

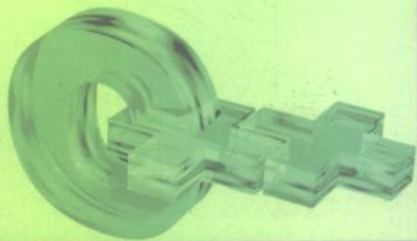
康博创作室 编著 梁书斌 审校

Visual C++ 6.0

高级编程



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



TP312

K19-2

Visual C++ 6.0 高级编程

康博创作室 编著
梁书斌 审校

7

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。

图书在版编目(CIP)数据

Visual C++ 6.0 高级编程/康博创作室编著. —北京:清华大学出版社,1999.5
ISBN 7-302-03507-5

I. V... II. 康... III. C 语言-程序设计 IV. TP312

中国版本图书馆 CIP 数据核字(1999)第 17167 号

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 5 月第 1 版 1999 年 7 月第 2 次印刷

书 号: ISBN 7-302-03507-5/TP·1507

印 数: 6001~14000

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 with 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

目 录

第 1 章 Visual C++ 6.0 概述	(1)
1.1 Visual C++ 6.0 的特点	(2)
1.2 Visual C++ 6.0 的新增功能	(2)
1.3 Visual C++ 6.0 开发环境	(3)
1.4 Visual C++ 6.0 的安装	(3)
第 2 章 Visual C++ 6.0 开发环境	(9)
2.1 Visual C++ 6.0 主窗口	(10)
2.2 Visual C++ 6.0 工具栏	(10)
2.2.1 Standard 工具栏	(11)
2.2.2 Build 工具栏	(11)
2.2.3 Build Minibar 工具栏	(12)
2.2.4 Resource 工具栏	(12)
2.2.5 WizardBar 工具栏	(13)
2.2.6 Debug 工具栏	(13)
2.2.7 Edit 工具栏	(14)
2.2.8 Database 工具栏和 Atl 工具栏	(14)
2.3 Visual C++ 6.0 菜单栏	(15)
2.3.1 File 菜单	(15)
2.3.2 Edit 菜单	(17)
2.3.3 View 菜单	(19)
2.3.4 Insert 菜单	(19)
2.3.5 Project 菜单	(19)
2.3.6 Build 菜单	(19)
2.3.7 Debug 菜单	(20)
2.3.8 Tools 菜单	(22)
2.4 项目与项目工作区	(23)
2.5 资源与资源编辑器	(25)
2.5.1 资源编辑器	(25)
2.5.2 对话框编辑器	(27)

2.5.3	菜单编辑器	(28)
2.5.4	加速键编辑器	(30)
2.5.5	串表编辑器	(31)
2.5.6	版本信息编辑器	(31)
2.5.7	图形编辑器	(32)
2.5.8	工具栏编辑器	(32)
第3章	C++ 编程基础	(35)
3.1	C++ 语言的特征	(36)
3.1.1	一个简单的 C 程序	(36)
3.1.2	一个简单的 C++ 程序	(37)
3.1.3	C++ 对 C 的扩展	(37)
3.1.4	C++ 的面向对象特征	(39)
3.2	C++ 标识符与关键字	(41)
3.2.1	标识符	(41)
3.2.2	关键字	(41)
3.3	C++ 数据类型	(42)
3.3.1	类型 char	(42)
3.3.2	类型 short	(43)
3.3.3	类型 int	(44)
3.3.4	类型 long	(44)
3.3.5	类型 _intn	(45)
3.3.6	浮点类型	(45)
3.3.7	常量	(46)
3.4	数据类型转换	(47)
3.4.1	自动数据类型转换	(47)
3.4.2	强制类型转换	(47)
3.5	C++ 存储类与变量的作用域和可见性	(48)
3.5.1	auto 存储类	(48)
3.5.2	register 存储类	(48)
3.5.3	extern 存储类	(48)
3.5.4	static 存储类	(49)
3.5.5	内部变量和外部变量	(50)
3.5.6	变量作用域总结	(51)
3.6	操作符	(51)
3.6.1	算术运算操作符	(51)
3.6.2	关系运算操作符	(52)
3.6.3	逻辑运算操作符	(53)
3.6.4	位操作符	(53)

3.6.5	增量和减量操作符	(53)
3.6.6	赋值和复合赋值操作符	(54)
3.6.7	条件操作符	(54)
3.6.8	逗号操作符	(55)
3.6.9	操作符的优先级和结合性	(55)
3.6.10	表达式	(57)
3.7	高级数据类型	(57)
3.7.1	枚举类型	(57)
3.7.2	联合(union)	(58)
3.7.3	结构声明	(60)
3.7.4	数组和字符串	(62)
3.7.5	指针	(66)
3.7.6	typedef 声明语句	(68)
3.8	控制结构	(68)
3.8.1	条件语句	(68)
3.8.2	循环语句	(70)
3.8.3	转移语句	(71)
3.9	函数	(71)
3.10	类与对象	(72)
3.10.1	类的定义	(73)
3.10.2	对象的定义	(74)
3.10.3	构造函数与析构函数	(76)
3.10.4	内联成员函数	(78)
3.10.5	this 指针	(79)
3.10.6	静态数据成员和静态成员函数	(80)
3.10.7	友元函数	(81)
3.11	类继承	(82)
3.11.1	派生类与基类	(82)
3.11.2	派生类的对象和构造函数	(84)
3.11.3	多重派生类	(85)
3.11.4	类的多重继承	(87)
3.11.5	多层派生类	(90)
3.12	重载(overload)	(92)
3.12.1	函数重载	(92)
3.12.2	操作符重载	(93)
第 4 章	Visual C++ 6.0 工具程序	(97)
4.1	WinDiff 应用程序	(98)
4.1.1	WinDiff 命令行	(98)

4.1.2 WinDiff 应用程序界面	(98)
4.2 Spy++ 应用程序	(100)
4.2.1 Spy++ 工具条	(100)
4.2.2 Spy++ 的视图	(101)
4.3 Help Workshop 工具	(105)
4.3.1 帮助生成器的构成	(105)
4.3.2 帮助工程文件结构	(105)
4.3.3 创建帮助主题文件	(107)
第 5 章 MFC 基础类库	(109)
5.1 MFC 的特点	(110)
5.2 MFC 的类层次	(110)
5.2.1 CObject 类	(110)
5.2.2 CMenu 类	(112)
5.2.3 文档结构子层次	(113)
5.2.4 边框窗口子层次	(114)
5.2.5 控件栏子层次	(115)
5.2.6 视图子层次	(115)
5.2.7 对话框子层次	(115)
5.2.8 控件子层次	(116)
5.2.9 绘图对象子层次	(116)
5.2.10 CDC 子层次	(117)
5.2.11 其他子层次	(117)
5.3 MFC 的通用集合类	(119)
5.3.1 CString 类	(120)
5.3.2 CTime 类	(120)
5.3.3 CStringArray 类	(121)
5.3.4 CStringList 类	(122)
5.3.5 CMapStringToString 类	(123)
第 6 章 Windows 应用程序开发基础	(125)
6.1 生成 MDI 应用程序	(126)
6.2 向导生成的文件及作用	(133)
第 7 章 边框窗口、文档和视图	(145)
7.1 边框窗口	(146)
7.1.1 边框窗口和视图	(146)
7.1.2 边框窗口与文档边框窗口	(146)
7.1.3 边框窗口的创建和销毁	(146)

7.1.4	管理当前视图	(147)
7.1.5	管理菜单、控件栏和加速键	(147)
7.1.6	定制边框窗口	(148)
7.2	文档与视图	(149)
7.2.1	文档与视图之间的联系	(149)
7.2.2	使用文档管理数据	(150)
7.2.3	在视图中显示数据及与用户交互作用	(150)
7.2.4	滚动与缩放	(151)
7.2.5	多文档类型	(152)
7.2.6	多视图	(152)
7.2.7	分割窗口	(152)
7.3	SDI应用程序编程步骤	(152)
7.3.1	建立应用程序基本框架	(153)
7.3.2	处理视图	(153)
7.3.3	处理文档	(158)
7.3.4	串行化处理	(163)
7.3.5	SDI应用程序编程思路	(165)
7.3.6	完整程序清单	(166)
第8章	绘图程序的实现	(189)
8.1	生成初始的应用程序框架	(190)
8.2	基本概念	(190)
8.2.1	设备描述表	(190)
8.2.2	坐标系统	(191)
8.2.3	绘图工具	(191)
8.2.4	绘图函数	(193)
8.2.5	映射模式	(196)
8.2.6	绘图模式	(199)
8.2.7	设置背景颜色	(200)
8.2.8	设置背景模式	(200)
8.3	文本处理	(201)
8.3.1	文本输出函数	(201)
8.3.2	设置文本属性	(202)
8.3.3	获取字符属性	(203)
8.3.4	使用字体	(203)
8.4	点位图及其操作	(205)
8.4.1	点位图操作常用函数	(205)
8.4.2	点位图的装入	(207)
8.4.3	点位图的显示	(209)

8.5	图标	(209)
8.6	光标	(210)
8.7	打印和打印预览	(211)
8.8	绘图功能的实现	(211)
第9章	对话框和控件	(221)
9.1	对话框	(222)
9.1.1	模态对话框	(222)
9.1.2	非模态对话框	(223)
9.1.3	创建对话框资源	(223)
9.1.4	创建对话框类	(224)
9.1.5	创建并显示对话框	(225)
9.1.6	处理对话框消息	(225)
9.1.7	关闭对话框	(225)
9.1.8	管理对话框的 MFC 函数	(226)
9.2	控件	(226)
9.2.1	标准控件	(227)
9.2.2	附加的控件类	(232)
9.2.3	管理对话框控件的 MFC 函数	(233)
9.3	控件栏	(235)
9.3.1	工具栏	(235)
9.3.2	状态栏	(236)
9.3.3	对话框	(236)
9.3.4	CControlBar 类	(237)
9.4	对话框编程	(237)
9.4.1	创建对话框模板资源	(237)
9.4.2	创建对话框类	(238)
9.4.3	定义成员函数	(239)
9.4.4	定义消息处理函数	(242)
9.4.5	修改对话框类代码	(243)
9.4.6	创建对话框类对象并显示对话框	(245)
9.4.7	创建并运行应用程序	(249)
9.4.8	完整的源程序清单	(250)
第10章	动态链接库	(283)
10.1	什么是动态链接库	(284)
10.2	利用向导开发动态链接库	(285)
10.3	DLL 应用示例(DLL 部分)	(286)
10.4	DLL 应用示例(调用部分)	(290)

10.5 动态链接库与静态链接库应用的进一步比较	(291)
第 11 章 ActiveX 控件及应用	(293)
11.1 ActiveX 控件应用程序的开发	(294)
11.1.1 开发 ActiveX 控件	(294)
11.1.2 事件、方法和属性	(296)
11.1.3 ActiveX 控件与 MFC 类库	(298)
11.2 程序代码及其作用分析	(298)
11.3 定制实现特定功能的 ActiveX 控件	(312)
11.3.1 改变控件的外观	(312)
11.3.2 实现特定事件处理函数	(315)
11.4 测试 MyActiveX 控件	(318)
附录 命令与快捷键	(321)

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 Facilities)、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(Class)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。

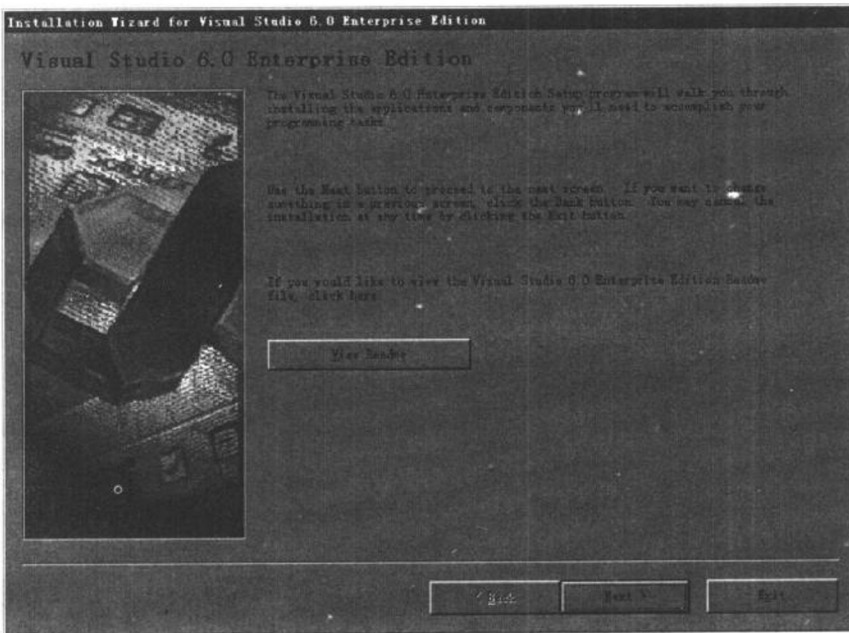


图 1-1 Microsoft Visual Studio 安装初始屏幕

(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。

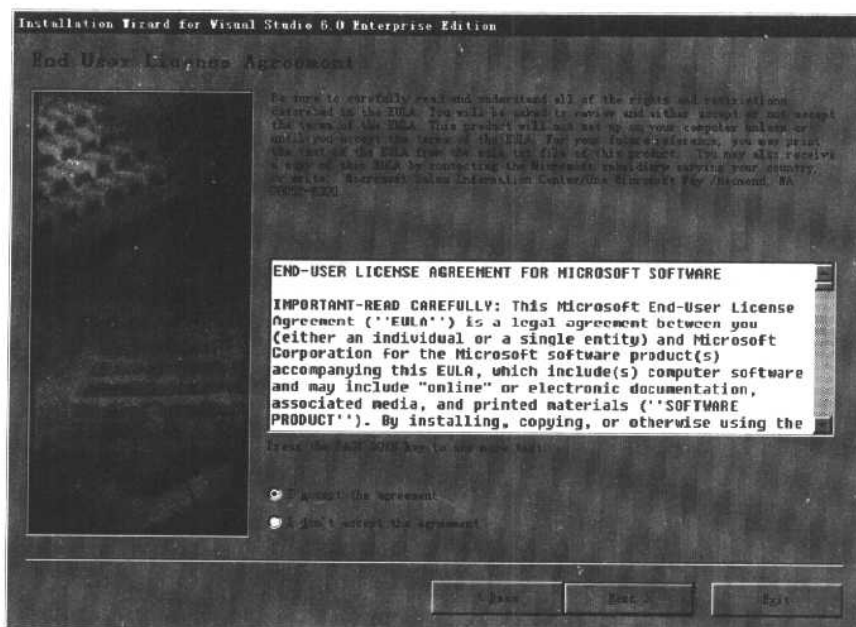


图 1-2 用户授权协议

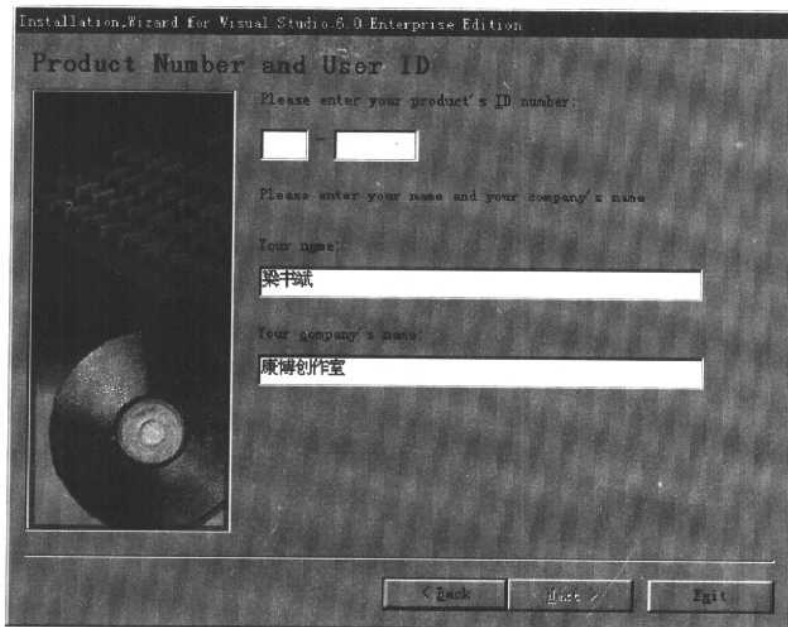


图 1-3 用户授权协议

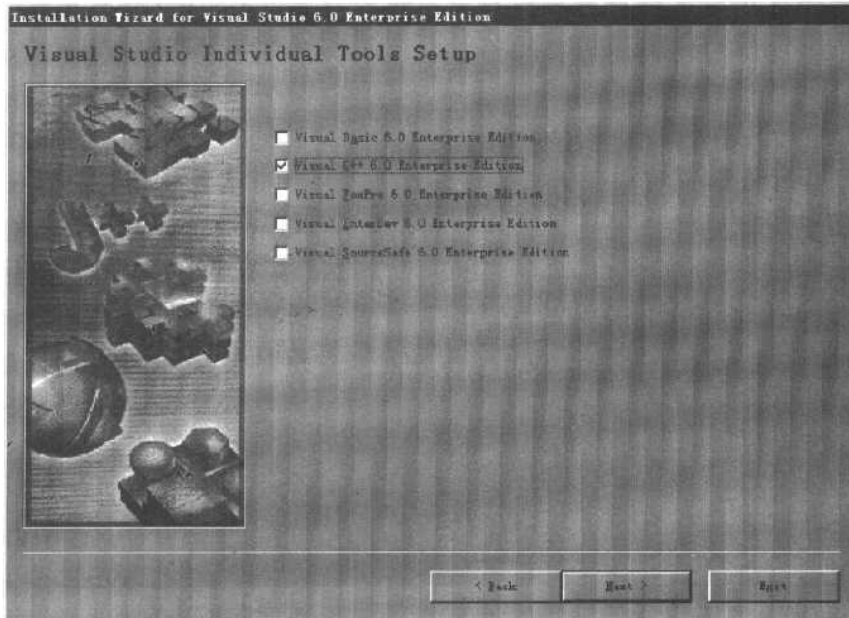


图 1-4 产品清单

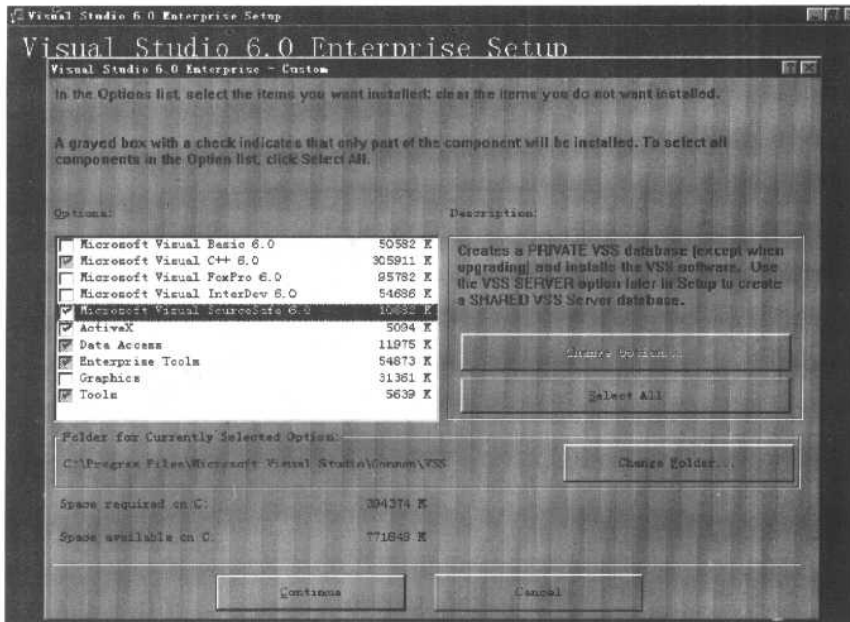


图 1-5 组件清单