

Visual C++ 6.0

高级编程

康博创作室 编著
梁书斌 审校

清华大学出版社

(京) 新登字 158 号

内 容 简 介

本书全面介绍了可视化编程语言 Visual C++ 6.0 的语言基础和编程技巧。

全书共 11 章, 主要内容包括: Visual C++ 6.0 的安装及特征, Visual C++ 6.0 的工作环境 with 用户界面, Visual C++ 6.0 的编程基础知识, Visual C++ 6.0 的常用工具, MFC 类库的结构, 创建 Windows 应用程序的基础, 创建边框窗口、文档和视图, 绘图程序的实现, 创建对话框与控件, 动态链接库, ActiveX 控件及其应用。

全书内容翔实, 结构合理, 编程示例丰富, 技术剖析深入浅出, 是广大计算机用户学习 Visual C++ 必备的入门参考书, 也可作为各类大中专院校的教材。

版权所有, 翻印必究。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

书 名: Visual C++ 6.0 高级编程

作 者: 康博创作室

出版者: 清华大学出版社(北京清华大学校内, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市丰华印刷厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 22.5 **字数:** 531 千字

版 次: 1999 年 4 月第 1 版 1999 年 4 月第 1 次印刷

书 号: ISBN 7-302-03507-5 TP·1507

印 数: 0001~6000

定 价: 29.80 元

前 言

Visual C++ 6.0 是 Microsoft 公司的 Developer Studio 6.0 工具集的重要组成部分,是一种用于开发 Windows 95、Windows 98 或 Windows NT 应用程序的可视化开发工具。它改善了传统的编程手段,使得程序员可以直接在用户界面良好的可视化开发环境中进行工作。Visual C++ 6.0 还集成了多种有用的工具与功能,从而大大提高了应用程序的开发效率。

本书全面介绍了 Visual C++ 6.0 的语言基础和编程技巧。全书共分 11 章。第 1 章介绍了 Visual C++ 6.0 的安装与特征;第 2 章介绍了 Visual C++ 6.0 的工作环境与用户界面;第 3 章详细论述了 Visual C++ 6.0 的语言基础;第 4 章介绍了几种常用的工具程序;第 5 章介绍了 MFC 类库的结构;第 6 章介绍了如何利用向导创建 Windows 应用程序框架;第 7 章介绍了如何创建边框窗口、文档和视图;第 8 章介绍了绘图程序的实现方法;第 9 章介绍了如何创建对话框与控件;第 10 章讨论了动态链接库;第 11 章介绍了 ActiveX 控件及其应用技巧。

本书属集体创作,参加本书编写和录排工作的有时花玲、张磊、黄斌、马占魁、张红军、孙全党、刘利平、许书明、王维、孔祥峰、刘春晓、李增民、张巍、常征、何君等人。梁书斌先生认真审阅了全书,并提出了许多宝贵的修改意见,在此表示感谢。由于我们水平有限,错误和疏漏之处在所难免,诚望广大读者多提宝贵意见。

作 者
1999 年 1 月

第 1 章

Visual C + + 6 .0 概述

随着计算机多媒体技术和图形图像技术的发展,可视化技术受到了广泛重视,直接面向对象的可视化技术受到广大计算机专业人员和非专业人员的喜爱,越来越多的计算机程序员开始研究和应用可视化技术。

可视化技术包含两方面的含义:一是软件开发阶段的可视化,即编程可视化。二是利用计算机图形技术和方法,对大量的数据进行处理,并用图形图像的方法形象地加以显示。

1.1 Visual C++ 6.0 的特点

Visual C++ 6.0 提供了用于开发 Windows(包括 Windows 95 和 Windows NT)环境下的应用程序的简捷、快速、实用的开发环境。利用 Visual C++ 6.0 开发 Windows 应用程序具有很高的效率。

Visual C++ 6.0 提供了 MFC 类库,开发者只需做少量工作即可得到功能齐全的 Windows 应用程序。与使用 C 和 Windows SDK 开发 Windows 应用程序相比,使用 Visual C++ 6.0 建立一个完美的 Windows 应用程序所花费的时间要少得多。

Visual C++ 6.0 提供了一个高度集成的工具集,使得在开发应用程序的全过程中都保证了较高效率。集成化便于程序开发,开发者可以同时诸如编辑、建立、调试等不同任务之间快速切换,甚至可以同时执行。

图形化的可视特性使得 Visual C++ 6.0 简单易学。由于 Visual C++ 6.0(包括所有工具)是完全基于 Windows 的,因此,Windows 的所有优越性(一致性运算、多任务、多线程以及可嵌入字体等)它都具备。另外,丰富的文档、样本代码、联机信息等可以帮助各个层次的开发人员。

Visual C++ 6.0 包括了 Microsoft 的代码优化技术,此项技术保持了 Microsoft C++ 所有的优化属性,并作了进一步的完善,因而,用 Visual C++ 6.0 开发的程序运行速度更快。

1.2 Visual C++ 6.0 的新增功能

Visual C++ 6.0 在界面上与 Visual C++ 5.0 没有太大的差别,但在实现技术上与 Visual C++ 5.0 相比作了许多改进,主要表现在以下几个方面:

1. 编译器和连接器的改进

Visual C++ 6.0 编译器不仅支持 COM(对象控件模型)应用程序开发,还进一步简化了 COM 应用程序的开发过程;Visual C++ 6.0 编译器在代码生成方面进一步作了优化,使得目标程序代码更紧凑,运行速度更快;Visual C++ 6.0 增加了对不同微处理器的支持,在生成应用程序时可以根据微处理器的不同产生不同目标代码,这可通过新增加的编译开关/ G3,/ G4,/ G5,/ G6 来实现;新增开关/ EH,则增强了 Visual C++ 6.0 对异常的处理能力。

2. 更加完善的 MFC 库

Visual C++ 6.0 的类库在对于 Internet 和数据库的支持方面也作了较大改进。首先 Visual C++ 6.0 允许开发典型的基于 Internet 的应用程序,允许异步下载文件和设置应用程序的属性,并且在任务完成后,应用程序会自动释放系统资源供其他应用程序使用。总之 Visual C++ 6.0 使得用户的应用程序和 Internet 紧密地结合在一起;其次 Visual C++ 6.0 的 MFC 库增加了对 DAO 的支持,并将原来的 ODBC API 函数进行了封装,提供了一系列 ODBC 类,以支持 ODBC 3.0 标准。

3. 与开发环境同步进行的改进和提高

Visual C++ 6.0 与 Windows 98 和 Internet Explorer 4.0 相伴而生,所以 Visual C++ 6.0 也提供了对它们的支持,主要表现在支持 Internet Explorer 4.0 控件,完善了对 OLE, ODBC, ADO 的支持,支持动态 HTML、HTML 帮助系统以及类似于 Internet Explorer 的应用程序和界面。

1.3 Visual C++ 6.0 开发环境

Visual C++ 6.0 开发环境 Developer Studio 由在 Windows 95 或者 Windows NT 环境下运行的一套集成工具所组成,包含文本编辑器(Text Editor)、资源编辑器(Resource Editor)、项目建立工具(Project Build Faculties)、优化编译器(Optimizing Compiler)、增量连接器(Incremental linker)、源代码浏览器(Source Code Browser)、集成调试器(Integrated Debugger)和图形浏览器条(the Visual Wordbranch)。此外,Visual C++ 6.0 还提供了几个附加的工具,如进程观察器 Spy++, 文件夹比较工具 Windiff 等。

有了 Microsoft Visual C++ 6.0,很容易在个人计算机上开发基于 Windows 的应用程序、数据库应用程序、Internet 应用程序。因为编写基于文本的应用程序利用了原 C++ 的方法,而 Visual C++ 6.0 中又提供了预先写好的功能很强的类(Classe)集,这样极大地简化了 Visual C++ 6.0 程序的编写过程。

借助 Visual C++ 6.0 的 MFC 类库和 Visual C++ 6.0 的交互工具,用户不必编写代码就能创建一个 C++ 应用程序。

1.4 Visual C++ 6.0 的安装

运行 Visual C++ 6.0 所需的软、硬件配置应满足以下要求:

Windows 95 或 Windows NT 及更高版本的 Windows 系统且安装有 4.0 以上版本的 Internet Explorer。

个人计算机,配有 Pentium 处理器。

8MB 以上内存,建议使用 16MB 内存或更大内存配置。

最少 200MB 的可用硬盘空间,更大的硬盘空间,可以安装更多的选件。

高密软盘驱动器。

800 * 600VGA 以上的显示器。

CD-ROM 驱动器。

使用 Visual C++ 6.0 开发应用程序之前,必须先用安装程序将 Visual C++ 6.0 的开发环境安装到计算机上。

在 Windows 95 中安装 Visual C++ 6.0 企业版(Enterprise Edition)的过程如下:

(1) 启动 Windows 95。

(2) 在 CD-ROM 驱动器中插入 Microsoft Visual Studio 6.0 系统光盘(Visual C++ 6.0 是其组件之一)。

(3) 单击“开始”按钮,从“开始”菜单选择“运行”命令,弹出“运行”对话框。在“打开”文本框中键入 D:Setup.exe(这里的 D 是用户的 CD-ROM 驱动器符),单击“确定”按钮,出现如图 1-1 所示的安装初始屏幕。在此屏幕中单击 Next 按钮,进入下一步。

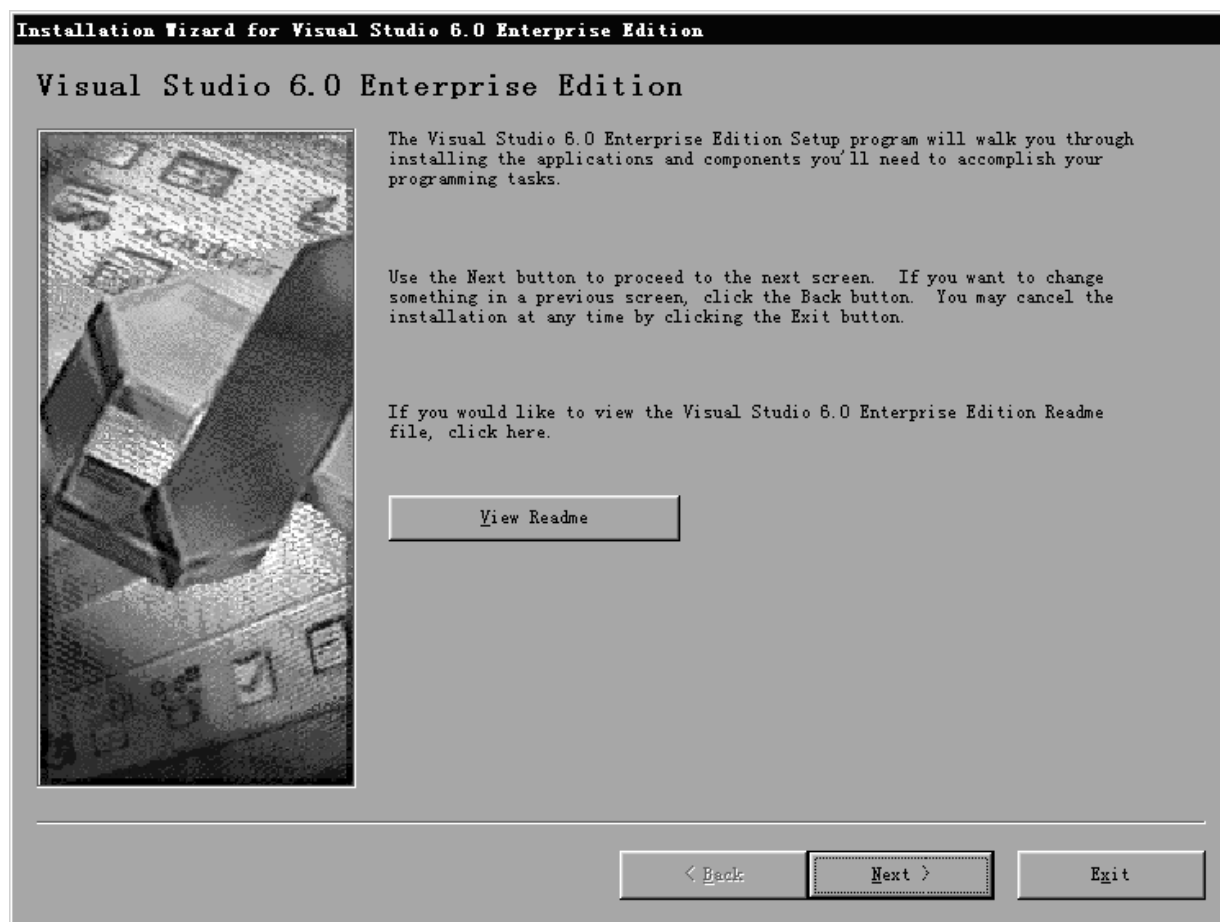


图 1-1 Microsoft Visual Studio 安装初始屏幕

(4) 在用户授权协议中(如图 1-2 所示),选择 I accept the agreement,单击 Next 按钮,在出现的如图 1-3 所示的屏幕中输入序列号、用户名和用户所在单位,单击 Next 按钮,在新出现的安装选择屏幕中选择 Product,即按产品进行安装,然后单击 Next 按钮。进入下一步产品选择。

(5) 在产品清单(如图 1-4 所示)中选择 Visual C++ 6.0 Enterprise Edition 后单击 Next 按钮,在接着出现的版权信息对话框中单击 Continue 按钮。

(6) 在组件清单(如图 1-5 所示)中根据需要,改变缺省的组件选项,建议增加 Graphics

选项,而不修改其他选项。单击 Continue 按钮即开始安装 Visual C++ 6.0,安装完成后,请重新启动 Windows。

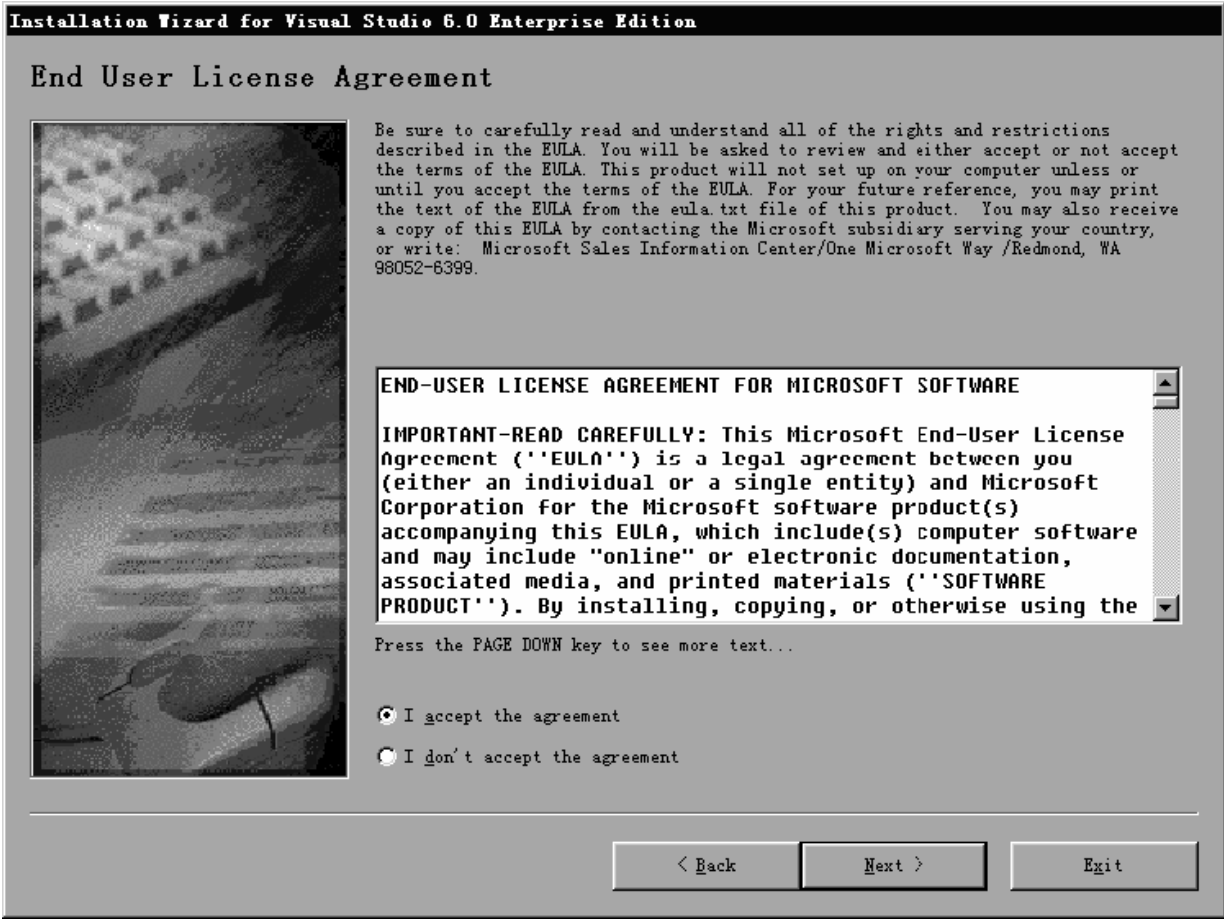


图 1-2 用户授权协议

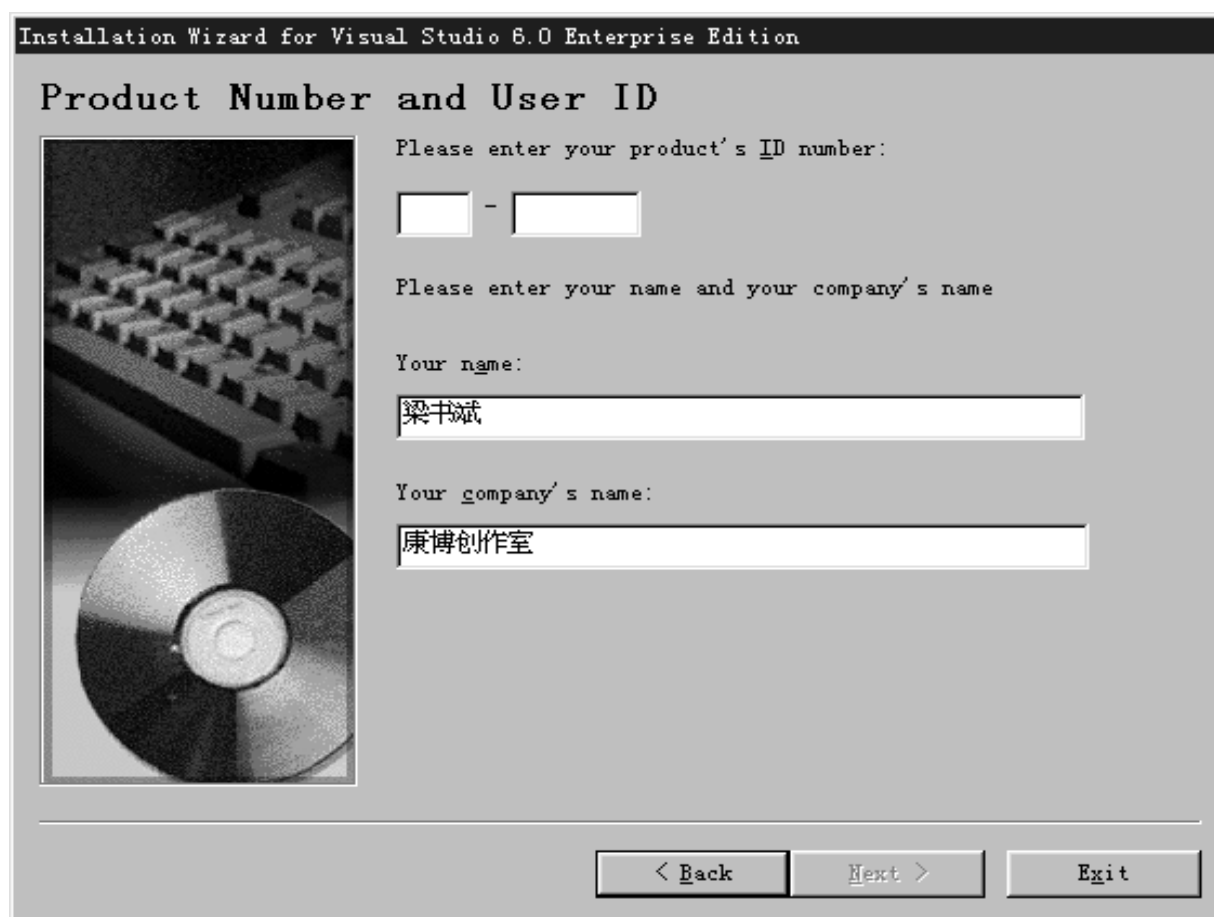


图 1-3 用户授权协议



图 1-4 产品清单

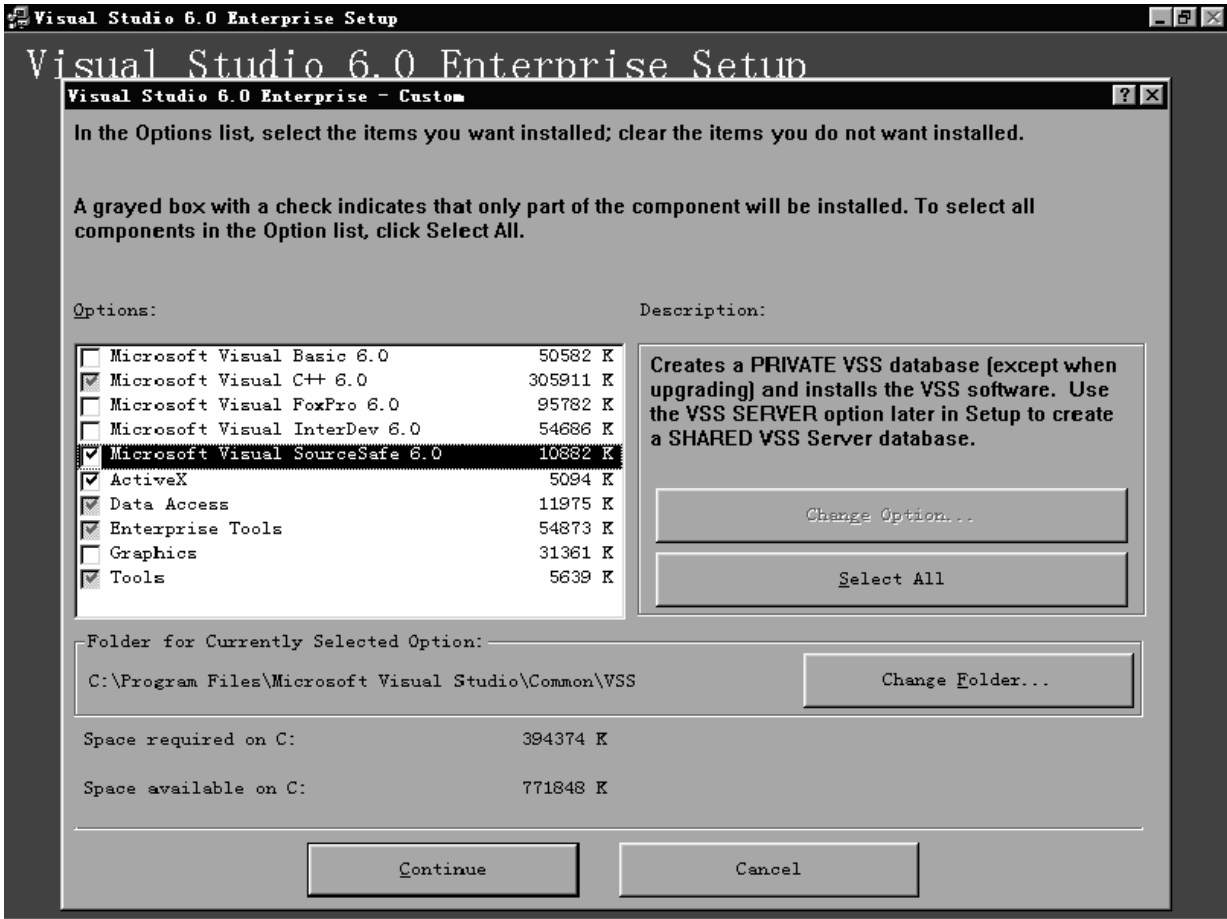


图 1-5 组件清单

注释：

安装前应检查硬盘是否有足够的空间来安装 Visual C++ 6.0, 按照前面的安装选项需要大约 250MB 的硬盘空间。如果安装了更多的产品或组件则需要更多的硬盘空间。

注释：

Visual C++ 6.0 支持多语言, 如果在安装过程中遇到所安装的语言支持程序与系统的不同, 安装程序会询问用户是否替换这些支持程序。

第 2 章

Visual C + + 6 .0 开发环境

Visual C++ 6.0 提供了良好的可视化编程环境,使得在 Visual C++ 应用程序中可进行的各种操作,包括建立、打开、浏览、编辑、保存、编译、链接和调试等,点按鼠标按钮即可进行,故而非常方便。

2.1 Visual C++ 6.0 主窗口

当正确安装 Visual C++ 6.0 到 Windows 系统之后,单击“开始”按钮,指向“程序”(Programs),“Microsoft Visual Studio 6.0”,最后单击“Microsoft Visual C++ 6.0”,即可启动 Visual C++ 6.0,进入集成开发环境 Developer Studio,参见图 2-1 所示。

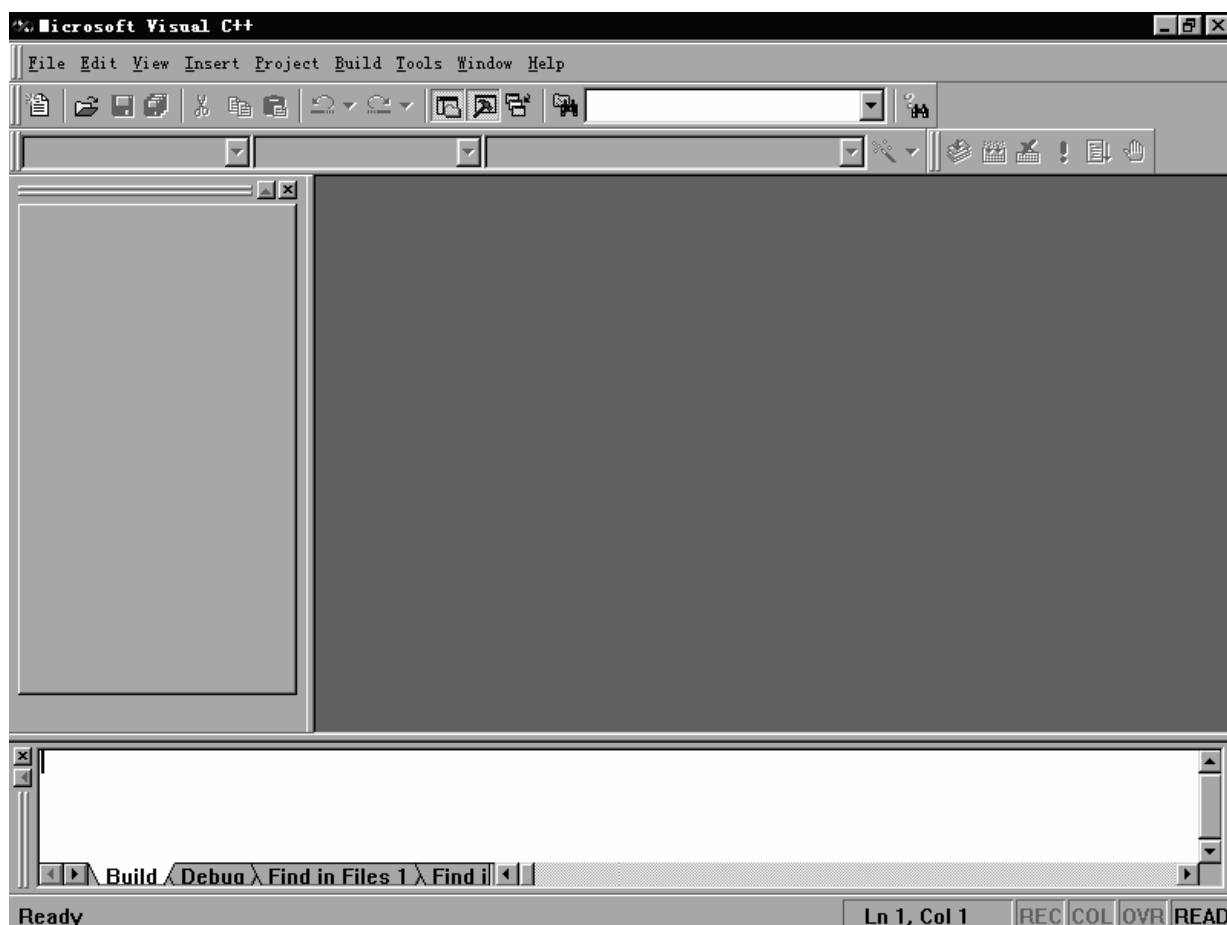


图 2-1 Visual C++ 6.0 开发环境

集成开发环境的主窗口包括标题栏、菜单栏、工具栏、工作区窗口(缺省时处于船坞化状态)、源代码编辑窗口、输出窗口(缺省处于船坞化状态)和状态条。

从上图可以看出,构成 Visual C++ 集成开发环境的主窗口由 3 个子窗口组成,它们分别是:编辑窗口和船坞化的工作窗口及输出窗口。如果试着用鼠标拖动这两个船坞化窗口的边框也可以使它们成为普通的窗口。

下面分别介绍 Visual C++ 6.0 集成开发环境的界面元素。

2.2 Visual C++ 6.0 工具栏

为了使用工具栏,必须使工具栏出现在 C++ 的主窗口中,缺省时,C++ 已经使船坞化的 Standard 和 Build 两个工具栏显示在主窗口中。要使用其他工具栏,只需移动鼠标指向

工具栏的位置,然后单击右键,屏幕上弹出一快捷菜单,单击要使用的工具栏,使其前面出现√,则相应的工具栏就会出现在 C++ 主窗口中。

下面逐一介绍每一工具栏所含的各工具的作用与用法。

2.2.1 Standard 工具栏

Standard 工具栏含有 15 个工具,主要用于建立项目工作区及项目,如图 2-2 所示。



图 2-2 Standard 工具栏

各按钮的具体功能如下:

- New Text File: 创建新的文本文件。
- Open: 打开已有的文档。
- Save: 保存文档。
- Save All: 保存所有打开的文件。
- Cut: 剪切选定的内容到剪贴板中。
- Copy: 复制选定的内容到剪贴板中。
- Paste: 在当前插入点处插入剪贴板中的内容。
- Undo: 取消最后的操作。
- Redo: 重复先前的操作。
- Workspace: 显示或隐藏工作区窗口。
- Output: 显示或者隐藏输出窗口。
- Windows list: 管理当前打开的窗口。
- Find in Files: 在多个文件中搜索字符串。
- Find: 激活查找工具。
- Search: 搜索联机文档。

2.2.2 Build 工具栏

Build 工具栏含有 8 个工具,如图 2-3 所示。当用户建立好应用程序项目后,可利用这些工具对项目进行编译、连接以及调试和运行。



图 2-3 Build 工具栏

各工具按钮的具体功能如下:

- Select Active Project: 选择活动项目。
- Select Active Configuration: 选择活动的配置。Visual C++ 提供了两种活动配置,

它们分别是 Win32 Release(基于 Win32 平台的发行版)和 Win32 Debug(基于 Win32 平台的调试版)。

Compile:编译文件。要编译的文件必须是构成项目的一部分。

Build:建立项目。

Stop Build:停止建立项目。

Execute Program:执行程序。

Go:启动或继续程序的执行。

Insert/ Remove Breakpoint:插入或删除断点。

2.2.3 Build Minibar 工具栏

该工具栏包含有 6 个工具按钮,如图 2-4 所示。其功能分别相当于 Build 工具栏中的后 6 个工具按钮,主要用于建立较小的项目。



图 2-4 Build Minibar 工具栏

2.2.4 Resource 工具栏

Resource 工具栏含有 9 个创建资源的工具和 1 个显示资源状况的工具,如图 2-5 所示。这些工具用来帮助用户创建应用程序所需的各种资源。



图 2-5 Resource 工具栏

下面列出每个工具的名称及其快捷键,具体功能将在本章的 2.5 节中详细介绍。

New Dialog:创建新对话框。

New Menu:创建新菜单。

New Cursor:创建新光标。

New Icon:创建新图标。

New Bitmap:创建新位图。

New Toolbar:创建新工具栏。

New Accelerator:创建新快捷键。

New String Table:创建新串表。

New Version:创建新版本信息。

Resource Symbols:显示资源符号。

2.2.5 WizardBar 工具栏

WizardBar 工具栏含有 4 个工具,如图 2-6 所示。其中前 3 个实际上是列表框,最后一个带缺省动作的列表框。利用该组工具可以方便地建立和修改应用程序中涉及的所有类。下面只列出各工具的名称及简单作用,详细功能描述见本章的 2.5 节。

WizardBar C++ Class: 在此列表框中选择要修改的类。

WizardBar C++ Filter: 在此列表框中选择要过滤的事件。

WizardBar C++ Members: 选择类成员。

Wizard Action(Default): 根据前 3 个列表框中的选择执行不同的操作,缺省时的具体操作因前三者的不同而有所区别。



图 2-6 WizardBar 工具栏

2.2.6 Debug 工具栏

Debug 工具栏含有 16 个工具,如图 2-7 所示。利用这些工具可以调试已开发的项目,以便找到问题所在。只有处于调试运行状态时这一组工具才有效。



图 2-7 Debug 工具栏

Restart: 启动程序的执行,并处于调试状态。

Stop Debugging: 停止调试执行程序。

Break Execution: 中断程序的执行。

Apply Code Change: 认同程序代码的改变,即当用户在调试过程中改变了程序代码后,使用该工具使代码有效。

Show Next Statement: 显示下一条要执行的语句。

Step Into: 进入被调函数内单步执行。

Step Over: 单步执行,但跳过被调用的函数。

Step Out: 从被调用的函数内跳出,继续执行调用语句的下一条语句。

Run to Cursor: 运行到当前光标处。

Quick Watch: 快速查看当前的调试状态。

Watch: 打开一独立窗口,在其内显示用户要查看的变量的值和类型。用户需输入变量名,调试程序自动显示变量类型及其值。

Variables: 打开一独立窗口,在该窗口内有 3 个选项卡,分别显示当前语句和上一条语句所用变量、正在执行函数的局部变量以及 this 指针所指对象的信息。

Registers: 打开一独立窗口,在该窗口内显示 CPU 各寄存器状态。

Memory: 打开一独立窗口,在该窗口内显示内存状态。

Call Stack: 打开一独立窗口,在该窗口内显示当前语句调用的所有函数,当前函数在栈顶。

Disassembly: 打开一独立窗口,在该窗口内显示反汇编代码。

2.2.7 Edit 工具栏

Edit 工具栏包含 8 个工具,如图 2-8 所示。这些工具用于源程序的编辑,提高用户查找和定位信息的效率。



图 2-8 Edit 工具栏

Toggle Bookmark: 在当前光标位置设置或取消书签。

Next Bookmark: 光标定位到下一个书签。

Previous Bookmark: 光标定位到上一个书签。

Clear All Bookmark: 清除所有书签。

Find: 在当前编辑窗口内查找。

Increase Indent: 增加缩进量。

Decrease Indent: 减少缩进量。

Toggle Whitespace Display: 锁定空白显示。

2.2.8 Database 工具栏和 Atl 工具栏

Database 工具栏有 3 个工具,而 Atl 工具栏只有一个工具,如图 2-9 所示。



图 2-9 Database 和 Atl 工具栏

Database 工具栏中的工具用于连接远程数据库服务器的操作。Atl 工具栏中的工具用于向用户项目添加 Atl 对象。

Run: 运行所选择的数据库对象。

Add Data Connect: 向活动项目中添加数据库连接。

Add New Database Item: 向数据库中添加新项目。

New Atl Object: 向当前项目添加新的 Atl 对象。