

Visual C++ 6: In Record Time

Visual C++ 6

轻松进阶

In Record Time

[美] Steven Holzner 著
王岚波 等译

轻松掌握，迅速提高

以实例为基础

以应用为目标



电子工业出版社

Publishing House of Electronics Industry

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

TP212
HEZ/1

Visual C++ 6: In Record Time

Visual C++ 6轻松进阶

〔美〕 S. 霍尔兹纳 著
Steven Holzner

王岚波 等译



电子工业出版社

Publishing House of Electronics Industry

0050049

内 容 提 要

Visual C++6是一个功能极其强大的工具，是许多工具的集合，这些工具有机地组合在一起，形成一个动态的即时可用的软件包。本书将介绍这些工具，并且使用它们开发专业级Windows程序，涉及的范围从显示图形到使用菜单，从使用按钮和正文框到处理文件，从Internet接入到建立自己的ActiveX控件。除此之外，本书还讨论了如何建立功能丰富的Web浏览器，这些功能包括链接、图形等等，还介绍如何开发具有剪切和粘贴、打印、存储文件、字对齐和其它功能的字处理器。

本书适合于准备学习C++编程的编程人员和计算机专业的师生。



Copyright©1998 SYBEX Inc., 1151 Marina Village Parkway, Alameda, CA 94501. World rights reserved. No part of this publication may be stored in a retrieval system, transmitted, or reproduced in any way, including but not limited to photocopy, photograph, magnetic or other record, without the prior agreement and written permission of the publisher.

本书英文版由美国SYBEX公司出版，SYBEX公司已将中文版独家版权授予中国电子工业出版社及北京美迪亚电子信息有限公司。未经许可，不得以任何形式和手段复制或抄袭本书内容。

书 名: Visual C++ 6轻松进阶

著 者: [美] Steven Holzner

译 者: 王岚波 等

责任编辑: 林晶

印 刷 者: 北京天竺颖华印刷厂

装 订 者: 三河金马印装有限公司

出版发行: 电子工业出版社出版、发行

北京市海淀区万寿路173信箱 邮编: 100036 发行部电话: 68279077

北京市海淀区万寿路甲15号南小楼一层 邮编: 100036 发行部电话: 68215345

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

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 28.5 字数: 750千字

版 次: 1999年2月第1版 1999年2月第1次印刷

书 号: ISBN 7-5053-4848-5/TP·2369

定 价: 48.00 元

著作权合同登记号 图字: 01-98-2472

凡购买电子工业出版社的图书，如有缺页、倒页、脱页者，本社发行部负责调换

版权所有·翻版必究

JS256/26
22

前言

欢迎学习和使用Visual C++

目前, Visual C++是最受用户青睐的Windows程序设计产品。Windows中的任何功能,都可以在Visual C++中实现。本书将向读者介绍Visual C++编程方面的有关知识。

在过去的几年中, Visual C++已经发展成为一组功能非常强大的Windows编程工具。若干年前, Windows编程可用的唯一产品是Windows软件开发工具(SDK)。而今天, Visual C++为我们提供了功能丰富的编程工具, 包括编辑器、工具、测试容器、类库和调试技术等等。

实际上, Visual C++是一个十分庞大的软件包。有时, 它似乎太大了。就在你认为已经掌握了许多Visual C++编程技术时, 在你前面又呈现出一片全新的景色。你意识到, 需要花费很长时间才能把这些新内容融汇到已经掌握的知识中。当你刚刚完成这个过程时, 又一个新景色出现。你需要再次经历整个学习过程。

本书能够使这个学习过程变得十分容易。我们将以一种清晰、详细的方式介绍如何使用工具、向导、编辑器和Visual C++资源, 还将详细阐述Visual C++的许多方面, 最后驯服它, 使它为我们工作。

我们将学习如何快速建立程序、如何增加对话框、菜单等, 也将学习如何在Windows程序中使用文本框、按钮、复选框、列表框和各种控件。我们甚至还学习如何连接到Internet。现在, 让我们花费一点时间看看本书将要介绍的内容。

本书的内容

本书的目标是介绍Visual C++以及高效使用Visual C++所需的技巧。最有效的学习方法是从阅读实际源代码中学习。许多书是从阐述抽象概念、程序设计结构和理论开始, 而我们认为Visual C++是一组功能强大的工具。程序员使用这些工具编写他们感兴趣的程序, 这些工具本身并不是最终目标。

通常, 程序设计是从程序设计的观点——即以程序设计结构, 诸如循环、条件和继承性等方式介绍。这使得学习过程枯燥无味。在本书中, 我们从一个不同的角度阐述Visual C++, 即从程序员的角度。本书不是由“If语句”、“访问修改符”或“抽象C++类”这样的章组成。相反, 本书中的各章实际上是技巧, 这些技巧是实际使用Visual C++时获得的经验。本书由“文件处理”、“图形和鼠标驱动绘图程序”、“建立复选框和单选钮”等章节组成。

各章的组织形式从简单到复杂, 以便读者在阅读本书时能够从前面的各章中积累经验。在掌握了基本技巧后, 诸如如何从键盘读取击键, 读者将学习如何建立菜单、工具栏按钮和快捷键。在第6章和第7章中, 读者将学习如何设计子菜单、复选框和对话框。第8章将向读者介绍如何使用列表框、组合框和滑动条。此时, 读者已经能够建立功能丰富的鼠标驱动绘

图程序（这个程序能够把文件存储到磁盘）和字处理程序，并且能够把数据库连接到Visual C++程序。在后面的各章中，将向读者介绍如何建立Internet程序，包括Web浏览器，还将介绍如何建立ActiveX控件。最后（也是必不可少的），我们将向读者介绍如何在调试程序时修复错误。

读者也将看到大量短小的例子程序，因为不实际运行程序而学习Visual C++，就像通过阅读飞机部件手册就想学习飞行一样。这些例子短小但切中要点，并且没有过多额外的细节。在建立程序的过程中，使用所有Visual C++向导和编辑器，让Visual C++程序为你工作。

Visual C++软件包具有许多工具，在建立自己的程序时可以充分使用它们。此外，本书还包括“说明”、“提示”和“警告”，以便提供更多的知识和经验。

提示：提示看起来像这样。编写它们的目的是节省读者的时间，或提示可能出现的额外信息。一般情况下，提示可提供一些附加信息。

说明：这是一个说明。我们使用说明提供一些特殊要点。例如，为什么在一种情况下使用鼠标，而在另一种情况下不使用鼠标。在特定时刻，这是十分重要的。

警告：警告看起来像这样。警告用于提示你可能遇到的一些常见的陷阱。

同时，我们按照递增的方式构造例子，因此当增加一行或若干行新代码时，使用箭头指出这些新行：

```
void CWelcomeView::OnDraw(CDC* pDC)
{
→   CString welcome_string = "Welcome to Visual C++";
    CWelcomeDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
→   pDC->TextOut(0, 0, welcome_string);
}
```

此外，本书中的代码（每个程序的完整代码都在特殊代码列表标题下给出）在Sybex的Web站点上给出。每个例子程序具有自己的子目录。这意味着，开发例子程序时，不必键入任何内容。全部代码都在Sybex Web站点上。

对读者的要求

尽管Visual C++是面向对象的语言，但是阅读本书不需任何面向对象的程序设计经验。我们将在第1章介绍面向对象程序设计的工作原理。如果读者具有一些C++编程经验，已经具有优势，因为你具有面向对象程序设计的初步知识。

为了使用本书，读者需要Microsoft Visual C++。Visual C++提供了完成本书中例子所需的全部工具。此外，如果读者要执行面向Internet的程序，应该通过某种方式连接到Internet，比如通过Internet服务提供商（ISP）。这就是读者需要具备的全部知识。

读者能够在Internet上找到有关Visual C++的更多帮助信息，站点地址是[http://www.microsoft.com/visual c](http://www.microsoft.com/visual_c)；在Visual C++提供的联机文档中也有许多帮助信息。

我们已经概述了本书内容。下面让我们开始第1章的学习，编写第一个C++程序。

//

目 录

第1章 欢迎使用Visual C++	1
Visual C++概述	2
第一个例子	2
Visual C++工作区和项目	3
建立源代码文件	4
向程序增加C++代码	6
C++流	6
运行第一个程序	7
C++类和对象	8
对象	8
类	9
第一个C++类和对象例子	10
访问标识符	10
使用构造器初始化类中的数据	12
使用DataClass类	14
一个真正的C++例子: SchoolRoom	17
C++销毁器	18
存储、检索数据以及计算数据的平均值	19
使用AddScore()存储数据	19
使用GetScore()检索数据	20
使用AverageScore()计算数据的平均值	21
在程序中使用SchoolClass	23
继承和替代	27
派生新类: 继承	28
改变方法: 替代	29
C++函数重载	31
本章要点	33
第2章 开始学习Visual C++	34
第一个真正的Windows Visual C++程序	34
Visual C++程序的组成部分	39
应用程序对象	39
主窗口对象	40
视图对象	40
文档对象	40

	显示欢迎消息	40
	应用程序对象剖析	42
	主窗口对象剖析	49
	视图对象剖析	54
	Windows面向事件的程序设计	54
	在视图中显示消息	54
	设备说明表	55
	文档对象剖析	61
	从视图使用文档	61
	把数据存储到磁盘	62
	本章要点	66
第3章	从键盘读取输入	67
	使用键盘	67
	为键盘数据建立存储区	68
	读取击键	68
	在文档中记录字符	70
	显示文本	71
	在窗口中心显示文本	79
	计算窗口的尺寸	81
	计算被显示文本字符串的大小	82
	本章要点	91
第4章	在Visual C++中使用鼠标	92
	向窗口增加光标	92
	利用文本测量术计算文本大小	94
	设定光标位置	96
	当失去或获得焦点时显示和隐藏光标	100
	使用鼠标	108
	使用ClassWizard鼠标方法	109
	在新的鼠标位置绘制文本	112
	本章要点	122
第5章	菜单、工具条按钮及状态条提示	123
	第一个菜单例子	123
	使用菜单编辑器	124
	增加新的菜单项	125
	将菜单选项连接到程序	125
	建立全新菜单的例子	134
	增加快捷键	136
	增加状态条提示	137
	向Demo菜单增加子菜单	137

增加加速键	138
向工具条增加工具	139
使菜单选项变灰	140
选择菜单选项	141
向子菜单选项增加代码	142
本章要点	153
第6章 对话框：使用按钮和文本框	154
建立第一个对话框	154
建立对话框	156
向对话框增加控件	157
标记控件	158
建立对话框类	159
将方法连接到对话框控件	160
将变量连接到对话框控件	162
替代OK按钮	164
显示对话框	165
使用对话框作为主窗口	178
本章要点	187
第7章 建立复选框和单选钮	188
使用复选框	188
向程序增加复选框	190
在对话框编辑器中进行控件对齐	190
将复选框连接到程序	191
使用单选钮	199
将单选钮连接到程序	200
安排单选钮	201
组合使用复选框和单选钮	207
使用组框	208
向复选框控件增加成员变量	210
本章要点	218
第8章 列表框、组合框和滑动条	219
处理列表框	219
使用标签显示文本	221
向列表框赋予成员对象	221
初始化列表框中的数据	223
处理列表框双击	224
确定选择了列表框中的哪个选项	225
使用组合框	231
初始化组合框	231

确定用户所做的选择	234
使用滑动条增加滚动功能	240
初始化滑动条控件	242
处理滑动条事件	243
显示文本框中的数字	244
本章要点	250
第9章 图形和鼠标驱动绘图程序	251
设计绘图程序	252
建立Painter程序的用户友好界面	252
设置Painter标志	253
建立工具菜单和工具条按钮	255
将标志连接到绘图工具	256
在菜单中设置复选标记	257
处理鼠标按下事件	258
绘制线段	258
绘制长方形	260
绘制椭圆和圆	262
使用颜色填充图形	263
使用鼠标自由绘图	265
改变鼠标光标的外观	267
缩放图形	268
二进制光栅运算	269
刷新Painter程序的显示	271
在元文件中镜像图形调用	273
重放元文件	273
保存图形文件	275
打开图形文件	276
建立新文档	277
Painter程序顺利终结	278
本章要点	290
第10章 文 件 处 理	291
Visual C++串行化	291
建立Writer程序	291
串行化StringData对象	293
串行化用户对象	303
向类增加串行化特性	305
普通日常文件的处理	315
将文件分为记录	316
初始化filer程序的数据	317

写文件	318
读文件	320
移动文件指针	321
本章要点	328
第11章 多文档和多视图	329
第一个多文档、多视图例子	330
MDI程序与其它程序的区别	331
两个窗口类型、两个菜单类型	332
在多视图程序中读取击键	333
使用视图提示	335
文档修改标志	335
滚动互不关联的视图	343
使视图变为可滚动	344
给出文档大小	344
调整被滚动的设备说明表	347
使用文本框作为视图	355
访问Editor程序的内部数据	356
本章要点	360
第12章 建立Internet程序: Web浏览器	361
建立Web浏览器	361
使用HTTP访问Internet	368
通过HTTP下载	370
在Internet上使用FTP	378
使用FTP连接到Internet	379
使用Visual C++处理数据库	387
设计db程序	387
访问当前记录	389
本章要点	394
第13章 创建ActiveX控件	395
方格游戏ActiveX控件	395
绘制ActiveX控件	396
向ActiveX控件增加事件处理器	398
测试ActiveX控件	400
在Visual C++程序中使用ActiveX控件	400
创建基于ActiveX控件的按钮	408
定制Buttoner	408
增加ActiveX控件事件	410
增加ActiveX控件属性	411
增加ActiveX控件方法	412

在其它程序中嵌入Buttoner控件	413
将ActiveX控件连接到代码	414
本章要点	427
第14章 调试Visual C++程序	428
Buggy程序	428
设置断点	432
运行到断点	434
单步执行代码	434
在程序运行时检查变量	436
Auto和Locals窗口	436
本章要点	446

第1章 欢迎使用Visual C++

- 初识Visual C++
- 类
- 对象
- Visual C++继承
- 函数替代
- 函数重载
- 对象初始化（构造器）

欢迎使用Visual C++！对于Windows程序设计来说，Visual C++是一个功能极其强大的工具——事实上，许多程序员认为，它是目前功能最强大的Windows程序设计工具。实际上，Visual C++不是一个单一工具，它是许多工具的集合；这些工具有机地组合在一起，形成一个动态的即时可用的软件包。

本书将介绍这些工具，并且使用它们开发专业级Windows程序，涉及的范围从显示图形到使用菜单、从使用按钮和文本框到处理文件、从Internet接入到建立自己的ActiveX控件。除了这些内容之外，我们还讨论如何建立功能丰富的Web浏览器，包括超链接、图形等等；还介绍如何开发具有剪切和粘贴、打印、存储文件、字对齐和其它功能的字处理器。我们还开发了一个功能强大的绘图程序，这个程序具有使用鼠标画图形、填充颜色、存储文件等更多的功能。我们将介绍更多的例子。

实质上，本书介绍三个主要方面的内容：其一是C++程序设计（如何使用类、对象和其它面向对象的程序设计技巧）；其二是Visual C++（主要介绍Microsoft基础类（MFC）中包括的内置资源）；其三是Windows程序设计（如何在Windows程序中挥洒自如地工作：使用窗口、菜单、按钮和其它特性的基本技巧）。在本章中，我们将介绍C++程序设计的基本要素：类、对象、函数替换等等。

为了简单起见，前几个例子程序不创建窗口。相反，这些程序将使用Visual C++内置提供的窗口。这些程序说明如何使用C++，并且向我们展示Windows程序设计所需要的重要知识。在下一章中，我们将使用Visual C++在窗口环境中使用和建立窗口。下面先介绍第1章的内容。首先快速概述一下Visual C++软件包。

说明：如果你已经熟悉C++概念，诸如继承、替代（overriding）、重载（overloading）、类和对象，可以跳过本章到下一章，开始使用书中的程序创建窗口。但是，如果对以上概念不十分了解，请继续阅读本章，因为要理解本书其余部分中的程序代码，必须对这些内容深入掌握。注意，我们假设你已经具有一些程序设计知识，诸如了解什么是变量和函数。如果你没有任何程序设计经验，可能需要花费一些时间理解基本概念，诸如什么是变量和函数等。

Visual C++概述

在Visual C++问世以前,如果你进行过Windows程序设计,大概不会忘记苦涩的经历。编写Windows程序的第一个方法是使用C(而不是C++)开发大型复杂的程序。事实上,在原始Windows程序设计中,即使当时认为是正规的许多编程方法,现在看来也是十分有害的,这包括一个语句(C switch语句)延续许多代码页。五页难懂神秘的代码,只是简单地在屏幕上显示一个空白窗口。

这个时代已经一去不复返了,这是因为C++的问世。C++是Windows程序设计最理想的工具。正如你将要看到的那样,可以把大部分代码放入自包含的C++对象中,这使得较长的程序看起来非常整洁(C++是第一个用于开发大型程序的语言,它使庞大的程序易于管理)。这意味着进行Windows程序设计更加容易,原始的不良程序设计习惯被优秀的程序设计方法所取代(这使得C++程序更易于调试排错)。不再是冗长庞大的程序,现在的Windows程序分为整洁的、易于管理的若干部分。

此外,Microsoft引入了Microsoft基础类(MFC)库。尽管出乎所料,但是,引入MFC就象在Windows中引入C++一样重要。MFC是预先编写的用于创建新窗口的全部代码(这需要很长一段代码),我们可以依赖MFC CWnd代码包,它为我们处理全部细节。使用本章学习的标准C++技术,能够随心所欲地定制MFC软件包,比如CWnd(例如,你可能想改变窗口标题条的文本)。

Visual C++不仅使用MFC,也使Windows程序设计更加容易,这是由于Visual C++引入了许多程序设计工具,诸如用于设计菜单的菜单编辑器,用于设计对话框的对话框编辑器。Visual C++为我们提供了一个集成的设计环境,在这个环境中能够编写和运行程序。除此之外,Visual C++把Windows程序需要的许多文件组织成项目(project),它为我们处理许多工作。事实上,Visual C++向导(Wizard)(一组特殊的程序设计工具)甚至能够为我们编写大部分程序代码。下一章将介绍Visual C++向导。

上面对Visual C++作了一个概述。在具有这些知识后,我们便可以开始学习和使用它。在本章中,我们将使用Visual C++,但不是编写创建它们自己窗口的程序。相反,我们将以尽可能简单的方式使用Visual C++,这样可以避免使用大量无关紧要的细节破坏C++程序设计例子。通过复杂的Visual C++程序学习C++程序设计是极其困难的,因此,我们按阶段介绍Visual C++程序设计。

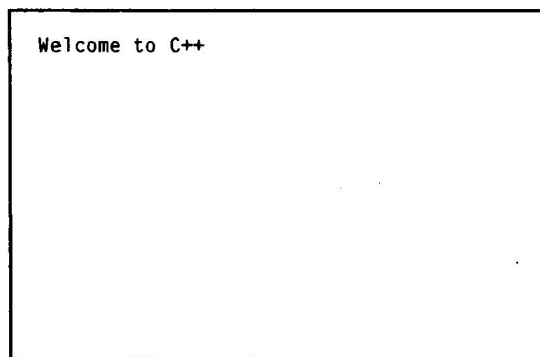
在本章的其余部分,我们将集中精力学习C++的基本概念,这包括让C++为我们的程序创建窗口并以最简单的方式运行。在下一章,我们将学习如何在Windows环境中使用Visual C++。

第一个例子

如果不留心,Visual C++程序可能变得非常复杂和冗长,因此我们将十分注意使例子程序简单,集中精力介绍C++程序设计概念,诸如类和对象。

并非许多程序员都注意到了这一点,但是可以使用Visual C++编写非窗口化(即“控

制台”）程序。现在开始编写例子程序。这些程序在看起来像DOS窗口的窗口中运行。第一个程序非常简单，只是在屏幕上输出消息“Welcome to C++”。



让我们看一下如何完成这项工作。启动Visual C++，如图1.1所示。

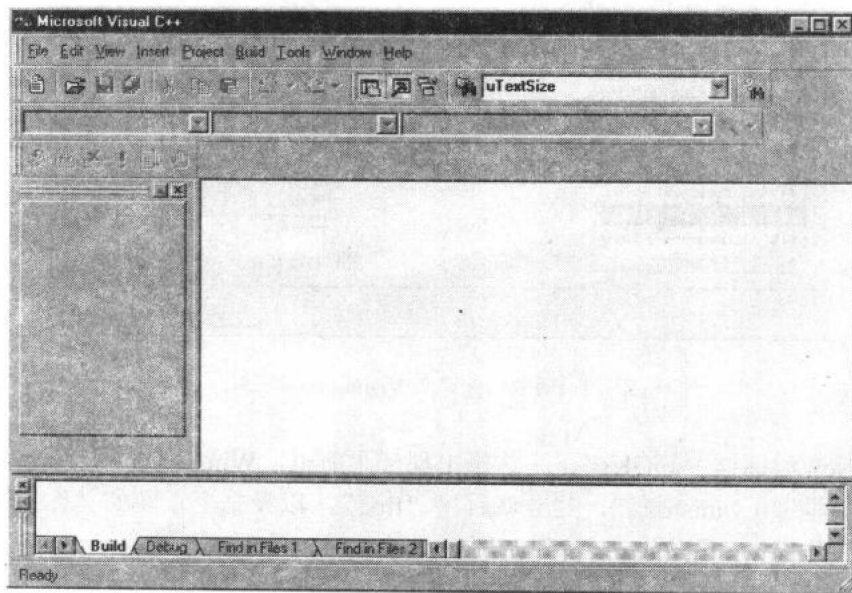


图1.1 Visual C++环境

在本书的大部分章节中，我们将编写处理程序窗口的C++程序，但是本章例外。在这里，我们将编写Win32控制台程序。这些程序在它们自己的窗口中运行，这些窗口是Visual C++为它们设置的。

Visual C++工作区和项目

第一个程序简单地称为“first”，现在作为一个新的Visual C++项目，我们编写这个程序。Visual C++把程序设计任务组织成项目（projects）。通常，每个程序对应一个项目。项目是一组文件的集合，这些文件对于一个可运行的程序是必不可少的。

此外，项目本身放在工作区（workspace）中。在一个工作区中，可以有若干个项目（在Visual C++的Project菜单中，选择Set Active Project项设置“活动”项目）。我们开发的每个项目都具有其自己的工作区。在本书中，我们不介绍工作区，因为在每次开始一个新程

序时，Visual C++都建立一个缺省工作区。本章主要介绍项目，而不是工作区。即使是这样，我们应该认识到，工作的主环境就是Visual C++工作区，项目位于工作区中。

现在开始建立和运行第一个项目。

1. 在Visual C++的File菜单中，选择New选项；这将出现New对话框，如图1.2所示。
2. 在New对话框中，选择Project标签和Win32 Console Application选项。
3. 在Project name框中输入名字first，在Location框中输入first项目的路径。对于我们的项目，使用目录c:\vcpp\first。

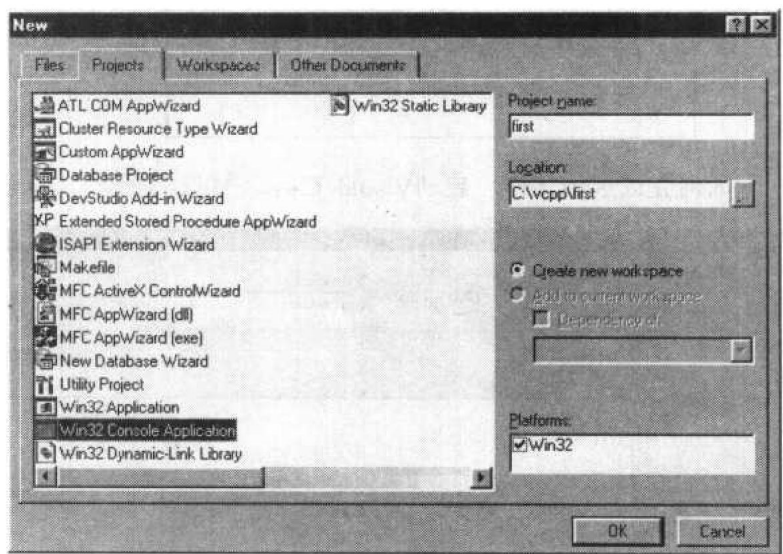


图1.2 建立一个新的Visual C++项目

4. 单击New对话框中的OK按钮。这将出现一个新框，Win32 Application框。
5. 单击该框中的Finish按钮，建立新程序“first”，以及二个重要的文件：first.dsw（管理新工作区）和first.dsp（管理新项目）。

提示：当打开以前建立的程序时，在Visual C++的File菜单中使用“Open Workspace...item”，并且打开程序的.dsw工作区文件。

至此，我们有一个新工作区，并且在这个工作区中有一个名为“first”的新项目。接下来，我们将对这个程序增加实际代码。这里，我们只是想显示消息“Welcome to C++”

建立源代码文件

要向这个项目增加程序设计代码，我们需要一个新文件：first.cpp。扩展名.cpp表示C++源文件。

说明：除了源文件之外，C++程序通常有若干头文件。这些头文件使用扩展名.h。头文件包含有效变量和函数说明。下一章将介绍头文件。

现在，建立first.cpp并且把它加到项目中。

1. 在Visual C++菜单中，使用New项建立first.cpp。但是，这时单击Files标签，如图1.3所示。

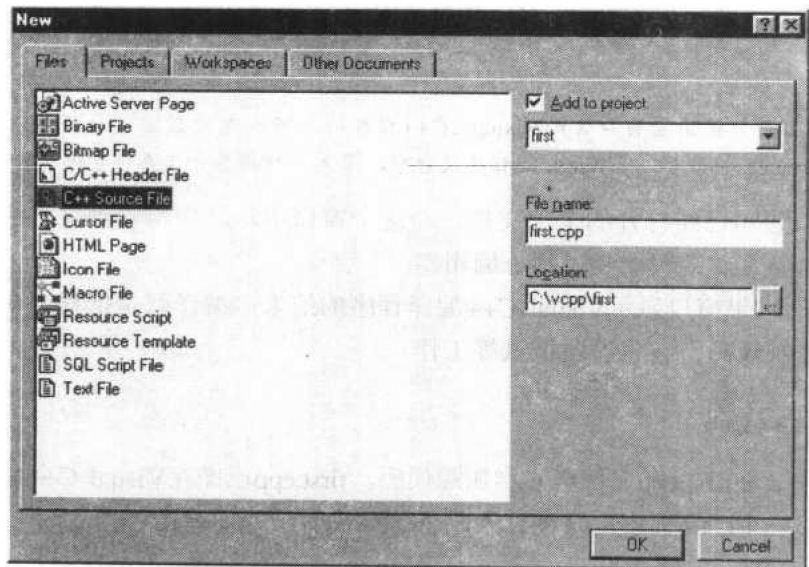


图1.3 建立一个新Visual C++源代码文件

2. 选择C++ Source File选项并且在File name框中给出文件名first.cpp。
3. 确保选择了标有“Add to project”的复选框。然后单击OK按钮。
4. 这样就建立了文件first.cpp，并且在Visual C++环境中打开它，如图1.4所示。

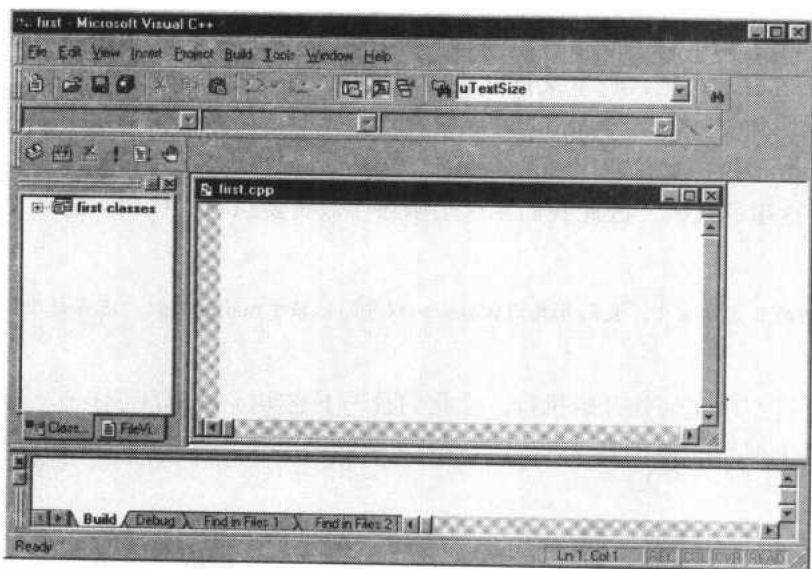


图1.4 在Visual C++中打开一个新文件

现在，在Visual C++中可以看到三个主窗口：左侧的查看器窗口，在其底部具有ClassView和FileView标签；右侧的编辑器窗口，在这个窗口中可编辑文档（现在可以看到打开的空first.cpp文件）；底部的结果窗口，它具有Build、Debug和Find in Files标签。

查看器窗口向我们展示项目的概况。这些概况的组织取决于单击的标签——ClassView，还是FileView（后面将介绍其它标签）。ClassView标签在工作区中显示类（Classes）。我们

将在不久介绍类的概念。**FileView**标签在工作区中显示文件。图1.4显示了工作区、项目和**first.cpp**文件。

说明：当在下一章使用全窗口化的Visual C++程序时，将在查看器窗口看到另一个标签：资源（Resources）标签。在Windows程序设计中，菜单、对话框和工具条等被认为是资源。

编辑器窗口显示已经打开的全部文档。在这个窗口中，我们将看到文本编辑器，以及其它编辑器，诸如菜单编辑器和对话框编辑器。

图1.4底部的结果窗口显示Visual C++编译程序的结果；编译就是把程序转换为计算机能够运行的二进制代码。它也处理调试等工作。

向程序增加C++代码

现在向程序（在**first.cpp**文件中）增加源代码，**first.cpp**已经在Visual C++编辑器窗口中打开。要建立第一个C++程序，只需键入以下C++代码：

```
#include <iostream.h>

void main()
{
    cout << "Welcome to C++ \n";
}
```

这是第一个C++程序。在这里，我们使用标准C++技术显示消息“Welcome to C++”。

需要特殊说明的是，我们包含了**iostream.h**文件。这个文件允许我们使用标准C++写向屏幕的技术（以这种方式包含一个文件就把该文件的内容插入到主文件中）。

接下去，我们建立一个名为**main()**的函数。当程序运行时，要调用这个实用函数。也就是说，在运行一个可执行文件，比如**first.exe**时，Windows在**.exe**文件中查找**main()**函数，并且首先开始执行这里的代码。因此我们在这个函数中放置要执行的代码。过一会将详细介绍**main()**函数。

说明：在本书的其它章节中，我们开发的Windows程序不依赖于**main()**函数，这些程序使用**WinMain()**函数。

由于已经知道程序在何处开始执行，让我们看一下它实际完成的工作——使用C++输入/输出技术在屏幕上显示一条消息。

C++流

在程序中，我们使用<<运算符向C++流（stream）**cout**输出，代码是这样的：**cout<<“Welcome to C++\n”；**：

```
#include <iostream.h>

void main()
{
    → cout << "Welcome to C++ \n";
}
```

这段代码的含意是向C++输出流发送文本“Welcome to C++\n”，该文本直接发送