



精华版编程高手时尚丛书

Visual C++ .NET

编程基础专家教程

陈坚 主编
陈鸿飞 孙志月 参编

高级程序员教你编程，成为编程高手不是梦



西安电子科技大学出版社

<http://www.xduph.com>

✦ 精华版编程高手时尚丛书

Visual C++.NET 编程基础专家教程

陈 坚 主编
陈鸿飞 孙志月 参编

西安电子科技大学出版社

2002

内 容 简 介

本书详尽介绍了使用 Visual C++.NET 设计 Windows 应用程序的各种技术,并通过丰富的范例帮助读者迅速掌握这一技术。

本书分七章介绍了 Windows 编程的基本技术和应用,主要包括 AppWizard、AppStudio 和 ClassWizard 编程工具,基本类库以及如何使用 AppWizard 生成应用程序框架和用户界面设计技术(包括对话框、上下文菜单、工具条、状态条、对话条、通用控件、图形图像显示等)。最后通过一个完整的综合实例来帮助读者更好地掌握本书介绍的编程技术。

本书主要面对初、中级读者,适用于熟悉 C 语言、掌握基本 C++ 知识、了解 C++ 继承特性的读者。另外,熟悉 Microsoft 公司的 Windows SDK、Visual Basic 或 Borland 公司的 Delphi、C++ Builder 编程的程序员都能从本书中找到感兴趣、有用的章节。

图书在版编目(CIP)数据

Visual C++.NET 编程基础专家教程 / 陈坚主编. — 西安: 西安电子科技大学出版社, 2002.4
(精华版编程高手时尚丛书)

ISBN 7-5606-1118-4

I. V… II. 陈… III. C 语言 - 程序设计 - 教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 013427 号

策 划 李惠萍 毛红兵

责任编辑 马晓娟

出版发行 西安电子科技大学出版社 (西安市太白南路 2 号)

电 话 (029) 8227828 邮 编 710071

<http://www.xduph.com>

E-mail: xdupfxb@pub.xaonline.com

经 销 新华书店

印 刷 西安兰翔印刷厂

版 次 2002 年 3 月第 1 版 2002 年 3 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 14

字 数 332 千字

印 数 1~4 000 册

定 价 18.00 元

ISBN 7-5606-1118-4/TP·0565

XDUP 1389001-1

如有印装问题可调换

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

前 言

Visual Studio.NET (又称为 Visual Studio 7.0) 是微软继 Visual Studio 6.0 后推出的又一代可视化开发工具。所谓 .NET 就是指 .NET 框架 (.NET Framework), 是一种用于构建、配置和运行 Web 服务和应用程序的多语言环境。

Visual C++.NET 是 Visual Studio.NET 的重要组成部分之一, 是 Microsoft 公司推出的目前使用极为广泛的基于 Windows 平台的可视化编程环境。相对于 Visual C++ 6.0 在 ATL、DCOM、MFC、数据库等方面, 功能有很大增强, 尤其在开发环境界面上变化很大, 采用了全新的平面化操作界面, 具有很强的亲和性。

“Visual”就是“可视的”, Visual C++ 就是可视编程语言。利用 Visual C++ 可以简化程序员的编程, 提高工作效率。这样说主要是基于以下几个方面。

- 利用 AppWizard 可以简单快速地生成一个应用程序框架, 而不需要程序员加入一句程序代码。
- 利用 AppStudio 资源编辑器可以使用户界面的创建简单直观, 所见即所得。
- 利用 ClassWizard 可以使消息或命令自动与消息处理函数或控制函数相联系, 程序员只需考虑函数核心代码的实现。

Visual C++ 贯穿了面向对象的程序设计思想, 其核心是 Microsoft 基本类库 (Microsoft Foundation Classes, MFC), 称之为“应用程序框架”。它一方面用类封装了 Windows API, 另一方面使用称为“消息映射”的机制把 Windows 消息和命令传递到窗口、文档、视图以及 MFC 应用程序中的其它对象。因而 MFC 成功地将面向对象和事件驱动编程概念联系了起来, 并得到了很好的配合。使用 MFC 编写 Windows 应用程序简单方便, 代码量小。

本书分为七章:

第一章介绍了 MFC 应用程序结构、编程工具以及 MFC 类的层次结构。

第二章介绍了 AppWizard 生成应用框架的步骤以及自动生成的各文件的内容。

第三章介绍了 Windows 极为重要的用户接口——对话框的构成、种类、创建和实现的整个过程。

第四章介绍了另外两种非常重要的用户接口——菜单和控制条。其中特别介绍了提供简单快捷操作的上下文菜单, 它在 Windows 中得到广泛的应用。此外还介绍了一种全新的控制条——对话条的设计技术。

第五章介绍了 Windows 通用控件特性与编程。

第六章介绍了作为多媒体技术核心的图形图像显示处理技术。

第七章首先介绍了文档模板、拆分窗口和“画中画”等另外一些深层次的内容, 然后通过一个完整的综合实例来帮助读者更好地掌握前面各章介绍的编程技术。

10/10/05

为了帮助读者迅速掌握介绍的内容，本书给出了丰富的范例程序。所有程序全部经严格测试，无需调试。读者完全可以即拿即用，从而节省了大量宝贵的时间，提高了工作效率。另外，有关 Visual C++.NET 的高级编程技术，诸如数据库、TCP/IP、Internet 编程技术等可参见西安电子科技大学出版社出版的《Visual C++.NET 深入编程与实例剖析》。

从本书的酝酿到编写，自始至终都得到西安电子科技大学出版社李惠萍老师的关心和指导，在此表示衷心地感谢。

限于作者学识水平有限和时间仓促，书中难免有不足和错误之处，恳请各位同仁及广大读者给予批评指正。欢迎将对本书的任何指教或宝贵意见通过 E-mail 发送给作者，以便作者在再版时改进。

作者的 E-mail 地址是：chinj@163.net，网址是：<http://chinj.yeah.net>。

作 者
2002 年 3 月

目 录

第一章 Visual C++和基本类库 1	
1.1 Visual C++重要特性..... 1	
1.1.1 功能高度集成的用户接口..... 1	
1.1.2 编译调试性能优越..... 3	
1.1.3 应用程序的开发效率高..... 4	
1.2 MFC 框架、文档和视图结构..... 5	
1.3 MFC 编程工具及其相互关系..... 7	
1.3.1 AppWizard..... 7	
1.3.2 AppStudio..... 7	
1.3.3 ClassWizard..... 8	
1.3.4 MFC 应用程序开发过程..... 9	
1.4 MFC 应用程序调试技术..... 10	
1.4.1 Visual C++内置的调试器..... 10	
1.4.2 TRACE 宏..... 10	
1.4.3 ASSERT 宏..... 11	
1.4.4 VERIFY 宏..... 12	
1.4.5 消息框..... 12	
1.5 MFC 类库层次结构..... 12	
1.5.1 基类..... 14	
1.5.2 应用体系结构类..... 14	
1.5.3 窗口、对话框和控件类..... 15	
1.5.4 绘图打印类..... 18	
1.5.5 简单数据类型类..... 19	
第二章 AppWizard 和应用框架 20	
2.1 使用 AppWizard 生成应用程序框架..... 20	
2.1.1 选择文件类型..... 20	
2.1.2 项目名和项目类型..... 20	
2.1.3 AppWizard 选项一——指定应用程序类型..... 21	
2.1.4 AppWizard 选项二——指定复合文档选项..... 22	
2.1.5 AppWizard 选项三——指定文档模板字符串选项..... 23	
2.1.6 AppWizard 选项四——指定数据库选项..... 23	
2.1.7 AppWizard 选项五——指定用户界面选项..... 24	
2.1.8 AppWizard 选项六——指定应用程序其它选项..... 24	
2.1.9 AppWizard 选项七——指定类名和对应的文件名..... 25	
2.2 AppWizard 生成的文件..... 26	
2.2.1 自述文件..... 27	
2.2.2 项目文件..... 27	
2.2.3 应用程序源文件和头文件..... 27	
2.2.4 资源文件..... 44	
2.2.5 预定义的头文件..... 45	
2.2.6 帮助文件..... 45	
2.2.7 按可选项增加的 AppWizard 文件..... 45	
第三章 控件与对话框 47	
3.1 Windows 标准控件..... 47	
3.1.1 静态控件..... 47	
3.1.2 编辑控件..... 48	
3.1.3 按钮控件..... 48	
3.1.4 列表框控件..... 49	
3.1.5 组合框控件..... 50	
3.1.6 滚动条控件..... 50	
3.2 对话框综述..... 50	
3.2.1 对话框的种类..... 50	
3.2.2 对话框的创建和显示..... 51	
3.2.3 CDialog 对话框类..... 52	
3.2.4 对话数据交换/对话数据验证..... 53	
3.3 对话框程序范例..... 54	
3.4 文件对话框的实现技术..... 66	
第四章 菜单和控件栏 74	
4.1 菜单..... 74	

4.1.1 菜单和资源	74	5.5.1 日期时间控件	134
4.1.2 CMenu 类	75	5.5.2 IP 地址控件	134
4.1.3 菜单消息映射和命令处理	76	5.5.3 应用实例	135
4.1.4 更新命令用户接口 (UI) 消息	77	5.6 其它通用控件	137
4.1.5 扩展命令和范围命令处理	78	5.6.1 滑块控件	137
4.1.6 上下文菜单	78	5.6.2 数值调节钮控件	138
4.1.7 应用实例	79	5.6.3 树控件	138
4.2 工具栏	89	5.6.4 工具提示控件	139
4.2.1 类 CToolBar	89	5.6.5 应用实例	140
4.2.2 工具栏的创建和处理	90	第六章 Windows 图形图像编程	148
4.2.3 工具栏泊位和漂浮	92	6.1 图形设备接口	148
4.2.4 工具提示	93	6.1.1 GDI 对象	148
4.2.5 工具栏控件	94	6.1.2 设备描述表	149
4.2.6 应用实例	97	6.2 位图	152
4.3 状态栏	103	6.2.1 图像处理主要函数	152
4.3.1 类 CStatusBar	104	6.2.2 兼容设备描述表	154
4.3.2 状态栏的创建和处理	104	6.2.3 位图的旋转	154
4.3.3 状态栏通用控件	106	6.2.4 位图的缩放	154
4.3.4 应用实例	106	6.3 图像显示技术	154
4.4 对话框	111	6.3.1 利用文件信息显示各种位图文件	155
4.4.1 类 CDialogBar	111	6.3.2 灰度位图显示高级技术	156
4.4.2 对话框的创建和处理	112	6.3.3 真彩色位图显示高级技术	165
4.4.3 应用实例	113	6.4 位图按钮	172
第五章 Windows 通用控件	117	第七章 综合实例	176
5.1 通用控件概述	117	7.1 文档模板	176
5.2 动画控件和进度控件	118	7.1.1 文档模板的构成	176
5.2.1 动画控件	118	7.1.2 文档模板的创建	177
5.2.2 进度控件	119	7.2 拆分窗口	178
5.2.3 应用实例	120	7.3 “画中画”技术	180
5.3 选项卡控件和属性对话框	122	7.4 闪烁窗口	181
5.3.1 选项卡控件	122	7.5 综合实例	181
5.3.2 属性对话框	123	7.5.1 创建应用程序	182
5.3.3 应用实例	124	7.5.2 加入闪烁窗口	182
5.4 图像列表和列表控件	127	7.5.3 应用界面编程	185
5.4.1 图像列表	127	7.5.4 手机制作	191
5.4.2 列表控件	128	7.5.5 本地图像、远地图像显示	198
5.4.3 应用实例	129	7.5.6 画中画显示	204
5.5 日期时间控件和 IP 地址控件	134	7.5.7 多媒体浏览器	214

第一章 Visual C++和基本类库

Microsoft推出的Visual C++和Microsoft基本类库（Microsoft Foundation Classes，MFC）成功地将面向对象和事件驱动编程概念联系起来，并得到很好的配合，使得编写Windows应用程序的过程变得简单、方便，且代码量小。为了使读者对MFC编程有一个整体了解，本章首先概述MFC应用程序结构等一些基础知识。

本章主要内容

- Visual C++重要特性
- MFC框架、文档和视图结构
- AppWizard、AppStudio和ClassWizard
- 利用MFC开发应用程序过程
- 调试技术
- MFC类的层次结构

1.1 Visual C++重要特性

Visual C++是Microsoft Visual Studio（可视工作室）的重要组成部分，并且可以同时和其它Microsoft工具协同工作来达到高效率的开发。

1.1.1 功能高度集成的用户接口

Visual C++包括四个最常用的视图窗口：解决方案资源管理器（FileView）、类视图（ClassView）、资源视图（ResourceView）和属性窗口，这些视图窗口提供对Visual C++工具的快速访问。

1. 解决方案资源管理器

解决方案资源管理器列出所有文件以及文件之间的相互关系，如图1.1所示。一个解决方案可以包括一个或多个项目，一个项目由源文件、头文件和资源文件构成。

源文件和头文件使用Visual C++编辑器进行编辑，该编辑器是一个强有力的源代码编辑器，提供了键盘宏指令、重映射的键、多文件查找、列选择、用户自定义的语句高亮显示、递增查找、自动缩格、可分割表格以及复杂模板匹配等功能。

2. 类视图

类视图将源代码组织为类，显示项目的类和组成部件，如图1.2所示。

利用类视图，程序员可以将精力集中于类上，通过类名和成员函数进行工作，而不需要操作源文件和头文件列表。利用一个类似Explorer的界面，可以跳到、增加或移去类、成员函数和成员变量，使得代码的查找、编辑也更为容易。由于类视图不必通过建立而自

动修改，因而程序员总能获得一个项目的精确视图。

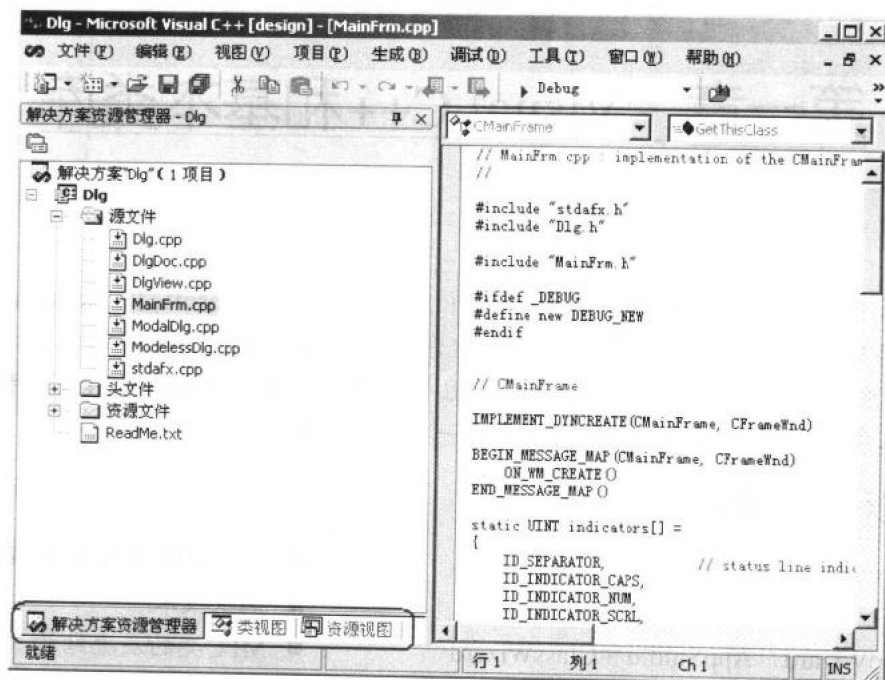


图 1.1 解决方案资源管理器

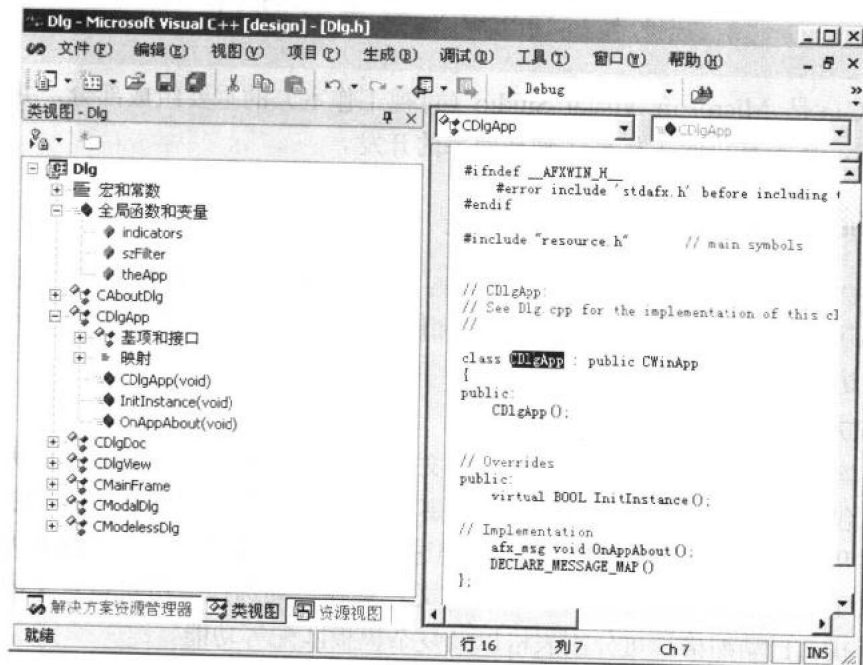


图 1.2 类视图

3. 资源视图

资源视图允许程序员查看 Windows 资源，如位图、菜单、对话框和字符串表等，如图 1.3 所示。

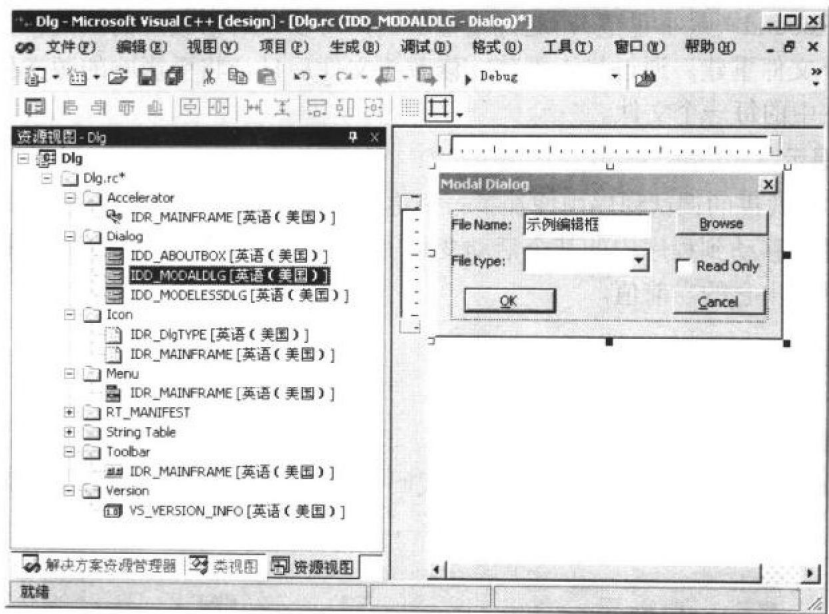


图 1.3 资源视图

通过资源视图和各资源编辑器就可以完成各类资源的创建、修改、删除等操作。

4. 属性窗口

属性窗口用于显示文件、类、资源的各种属性，并可用于添加类的事件处理函数、消息处理函数、重载函数等，如图 1.4 所示。

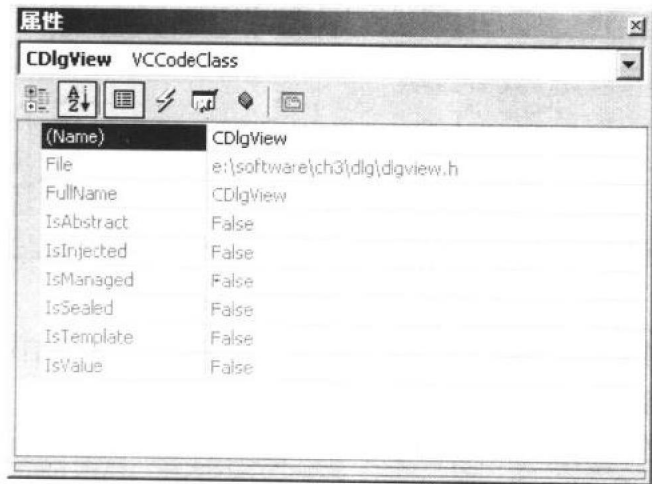


图 1.4 属性窗口

1.1.2 编译调试性能优越

1. 增量编译和最小重建

当程序员开发应用程序时，需要经常对它进行重建。以前，如果一个源文件以任何方式进行了修改，则需要在下次建立进程中重新编译整个源文件。而 Visual C++ 则只编译或链接修改过的部分，缩短了编译时间，采用的技术为 **Incremental Compilation**（增量编译）

和 Minimal Rebuild（最小重建）。通过增量编译，可以只对修改过的函数和类进行重建，而不是对整个文件重建。通过最小重建，修改头文件后只会将受影响的源文件重新编译，而不是头文件中的每一个文件。

2. 数据提示

Visual C++改进的调试程序中增加了一个称为 Data Tips（数据提示）的功能。在调试时，当你将光标移动到程序中的某个活动变量上时，一个小的 ToolTip 窗口将会在该变量右侧出现并显示出它的当前值。

1.1.3 应用程序的开发效率高

1. “编辑并继续”特性

该特性实现了在调试时，如果修改了代码，则继续执行时，系统将弹出“编辑并继续”对话框，询问程序员是否执行该特性操作，如图 1.5 所示。

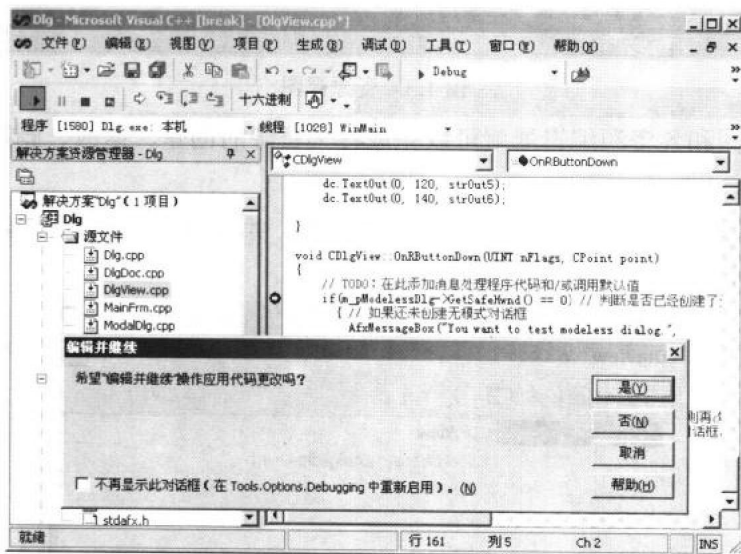


图 1.5 “编辑并继续”特性

如果选择“是”，则无需进行退出调试会话、重新编译、重新启动调试器、再设断点到当前问题的语句等工作，就可以继续执行，如图 1.6 所示。

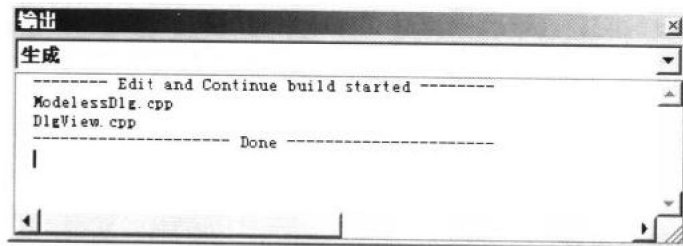


图 1.6 重新编译并继续执行

2. 智能输入（IntelliSense）技术

智能输入，也称编辑器自动感应输入，如提示类对象的成员函数、函数参数信息、类

型信息以及自动完成输入单词等（如图 1.7 所示）。这使得开发人员无需记忆复杂的语法、参数等。

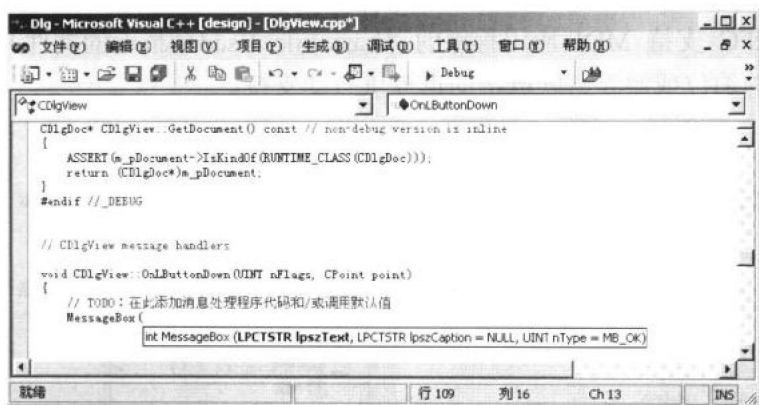


图 1.7 智能输入

3. 参数自动弹出功能

当鼠标光标放在某个参数上时，就可以得到该参数的数据类型信息，如图 1.8 所示；当鼠标光标放在某个函数上时，就可以得到该函数的函数原型。

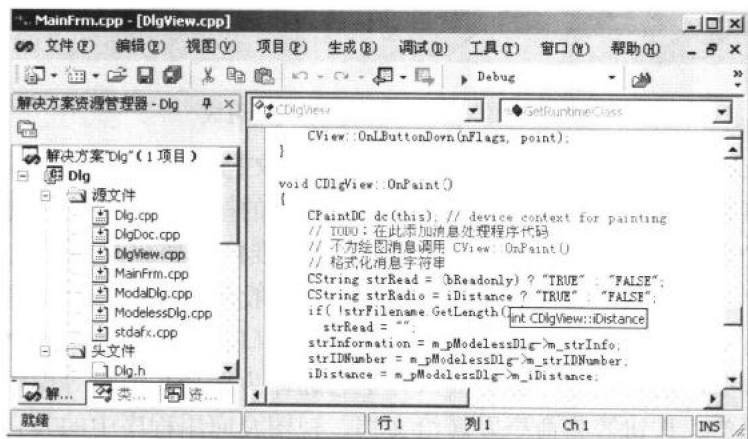


图 1.8 参数自动弹出功能

4. 动态更新类视图

在程序中新增的成员变量或者方法，无需重新编译就能立即反映在类视图中，从而大大方便了程序员。

1.2 MFC 框架、文档和视图结构

MFC 实现了两项功能：一是定义了一个应用程序的初始行为，二是对 Windows API 进行了类封装。MFC 不仅封装处理 Windows API 的所有细节，还简化了 OLE（对象链接与嵌入）和 ODBC（开放式数据库互连）等处理。

MFC 既支持单文档接口 (SDI) 应用程序, 还支持多文档接口 (MDI) 应用程序。SDI 应用程序 (如 NOTEPAD 记事本) 一次只允许打开一个文档框架窗口, 而 MDI 应用程序 (如 Visual C++ 工作平台, 如图 1.9 所示) 允许在同一个应用实例中打开一个或多个文档框架窗口。MFC 支持 MDI 应用程序的 “Last File Used (最近的文件)” 列表。另外, MFC 对工具栏的泊位或漂浮、联机帮助也提供自动支持。



图 1.9 MDI 应用程序及其组成

一个 SDI 或 MDI 程序主要由框架 (Frame)、文档 (Document) 和视图 (View) 构成。单文档窗口和 MDI 各子窗口都是框架窗口, 它的核心是 “文档 - 视图” 结构。用户处理的数据单位就是一个文档对象。文档对象用于维护、加载及保存数据。它是由 “文件” 菜单中的 “新建” 或 “打开...” 命令创建的, 一般都保存在一个文件中。用户通过文档之上的视图与文档进行交互。视图是镶嵌在框架窗口客户区域内的一个窗口, 它显示其关联的文档的数据, 接收鼠标与键盘输入并把它翻译为选择与编辑等操作。另外通过用户接口对象, 如菜单、按钮等将命令发送给文档、视图及应用程序中的其它对象, 然后由这些对象完成命令的执行。

具体而言, 我们可以认为应用程序主要由以下 6 个对象构成:

(1) 应用程序对象。应用程序类 (从 `CWinApp` 派生而来) 完成应用程序的初始化和清除工作。有且只有一个应用程序对象, 它创建和管理所有文档类型的文档模板。

(2) 线程对象。线程对象是指应用程序的主执行线程 (称为主进程), 其它独立的执行线程 (称为辅进程) 需从 `CWinThread` 派生。`CWinApp` 的基类也是 `CWinThread` 类。

(3) 文档模板。文档模板定义了文档类、视图类和框架窗口类三者之间的关系。`CSingleDocTemplate` 类实现 SDI 文档模板, `CMultiDocTemplate` 类实现 MDI 文档模板。

(4) 文档。文档类 (从 `CDocument` 派生而来) 指定了应用程序的数据。若应用程序支持 OLE 功能, 则文档类可从 `COleDocument` 或其派生类中派生。

(5) 视图。视图类 (从 `CView` 派生而来) 是用户的 “数据窗口”。视图类指定了用户

查看文档数据的方式和交互方式，如见到的文档可能是文本，也可能是图表或图像等。通常情况下，一个文档只有一个视图，但在有些情况下一个文档可能要有多个视图。

MFC 已定义了许多视图类，应用程序可以根据不同需要从不同的视图类派生；如果需要滚动，则视图类可从 `CScrollView` 派生；如果视图中有一个基于对话框资源的用户接口，则视图类可从 `CFormView` 派生。对于简单的文本数据，可从 `CEditView` 派生；对于基于格式的数据访问应用程序，可从 `CRecordView` 派生。

(6) 框架窗口。视图是显示在“文档框架窗口”中的。在 SDI 应用程序中，文档框架窗口也是该应用程序的“主框架窗口”。在 MDI 应用程序中，文档窗口是显示在主框架窗口中的子窗口，所派生的主框架窗口类指定了包含视图的框架窗口的格式及其它特性。

SDI 应用程序的文档框架窗口类的基类为 `CFrameWnd` 类。MDI 应用程序的主框架窗口类的基类为 `CMDIFrameWnd` 类，文档框架窗口类的基类为 `CMDIChildWnd` 类。

总之，应用程序的这些对象通过命令和消息相互联系，共同对用户动作进行响应。一个应用程序对象管理一个或多个文档模板；每个文档模板创建并管理一个或多个文档；用户通过包含在框架窗口中的一个视图观察和处理文档。

1.3 MFC 编程工具及其相互关系

Visual C++ 的 MFC 应用框架将编辑器、编译器、连接器、调试器、AppWizard、AppStudio、ClassWizard 等工具集成在同一环境中，极大地方便了程序员开发 Windows 应用程序。其中以下三个工具尤为重要：

- ① AppWizard;
- ② AppStudio;
- ③ ClassWizard。

1.3.1 AppWizard

AppWizard 又称为 MFC 应用程序向导，是一个代码发生器，用于创建 MFC 应用程序。它按照用户通过对话框指定的特性、类名及源代码文件名来创建一个可启动的 Windows 应用程序框架。在此基础上程序员可完成自己应用程序的特殊功能。生成的应用程序项目文件包含应用程序、文档、视图和框架窗口等类实现文件；资源文件，包含标准菜单和可选择的工具栏；其它必需的 Windows 文件，以及可选的含有 Windows Help 主题的 .RTF 文件或.htm 文件等。AppWizard 还可以包含对 OLE 和数据库的支持文件。

在开发应用程序的进程中，必须通过修改菜单资源，加入处理函数等，使程序完全“有血有肉”。AppWizard 的“// TODO”注释表示在此处需要程序员加入实际操作代码。

1.3.2 AppStudio

AppStudio 是一个资源编辑器，用于创建用户界面和编辑程序资源，它可用来创建包含 #define 常量在内的头文件。具体地讲，AppStudio 可以创建并编辑菜单、对话框、定制的控件框、加速键、位图、图标、光标、字符串和版本等资源。它对应于 Studio

Workspace 的“资源视图”，如图 1.3 所示。

通常，该资源编辑器与 ClassWizard 一同使用，例如，创建了对话模板资源后，就可以利用 ClassWizard 创建新的对话框类，然后可以映射对话框中各命令按钮的消息处理函数以及增加的成员变量。

AppWizard 建立含有基本资源（标准菜单、About 对话框、含有预定义字符串的字符串表等）的.RC 文件。其它文件包含缺省图标，版本资源等。如果选择包括初始工具栏的选项，则显示含有工具栏按钮图像的位图。

在编程时常常需要建立新的资源和修改原有资源。对于菜单、工具栏按钮和对话框等资源，在创建之后可以使用 ClassWizard 将它们链接到代码中。

1.3.3 ClassWizard

在编写程序过程中，最频繁使用的 MFC 工具就是 ClassWizard，它包括 MFC 类向导（如图 1.10 所示）、事件处理程序向导（如图 1.11 所示）、添加成员变量向导（如图 1.12 所示）等。

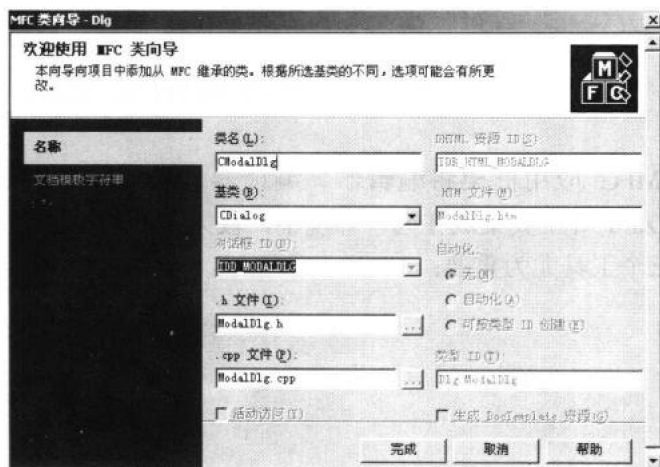


图 1.10 MFC 类向导

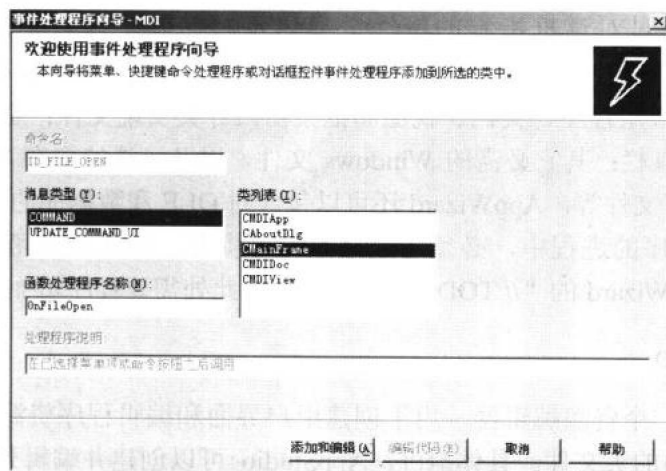


图 1.11 事件处理程序向导

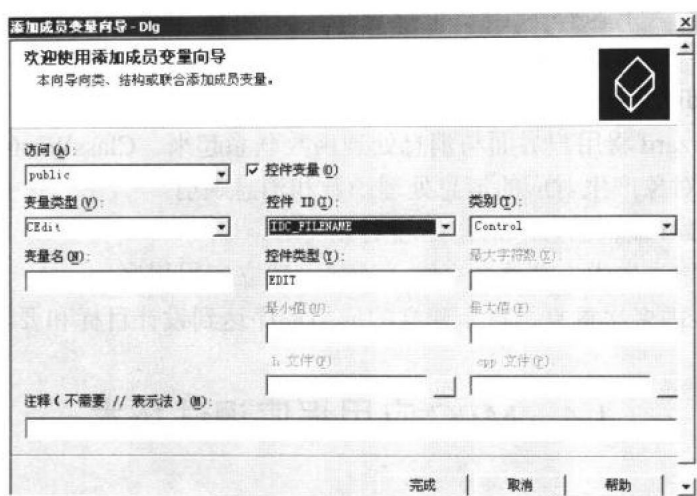


图 1.12 添加成员变量向导

程序员利用 ClassWizard 填写实施细节，选择把哪些消息映射给哪些对象，然后实现这些映射。即利用它管理类和 Windows 消息，将 Windows 消息及用户界面对象（例如菜单）与程序代码联系起来。

利用它可以往 AppWizard 所产生的工作框架中添加新的类或函数，也可以对现存的类进行修改，这就避免了一般代码发生器经常遇到的一致性问题。

另外还可以创建类成员变量、OLE Automation 的方法和属性，数据库类和覆盖 MFC 类中的虚函数等，以实现应用程序工作的细节。

Visual Workspace 提供的类视图将源代码组织为类，使得代码的查找、编辑更为容易。通过利用 ClassView，程序员可以将精力集中于类上，通过类名和成员函数进行工作，而不需要操作源文件和头文件列表。类视图是可缩放项目窗口的一个弹出对话框，用来显示项目的类和组成部件，使得程序员不必再手工打开它们。利用一个类似 Explorer 的界面，可以跳到、增加或移去类、成员函数和成员变量。由于类视图不必通过建立而自动修改，因而程序员总能获得一个项目的精确视图。

ClassWizard 允许程序员浏览自己应用程序中所有与用户接口对象相关的消息，并且允许为这些消息快速地定义消息处理函数，同时 ClassWizard 会自动更新消息映射。用户接口对象的类型可分为三类，与消息类型一一对应，如表 1.1 所示。

表 1.1 对象类型与对应的消息类型

对象类型	消息类型
窗口类	Windows 消息，如 WM_CREATE 消息
菜单或加速键标识符	COMMAND 消息（命令处理消息） UPDATE_COMMAND_UI 消息（更新命令用户接口消息）
控制标识符	选中的控制类型的通知消息，如 EN_CHANGE 消息

1.3.4 MFC 应用程序开发过程

使用 MFC 开发应用程序的一般过程为：

(1) 用 AppWizard 创建初始应用程序的框架文件。这些文件中的类都是由类库中的类

派生的，其中包含一个文档类、一个视窗类、一个框架窗口类和一个应用程序类。AppWizard 还创建一个初始的资源文件及其它的辅助文件。

(2) 用 AppStudio 创建和编辑资源，从而创建用户界面对象，如菜单、对话框等。

(3) 用 ClassWizard 将用户界面与消息处理函数联系起来。ClassWizard 为 AppStudio 产生的每个用户接口对象产生对应的消息处理函数和消息映射。

(4) 用 Visual C++ 编辑器加入消息处理函数代码。

(5) 用“生成”|“生成”菜单命令编译连接，建立应用程序。

以上(2)~(5)步需多次重复，直至建立的应用程序达到设计目标和要求。

1.4 MFC 应用程序调试技术

除非是特别小的应用程序，或是您非常仔细，否则不可能将程序一次建立成功，且符合要求。在设计过程中，程序发生错误是不可避免的，且程序越大越复杂，错误发生的可能性也就越大。通常可分为两种错误：语法错误和执行错误。程序员可以很容易地发现并改正程序语法错误，但不易发现和改正执行错误。因此本节主要讨论如何调试应用程序执行中的错误。Visual C++ 和 MFC 提供了多种方法供程序员调试应用程序。在此只介绍主要的几种方法。

1.4.1 Visual C++ 内置的调试器

与 DOS 程序调试器不同，不能使用 Visual C++ 内置调试器来从头到尾跟踪 Windows 程序。因为 Windows 程序是消息驱动的，而非顺序驱动。我们只能在需要跟踪的消息处理函数的入口设置断点。

Visual C++ 内置调试器与 Visual C++ 工作平台紧密配合，可以设置断点、控制单步执行，可以查看变量、修改变量、查看寄存器，也可通过 Quick Watch 快速查看变量值。菜单命令“调试”|“逐语句”可跟踪进入类库，“调试”|“逐过程”可以避免进入类库中的代码中。

只有将程序设置为 DEBUG 模式（含有调试信息），才能进行调试。

1.4.2 TRACE 宏

TRACE 宏为程序员提供与 DOS 环境下 printf 函数类似的功能，它将格式化字符串输出并送到工作台的调试窗口（如图 1.13 所示）。TRACE 宏是在程序执行时跟踪变量的值的简便方法。

TRACE 宏用法如下：

TRACE(exp)，其中 exp 指定可变数目的参数，这些参数与 printf 函数中使用的可变数目的参数的用法相同。例如：

```
int iLen = 3;
char strBuf[] = "abc";
TRACE("Length = %d, String = %s\n", iLen, strBuf);
```