

# Visual C++ 实用程序设计教程

陈东明 主编



東北大學出版社  
Northeastern University Press

# Visual C++ 实用程序设计教程

陈东明 主编

东北大学出版社

· 沈 阳 ·

© 陈东明 2017

图书在版编目 (CIP) 数据

Visual C++ 实用程序设计教程 / 陈东明主编. — 沈阳: 东北大学出版社, 2017. 12

ISBN 978-7-5517-1741-0

I. ①V… II. ①陈… III. ①C++ 语言—程序设计  
IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 306953 号

---

出版者: 东北大学出版社

地址: 沈阳市和平区文化路三号巷 11 号

邮编: 110819

电话: 024-83683655(总编室) 83687331(营销部)

传真: 024-83687332(总编室) 83680180(营销部)

网址: <http://www.neupress.com>

E-mail: [neuph@neupress.com](mailto:neuph@neupress.com)

印刷者: 沈阳航空发动机研究所印刷厂

发行者: 东北大学出版社

幅面尺寸: 185mm × 260mm

印张: 18.75

字数: 468 千字

出版时间: 2017 年 12 月第 1 版

印刷时间: 2017 年 12 月第 1 次印刷

组稿编辑: 罗鑫

责任编辑: 刘莹

责任校对: 梦彤

封面设计: 潘正一

责任出版: 唐敏志

---

ISBN 978-7-5517-1741-0

定价: 58.00 元

# 前言

目前,市场上有很多关于程序设计方面的书籍。其中,有关 Visual C++ 程序设计的参考书也占据一定的比例,但它们大都是系统地介绍语法和基础知识。这些基础理论固然重要,但很多人希望能有介绍语言类使用方法和技巧的读物。为了弥补这类书籍的不足,我们编写了本书,重点关注 Visual C++ 的使用方法、开发技巧和值得关注的问题,强调实践操作。

虽然微软的 Visual Studio 有不同的版本,但本书仍然选择 Visual C++ 6.0 为对象,因为这对于一般的应用来说已经足够了,并且占用的系统资源较少。本书既不介绍 C++ 的基础知识,也不详细地介绍微软基础类库(MFC)的内容及使用,因为这些知识在很多地方都很容易获得和学习。本书重点关注 Visual C++ 的实用性,主要内容如下。

第1章为“Visual C++ 集成开发环境”。详细地介绍了 VC++ 集成开发环境(IDE),介绍了一些常见且关键的菜单操作及系统功能配置,给出了调试程序的方法及技巧,介绍 Visual C++ 程序 Debug 版和 Release 版的不同特点,以及 Visual C++ 程序发布需要注意的问题。

第2章为“Windows 编程机制”。详细地介绍了 Windows 的消息驱动机制,阐述 Visual C++ 创建窗口的过程、MFC 及其编程机理。

第3章为“Visual C++ 应用程序框架”。详细地介绍了 Visual C++ 开发不同类型应用程序的操作过程,包括控制台程序(Win32 Console Application)、MFC AppWizard (exe)、动态链接库(DLL,含 Regular DLL、MFC 扩展 DLL 和 Win32 DLL)、ActiveX 控件、ATL COM AppWizard,同时给出了动态库、组件的调用过程。

第4章为“数据库编程”。以 ODBC 和 ADO 两种方式连接 Access、SQL Server 等数据库,详细地介绍了数据库编程的基本操作过程。

第5章为“网络编程”。基于 TCP 和 UDP 套接字编程,实现了包括客户端和服务端(发送端和接收端)的网络编程,同时给出了 MFC WinSock 类编程案例。

第6章为“Office 导入导出编程”。通过 Visual C++ 读写微软 Word 和 Excel。

第7章为“MFC 异常处理”。阐述了 Visual C++ 的异常处理机制,介绍了 MFC 中的异常类。

第8章为“文件操作”。介绍并比较了 C 语言文件操作、C++ 流文件操作、Win32 API 文件操作和 MFC 类文件操作。

第9章为“进程与线程”。介绍进程、线程和线程同步的基本概念及其关系,并给

出了多线程编程实例。

第10章为“Visual C++项目开发案例”。以《学生管理系统》为案例，遵循软件开发生命周期，系统地介绍了需求分析、系统设计、系统实现和系统测试全过程，为使用 Visual C++ 开发实际应用系统提供了通用的方法。

第11章为“Visual SourceSafe”。介绍了 Microsoft 源代码控制系统 Visual SourceSafe 的使用方法，并以实际的 VC++ 工程为例，阐述了源代码的共享管理。

第12章为“Visual C++ 实用技巧集锦”。提供作者认为对 Visual C++ 程序开发有用的小技巧和经验分享。

第13章为“Visual Studio 其他版本介绍”。对 Visual Studio 2017 集成开发环境做了简单介绍。

本书由陈东明主编，王冬琦、马毅参编，王云开、袁泽枝、黄新宇参与了大量的文字校对及处理工作，全书由陈东明负责统稿。

本书可能会存在错漏之处，敬请读者批评指正。

作者

2017年10月于东北大学

# 目 录

<b>1 Visual C++ 集成开发环境</b> .....	<b>1</b>
1.1 Visual C++ 6.0 安装配置 .....	1
1.2 第一个 VC++ 程序 .....	9
1.3 Debug \ Release 版本 .....	11
1.4 如何调试 VC++ 程序 .....	12
<b>2 Windows 编程机制</b> .....	<b>15</b>
2.1 VC++ 窗口的创建过程 .....	15
2.2 MFC 简介 .....	26
<b>3 Visual C++ 应用程序框架</b> .....	<b>30</b>
3.1 控制台程序 (Win32 Console Application) .....	30
3.2 MFC 应用程序 [MFC AppWizard (exe)] .....	33
3.3 动态链接库 .....	46
3.4 ActiveX 控件 .....	75
3.5 ATL COM AppWizard .....	87
<b>4 数据库编程</b> .....	<b>92</b>
4.1 创建 Database Project .....	92
4.2 ODBC 数据库编程 .....	98
4.3 ADO 数据库编程 .....	124
<b>5 网络编程</b> .....	<b>144</b>
5.1 WinSock API 编程 (TCP/UDP) .....	144
5.2 MFC WinSock 类编程 .....	163
<b>6 Office 导入导出编程</b> .....	<b>171</b>
6.1 VC++ 6.0 实现对 Word 操作 .....	172

6.2	VC++6.0 实现对 Excel 操作 .....	177
<b>7</b>	<b>MFC 异常处理 .....</b>	<b>183</b>
7.1	Visual C++ 的异常处理机制 .....	183
7.2	MFC 宏异常处理 .....	185
<b>8</b>	<b>文件操作 .....</b>	<b>190</b>
8.1	C 函数实现文件操作 .....	190
8.2	C++ 文件操作 .....	192
8.3	Win32 API 实现文件操作 .....	194
8.4	MFC 类实现文件操作 .....	196
<b>9</b>	<b>进程与线程 .....</b>	<b>200</b>
9.1	基本概念 .....	200
9.2	多线程程序 .....	202
9.3	多线程编程实例 .....	206
<b>10</b>	<b>Visual C++ 项目开发案例 .....</b>	<b>214</b>
10.1	需求分析 .....	214
10.2	系统设计 .....	214
10.3	系统实现 .....	218
<b>11</b>	<b>Visual SourceSafe .....</b>	<b>248</b>
11.1	Microsoft Visual SourceSafe 简介 .....	248
11.2	Microsoft Visual SourceSafe 使用举例 .....	249
<b>12</b>	<b>Visual C++ 实用技巧集锦 .....</b>	<b>266</b>
<b>13</b>	<b>Visual Studio 其他版本介绍 .....</b>	<b>280</b>
13.1	Visual Studio 2017 功能简介 .....	280
13.2	Visual Studio 2017 项目创建过程 .....	282
13.3	Visual Studio 2017 中相关应用 .....	288

# 1 Visual C++ 集成开发环境

Visual C++ (简称 VC++) 是 Microsoft 公司的 Visual Studio 开发工具箱中的一个 C++ 程序开发环境, 凭借着 C++ 语言的强大威力、开发环境的良好支持, 以及与 Windows 操作系统的“血缘”关系, 一直是 Windows 操作系统环境下最主要的开发工具之一。VC++ 是一个面向对象的可视化集成开发系统, 它不但具有程序框架自动生成、灵活方便的类管理、代码编写和界面设计集成交互操作等优点, 还可以开发多种应用程序, 从底层软件到上层直接面向用户的软件, 都可以利用 VC++ 开发, 而且通过简单的设置就可以使其生成的程序框架支持数据库接口、OLE2、WinSock 网络、3D 控制界面等不同的应用需求。

## 1.1 Visual C++ 6.0 安装配置

自从微软于 1993 年推出 Visual C++ 1.0 工具以后, Visual C++ 的版本不断升级。虽然已经推出了更新的版本, 但 Visual C++ 6.0 依然占有较大的市场。

Visual C++ 6.0 提供了良好的可视化编程环境, 它由许多组件组成, 包括编辑器、调试器, 以及程序向导 AppWizard、类向导 Class Wizard 等开发工具。这些组件通过一个名为 Developer Studio 的组件集成在一起, 形成一个统一的集成开发环境 (Integrated Development Environment, IDE)。Visual C++ 6.0 不仅为用户提供了编辑代码、编译、链接、调试等基本功能, 更提供多种辅助开发工具, 以使用户的开发过程更简单、更方便。

首先应安装 Microsoft Visual C++ 6.0, 安装完成后, 启动该集成开发环境, 其主界面如图 1.1 所示。

在此集成开发环境中, 对一些菜单及其功能做出说明。

### 1.1.1 工作区

一个工作区 (Workspace, 扩展名为 .dsw) 可以加载多个工程, 工作区这个概念就是为了管理多个工程设计的, 比如要开发一个网络程序, 有服务器和客户端两个程序 (工程), 那么, 把两个工程放在同一个工作区, 以方便管理。当用户新建一个工程时, 系统会自动地将该工程放在一个缺省的工作区中。程序编译执行的最小单位还是工程, 与工作区无关。

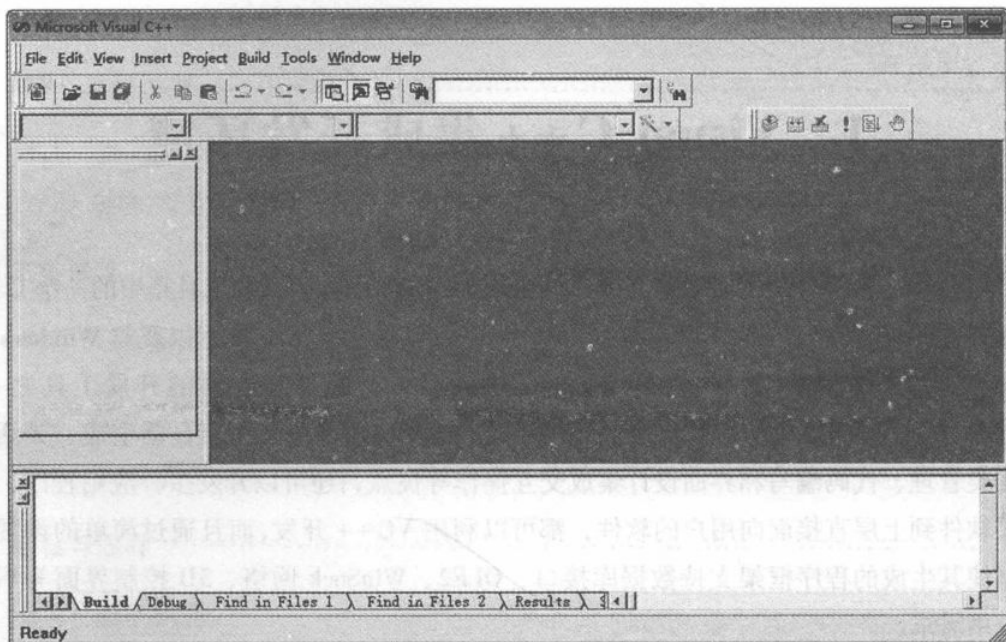


图 1.1 Microsoft Visual C++6.0 主界面

### 1.1.2 工程（项目）

一个工程（Project，扩展名为.dsp）的最终生成结果就是一个EXE文件，或者其他类型的可执行文件，一个工程中可以有多个源程序文件。

在VC++集成开发环境中，可以在一个工作区中同时加载多个Project，可以选择菜单Project→Set Active Project来设置当前活动的Project（项目/工程）。这样，在集成开发环境中进行的操作只是针对当前活动的Project。

当多个工程被同时装载在一个工作区中时，可以实现程序联调。若是EXE类型的应用程序，则自身能够运行；但若是像动态链接库这种自身不能运行的应用程序需要进行调试时，则要借助于其他可执行程序进行联调。

### 1.1.3 部分菜单功能简介

#### (1) 菜单Project→Settings (Alt + F7)

通过选择该菜单，打开如图1.2所示对话框。

对该对话框的相关功能介绍如下。

##### ① General 标签。

Microsoft Foundation Classes：提示应用程序支持（使用）MFC的方式

Not Using MFC：不使用MFC

Use MFC in a Static Library：使用MFC静态链接

Use MFC in a Shared Library：使用MFC动态链接

Output directories：提示输出目录

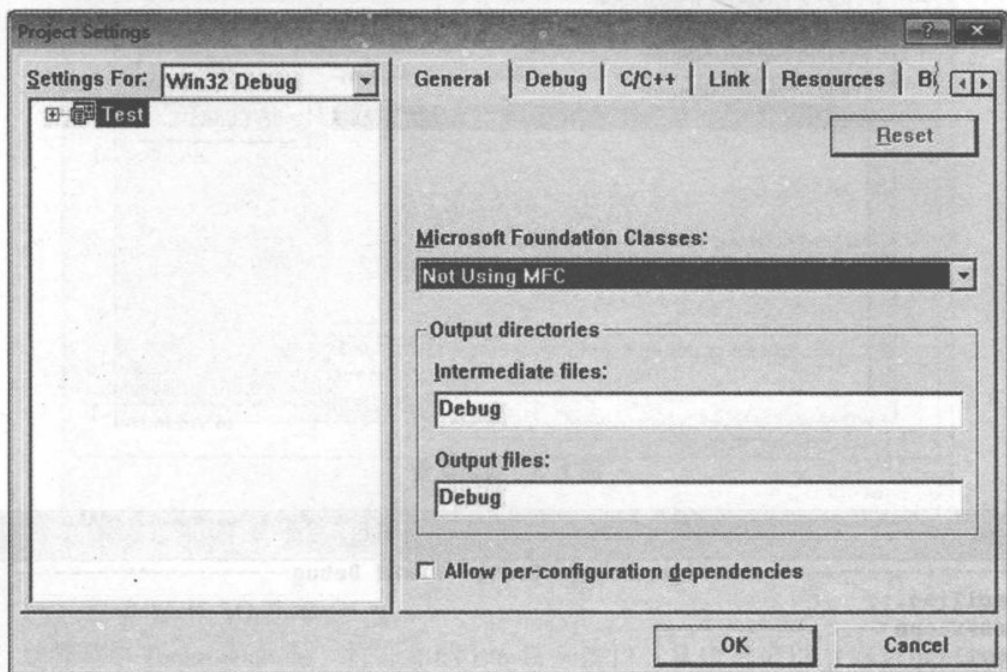


图 1.2 Project Settings

Intermediate files: 中间 (临时) 文件的输出路径

Output files: 生成的输出文件路径 (比如可执行文件、动态链接库等)

## ② Link 标签。

Output file name: 项目生成的输出应用程序文件名称

Object/library modules: 当前工程需要包含的库文件 (如 .lib)

## (2) 菜单 Project→Insert Project into Workspace

在当前工作区中已经有项目打开的情况下, 插入一个新的项目到工作区。这样, 在一个工作区中, 就可以加载多个项目, 但只有一个是当前活动的项目。

## (3) 菜单 Project→Set Active Project

通过该菜单选项, 可以将某一项目设置为当前活动项目。也可以在工作区中项目名称上单击鼠标右键, 从弹出的菜单中选择 Set as Active Project 来设置。一旦设置了当前活动的项目, 那么所做的编译、链接、调试和运行等操作都是针对该活动项目的。

## (4) 菜单 Build→Set Active Configuration

通过该菜单选项, 打开如图 1.3 所示对话框, 用来设置项目的配置, 一般是 Debug 和 Release, 系统根据该配置来进行编译和链接。

图 1.4(a) 和图 1.4(b) 所示分别显示当前配置为 Debug 和 Release 时的编译结果输出窗口, 从输出窗口可以看到编译文件的过程以及出现的错误和警告数量。

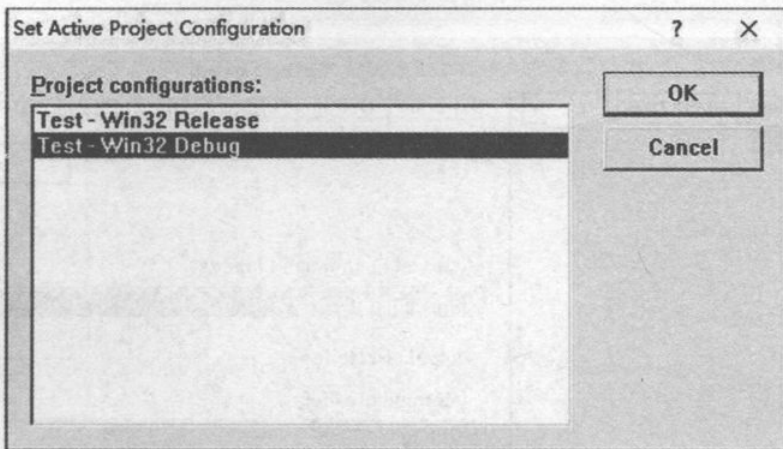
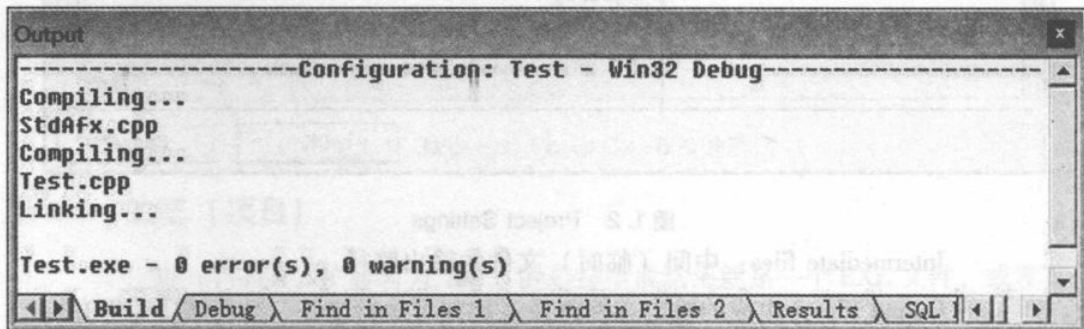
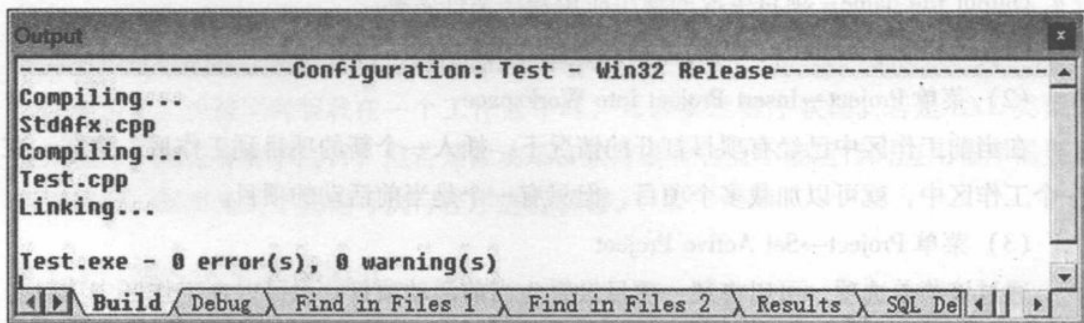


图 1.3 项目配置



(a) 项目配置为 Win32 Debug 时输出信息



(b) 项目配置为 Win32 Release 时输出信息

图 1.4 输出信息

#### (5) 菜单 Tools→Customize

通过选择该菜单选项，打开如图 1.5 所示窗口，用户可以定制集成开发环境 (IDE) 及编辑器中的相关配置，以满足用户个性化的需要。

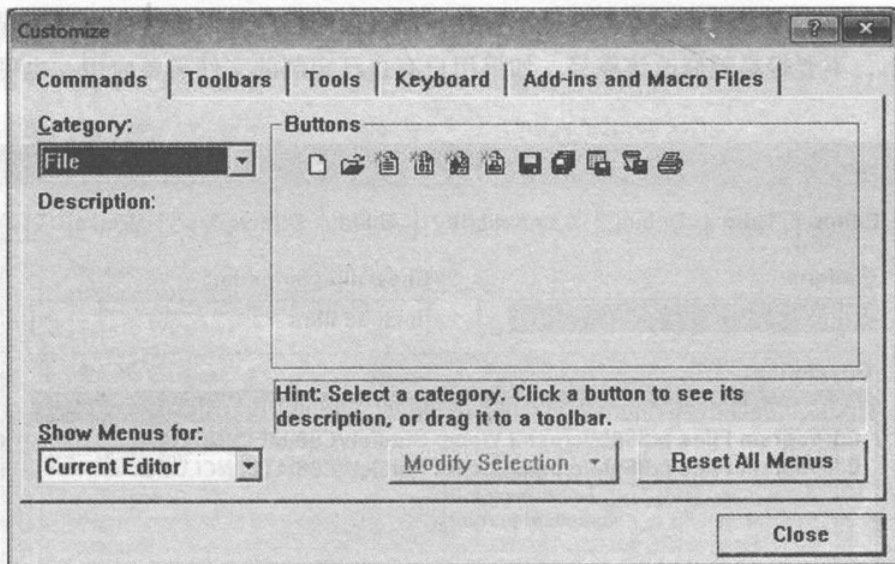


图 1.5 个性化相关配置

## (6) 菜单 Tools→Options

选择菜单 Tools→Options，打开如图 1.6 所示窗口，从这里可以对编辑、调试、路径、工作区、格式等进行设置。

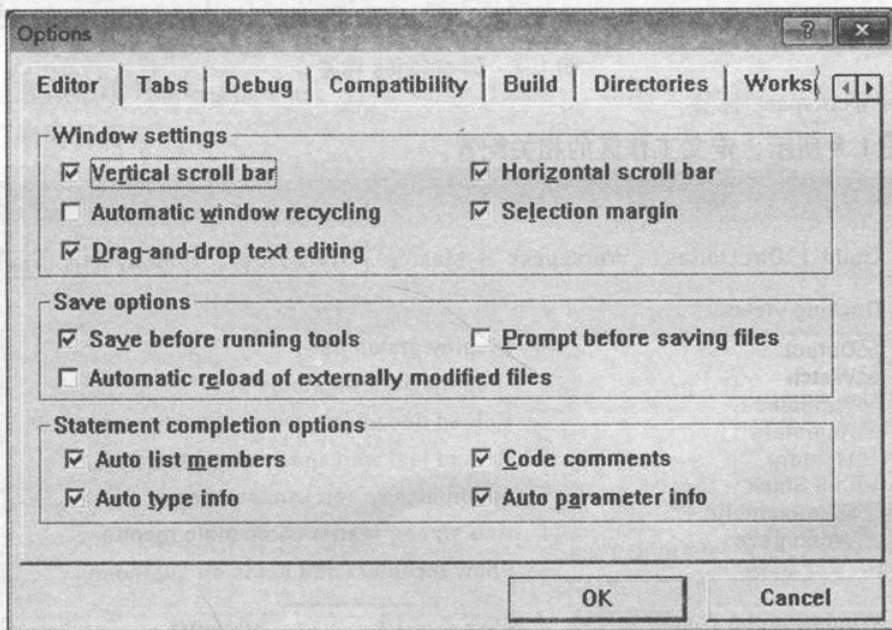


图 1.6 Options 选项

## (7) Directories 标签

如图 1.7 所示，Directories 标签里记录 Visual Studio 系统安装的相关文件路径，包括可执行文件 (Executable files)、包含文件 (Include files)、库文件 (Library files)、源文件 (Source files) 等。这些路径里包含大量程序编译、链接时需要的文件，如果因为用户操作不当，修改了这里的路径或者挪动了磁盘相关文件，造成该配置的路径和实际文

件路径不一致，或者该路径下的文件不存在，那么应用程序编译链接会出现错误。在一般情况下，不要轻易地修改该路径。如果用户有自己定制的文件需要应用，也可以添加到此处。

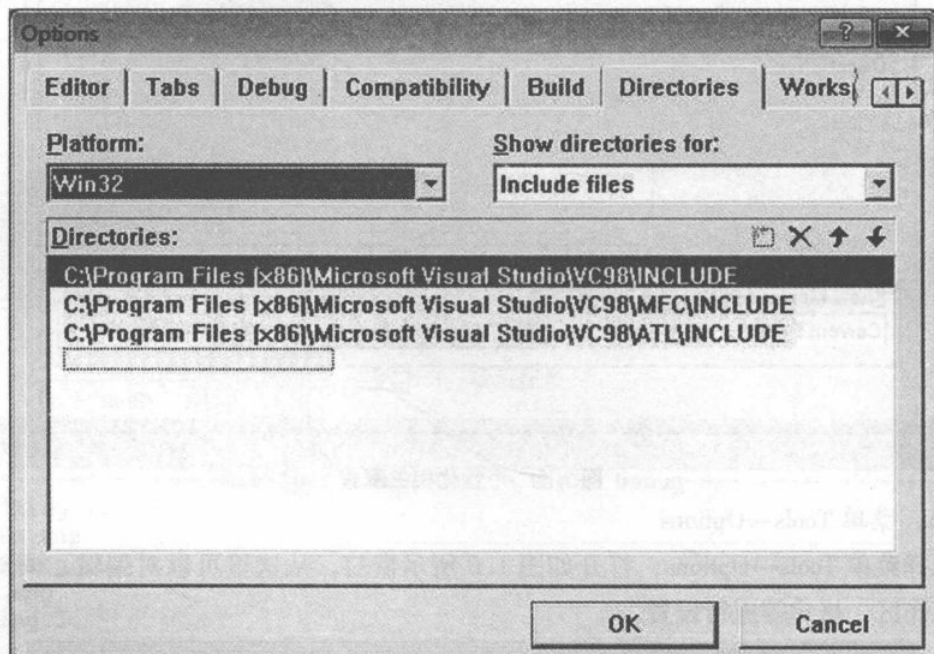


图 1.7 Directories 标签

#### (8) Workspace 标签

如图 1.8 所示，定义工作区的相关配置。

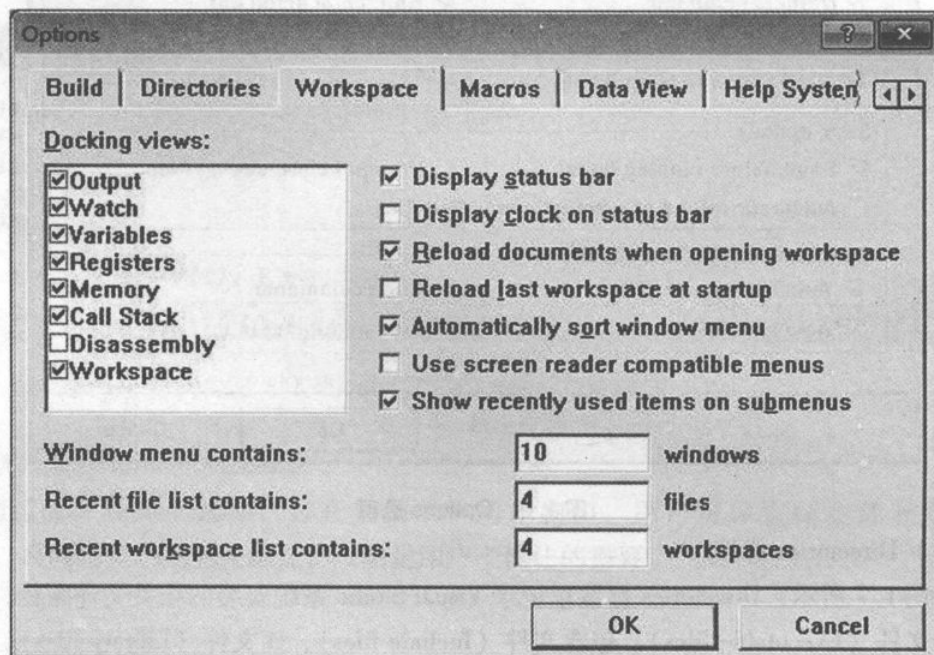


图 1.8 Workspace 标签

## (9) Format 标签

如图 1.9 所示, Format 标签定义相关窗口和编辑器的相关格式, 以满足不同用户的个性化设置。

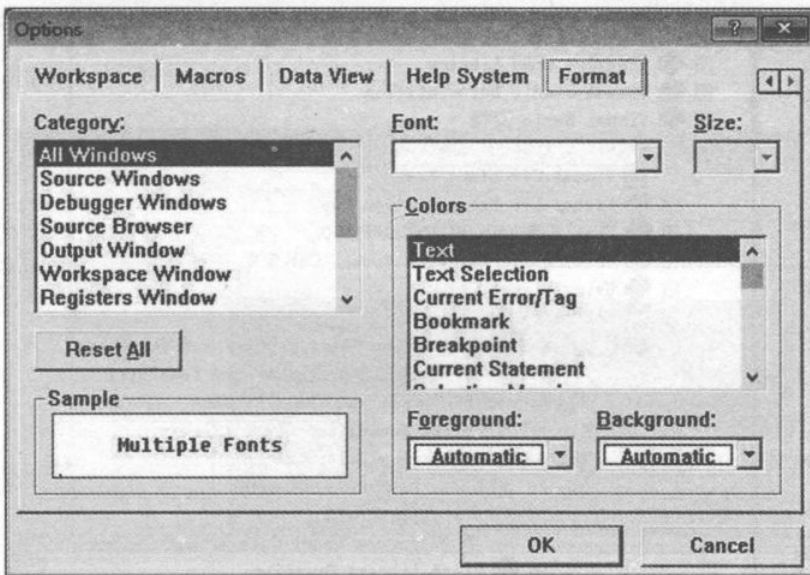


图 1.9 Format 标签

## (10) 菜单 Help→Contents

通过选择菜单 Help→Contents, 打开 MSDN Library, 如图 1.10 所示 (不同版本的显示方式可能会有所区别)。

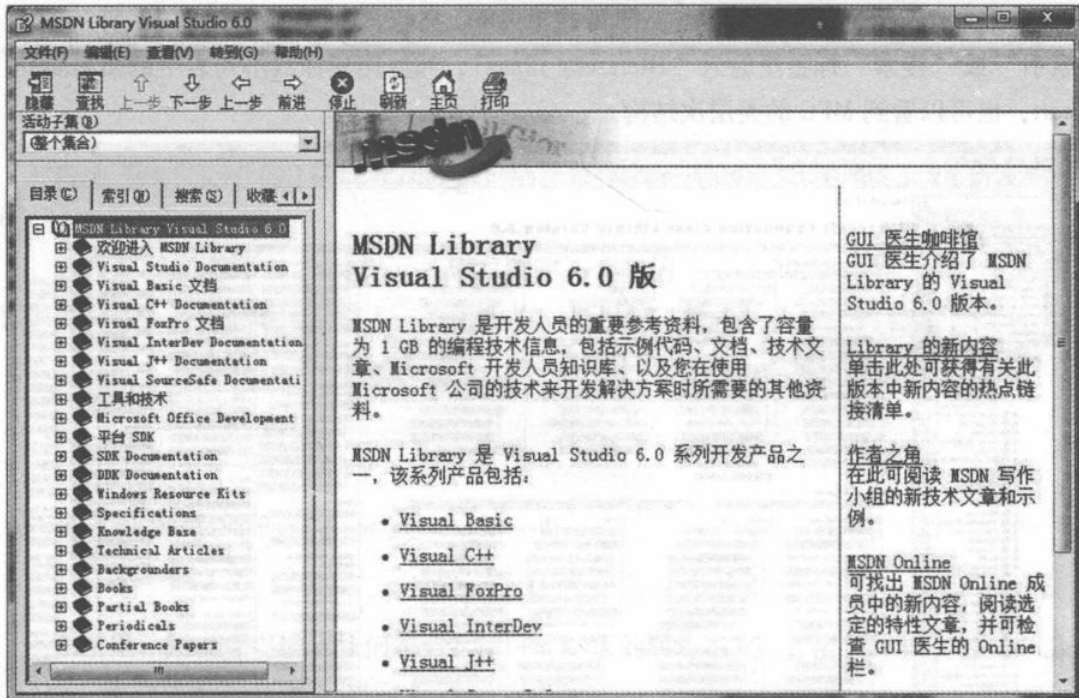


图 1.10 MSDN Library 主界面

MSDN 的内容非常全面,可以为初学者学习 VC++ 提供很大的帮助。打开 MSDN 界面,在“目录”标签页里找到“Hierarchy Chart”节点,如图 1.11 所示。

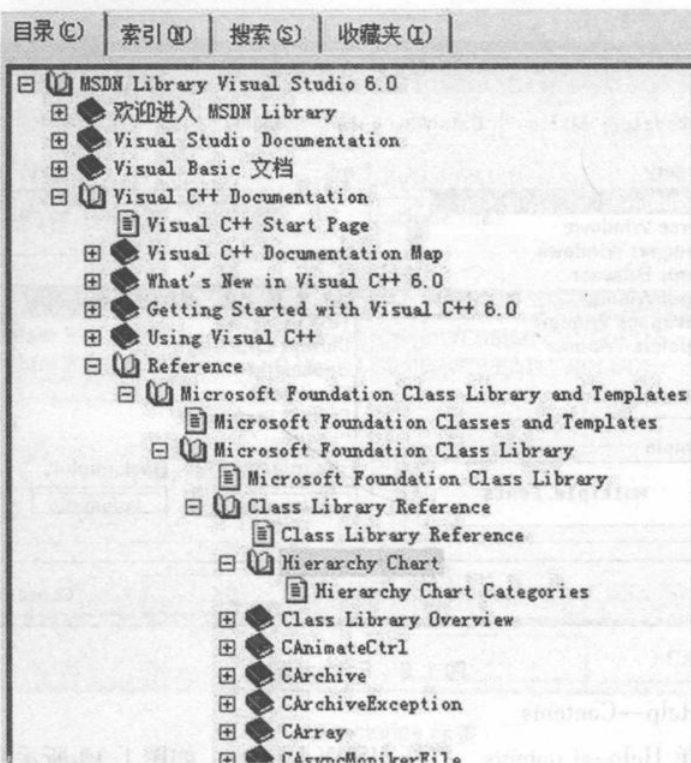


图 1.11 MSDN 目录页

单击“Hierarchy Chart”节点,即可打开 MFC 类层次结构,如图 1.12 所示。在“索引”或“搜索”标签里输入“Hierarchy Chart”,然后用鼠标双击列表中的 Hierarchy Chart,也可以看到 MFC 的类层次结构。

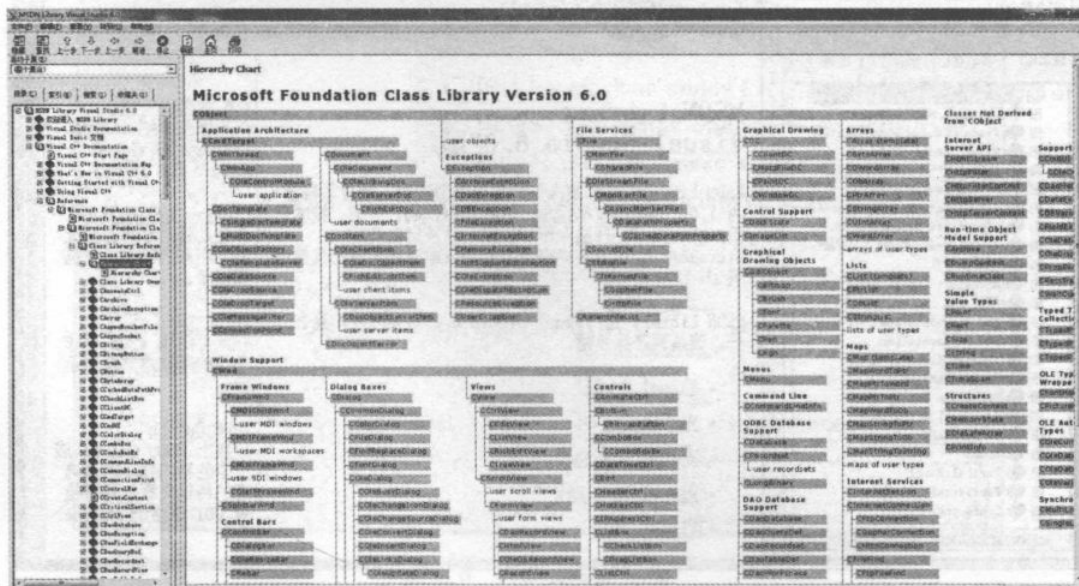


图 1.12 MSDN 类层次结构图

### (11) Build MiniBar 工具条

该工具条按钮包括 Compile (编译)、Build (生成/构建)、Stop Build (停止生成)、Execute Program (执行程序)、go (调试运行)、Insert/Remove Breakpoint (插入/删除断点), 可执行应用程序的编译、调试和运行等工作。Build 菜单中一般涵盖 MiniBar 工具条的功能, 如图 1.13 所示。

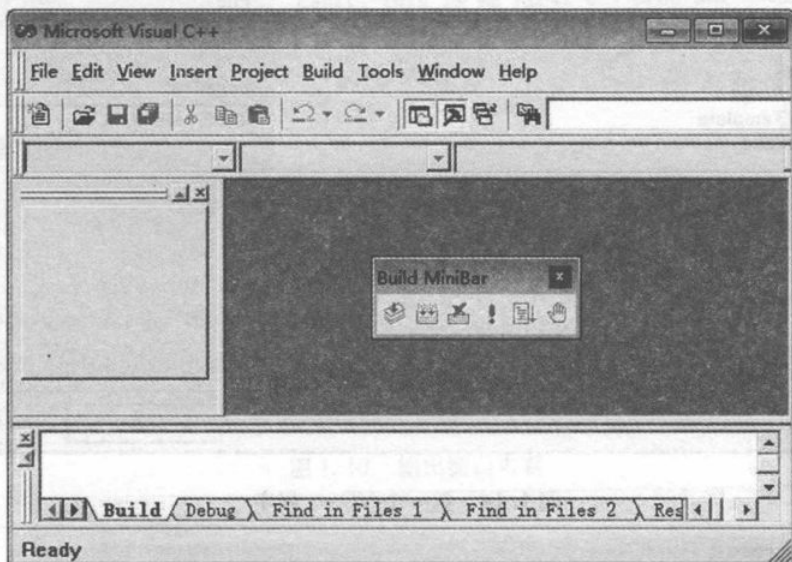


图 1.13 Build MiniBar 工具条

## 1.2 第一个 VC++ 程序

启动 Visual C++ 6.0 集成开发环境, 选择菜单 File→New, 出现如图 1.14 所示对话框。

选择 Files 标签下的 C++ Source File, 创建一个空的 .cpp 文件, 缺省文件名为 Cpp1, 扩展名为 .cpp, 在编辑区域输入一个简单的 C++ 程序。Cpp1.cpp 文件内容如下:

```
#include "iostream.h"
void main()
{
    cout << "This is my first C++ program." << endl;
}
```

在这里, 用户可以输入任何符合 C/C++ 要求的程序代码。直接编译该文件 [Build (F7)], 出现如图 1.15 所示提示:

提醒是否创建缺省工作区, 选择“是 (Y)”。编译、链接后, 在输出窗口可以观察

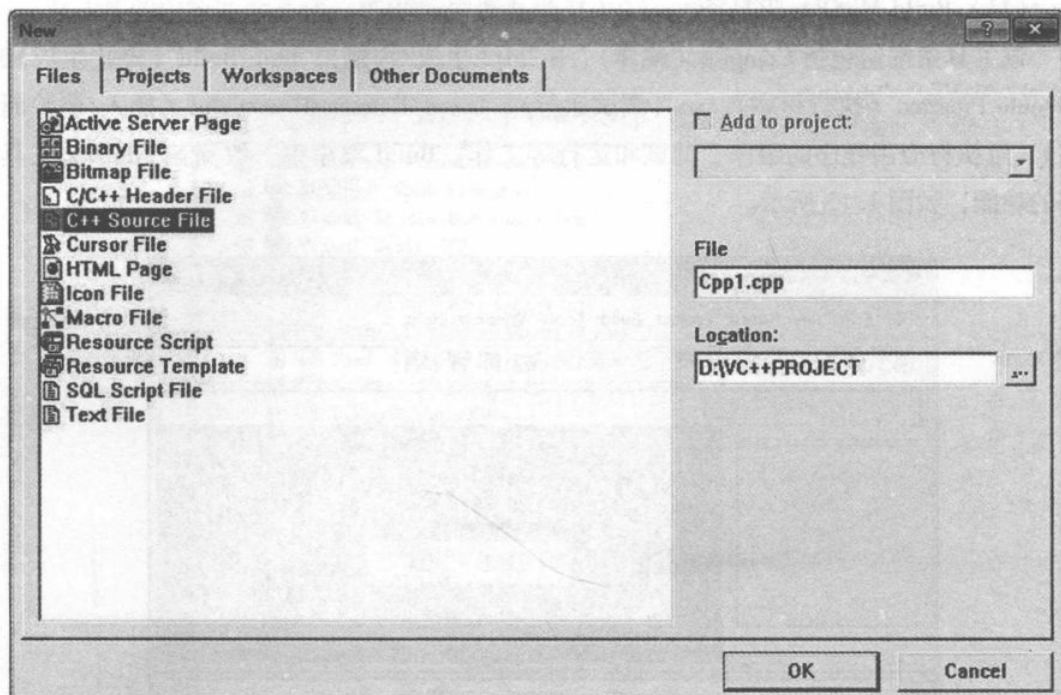


图 1.14 第一个 VC++ 程序

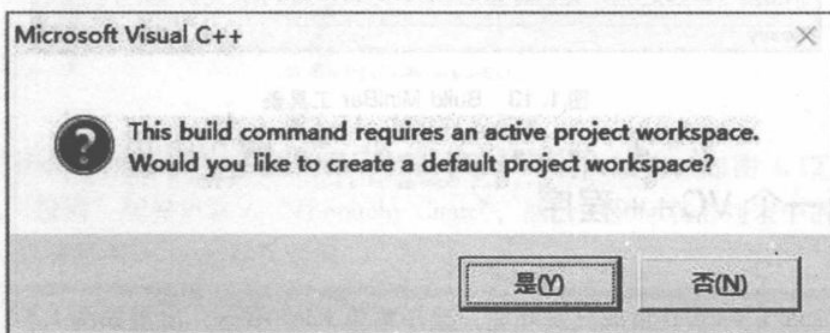


图 1.15 编译时出现的提示

到如图 1.16 所示编译、链接进度及结果窗口，并给出错误（error）和警告（warning）的数量。

在没有错误的情况下，运行上述程序 [Execute Program (Ctrl + F5)]，得到如图 1.17 所示输出结果。

这里通过简单的操作即可得到一个完整的应用程序，并获得正确的运行结果。用户可以在此文件中输入任何 C/C++ 程序，在 Visual C++ 6.0 集成开发环境中进行编辑、编译、链接和运行操作，并获得程序的隐形结果。