

简介

本书有两个主要目的：第一是教你用 C++ 语言编程，第二是教你使用 Turbo C++ for Windows 建立 Windows 应用程序。它不要求你以前有任何编程经验。但是，如果你知道用其他语言编程，比如 BASIC 或 Pascal，那么对你的学习肯定会有所帮助的。本书不是为懦弱的读者而编写的，因为学会用 C++ 语言编程和用 C++ 语言编写 Windows 应用程序并不是两件很简单的事情！

这本书共包含二十一章，你每天可以学习一章，因为该书的目的就是从某种程度上加快你学习的速度。书中的每一章都包含一个问与答章节、一个测验章节和一个练习章节。附录 B 包含了所有测验的答案和一些练习的答案。

第一天简要地介绍了开发 C++ 程序所需要的 Windows 环境—Turbo C++ IDE，同时还向你展示了第一个 C++ 程序，用来说明一个非 Windows C++ 程序的基本组成部分。

第二天更详细地介绍 C++ 程序的各部分。这章讨论了变量、常量和函数的命名与说明，并且文中首先集中说明了 C++ 函数，因为它们是最重要的程序构件。

第三天介绍各种 C++ 运算符和表达式。运算符使你能够操作数据，形成支持更复杂的数据操作的表达式。

第四天讨论格式化的流输入与输出，以及著名的 printf 函数。后面的这个函数支持各种用途的格式化输出。

第五天描述了 C++ 的判断结构。这些结构包括了各种 if 语句，以及 switch 语句。

第六天讨论 C++ 循环语句，它们包括 for, do-while, while 循环语句，并且还说明了如何使用 for 循环语句作为一个开循环。另外，本章讨论了如何跳过循环，退出循环和嵌入循环。

第七天介绍 C++ 中的数组。这章包含了一维数组和多维数组，并且讨论了如何说明和初始化它们。另外，该章还讨论了对一维数组的排序与搜索。

第八天讨论了用户自定义类型和指针。这章包含了枚举数据类型、结构、联合、参量和指针。本章说明了如何声明和使用简单变量、数组、结构及动态内存的指针。

第九天集中在字符串和从 C 语言中继承的 STRING.H 库。这章包括了像分配、连接、比较、转换及颠倒字符串这样的主题。另外，该章还讨论了如何在字符串中寻找字符和子串。

第十天讨论高级函数参数，主要包括数组、字符串、结构及指向函数的指针的参数。这章还讨论了把结构作为参数传递的各种方法，并且介绍了递归函数。

第十一天向你介绍了面向对象编程(OOP)的世界。这章介绍了 OOP 的基础和 C++ 的类。文中讨论了 C++ 类的基本组成部分，以及与使用这些部分相关的规则。

第十二天讨论基本 I/O 流文件，它是由 C++ 流库所支持的。这章包含了普通的流函数、顺序文本 I/O 流、顺序二进制 I/O 流及随机访问 I/O 流。

第十三天向你介绍使用 ObjectWindows 库(OWL)编写 Windows 应用程序。这章包括了广泛应用的 Windows 数据类型,以及 OWL 的简要概述、Windows API 函数和 Windows 的消息。

第十四天介绍非常简单的、基于 OWL 的 Windows 应用程序。这章从提供一个最小型的 OWL 程序开始,然后由它逐渐引伸到菜单,以及对鼠标点取的响应。

第十五天讨论在一个窗口中描绘文本。这章介绍了不滚动窗口和滚动窗口,并说明了如何在窗口中绘制文本(像图形那样)。

第十六天介绍 OWL 库的类,它们是静态文本控制、编辑控制和按钮控制的模型。这章还介绍了一个特殊的、面向命令行的计算器,作为使用这些控制的一个例子。

第十七天介绍作为复选框控制、单选按钮控制和成组控制模型的 OWL 库类。这章显示了如何组织这些控制,并且介绍了计算器应用程序的更新版。

第十八天讨论作为列表框模型的 OWL 库类。这章讨论了单选项列表框和多选项列表框。同时,该章的程序也说明了这两种列表框。

第十九天介绍作为滚动条控制和组合框控制模型的 OWL 库类。这章还讨论了如何使用组合框建立过去事情的记载框。另外,该章还介绍了使用组合框的计算器应用程序。

第二十天介绍建立和使用对话框。这章向你显示了如何使用资源文件来定义原型对话框和非原型对话框。另外,该章还讨论了对话框和它的父窗口之间的数据传输。

第二十天介绍多文本界面(MDI)窗口。这章介绍了支持 MDI 应用程序的类,并说明了如何管理 MDI 子窗口。

附录 A 介绍了用来创建菜单和对话框的资源描述语句。附录 B 包含了所有测验的答案和一些练习的答案。

注意:本书包含了讲解各方面编程的 Windows 程序,它超越了平常使用的各种可视化控制的方面。一定要认真研究这些程序;它们包含了可以丰富你的 Windows 编程的技术和诀窍。通过察看这些例子(包括特殊的例子)并向朋友请教其中不懂的问题

第一周浏览

学习编写 Windows 应用程序旅程的第一周是以介绍 Turbo C++ 环境——IDE(集成开发环境)开始。在这一周的剩余的几天里向你介绍 C++ 语言的基础。你可以学习到预定义的数据类型、常量、变量和函数的命名、C++ 运算符和表达式、基本输入和输出管理、判断、编写循环、以及说明和使用数组。因此,这周覆盖了 C++ 语言的所有基本组成部分。

第一章 入门(第一天)

欢迎你进入 C++ 和 Windows 编程世界。今天开始进入了那激动人心的冒险旅程。今天的课程的大部分信息是使你熟悉 Turbo C++ 集成开发环境 (IDE)。你将会学到下面的内容:

- ☐ C++ 程序的基础和历史。
- ☐ 如何装载和使用 Turbo C++ IDE。
- ☐ IDE 中的各种菜单。
- ☐ EasyWin 应用程序。
- ☐ 输入和运行你的第一个 C++ 程序。

1.1 C++ 程序基础

用本书学习 Turbo C++ 编程,不需要任何预备知识;但是,如果你以前用 Turbo C++ 编过程序,那么,一切事情就很简单了。像其他语言一样,C++ 也是由说明和语句组成的,这些说明和语句在程序运行时,指定了正确的执行指令。

C++ 语言是由贝尔实验室的 Bjarne Stroustrup 开发的。该语言试图取代目前流行的 C 语言,并在 C 语言基础上通过增加面向对象的语言向最高峰发展。

新术语: 面向对象(object-oriented)语言表示对象的属性和操作。

另外,C++ 语言对不是面向对象的 C 语言作了大量的增强工作。然而,学会 C++ 语言也有助于你对 C 语言的进一步了解。但是,并不是像标准 C 语言一样,C++ 语言还要经受标准化的过程。

用 C++ 编程要求你必须了解那些支持库,它们用来完成各种不同的任务,比如输入/输出、文本操作、数学运算、文件 I/O(输入/输出)等等。而在像 BASIC 这样的语言中,对这样一些操作的支持显而易见是由程序实现的,也就是这些程序本身能够自动完成这些操作。所以,许多程序相互并列作为组成部分,它们独立于任何其他的编程部分。相反,用 C++ 编程使你更加了解一个程序对各种库的依赖性。这种语言特征的优点是,你可以在类似的库—包括你自己开发的库—之间进行选择。所以,C++ 程序是模块化的。C++ 编辑器—包括 Turbo C++—使用工程文件和程序文件。而 Turbo C++ IDE 使用工程文件来处理程序的创建和更新。

新术语: 工程文件(project files)指定库。程序文件(program files)创建应用程序。

1.2 装载 Turbo C++ IDE

Turbo C++ IDE 是作为 C++ 编译器、连接器、调试器、以及其他用来创建、管理和维护 C++ 程序工具的可视化界面。你可以通过简单地单击 Turbo C++ 图标或者从 File Manager 中双击 TCW.EXE 程序来装载 IDE (文件 TCW.EXE 放在 \TCWIN\BIN 目录中)。

1.3 Turbo C++ IDE 概述

Turbo C++ IDE 是从属于 MDI 的应用程序,它有以下几个主要组成部分:

- ☐ 带有菜单系统,最小化图标和最大化图标的边框窗口。你可以缩放、移动、最大化和最小化 Turbo C++ IDE 窗口。该窗口有一个反映活动窗口名称的标题栏。
- ☐ 系统菜单提供了大量的选项。
- ☐ 速度条,包含了指定的位图按钮,用来提供特定命令的快捷键。IDE 能够使你在速度条中拥有专用的位图按钮。另外,这些按钮对上下文都起作用。它们的数字和类型可以改变,这依赖于当前的任务或活动窗口。IDE 支持良好的特性,当你把鼠标移动到一个位图按钮上时,能够显示该按钮做什么用(正文出现在状态行中)。
- ☐ 用户区域,包含了各种窗口,例如源代码编辑窗口、消息窗口、变量观察窗口等等。
- ☐ 状态行位于 IDE 窗口的底部。该行在你把鼠标移动到速度条中的按钮上时,显示出简要的在线帮助,为不同的菜单项提供简要说明,显示光标位置,以及显示插入/重写模式的状况。

图 1.1 显示 Turbo C++ IDE 的界面样本。

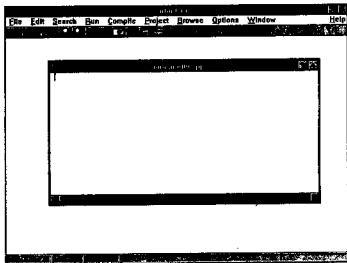


图 1.1 Turbo C++ IDE

注意:因为 IDE 目的是提供给软件开发者,所以如果你是一个初学者,那么许多选项可能对你来说是很高级的。但是,当你变的更有经验,这些选项和术语将成为你作为一个 Turbo C++ 程序员的部分知识了。

1.4 File 菜单

File 菜单提供了管理文件、打印文本和退出 IDE 的命令。表 1.1 摘要介绍了 File 菜单中的这些命令。同时,File 菜单还包含了最近打开的源文件的动态列表。

表 1.1 File 菜单中的命令摘要

命 令	快 键	功 能
New		打开一个新的编辑窗口。
Open...		把一个存在的源文件装入新的编辑窗口。
Save		存入活动编辑窗口的内容。
Save as ...		使用一个新的文件名存入活动编辑窗口的内容。
Save all		把所有打开的源代码窗口存入到它们各自的文件中。
Print...		打印源代码窗口的内容。
Print setup		设置打印机。
Exit	Alt+F4	退出 IDE。

1.4.1 New 命令

New 命令用来打开一个新的编辑窗口(也称作源代码窗口),并给它指定一个缺省的相关文件名。你打开第一个新窗口的缺省文件名是 NONAME00.CPP 你打开的第二个新窗口的缺省文件名是 NONAME001.CPP。新打开的窗口开始是空的,它具有与上次的活动窗口同样的尺寸和位置。换句话说,如果上次的活动窗口是最大型的,那么新窗口也将是最大型的。

1.4.2 Open... 命令

Open...命令使你能够把一个存在的源文件的内容装入到一个新的编辑窗口。事实上,IDE 能装载多个文件。该选项首先弹出 Open a File 对话框,见图 1.2 所示。对话框含有几个列表框的组合框控制,它们能使你定位于某个源文件,然后选择它。这些控制允许你选择驱动器,目录和文件名列表,帮助你确定所寻找的源文件的位置。

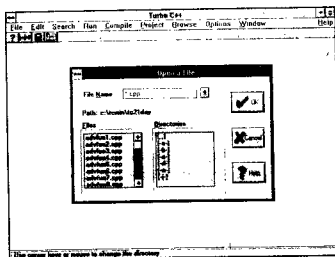


图 1.2 Open a File 对话框

1.4.2 Save 命令

Save 命令帮助你把活动编辑窗口的内容存入到它的一相关的文件里。如果你在新编辑窗口中激活这个选项,那么 Save 选项就会弹出 Save File As 对话框,见图 1.3 所示。这个对话框能够使你有选择地指定某个非缺省的文件名,同时还包括目的驱动器和目录。

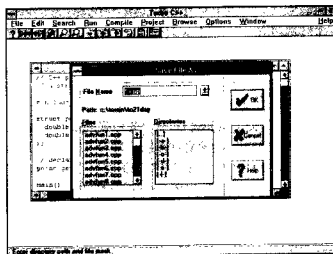


图 1.3 Save File As 对话框

1.4.3 Save as ... 命令

Save as ... 命令能够使你把活动编辑窗口的内容存入到与当前文件名不同的文件

中。事实上,新的文件名也就变成了活动窗口的新的文件。Save as ...命令弹出 Save File As 对话框,见图 1.3 所示。如果你选择一个存在的文件,那么该选项就会弹出一个消息对话框,询问你是否希望用活动编辑窗口的内容重写那个存在的文件。

1.4.4 Save all 命令

Save all 命令把所有修改过的编辑窗口的内容写入到它们各自相关的文件中。如果 IDE 含有新的编辑窗口,那么这个选项就会弹出 Save File As 对话框,存储这些新的窗口内容。

1.4.5 Print ...命令

Print ...命令能够使你打印活动编辑窗口的内容。

1.4.6 Print Setup ...命令

Print Setup ...命令能够使你在使用 Print...选项进行打印之前,首先设置好你的打印机。

Print Setup ...命令弹出 Setup 对话框,见图 1.4 所示(在这个图中的对话框是基于拥有 HP LaserJet II 系统)。这个对话框含有使你能够指定下列项目的控制。

- ☐ 纸张大小。
- ☐ 纸源。
- ☐ 打印的页数。
- ☐ 打印机内存的大小。
- ☐ 打印输出定位。
- ☐ 选择的字体库和字体。
- ☐ 保留额外内存用于打印一页的页保护。这一选项只有当你拥有 1M 以上的打印机内存时才能有效。

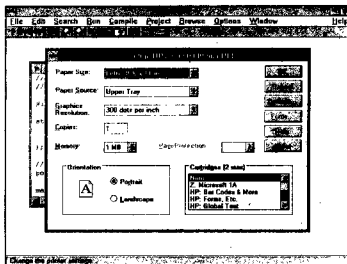


图 1.4 Setup 对话框

1.4.7 Exit 命令

Exit 命令能够使用户完全退出 Turbo C++ IDE。IDE 提示用户不能保存任何已修改编辑的窗口。

1.5 Edit 菜单

Edit 菜单含有使你能够在编辑窗口进行文本编辑的命令。表 1.2 摘要介绍了在 Edit 菜单中的命令。

表 1.2 Edit 菜单中的命令摘要

命 令	快 键	功 能
Undo	Alt + BkSp	撤消上次的编辑。
Redo	Alt + Shift + BkSp	保留上次 Undo 选项的编辑。
Cut	Shift + Del	删除选中的文本,并把它拷贝到剪裁板上。剪裁板以前的内容丢失。
Copy	Ctrl + Ins	拷贝选中的文本到剪裁板。剪裁板以前的内容丢失。
Paste	Shift + Ins	把剪裁板的内容插入到当前光标位置。
Clear	Ctrl + Del	删除选中的文本,但是不把它写到剪切板上。

1.5.1 Undo 命令

Undo 命令能够使你保存上次编辑任务的效果,并恢复活动窗口的内容。这个选项的快捷键是 Alt + BkSp。该选项使你能够快速而有效地处理编辑错误——尤其是在工作很长时间以后。

1.5.2 Redo 命令

Redo 命令能够使你保留 Undo 选项的活动。它的快捷键是 Alt + Shift + BkSp。Redo 命令能够使你在两种形式的编辑源代码之间进行转换。该命令对那些真正已经精疲力尽,不能决定源代码应该是怎样的程序员是很有益处的。

1.5.3 Cut 命令

Cut 命令用来删除选中的文本,并把放置到剪裁板上。剪裁板以前的内容被丢失掉。Cut 命令的快捷键是 Shift + Del。

1.5.4 Copy 命令

Copy 命令把选中的文本复制到剪裁板上。而剪裁板以前的内容被丢失。Copy 命令的

快捷键是 Shift+Ins。

1.5.5 Paste 命令

Paste 命令把剪裁板的内容插入到当前光标位置。而剪裁板的内容仍保持不受影响。所以你可以使用 Cut 和 Paste 命令,在同样的编辑窗口或交叉不同的编辑窗口中移动文本,你也可以使用 Copy 和 Paste 命令,在同样的编辑窗口或交叉不同的编辑窗口中复制文本块。Paste 命令的快捷键是 Shift+ Ins。

1.5.6 Clear 命令

Clear 命令清除选中的文本,并且不把它复制到剪裁板上。这并不意味着被删除的文本将完全丢失,因为你可以使用 Undo 命令来恢复被删的文本。Clear 命令的快捷键是 Ctrl+Del。

1.6 Search 菜单

Search 菜单含有使你能够定位各种信息的命令,比如正文、符号定义、函数说明和程序建立出错的信息。表 1.3 摘要介绍了 Search 菜单中的命令。

表 1.3 Search 菜单中的选项摘要

命 令	快 键	功 能
Find...		寻找活动编辑窗口中的文本
Replace...		代替活动编辑窗口中的文本。
Search again	F3	重复上次的 Find 或 Replace 操作。
Go to line number		立到某上指定行。
Previous error	Alt+F7	选择先前程序建立的出错信息,并把光标定位在编辑窗口中的出错行上。
Next error	Alt+F8	选择下一个程序建立的出错信息,把光标定位在编辑窗口的出错行上。

1.6.1 Find... 命令

Find ...命令支持在活动编辑窗口中寻找文本。这个选项的快捷键是 Ctrl+QF,它能弹出 Find Text 对话框,见图 1.5 所示。这个对话框具有下面的控制:

- ☐ Text to find 组合框控制,它能使你或者在寻找文本中嵌入,或者调用最近寻找过的文本。
- ☐ Options 复选框,包括下面几项:
 - ☐ Case Sensitive 复选框,能够使你选择相关型的或者相关型的文本搜寻。
 - ☐ Whole Words only 复选框,能够使你在匹配全部或者匹配字任何文本之间进行

选择。

- ☐ Regular expression 复选框, 打开或关闭 BRIEF 编辑器的正规表达特征的使用。这样的表达式将造成使用 Text to find 控制中的文本作为正式文本模型

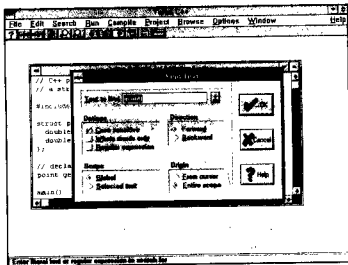


图 1.5 Find Text 对话框

- ☐ Direction diamond-shaped 单选钮控制。这些控制能够使你选择向前搜寻或者向后搜寻。
- ☐ Scope diamond-shaped 单选钮控制。这些控制能够使你在搜寻全部文本或者限制搜寻选中文本之间进行选择。
- ☐ Origin diamond-shaped 单选钮控制。这些控制能够使你在搜寻整个编辑窗口或者从光标的位置开始搜寻之间进行选择。
- ☐ OK, Cancel 和 Help 按钮。

1.6.2 Replace...命令

Replace...命令支持在活动编辑窗口中进行文本替换。这个选项拥有快捷键 Ctrl+QA, 它能弹出 Replace Text 对话框, 见图 1.6 所示。这个对话框含有下面的控制:

- ☐ Text to find 组合框控制, 它能够使你直接敲入要寻找的文本或者调用最近寻找的文本。
- ☐ New text 组合框, 它能够使你直接输入用来代替的文本或选择最近使用过的替换文本。
- ☐ Options 复选框包括下面几项:
 - ☐ Case sensitive 复选框, 它能够使你选择相关型的或非相关型的文本搜寻。
 - ☐ Whole words only 复选框, 它能够使你在匹配全部字或者匹配任何文本之间进行选择。
 - ☐ Regular expression 复选框, 打开或关闭 BRIEF 编辑器的正规表达形式的运用。

这样的表达式将造成使用 Text to find 控制中的文本作为文本模型。

- ☐ Prompt on replace 复选框,使你能够选择是否不经你确定就进行文本替换。
- ☐ Direction diamond-shaped 单选钮控制。这些控制允许你选择是向前搜寻还是向后搜寻。
- ☐ Scope diamond-shaped 单选钮控制。这些控制使你能够在搜寻全部文本或者限制搜寻选中的文本之间进行选择。
- ☐ Origin diamond-shaped 单选钮控制。这些控制使你能够选择,是搜寻整个的编辑窗口还是从光标位置开始搜寻。
- ☐ Change All 按钮,它能够使你替换所有的匹配文本。相反,如果你单击 OK 钮,那么你将只能替换下一个匹配的文本。
- ☐ OK, Cancel 和 Help 按钮。

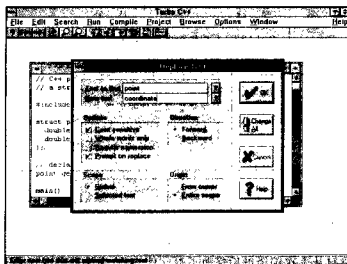


图 1.6 Replace Text 对话框

1.6.3 Search Again 命令

Search Again 命令使你能够重复上次的 Find... 或者 Replace... 选项。该选项的快捷键是 F3 功能键。

1.6.4 Go to Line Number 命令

Go to line number 命令能弹出简单的 Go to Line Number 对话框,它允许输入行数或者从输入组合框中选择一个以前输入的行数。

1.6.5 Previous Error 命令

Previous error 命令使你能够缩放出错的源文件行,该行是与先前在 Message 窗口中的出错信息相联系的。IDE 通过显示含有该出错行的编辑窗口来反映这个选项。这个命令

的快捷键是 Alt+F7。

1.5.6 Next Error 命令

Next error 命令使你能够缩放与 Message 窗口中的下一个出错信息相联系的出错源文件行。IDE 通过显示含有该出错行的编辑窗口来反映这个选项。这个命令的快捷键是 Alt+F8。

1.7 Run 菜单

Run 菜单提供了运行你的程序和启动 Turbo Debugger for Windows 的命令。表 1.4 摘要介绍了 Run 菜单中的命令。

表 1.4 Run 菜单的命令摘要

命 令	快 键	功 能
Run	Ctrl+F9	编辑和运行当前的工程文件。
Arguments		允许你从 IDE 内输入命令行数。
Debugger		启动 Turbo Debugger for Windows, 以便你能调试你的程序。
Debugger arguments		允许你从 IDE 内输入调试器参数。

1.7.1 Run 命令

Run 命令用来运行你的程序,它可以使用由 Run | Arguments 命令传递给它的任何参数。并且如果你自从上次编译以后修改了源文件,Run 命令就会激活 Project Manager,重新编译和连接的程序。该命令的快捷键是 Ctrl+F9。

1.7.2 Arguments 命令

Arguments 命令能够弹出 Program Arguments 对话框,你可以在运行你的程序之前输入命令行参数。IDE 能够准确地处理这些参数,就好像你已经在 DOS 命令行上输入了它们。

1.7.3 Debugger 命令

Debugger 命令能够激活 Turbo Debugger for Windows,以便你能调试你的程序,并且确定任何逻辑或运行错误的位置。Turbo C++ IDE 可以通知 Turbo Debugger 要调试哪个程序。

1.7.4 Debugger Arguments 命令

Debugger arguments 命令能够弹出 Debugger Arguments 对话框,它允许你把参数输

入到调试器。

1.8 Compile 菜单

Compile 菜单提供了编译活动窗口中的程序,以及建立你的工程文件的命令。表 1.5 摘要介绍了 Compile 菜单的命令。

表 1.5 Compile 菜单的命令摘要

命 令	快 键	功 能
Compile	Alt+F9	把活动文件编译成 .OBJ 文件。
Make	F9	更新工程文件。
Link		连接工程文件。
Build All		重新建立工程文件,不管它们是否更新。
Information		显示当前文件的信息。
Remor Messages		撤除 Message 窗口。

1.8.1 Compile 命令

Compile 命令用来把活动的源文件(C 或者 .CPP 文件)编译成 .OBJ 文件。这时,IDE 会弹出一个 Compile Status 信息框,显示出编译进展情况和结果。当编译过程结束时,单击 OK 按钮。如果发生错误,IDE 就会显示出 Message 窗口,并把第一个错误高亮度显示。

1.8.2 Make 命令

Make 命令能够激活 Project Manager 来制立且标工程文件。列出的 .EXE 文件名是由两个名称衍生出来的,这两个名称的顺序如下:

1. 由 Project | Open Project 命令指定的工程文件(.PRJ)。
2. 活动编辑窗口中的文件名。

如果没有定义工程文件,那么你就会获得由文件 TCDEFW.DPR 定义的缺省工程文件。Make 命令可以只重新建立那些当前的文件。

1.8.3 Build All 命令

Build All 命令重新建立用户工程文件中的所有文件。这个命令与 Compile | Make 命令相似,只是除为它能重新建立当前工程文件中的所有文件或非当前的文件以外。如果你通过按 Esc 键或单击 Cancel 按钮取消 Build 命令,或者如果你出错而停止了建立,那么你可以通过激活 Compile|Make 命令来恢复继续建立。

1.8.4 Link 命令

Link 命令用来连接当前工程文件中定义的文件,或者缺省工程文件中定义的文件。

1.8.5 Information 命令

Information 命令弹出 Information 对话框,显示出有关当前文件的统计信息。

1.8.6 Remove Message 命令

Remove message 命令从 Message 窗口中清除所有的信息。

1.9 Project 菜单

Project 菜单提供了创建、打开或关闭一个工程文件的命令,增加或删除一个工程文件中的文件的命令,以及察看工程中一个指定文件所包含的文件的命令。表 1.6 摘要介绍了 Project 菜单的命令。

表 1.6 Project 菜单的命令摘要

命 令	快 键	功 能
Open project		打开一个新的或已存在的工程文件,并自动关闭当前的工程文件。
Close project		关闭当前的工程文件。
Add item		添加一个文件到工程文件列表中。
Delete item		从工程文件列表中删除一个文件。
Include files		显示出一个列有所包含文件的对话框。

1.9.1 Open Project 命令

Open project 命令显示出 Open Project File 对话框,你可以在其中选择和装载一个存在的工程文件,或者创建一个新的工程文件。Open Project File 对话框与 Open File 对话框相似,只是除了前者的对话框专用于打开工程文件以外。

1.9.2 Close Project 命令

Close Project 命令用来撤消你当前的工程文件,而返回到缺省的工程文件 TCDEF、DPR。

1.9.3 Add Item 命令

Add Item 命令弹出 Add to Project List 对话框,你可以在其中把一个文件增加到工程文件列表中。Project List 对话框与 Open File 对话框相似。

1.9.4 Delete Item 命令

Delete item 命令用来删除 Project 窗口中选中的文件。当 Project 窗口是活动的时候，一个文件，然后按下 Del 键，删除那个文件。

1.9.5 Include Files 命令

Include files 命令弹出 Include File 对话框，见图 1.7 所示。这个对话框显示出你从 Project 窗口选择的那个文件所包含的文件列表。

如果你没有建立一个工程文件，那么 IDE 就使 Include files 命令无效。

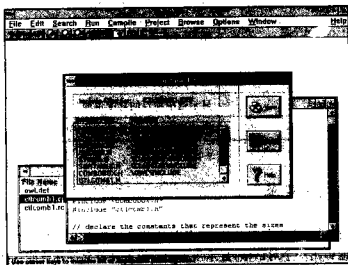


图 1.7 Include File 对话框

1.10 Browse 菜单

Browse 菜单提供了激活 ObjectBrowser 的命令，ObjectBrowser 允许你浏览各级类、函数和变量。一定要记住在你使用 ObjectBrowser 以前，需要首先编译你的程序。如果你编译的程序是由多种源文件组成，那么你在使用 ObjectBrowser 以前，必须打开 IDE 中相关的工程文件。表 1.7 摘要介绍了 Browse 菜单的命令。因为这些命令相对都是比较高级的，所以我显示的这个表将作有限的讨论。