/* *
 * This class reads PARAM tags from its HTML host page
 * and sets the color and label properties of the applet.
 * Program execution begins with the init() method.
 * /
public class HelloWorld4 extends Applet
{
    /* *
     * The entry point for the applet.
     * /
    public void init()
    {
        initForm();

        usePageParams();
    }

    private final String labelParam = "label";
    private final String backgroundParam = "background";
    private final String foregroundParam = "foreground";

    /* *
     * Reads parameters from the applet's HTML host and
     * sets applet properties.
     * /
    private void usePageParams()
    {
        final String defaultLabel = "Default label";
        final String defaultBackground = "COCOCO";
        final String defaultForeground = "000000";
        String labelValue;
```