

高手速成系列丛书

Microsoft

Visual C++ 6.0

Enterprise Edition



Microsoft

Visual C++ 6.0

高手速成

步行者工作室 编著

兵器工业出版社

高手速成系列丛书

Visual C++ 6.0

高手速成

步行者工作室 编著

兵器工业出版社

内 容 简 介

本书介绍了 Visual C++ 6.0 集成开发环境的使用及 MFC 的概念及使用。提供了大量详细的示例, 汇集了笔者使用 Visual C++ 6.0 的经验。书中使用生动的例子展示了 Visual C++ 6.0 和 MFC 的强大功能, 对于初学者和使用 Visual C++ 6.0 进行商业软件开发的人员都是很好的参考资料。

图书在版编目 (CIP) 数据

Visual C++ 6.0 高手速成/步行者工作室编著. - 北京:
兵器工业出版社, 1999.6 (高手速成系列丛书)
ISBN 7-80132-675-X

I. V... II. 步... III. C 语言 IV. TP312

中国版本图书馆 CIP 数据核字 (1999) 第 11014 号

出版发行: 兵器工业出版社
责任编辑: 罗正云 刘莹
责任技编: 刘燕丽
社址: 100089 北京市海淀区车道沟 10 号
经销: 各地新华书店经销
印刷: 兵器工业出版社印刷厂
版次: 1999 年 6 月第 1 版第 1 次印刷
印数: 1-5000

封面设计: 蒋宏
责任校对: 姚培新
责任印制: 张伟
开本: 787×1092 1/16
印张: 19.5
字数: 478.9 千字
定价: 30.00 元

(版权所有 翻印必究 印装有误 负责调换)

前 言

Microsoft 公司于 1998 年 9 月推出了 Visual Studio 98 系列开发工具, Visual C++ 6.0 是其中的一个, 在这个版本的 Visual C++ 中, 可视化设计开发的概念得到进一步拓展, 融合了 Microsoft 公司其它编译器产品的优点, 使开发者的开发工作变得更有趣、方便, 工作效率也大大提高。

本书主要目的在于介绍 Visual C++ 6.0 的使用以及新增的 MFC 类的使用。同时也用一定的篇幅兼顾介绍一些基础知识与背景知识。

很多读者希望计算机方面的书籍能有一些详细的示例、详尽的解说以及比较高的实用价值, 我们也希望读者在使用本书的时候能获得更多的知识, 能更快地理解先进的计算机技术, 本着这样的想法, 本书主要通过大量的例子对 Visual C++ 6.0 的使用进行解说, 示例在安排上尽量由浅入深、循序渐进, 又能深入浅出。因为篇幅所限, 书中没有给出所有的源程序, 但读者只要按照书中所述步骤一步一步做下去, 按照给出的程序片段正确输入, 最终都能完成本书中的例子。

MFC 编程是一个非常庞大的话题, 仅靠本书很难将其细述一遍, 但本书力争向读者介绍最新的 MFC 知识、技巧, 以适应各个层次的读者的需要。如果读者是初学者, 按照本书中各个示例的步骤从头到尾制作一遍, 相信一定能对 MFC 有一个比较清楚的认识, 对 Visual C++ 6.0 的集成开发环境的使用也能达到得心应手、操作自如的地步。如果读者已经对 MFC 有一些了解, 也使用过 Visual C++ 的集成开发环境, 则本书中介绍的 Visual C++ 6.0 使用, 以及各个章节中的例子、图表, 将使读者能掌握最新的 Visual C++ 6.0 使用技巧和最新的 MFC 类的使用。

计算机编程活动的实践性非常强, 特别是 C++ 的编程, 更需要对计算机语言艺术有深入的了解, 本书希望读者对 C++ 语言能有所了解, 如果先前没有接触过 C++ 语言, 希望读者至少能掌握类、重载、继承等几个面向对象编程常用的概念。

虽然作者已经竭尽所能, 但因时间所限, 书中一定还有不少未能纠正的错误和疏漏, 恳请读者指正。

步行者工作室

1999 年 3 月于清华园

目 录

第一章 初识 VC6.0	(1)
1.1 Visual C++ 6.0 的安装	(1)
1.2 帮助系统	(4)
1.3 Visual C++ 6.0 的最新特性	(6)
1.4 树型浏览窗口	(11)
1.5 文本编辑窗口	(13)
1.6 资源编辑器	(17)
1.7 输出窗口	(21)
1.8 调试窗口	(21)
第二章 使用 Visual C++ 6.0	(25)
2.1 Visual C++ 6.0 的“New”（新建）命令	(25)
2.2 使用 MFC 开发应用程序	(28)
2.3 生成一个基于对话框的应用程序——时钟	(28)
2.4 生成一个单文档应用程序——笑脸	(42)
2.5 生成一个多文档应用程序——更多的笑脸	(75)
第三章 从按钮开始	(86)
3.1 按钮简介	(86)
3.2 按钮基本用法示例	(87)
3.3 在对话框中使用位图按钮	(92)
3.4 在视图（View）中使用位图按钮	(94)
3.5 在按钮中播放动画	(98)
3.6 异形按钮	(107)
3.7 一个实用的自定义按钮类	(113)
第四章 列表框、组合框和编辑框	(125)
4.1 列表框控件类简介	(125)
4.2 在对话框中使用列表控件	(131)
4.3 使用动态创建的列表框	(135)
4.4 手工添加列表框的消息映射入口	(137)

4.5 编辑框控件类简介	(139)
4.6 制作一个小型文本编辑器	(143)
4.7 组合框控件类简介	(151)
4.8 位图组合框	(157)
第五章 图像列表、树视控件与列视控件	(166)
5.1 图像列表控件	(166)
5.2 树视控件简介	(171)
5.3 树视控件的基本用法	(184)
5.4 为树视控件添加拖放支持	(188)
5.5 在位编辑树视控件节点	(193)
5.6 父节点在展开和未展开时显示不同的图像	(194)
5.7 列视控件简介	(196)
5.8 列视控件基本使用	(213)
5.9 列视控件各种视图切换	(216)
5.10 使用状态图标	(219)
5.11 为题头控件加上图标	(220)
5.12 使用叠加图像	(223)
5.13 响应显示信息消息	(225)
5.14 在列视控件中拖放图标	(227)
第六章 工具条和状态条	(230)
6.1 控件条简介	(230)
6.2 工具条控件简介	(231)
6.3 CToolBar 类	(240)
6.4 工具条基本使用	(240)
6.5 为工具条加上按钮文字	(242)
6.6 工具条下拉按钮	(244)
6.7 使用热点图像	(247)
6.8 状态条简介	(248)
6.9 状态条基本使用	(249)
6.10 在状态条上显示图标	(251)
6.11 在状态条上显示进程条	(252)
第七章 CReBar 和 CComboBoxEx	(256)
7.1 CComboBoxEx 简介	(256)
7.2 与扩展组合框相关的结构	(259)
7.3 CComboBoxEx 类基本使用	(261)
7.4 CReBarCtrl 简介	(265)

7.5 rebar 控件的相关结构.....	(268)
7.6 CReBar 简介.....	(272)
7.7 rebar 控件基本使用.....	(273)
第八章 CHtmlView	(280)
8.1 CHtmlView 类简介	(280)
8.2 自己的浏览器	(282)
第九章 使用注册表	(289)
9.1 注册表及 CRegKey 类简介.....	(289)
9.2 注册表编程	(290)
附录 A MFC 类库开发指南	(293)
A.1 概览	(293)
A.2 MFC 类库的设计目标	(293)
A.3 MFC 扩展类库编写指南	(294)
A.4 遵循 MFC 惯例	(298)
A.5 使类有充分的灵活性	(300)
A.6 用 MFC 编写类	(301)
A.7 为用户编写详尽的文档	(303)
A.8 总结	(304)

第一章 初识 VC 6.0

Microsoft 公司于 1998 年推出了 Visual Studio 98 系列, 其中包含全新的 Visual C++ 6.0, 虽然在界面方面与 Visual C++ 5.0 看不出有什么大的差别, 但经过使用之后, 可以发现 Visual C++ 6.0 增加了很多提高开发者工作效率、方便开发者编程的功能。

1.1 Visual C++ 6.0 的安装

插入 Visual Studio 98 的第一张 CD-ROM, 使用系统的 AutoRun 功能或运行 CD-ROM 上的 Setup.exe, 将会出现如图 1.1 所示的安装向导的第一屏。

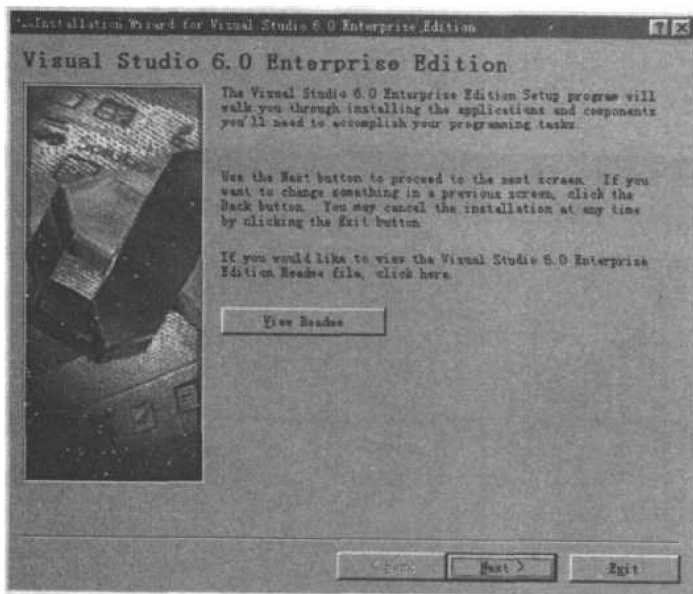


图 1.1 安装向导第一屏

看完 Readme 文件, 按下 Next 按钮, 出现如图 1.2 所示的用户许可协议, 用户许可协议阐述了用户的责任与权利, 如果认可, 选中“我接受协议”(I accept the agreement), 如果不接受, 选中“我不接受协议”(I don't accept the agreement), 将退出安装。

接受协议, 还需要输入产品序列号、确认等步骤, 最后将出现如图 1.3 所示的安装模式选择对话框, 通常情况下选 Custom 定制即可。

下一屏是询问安装路径(如图 1.4 所示), 安装基本的 Visual C++ 6.0 至少需要 350M 的空间。在系统目录中的文件需要占用 50M 的磁盘空间。如果需要安装帮助系统, 则需按安装帮助文件多少来决定还需要多少的磁盘空间。运行 Visual C++ 6.0 所需的磁盘空间比较多, 如果有一个比较大的硬盘, 如 4G 以上, 开发工作会进行得比较顺利。

确定了安装路径后, 出现如图 1.5 所示的定制对话框, 灰色的检查框表示只选中了这个

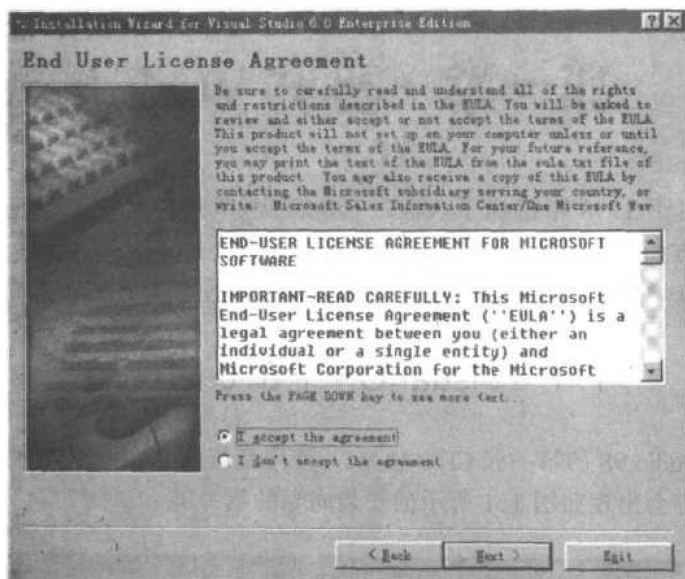


图 1.2 用户许可协议

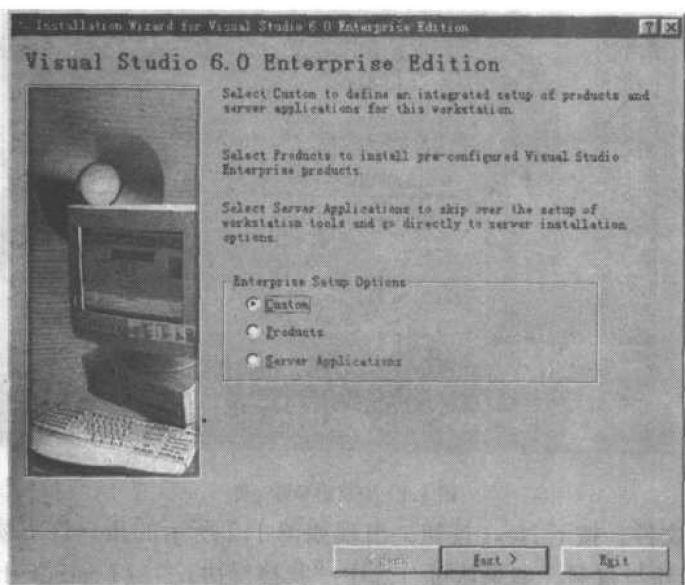


图 1.3 选择安装模式

组件的一部分，如果需要改变选项，可以按右边的 Change Option... 按钮进一步选择这个组件的具体内容。如果只使用 Visual C++ 6.0 进行常用软件的开发，可以选择不安装 Visual Basic、Visual FoxPro、Visual InterDev 和 Visual SourceSafe。

定制完毕，即可按 Continue 进入文件拷贝工序了，在文件的拷贝过程中，会替换系统原有的一些文件，在替换之前，安装程序会提问是否覆盖原文件。如果安装于中文系统上，安装程序会询问是否用英文版文件替换中文版的文件，这时应选择否，然后让安装程序对所有类似的问题都使用同样的处理。继续即可，如图 1.6 所示。

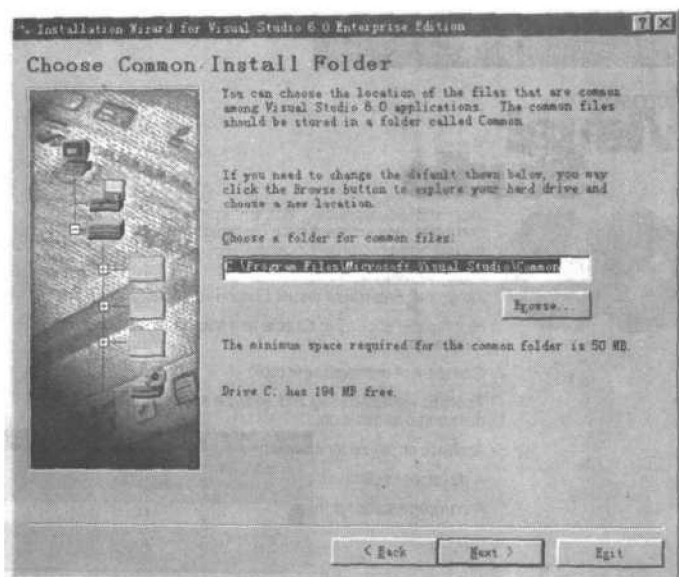


图 1.4 选择安装路径

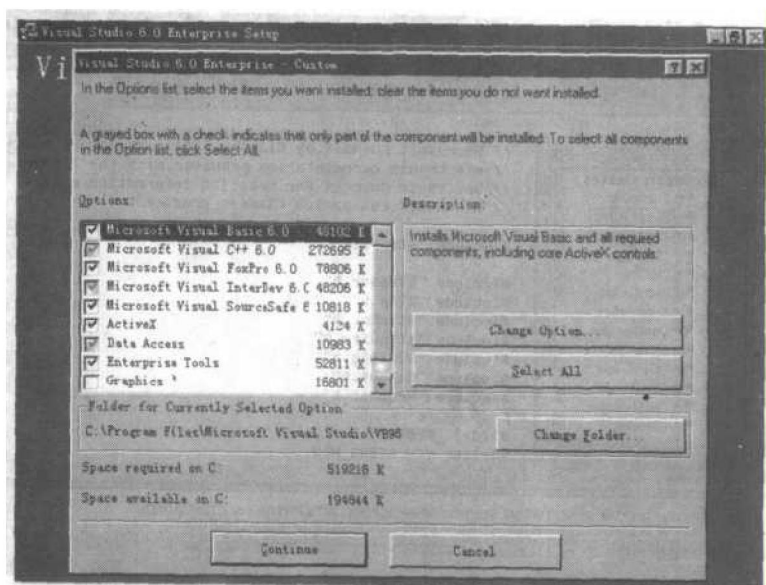


图 1.5 定制对话框

正确安装上 Visual C++ 6.0 以后，进入 Visual C++ 6.0 的集成开发环境，如图 1.7 所示，熟悉 Visual C++ 5.0 的用户会发现它的界面与 Visual C++ 5.0 大同小异。但实际上，这一版本的 Visual C++ 增添了很多先进的功能，这一点在后面的介绍中会体现出来。

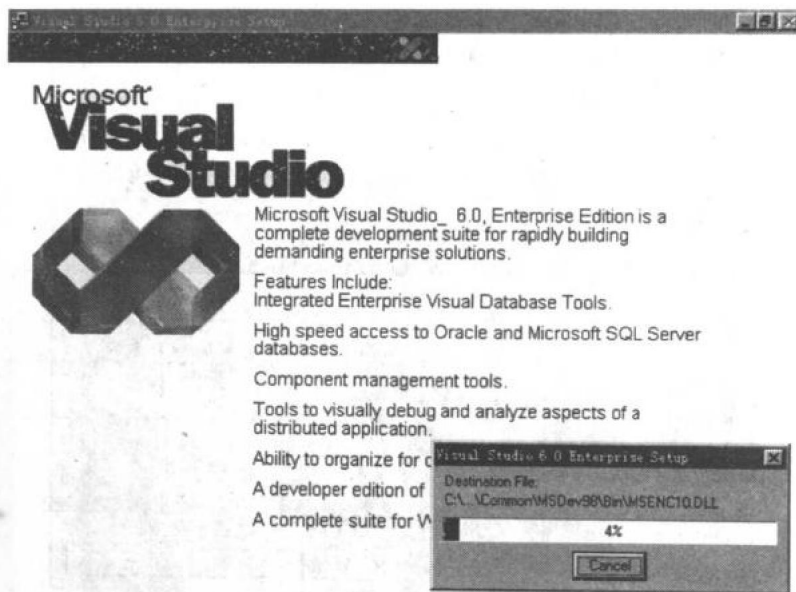


图 1.6 开始安装



图 1.7 Visual C++ 6.0 的集成开发环境界面示例

1.2 帮助系统

与 Visual C++ 5.0 版不同的是, Visual C++ 6.0 的帮助系统 (MSDN) 并不集成在集成开发环境 (IDE) 里, 而是做为一个单独的软件出现的, 需要另外购买, 完整的 MSDN 是两张 CD-ROM。所以在左边的树型浏览窗没有 Visual C++ 5.0 使用者所熟悉的 In-foviewer 栏, 但在正确安装了 MSDN 以后使用方法还是和 Visual C++ 5.0 一样的: 将光标

停在需要寻找帮助的文字上，或者拖动鼠标使文字高亮显示，然后按 F1 键，将出现一个帮助窗口——MSDN (Microsoft Developer Network)，如图 1.8 所示，左边窗口中显示的是目录、索引、搜索和书签，右边的窗口显示的就是所选中的关键字帮助的正文。这样的安排，一是可以在单独的窗口中浏览帮助信息，不必像在 Visual C++ 5.0 中那样，在一个集成环境下切换窗口；二是可以借助于微软新推出的 CHM (Compiled Html) 帮助文件的强大功能，使查询帮助更方便。

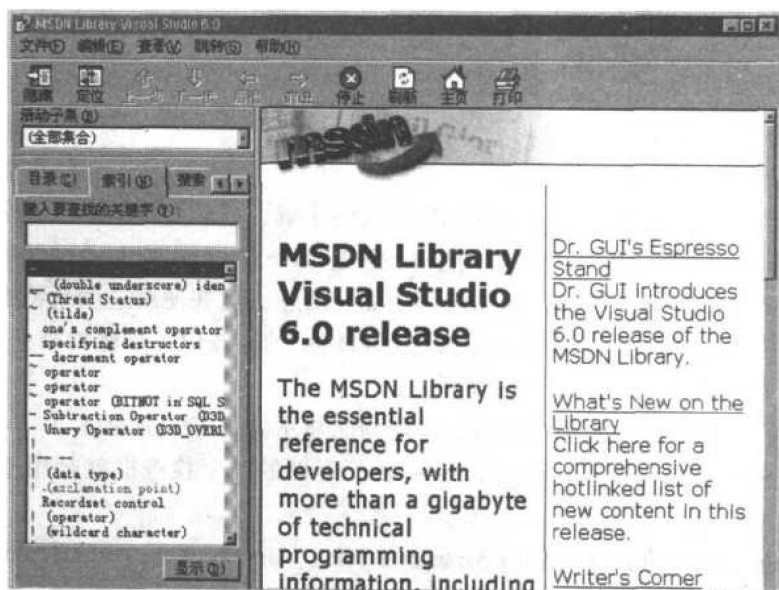


图 1.8 MSDN 窗口

Visual C++ 6.0 的帮助文件增加了大量实用价值很高的内容。

- 微软出版的 Inside OLE 2.0 完全版是一本公认对 OLE 技术阐述得最详实的著作，这一个版本实际上介绍的就是现在微软正大力宣传，并正在各个领域如办公软件、Internet 网页等广泛使用的 ActiveX。

- 值得大力推荐的是 110M 的 Knowledge Base。Knowledge Base 中有大量的微软官方问题解答，涵盖了使用 Visual C++ 6.0 进行软件开发的方方面面，内容包括编程技巧、Bug 汇报及解决，如果在开发过程中遇到疑难问题，可以先使用搜索功能在 Knowledge Base 中搜索相关的内容，通常都能得到正面的答案或相关的线索。具体方法是在活动子集中选 MSDN Books and Periodicals，然后在“键入要查找的单词”中键入欲取得帮助的单词，搜索结果将会按查找到的顺序出现在主题窗里，即可浏览查看。左下方还有三个选项：搜索上一次结果、匹配相似的单词、仅搜索标题。这些选项可以用于更深入、精确的查找。

Visual C++ 6.0 的在线帮助系统非常庞大，之所以在讲述 Visual C++ 6.0 的集成开发环境之前先介绍帮助系统，是希望本书的读者能学会自己寻找答案、自己解决问题，掌握了获得信息的方法，就相当于拥有了金手指，依靠自己的能力，就能点石成金、攻克难关。

下面介绍 Visual C++ 6.0 的集成开发环境，先介绍 Visual C++ 6.0 中常见的窗口。

1.3 Visual C++ 6.0 的最新特性

Visual C++ 6.0 的新特性可以使开发者的软件开发活动变得轻松、有趣，大大提高程序员的生产效率。主要体现在下面几个方面：

- 自动列出类成员 Auto list members
- 自动参数信息 Auto parameter info
- 代码注解 Code comments
- 自动类型信息 Auto type Info
- 补全单词 Complete Word
- 即改即运行 Edit and Continue

其中，自动列出类成员、自动参数信息、代码注解、自动类型信息、补全单词等特性总称为智能感应 (IntelliSense)，这个技术能将 MFC、Win32 以及 ATL 等一些库的资料、信息根据开发者的键入情况自动调出，具体的使用如后所述，但在使用这些新特性之前，首先要保证 Visual C++ 6.0 中这些功能是被设置为可以使用的，按以下步骤检查：

- (1) 选择主菜单上的 Tools，选择 Options
- (2) 弹出如图 1.9 所示的选项窗口，确定当前显示的是 Editor 标签。
- (3) 找到 Statement completion options 一栏，下面的四个检查框就是用于控制新特性的使用与否。
- (4) 将 Auto list members、Auto parameter info、Auto type info、Code comments 四个检查框全都选上，按 OK 退出即可

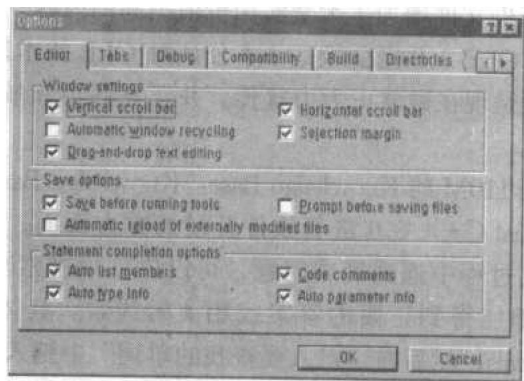


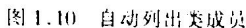
图 1.9 启用新特性

1. 自动列出类成员 Auto list members

在源程序的编辑过程中，编辑器会主动分析程序员键入的字符，如果程序员键入了一个小数点，若当前输入的单词不是一个浮点数，就会在名字空间 (Namespace) 中搜索小数点前面的单词，如果是一个类、结构或联合等的标识符，编辑器会弹出一个类成员、结构成员或联合成员的滚动列表。若是类，列表中也包括这个类的基类成员。程序员可使用上下箭头移动选择所需要的成员函数、数据成员。

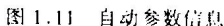
同样，如果输入的是指针操作符 “->” 或作用域操作符 “::”，情况也是一样的。

如果需要查看全局变量、函数等，可以在一个空行上按 Ctrl+Alt+T，滚动列表就会出现，其中显示的包括系统 API 函数、C++ 类、实例变量和局部变量。自动列出类成员如图 1.10 所示。



在源代码的输入过程中 如果程序员转

自动参数信息如图 1.11 所示。



自动列出成员时 如果在成

自动列出成员时，如果在成员函数、变量的定义文件中，成员函数、变量定义后面有注解，在编辑器列出成员函数的同时，当前高亮的成员函数的注解也会显示在一个工具提示窗口中，最多可以显示 12 行注解文字。这些注解有些时候对程序员来说是很具有指导意义的。比如指明某些函数的限制、是否过时等，特别是在开发一个大的工程项目时，这些问题是不

可避免的, 如果每个程序员在开发过程中都得回头去看自己或别人写的头文件, 则工作效率就很难得到提高。在使用 Visual C++ 6.0 的过程中, 程序员应该养成多写注解的好习惯, 以提高整个工作组的工作效率。

注解最好写在类定义文件中函数、变量声明的后面, 并在声明的同一行上, 便于 Visual C++ 6.0 查找。

代码注解如图 1.12 所示。

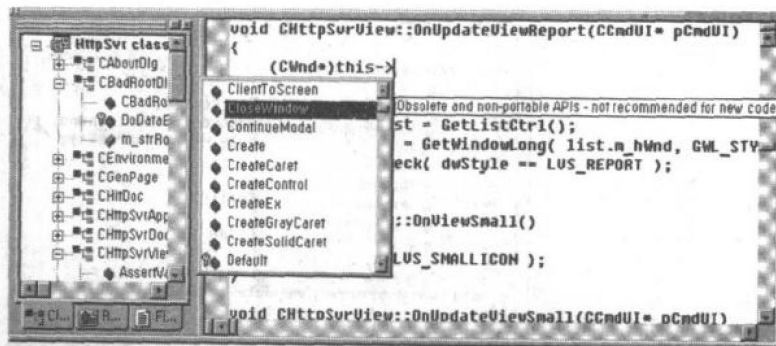


图 1.12 代码注解

4. 自动类型信息 Auto type info

在一个比较长的程序中, 变量、函数出现的地方可能距定义它的地方比较远, 这时若需要得到变量、函数的类型信息, 通常情况下就要使用查找功能进行搜索, 有了自动类型信息的功能, 只要将鼠标放在变量、函数名上方, 停留几秒钟, 将键盘光标放置于标识符中间, 按 Ctrl+T, 变量、函数的定义就会出现在一个工具提示中。

自动类型信息如图 1.13 所示。

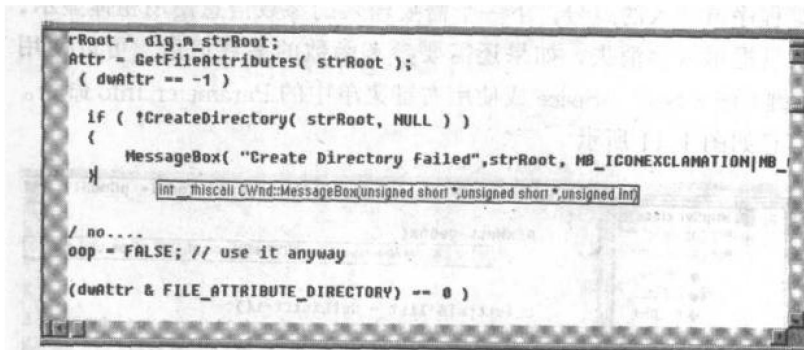


图 1.13 自动类型信息

5. 补全单词 Complete Word

源代码的书写力求逻辑性强、可读性好, 在很多情况下, 如果使用比较长的标识符来标识变量或函数, 可以使变量或函数的意义更加明显、直观, 比如 m-bIsThisConnectCanceled, 看起来就要比 m-bIsConCan 等要明了得多。但标识符比较长带来的明显缺点就是增加了程序员的工作量, 需要多键入不少字符, Visual C++ 6.0 带来的这个新特性很好地解决了这个问题, 只要键入标识符的头几个字母, 在这些字母的后面按下鼠标右键, 从弹出的菜单中

选择 Complete Word (如图 1.14 所示), 即可自动补全其余的单词, 如果有多个以这同样的字符串开头的标识符, 则会弹出一个类似于 Auto list members 的小窗口, 程序员可以从中选择正确的标识符。

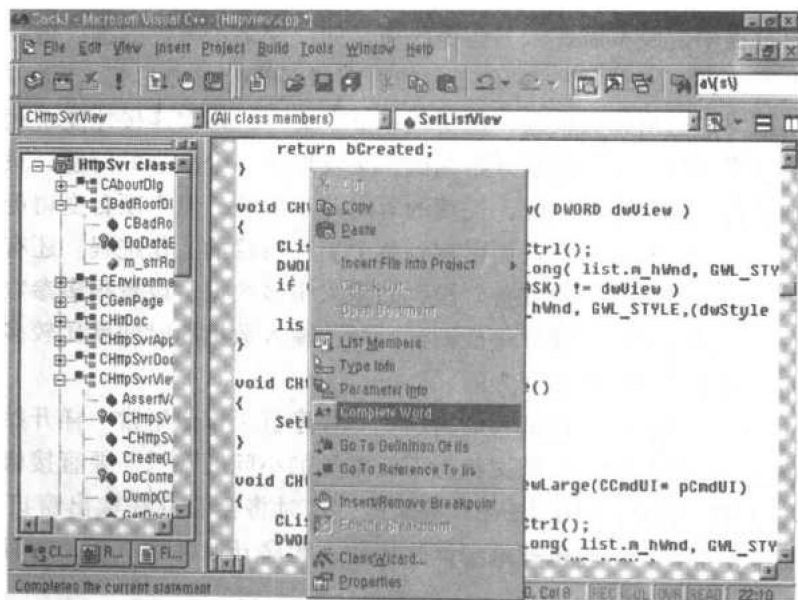


图 1.14 补全单词

此功能非常实用, 但每次都要使用鼠标右键, 如果使用键盘, Visual C++ 6.0 内部预设的快捷键是 Ctrl+Space, 正好与切换中英文输入法的快捷键冲突, 有些不便, 推荐的做法是自定义一个快捷键, 以解决这个问题。具体步骤如下:

- (1) 从主菜单中选择 Tools。
- (2) 选择 Customize。
- (3) 选择 Keyboard (如图 1.15 所示)。

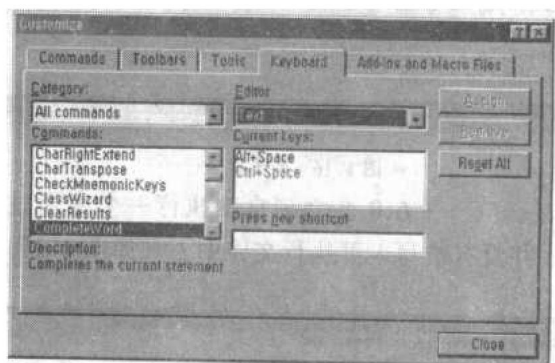


图 1.15 自定义补全单词的快捷键

- (4) 确定左上角的 Category 当前选项是 All Commands。
- (5) 确定中上方的 Editor 当前选项是 Text。
- (6) 从 Commands 中找到 CompleteWord。

(7) 将光标置于 Press New Shortcut 中, 按下需要的快捷键组合, 如 Alt + Space。

(8) 按右上角的 Assign 按钮, 退出此对话框即可。

6. 即改即运行 Edit and Continue

Visual C++ 6.0 最大的进步就是加入了即改即运行 (Edit and Continue) 的特色。程序员在调试程序的过程中, 经常需要做一些小改动, 如改变某个颜色的预设值, 改变一个判断语句的判断条件等, 在 Visual C++ 6.0 之前的版本和别的公司的 C/C++ 语言产品中, 这样一些小小的改动都要求退出调试状态, 重新编译整个文件, 再次连接才能继续调试。常见的情况是一个小问题解决了, 刚刚开始调试, 又发现了一个小小的问题, 只好重新编译连接, 如果正在进行的只是一个练习时, 耗费的时间并不显著, 但当项目变得庞大的时候, 重新编译就需要耗费比较多的时间, 进而影响到整个工程项目的工作进程。还有一种情况是正在进行的调试工作需要比较多的准备阶段, 如输入很多数据、设置很多参数, 需要比较长的时间运行到需要调试的地方, 这样每次调试时耗在输入数据的时间就比较多。而每次都在对话框中输入同样的内容也是非常烦人的。

有了 Visual C++ 6.0, 这一切都将成为历史。程序员可以像平常一样开始调试, 但是不需要在修改满意以后重新编译, 在调试状态下, 常见的小问题都只需要直接修改代码, 然后继续让调试器进行 F5 (Go)、F10 (Step) 等调试, 这时将会看见在输出窗口中文件正在重新被编译 (如图 1.16 所示)。在迅速地编译连接后, 程序开始继续执行, 所有的改动都将成为现实。

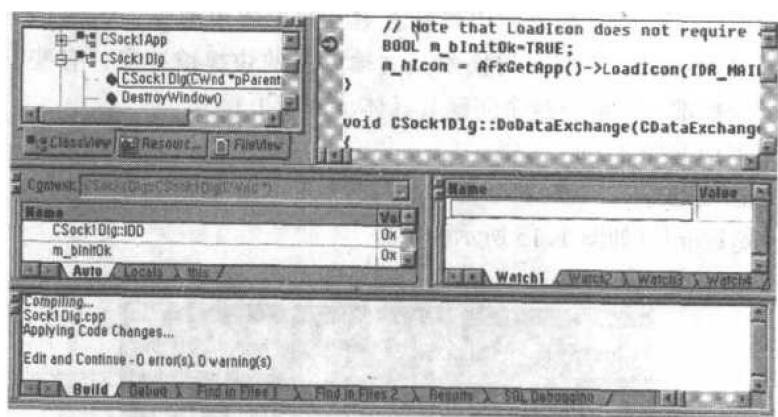


图 1.16 即改即运行

退出调试状态后, Visual C++ 6.0 会立刻自动执行一个编译过程, 因为此功能对程序的改变只保留在内存中, 并没有被写入可执行文件中 (如图 1.17 所示)。

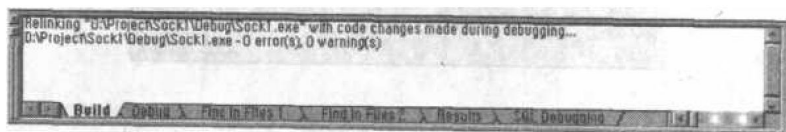


图 1.17 退出调试后重新连接

但如果这个功能没有办法正常工作, Visual C++ 6.0 会给出一个提示信息。这时就需要退出调试器重新编译文件。如果有可能继续进行下去, 就可以节约很多时间。