

Visual C++ .NET 程序设计教程

第2版

郑阿奇 主编
丁有和 等编著



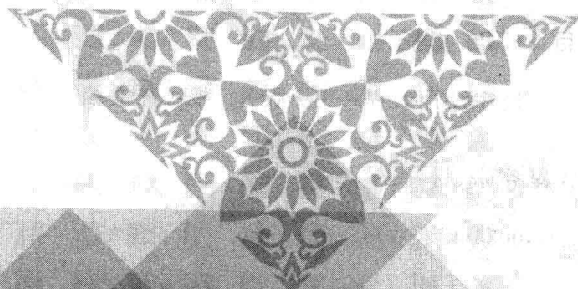
机械工业出版社
China Machine Press

计算机基础课程系列教材

Visual C++ .NET 程序设计教程

第2版

郑阿奇 主编
丁有和 等编著



机械工业出版社
China Machine Press

本书以 Visual C++ .NET 为平台介绍程序设计方法。第一部分为教程, 首先介绍 Visual C++ .NET 开发环境, 然后介绍 C++/CLI 语言基础和 C++/CLI 面向对象编程基础, 在这个基础上系统介绍 Visual C++ .NET 应用程序设计 (包括 Windows 窗体和对话框、窗体常用控件、基本界面元素、文档界面模型、GDI+ 和图像处理、文件和流及数据库)。第二部分为实验, 并在最后给出一个综合应用实验。为了方便教学, 本书配有 PPT 和配套的应用程序实例。阅读本书, 认真完成实验, 读者就能在较短的时间内基本掌握 Visual C++ .NET 及其应用技术。

本书可作为高等学校计算机及相关专业程序设计课程的教材, 也可供 Visual C++ .NET 软件开发人员参考。

封底无防伪标均为盗版

版权所有, 侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目 (CIP) 数据

Visual C++ .NET 程序设计教程 (第 2 版) / 郑阿奇主编. —北京: 机械工业出版社, 2013.1

(计算机基础课程系列教材)

ISBN 978-7-111-40084-4

I . V… II . 郑… III . C 语言—程序设计—高等学校—教材 IV . TP312

中国版本图书馆 CIP 数据核字 (2012) 第 245207 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 刘立卿

北京市荣盛彩色印刷有限公司印刷

2013 年 1 月第 2 版第 1 次印刷

185mm×260mm·21.75 印张

标准书号: ISBN 978-7-111-40084-4

定价: 36.80 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

购书热线: (010) 68326294; 88379649; 68995259

投稿热线: (010) 88379604

读者信箱: hzjsj@hzbook.com

前 言

Visual C++ .NET 除了原来的 MFC, C++/CLI 语言已经趋于完善, 它作为 Visual Studio .NET 的重要组成部分, 是目前最推崇的应用程序开发平台。

本书以 C++/CLI 为基础, 系统介绍 Visual C++ .NET。本书第一部分是 Visual C++ .NET 教程, 每章后面有习题。第二部分为实验, 最后一个实验是综合应用。

本书在内容安排上具有如下特点:

(1) 第 1 章介绍 Visual C++ .NET 开发环境, 包括安装和设置、认识开发环境、解决方案和资源管理及使用代码编辑器。

(2) 第 2 章到第 4 章是 C++/CLI 语言基础, 包括 C++/CLI 编程基础、C++/CLI 编程进阶和 C++/CLI 面向对象编程。

(3) 第 5 章到第 11 章系统介绍 Visual C++ .NET 环境下开发 Windows 应用程序, 包括 Windows 窗体和对话框、窗体常用控件、基本界面元素、文档界面模型、GDI+ 和图像处理、文件和流及数据库。

(4) 第二部分实验可作为学生上机操作指导, 最后一个实验是综合应用, 用于锻炼学生综合解决问题的能力。

本书配有教学课件和配套的应用程序实例, 需要者可以到 <http://www.hzbook.com> 网站免费下载。

实际上, 本书不仅适合教学, 也非常适合 Visual C++ .NET 的各类培训和程序开发人员学习和参考。只要阅读本书, 并结合实验进行练习和实习, 就能在较短的时间内基本掌握 Visual C++ .NET 及其应用技术。

本书主要由南京师范大学丁有和编写, 南京师范大学郑阿奇对全书进行了统编、定稿。另外梁敬东、顾韵华、朱毅华、时跃华、赵青松、王燕平、汤玫、刘毅、彭作民、崔海源、徐卫军等也参加了本书的编写工作。刘建、刘中、郑进等其他很多朋友对本书编写提供了帮助, 在此表示感谢!

由于作者水平有限, 不当之处在所难免, 恳请读者批评指正。

目 录

前言

第一部分 教 程

第 1 章 Visual C++ .NET 开发环境 1

1.1 安装和设置 1

1.1.1 Visual Studio 2008 安装 1

1.1.2 默认环境设置 2

1.1.3 重置默认环境 3

1.2 认识开发环境 4

1.2.1 开发环境简介 4

1.2.2 窗口操作 5

1.2.3 初试 Form 项目 6

1.3 解决方案和资源管理 10

1.3.1 项目和解决方案概念 10

1.3.2 解决方案资源管理器 10

1.3.3 类视图 11

1.3.4 属性管理器 11

1.4 使用代码编辑器 12

1.4.1 字体和颜色 12

1.4.2 选择和更改代码 13

1.4.3 代码大纲 13

1.4.4 书签和代码定位 14

习题 16

第 2 章 C++/CLI 编程基础 17

2.1 C++/CLI 框架 17

2.1.1 C++/CLI 概述 17

2.1.2 创建控制台程序 18

2.1.3 理解程序框架 19

2.1.4 修改并添加代码 20

2.1.5 生成和运行 20

2.2 数据及其类型 21

2.2.1 变量 21

2.2.2 基本值类型 23

2.2.3 基本引用类型 27

2.2.4 常量 28

2.3 运算符 31

2.3.1 算术运算符 31

2.3.2 赋值运算符 33

2.3.3 关系运算符 34

2.3.4 逻辑运算符 34

2.3.5 位运算符 35

2.3.6 条件和逗号运算符 36

2.4 基本语句 36

2.4.1 表达式语句、空语句和块语句 36

2.4.2 if 和 switch 语句 37

2.4.3 while 和 do-while 循环 40

2.4.4 for 循环 42

2.4.5 for each 循环 43

2.4.6 break 语句 44

2.4.7 continue 语句 44

2.5 函数 45

2.5.1 函数的定义 45

2.5.2 函数的调用和声明 47

2.5.3 全局变量和局部变量 48

2.5.4 函数的默认形参值 49

2.5.5 函数重载 50

习题 51

第 3 章 C++/CLI 编程进阶 53

3.1 跟踪句柄和引用 53

3.1.1 跟踪句柄和 ^ 53

3.1.2 跟踪引用和 % 54

3.2 C++/CLI 枚举 54

3.2.1 枚举定义 54

3.2.2 枚举常量的类型和值 56

3.2.3 枚举常量的运算 56

3.3 C++/CLI 数组 57

3.3.1 CLR 数组 57

3.3.2 数组的排序及查找 59

3.3.3 多维数组 61

3.3.4 数组的数组 62

3.4 C++/CLI 字符串 63

3.4.1 连接字符串 64

3.4.2 修改字符串	65	5.1.2 使用项目模板	114
3.4.3 比较字符串	66	5.1.3 窗体程序结构	115
3.4.4 搜索字符串	67	5.2 窗体属性及设置	116
3.4.5 子串和拆分	68	5.2.1 Form 类	116
3.4.6 函数和 main 实参	69	5.2.2 外观属性	117
3.5 控制台输入输出及格式	70	5.2.3 布局属性	117
3.5.1 控制台输入	70	5.2.4 窗口样式属性	117
3.5.2 Write 和 WriteLine	71	5.2.5 属性设置	118
3.5.3 格式项	72	5.3 窗体事件及处理	118
3.5.4 自定义格式和 ToString	74	5.3.1 窗体常用事件	118
习题	76	5.3.2 鼠标事件	119
第 4 章 C++/CLI 面向对象编程	77	5.3.3 键盘事件	120
4.1 C++/CLI 面向对象概述	77	5.4 窗体添加和使用	122
4.1.1 C++/CLI 类	77	5.4.1 向窗体添加控件	122
4.1.2 数值类和对象	78	5.4.2 控件拾取和操作	123
4.1.3 引用类和对象	80	5.4.3 更改控件的 Tab 键顺序	124
4.1.4 类、程序集和命名空间	81	5.4.4 向项目添加窗体	124
4.2 类成员	83	5.4.5 模式和无模式显示	125
4.2.1 属性	83	5.5 对话框	126
4.2.2 literal 和 initonly 字段	88	5.5.1 自定义对话框	126
4.2.3 拷贝和静态构造函数	90	5.5.2 通用对话框	127
4.2.4 终结器	91	5.5.3 消息对话框	128
4.3 继承和多态	93	习题	129
4.3.1 派生类	93	第 6 章 窗体常用控件	130
4.3.2 虚函数	94	6.1 控件概述	130
4.3.3 new 和虚函数重置	95	6.1.1 控件的添加和移除	130
4.3.4 抽象和密封函数	97	6.1.2 控件的外观属性	132
4.3.5 运算符重载	98	6.1.3 控件的布局属性	132
4.4 接口、委托和事件	100	6.1.4 控件的事件及其处理	133
4.4.1 接口和接口类	100	6.2 标签和组框	134
4.4.2 委托	102	6.2.1 标签 (Label)	134
4.4.3 事件及其处理	105	6.2.2 链接标签 (LinkLabel)	135
4.5 类型转换和异常处理	106	6.2.3 组框 (GroupBox)	137
4.5.1 类型转换	106	6.3 按钮、单选按钮和复选框	137
4.5.2 try 和 catch	107	6.3.1 按钮 (Button)	137
4.5.3 异常类	108	6.3.2 单选按钮 (RadioButton)	137
4.5.4 使用 throw	109	6.3.3 复选框 (CheckBox)	138
4.5.5 使用 finally 块	111	6.3.4 实例：制作问卷调查	138
习题	112	6.4 文本框和数字旋转控件	140
第 5 章 Windows 窗体和对话框	113	6.4.1 文本框 (TextBox) 属性	140
5.1 创建窗体应用程序	113	6.4.2 文本框事件和基本操作	141
5.1.1 Forms 命名空间和类	113	6.4.3 数字旋转控件 (NumericUpDown)	142

6.4.4 实例：学生成绩输入	142	7.4.2 编辑状态栏	178
6.5 列表框	143	7.4.3 实例：显示特殊键状态	179
6.5.1 列表框 (ListBox) 的属性	143	习题	180
6.5.2 列表框事件和基本操作	144	第 8 章 文档界面模型	181
6.5.3 实例：使用列表框	145	8.1 列表视图	181
6.6 组合框	147	8.1.1 ListView 属性和事件	181
6.6.1 组合框 (ComboBox) 属性	147	8.1.2 列表项操作	183
6.6.2 组合框事件和基本操作	148	8.1.3 列标头操作	185
6.6.3 示例：使用组合框	148	8.1.4 列表界面元素与类的关系	186
6.7 图片框和进展条	149	8.1.5 实例：学生成绩列表	186
6.7.1 图片框 (PictureBox)	149	8.2 树视图	190
6.7.2 进展条 (ProgressBar)	150	8.2.1 TreeView 属性和事件	190
6.7.3 实例：使用进展条	150	8.2.2 树视图的节点操作	191
6.8 滚动条和滑动条	151	8.2.3 实例：学生成绩树	193
6.8.1 滚动条 (ScrollBar)	151	8.3 选项卡和切分窗口	195
6.8.2 滑动条 (TrackBar)	152	8.3.1 面板 Panel	195
6.8.3 实例：调整窗体背景颜色	153	8.3.2 选项卡 TabControl	196
6.9 日期时间控件、月历控件和计时器	154	8.3.3 切分窗口 SplitContainer	200
6.9.1 日期时间控件 (DateTimePicker)	154	8.4 多文档界面	202
6.9.2 月历控件 (MonthCalendar)	154	8.4.1 MDI 及其创建	202
6.9.3 实例：使用日期时间和月历控件	155	8.4.2 合并菜单项	203
6.9.4 计时器 (Timer)	156	8.4.3 管理子窗体	206
6.9.5 实例：小游戏	156	习题	208
习题	158	第 9 章 GDI+ 和图像处理	209
第 7 章 基本界面元素	160	9.1 GDI+ 概述	209
7.1 图标和光标	160	9.1.1 GDI+ 新特性及功能	209
7.1.1 图像编辑器	160	9.1.2 GDI+ 类和结构	210
7.1.2 图标编辑和使用	161	9.1.3 常用基本数据结构	211
7.1.3 光标及其编程控制	164	9.2 绘图基础	213
7.2 菜单	165	9.2.1 使用 GDI+ 的一般方法	213
7.2.1 MenuStrip 和事件	166	9.2.2 颜色和颜色对话框	213
7.2.2 编辑菜单	166	9.2.3 画笔	214
7.2.3 实例：菜单切换	169	9.2.4 画刷	216
7.2.4 菜单的编程控制	170	9.2.5 基本绘图方法	221
7.2.5 使用快捷菜单	172	9.2.6 坐标空间及其变换	224
7.3 工具栏	173	9.3 文本绘制	226
7.3.1 ToolStrip 和事件	173	9.3.1 字体及字体对话框	226
7.3.2 编辑工具栏	174	9.3.2 文本输出和显示质量	228
7.3.3 工具条资源和图像列表	175	9.3.3 文本格式化属性	229
7.3.4 实例：使用多个工具栏	176	9.3.4 计算字符和文本尺寸	231
7.4 状态栏	178	9.4 图像处理	233
7.4.1 StatusStrip 类	178	9.4.1 图像格式和 Image 类	233

[illegible]

第二部分 实 验

实验 1	熟悉开发环境和建立控制台项目	299
实验 2	基本数据类型、表达式和基本语句	302
实验 3	数组和格式输入输出	303
实验 4	C++/CLI 类和对象	304
实验 5	多态和虚函数、运算符重载 ..	307
实验 6	Windows 窗体和对话框	309
实验 7	标签和按钮控件	311
实验 8	文本框、列表框和组合框	313
实验 9	其他常用控件	315
实验 10	菜单、工具栏和状态栏	316
实验 11	文档界面模型	319
实验 12	使用 GDI+	320
实验 13	文件和流操作	321
实验 14	数据库	322
实验 15	综合应用	326
附录 A	本书约定	332
附录 B	常用键代码	333
附录 C	Student.MDB 数据库表	335
附录 D	程序调试	336

第一部分 教程

第 1 章 Visual C++ .NET 开发环境

随着 .NET 技术的不断发展, .NET Framework 从早期的 .NET Framework 1.0 发展到现在较为流行的 .NET Framework 3.5, 并集成到 Visual Studio 2008 开发平台中。这个平台是面向 Windows Vista、Office 2007 以及 Web 2.0 等的新一代开发工具, 是对 Visual Studio 2005 的一次全面升级。

Visual Studio 2008 支持项目模板、调试器和部署程序, 它是快速建立企业级 Web 应用、Office 应用、Mobile 应用程序和高性能桌面应用程序的集成开发环境, 其中包括了 Visual C++、Visual Basic、Visual C# 和 Visual Web Developer 等开发工具。

本书以 Visual Studio 2008 中文专业版中的 Visual C++ 作为编程语言, 为区别于早期的 Visual C++, 这里(本书的第 2 版)仍旧称为 Visual C++ .NET, 所有程序均基于 .NET Framework, 且在 Windows XP 中调试通过。

1.1 安装和设置

Visual Studio 2008 引入了几百个新特性, 整合了对象、关系型数据、XML 的访问方式, 语言更加简洁。相应的设计器不仅可以实时反映代码的变更, 而且其中的智能感知功能可以更好地提高开发效率。这里先来看看 Visual Studio 2008 的安装和环境设置等内容。

1.1.1 Visual Studio 2008 安装

在安装的过程中, 为了减少可能遇到的问题, 应遵循一定的安装步骤。下面以 Visual Studio 2008 中文专业版为例, 介绍 Visual Studio 2008 在 Windows XP 下的安装过程。

① 关掉所有正在运行的程序, 尤其是杀毒监控程序。将安装程序加载到虚拟光驱或将 DVD 安装盘放入光驱, 此时将自动运行 Autorun.exe。或者, 将安装程序解包, 然后直接运行 setup.exe, 此时将弹出如图 1-1 所示的安装对话框, 其中包含几种安装选项。

② 单击“安装 Visual Studio 2008”选项, 对话框退出。安装程序将加载安装组件到本地计算机并检测当前配置, 待完成后进入开始安装界面, 如图 1-2 所示。



图 1-1 “安装程序”对话框



图 1-2 进入安装向导

③ 单击 **下一步(N) >** 按钮, 出现如图 1-3 所示的协议许可页面, 阅读许可协议中的条款, 并选择接受许可条款。

④ 单击 **下一步(N) >** 按钮, 出现如图 1-4 所示的安装选项页面, 从中可选择相应的安装方案: 默认值、完全和自定义。单击 **浏览(B)...** 按钮, 从弹出的对话框中指定相应的安装路径, 或直接在“产品安装路径”编辑框中输入其他安装路径名。

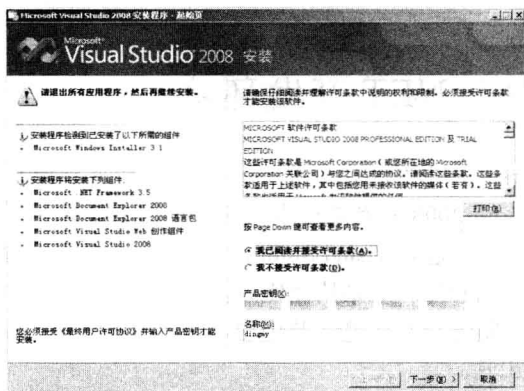


图 1-3 安装起始页(许可协议)



图 1-4 安装选项页

⑤ 选择“默认值”方案, 单击 **完成(F)** 按钮, 开始安装, 如图 1-5 (左) 所示。需要说明的是, 安装组件过程的长短与计算机的配置有关, 需要耐心等待。当出现如图 1-5 (右) 所示的安装页面时, 表明已经安装成功。

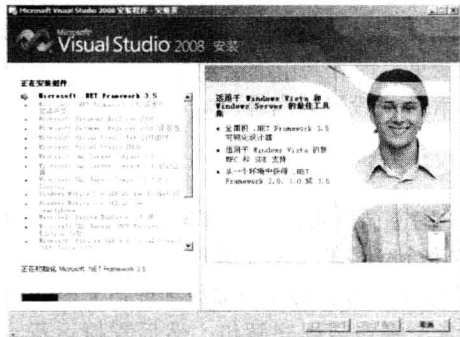


图 1-5 安装页和完成页

⑥ 单击 **完成(F)** 按钮, 又会出现“安装程序”第 1 个对话框页面, 只不过它的第 1 个安装选项变成了“更改或移除 Visual Studio 2008”。

⑦ 单击“安装产品文档”, 开始安装 MSDN 帮助文档(它是每一个开发人员必备的技术资源), 这里从略。

1.1.2 默认环境设置

Visual Studio 2008 成功安装后, 就可以单击 Windows XP 操作系统任务栏的 **开始** 按钮, 然后从弹出的菜单中选择“程序”→“Microsoft Visual Studio 2008”→“Microsoft Visual Studio 2008”, 启动 Visual Studio 2008。

每次启动时, 都会出现一个初始欢迎页面, 称为“Splash Screen”(闪屏, 常意译为初始页面), 如图 1-6 所示。与大多数初始页面一样, Visual Studio 2008 的初始页面也显示了该产品的版本、授

权的用户信息等。更为主要的是，初始页面的右边还显示了已安装的主组件内容，若还安装了第三方插件，则也会一并显示。

第一次运行 Visual Studio 2008 时，初始页面消失后会出现如图 1-7 所示的“选择默认环境设置”页面。对于 Visual C++ 用户来说，为了能延续以往的环境布局和操作习惯，应选中“Visual C++ 开发设置”，然后单击 **启动 Visual Studio(S)** 按钮。稍等片刻后，出现 Visual Studio 2008 开发环境。



图 1-6 初始页面

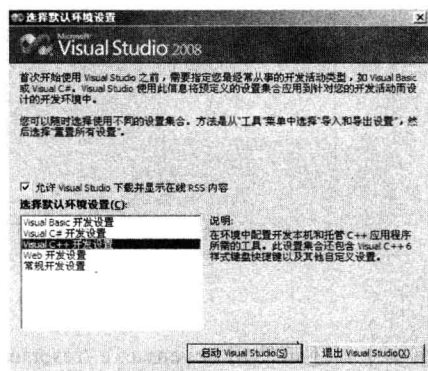


图 1-7 选择默认环境设置页面

1.1.3 重置默认环境

默认环境设置有时并不能满足自己的需要，因为常需要重置 Visual Studio 2008 的默认环境选项。如下列步骤：

① 启动 Visual Studio 2008 后，选择菜单“工具”→“导入和导出设置”命令，弹出如图 1-8（左）所示的“导入和导出设置向导”对话框页面。

② 选中“重置所有设置”选项，单击 **下一步(N) >** 按钮，出现如图 1-8（右）所示的“保存当前设置”向导页面。

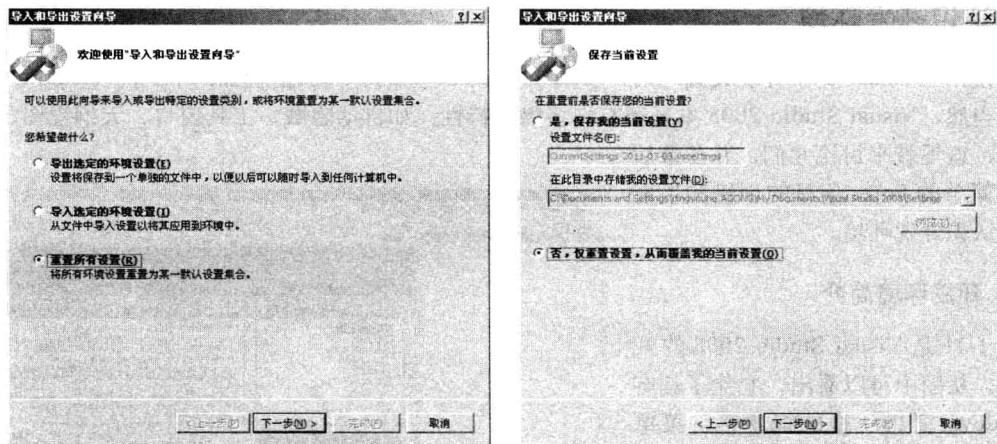


图 1-8 导入和导出设置向导页面

③ 选中“否，仅重置设置，从而覆盖我的当前设置”选项，单击 **下一步(N) >** 按钮，出现如图 1-9（左）所示的“选择一个默认设置集合”向导页面。

④ 选中“Visual C++ 开发设置”，单击 **完成(F)** 按钮，片刻之后，出现如图 1-9（右）所示的“重置完成”向导页面。单击 **关闭** 按钮，向导页面关闭，恢复到刚开始的 Visual Studio 2008 开发环境页面。

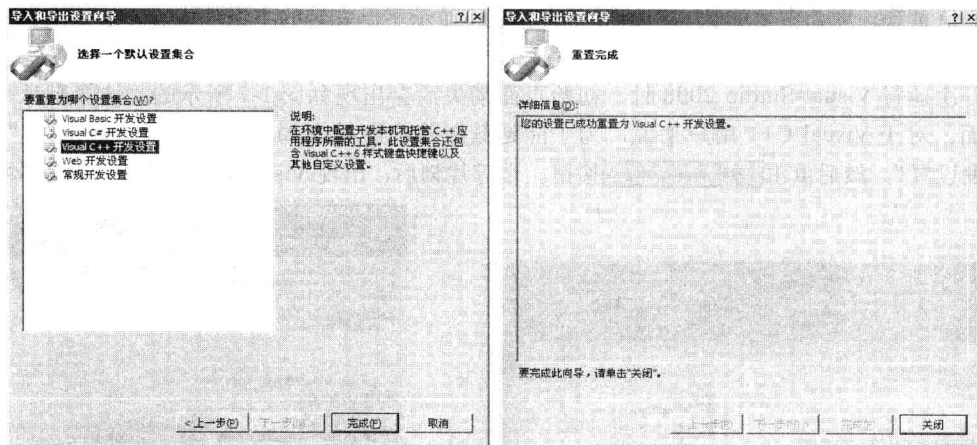


图 1-9 重置向导页面

事实上，还可使用 `devenv.exe /resetsettings` 来恢复 Visual Studio 2008 环境的初始状态，具体步骤如下：

① 关闭 Visual Studio 2008。单击任务栏的 **开始** 按钮，从弹出的菜单中选择“运行”命令，在弹出的对话框输入“`devenv.exe /resetsettings`”（双引号不输入），如图 1-10 所示。

② 单击 **确定** 按钮，上述命令就会运行，与此同时，Visual Studio 2008 清除设置并将其自身重置到其最初的状态。此时，可以打开“任务管理器”来检查 `devenv.exe` 进程是否仍在运行。在 `devenv.exe` 结束运行后，可以重新启动 Visual Studio 2008，此时会再次出现如图 1-7 所示的“选择默认环境设置”对话框页面。



图 1-10 “运行”对话框

1.2 认识开发环境

当使用“Visual C++ 开发设置”后，Visual Studio 2008 的界面风格就如同早期的 Visual C++ 6.0 那样。当然，Visual Studio 2008 本身还有许多界面特性，如自动隐藏、工具窗口、大纲视图、属性页面等，这里就来讨论它们，并在最后以一个简单的 Form 项目的创建和生成过程来认识开发环境。

1.2.1 开发环境简介

图 1-11 是 Visual Studio 2008 的开发环境。从图中可以看出，它除了具有和 Windows 窗口一样的标题栏、菜单栏、工具栏和状态栏外，最主要的是还有不一样的窗口区。窗口区是由中间的 Web 浏览区、左边的解决方案工作区、下面的输出窗口区以及其他窗口等组成。

Web 浏览区位于开发环境的中间，

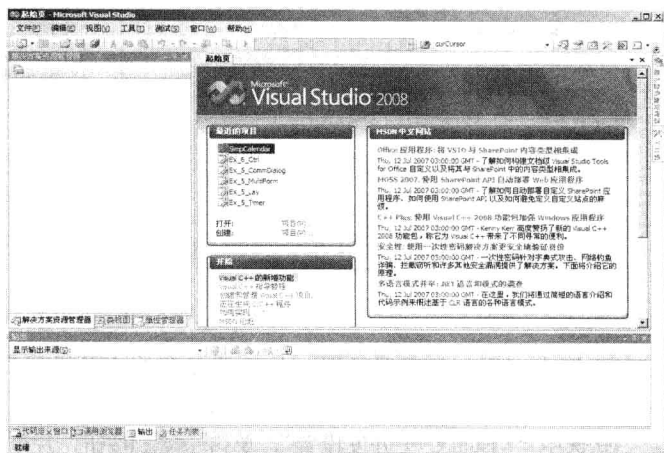


图 1-11 Visual Studio 2008 开发环境

占着较大的范围。它是一个多文档浏览窗口区，不仅可以显示各种程序代码的源文件、资源内容、文档等，而且还将 Web 浏览器（IE）嵌入其中，从而可以直接浏览 Web 页面。由于各文档页面均以标签形式列在窗口上方，因而文档内容的切换只需单击相应的标签即可，操作非常方便。

默认时，Web 浏览区窗口显示的是“起始页”。“起始页”除“最近的项目”外，几乎所有的内容来源于相关的网络资源，其左边还包括“开始”和“Visual Studio 标题新闻”等内容，而右边则是“MSDN 中文网站”。

在开发环境的左侧是解决方案工作区，它是由“解决方案资源管理器”、“类视图”及“资源视图”页面等组成，以“树结构”方式来显示解决方案中的一些信息和相应的操作项，包括类成员、解决方案项目文件节点以及资源节点等。

在开发环境的底部是输出窗口区，它由“代码定义窗口”、“调用浏览器”以及“输出”页面等组成，用来显示各种调用关系、编译连接信息、查找等内容。

1.2.2 窗口操作

事实上，组成 Visual Studio 2008 开发环境的窗口可以只分为两种类型，一种是“文档窗口”，另一种是“工具窗口”。文档窗口是动态产生的，当打开一个 C++ 文件时或在解决方案工作区查看类、资源等具体内容时，就会在 Web 浏览区中打开一个文档窗口用来显示相应的内容。而除文档窗口外的窗口都可称为工具窗口，如输出窗口和属性窗口等。

在 Visual Studio 2008 开发环境中，文档窗口的切换可直接在 Web 浏览区窗口上方单击相应的标签进行，或单击文档窗口最右上角的下拉按钮 ▾，从弹出的下拉文档列表中选择要显示的文档项即可。若单击下拉按钮右边的关闭按钮 ×，则退出当前文档。而对于工具窗口来说，窗口操作可以有浮动和停靠、选项卡式文档、自动隐藏等。

1. 浮动和停靠

Visual Studio 2008 刚开始运行时，窗口区中的各种窗口均处于停靠状态，任何时候用鼠标双击窗口标题栏，都会在浮动和停靠之间进行切换。若用鼠标单击某个窗口不放，可将其拖放到整个窗口区的任何位置，这个位置可以是任何一个窗口区的四边。被拖放的窗口既可单独显示在开发环境界面中的某处，也可与窗口区的其他窗口构成一组。

2. 选项卡式文档

前面说过，Visual Studio 2008 的文档窗口是采用标签式（又称选项卡式）的操作模式。对于工具窗口来说，可通过“选项卡式文档”命令使其按文档窗口模式来操作。

单击某个工具窗口后，选择“窗口”菜单中的“选项卡式文档”菜单命令，则将该工具窗口以标签的方式显示在 Web 浏览区中。当然，若此时选择“窗口”菜单中的“可停靠”菜单命令，则当前工具窗口恢复到上次停靠位置。

3. 自动隐藏

自动隐藏是 Visual Studio 引入 .NET Framework 后新增的界面特性，它和 Windows 任务栏的自动隐藏功能相类似。自动隐藏的功能能够使窗口显示的数量更多，凡是自动隐藏的窗口，都会在其靠近的那一侧边最小化，并只显示出窗口名称标签，参看图 1-11 右边的“服务器资源管理器”和“工具箱”两个默认窗口。当用户将鼠标移动到这种窗口的名称标签时，该窗口就会自动滑出，当该窗口具有输入焦点时（即该窗口标题栏高亮显示），它不会自动隐藏，一旦失去焦点，它又滑向屏幕的侧边，呈最小化状态。

4. 关闭和显示

在窗口区中，每个活动窗口的标题栏处都有下拉按钮、自动隐藏和关闭按钮，如图 1-12 所示。单击关闭按钮后，窗口被关闭，但可通过选择“视图”菜单下的菜单命令，或选择“视图”→“其

他窗口”菜单命令来恢复显示相应的窗口，如图 1-13 所示。



图 1-12 标题栏上的工具按钮



图 1-13 视图和其他窗口菜单项

需要说明的是，当用鼠标右击窗口的标题栏时，或单击下拉按钮，都会弹出一个快捷菜单，其菜单命令依次为“浮动”、“可停靠”、“选项卡式文档”、“自动隐藏”和“隐藏”，这些命令与“窗口”菜单同名命令功能一致。

1.2.3 初试 Form 项目

为了能更直观地操作 Visual Studio 2008 开发环境，这里先来建立一个 Form 项目。

Form 一词，早在 Visual C++ 6.0 之时就被引入，那时译为“表单”，以区别于“对话框” (Dialog) 一词。同时，Form 还意味着与 HTML 有关，是一种可拓展的界面概念。如今，由于 .NET Framework 的产生和发展，Form 找到了更为恰当的内涵，现在常常将 Form 译为“窗体”。

【例 Ex_Form】初试 Form 项目

① 在适当位置处创建一个工作文件夹“VCNET2008 程序”，在此文件夹下再创建一个子文件夹“1”。这样，本章所有创建的应用程序项目均放置在此子文件夹下，既便于管理，又容易查找。以后，第 2 章的程序就放在子文件夹“2”中，以此类推。

② 在“起始页”中的“最近的项目”栏，单击“创建”右侧的“项目”文本项，或选择“文件”→“新建”→“项目”菜单命令，弹出如图 1-14 所示的“新建项目”窗口。从项目类型可以看出，Visual C++ 2008 不仅支持 MFC、ATL，最主要的是它支持 CLR。

MFC (Microsoft Foundation Classes) 不仅是一套基础类库，更主要的还是一种编程方式。MFC 把传统的 Windows API 编程规范中的大多数内容封装成为各种类，使程序员从繁杂的编程中解脱出来，提高了编程和代码效率。

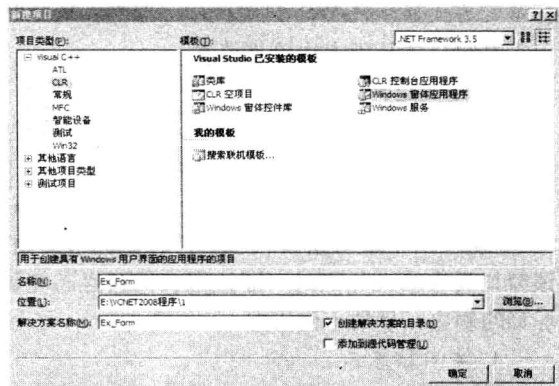


图 1-14 “新建项目”窗口

ATL 是 ActiveX Template Library 的缩写，它是一套 C++ 模板库。使用 ATL 能够快速地开发出高效、简洁的代码，同时对 COM（Component Object Model，组件对象模型）组件的开发提供最大限度的代码自动生成以及可视化支持。

CLR（Common Language Runtime，公共语言运行时）和 Java 虚拟机一样，它也是一个运行时环境，负责资源管理（内存分配和垃圾收集），并保证应用和底层操作系统之间必要的分离。它与 .NET Framework 数据、类和组件库一起，组成了 .NET Framework 整个开发环境。

③ 在“项目类型”栏内选中“CLR”，在“模板”栏内选中“Windows 窗体应用程序”。在下方的“名称”框中输入“Ex_Form”，此时下方的“解决方案名称”框内的内容会自动与输入的一致。单击  按钮，弹出“项目位置”对话框，从中选定以前创建的工作文件夹路径“VCNET2008 程序\1”，然后单击  按钮回到“新建项目”窗口页面，再单击  按钮，系统开始创建，同时开发环境变成如图 1-15 所示的界面。其中，界面中间是窗体设计器，是一种所见即所得的编辑方式。

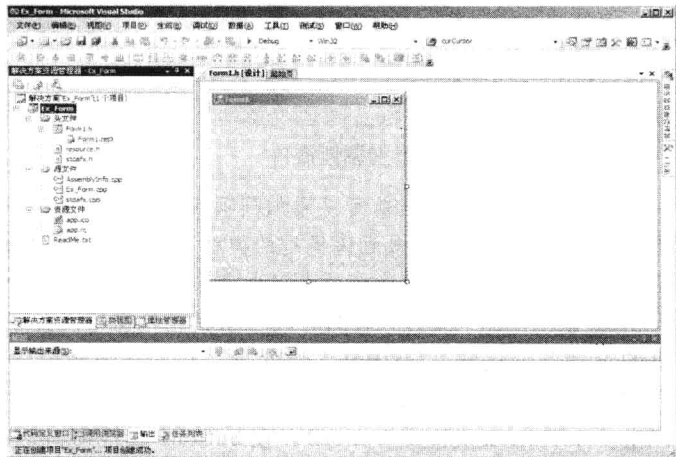


图 1-15 项目创建后的开发环境

④ 在窗体中右击鼠标，弹出相应的快捷菜单，从中选择“属性”菜单命令，则在界面右侧边显示出“属性”窗口，它是添加事件处理、更改对象选项参数的主要操作窗口。将鼠标移至“属性”窗口与设计器之间的竖直边，拾取并向左拖放其竖直边，使“属性”窗口的宽度变大一些。同样，将“解决方案资源管理器”窗口区的宽度也增大一些，使各项文本内容全部显示出来，如图 1-16 所示。

“属性”窗口显示了可编辑的不同类型的字段项，每个字段项均有字段名和可编辑的字段数据（值）。可编辑的字段数据可通过相应的编辑框、下拉列表等来指定或设定。不同的对象，其属性也是不一样的。一旦相应的属性值被更改，则用加粗的字体来显示。

在“属性”窗口的顶部有一个组合框，从中可选择当前容器（多个对象组成）中的某个对象。在组合框下方有几个工具按钮，依次为：按分类顺序显示 、按字母顺序显示 、属性 、事件处理  和属性页 。一旦选定具体的属性字段时，右侧的字段数据域具有输入焦点，同时在属性窗口的最下方显示出当前属性字段的说明。

⑤ 默认时，“属性”窗口中的“Text”属性自动具有输入焦点，以便直接修改其内容。在“Text”属性右侧的编辑框中输入“初试 Form 项目”，然后在设计器中的窗体上单击鼠标，就会发现此时的窗体标题已更改。关闭“属性”窗口。将鼠标移动到右侧边的“工具箱”标签项，则“工具箱”窗口滑出，单击标题上的“图钉”（）工具按钮，则“工具箱”窗口将“钉”在左侧边，并一直显示，如图 1-17 所示。需要说明的是，为了方便编程，通常将“工具箱”窗口设置成“自动隐藏”，而将“属性”窗口钉在左侧边。

从外观上看，“工具箱”是一个浮动的树控件，它与 Windows 资源管理器的工作方式非常类似，但没有网格或连接线。可以同时展开“工具箱”的多个段（称为“类型节点”），整个目录树可在“工具箱”窗口内部通过右侧滚动条来滚动。若要展开“工具箱”的某类型节点，则应单击类型

节点名称左侧旁的加号按钮。若要收起一个已展开的类型节点，则应单击类型节点名称左侧旁的减号按钮。



图 1-16 添加属性窗口

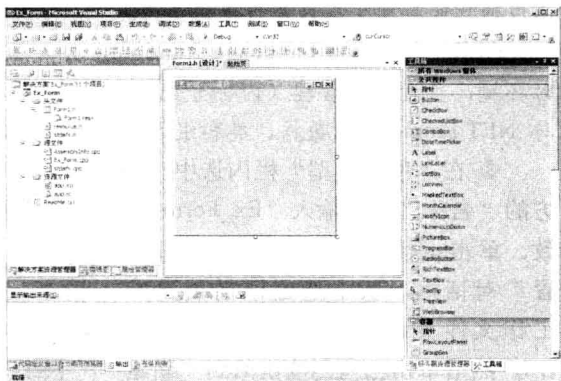


图 1-17 钉住“工具箱”窗口

“工具箱”显示了当前可以添加到项目中的控件或组件等项。每次返回编辑器或设计器时，“工具箱”都会自动滚动到最近选择过的类型节点和项。当焦点转移到其他编辑器、设计器或另一个项目时，“工具箱”中当前选定的内容也将随之改变。

⑥ 显示窗体“属性”窗口，将“工具箱”窗口设置成“自动隐藏”。滑出并滚动“工具箱”窗口的滚动条，找到“TextBox”控件项，将其拖放到窗体中。此时，系统将自动为程序创建一个 TextBox 类对象 textBox1。同样，找到“Button”控件项，将其拖放到窗体中，这样就会添加并产生一个 Button 类对象 button1。在 button1 属性窗口中，将其 Text 属性改为“Say”，如图 1-18 所示。

⑦ 单击“属性”窗口上部的事件按钮，切换到 button1 控件的事件处理属性页面，同时它会定位到按钮的 Click 字段项，在右侧组合框中输入 OnSay 并按【Enter】键，则此时打开 Form1.h 的代码窗口，并自动定位到该事件的处理函数上。需要说明的是，为了方便输入代码时，通常也将“属性”窗口设置成“自动隐藏”，如图 1-19 所示。

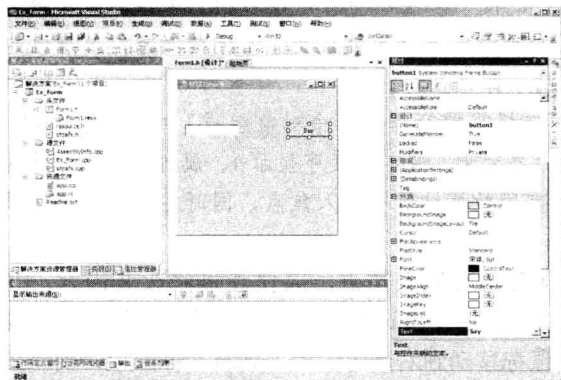


图 1-18 添加控件

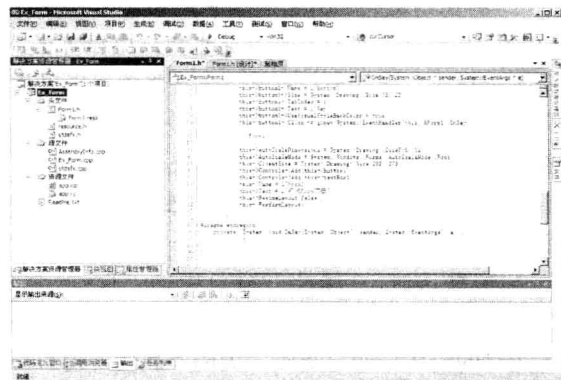


图 1-19 添加控件事件处理

有时，代码窗口中代码的字体和大小是不满足程序员的习惯的。这时，就需要选择“工具”→“选项”菜单命令，弹出“选项”对话框。在左侧“环境”节点下，选中“字体和颜色”，则右侧内容将随之改变。保留“显示其设置”的默认项“文本编辑器”，在“字体”下拉框中选择喜欢的字体，如幼圆，并设定其大小为 10，单击 **确定** 按钮，“文本编辑器”的字体和大小被重新

指定（后面还会详细说明）。

⑧ 在 OnSay 函数中添加下列代码（以后会解释）：

```
private: System::Void OnSay(System::Object^ sender, System::EventArgs^ e) {
    String^ strText = this->textBox1->Text->Trim();
    if ( String::IsNullOrEmpty( strText ) ) {
        MessageBox::Show( "编辑框中还没有内容!" );
        return;
    }
    MessageBox::Show( strText );
}
```

⑨ 选择“调试”→“开始执行（不调试）”菜单命令，或直接按【Ctrl+F5】快捷键，弹出项目已过期的重新生成的对话框，单击  按钮，系统开始生成创建，当在输出窗口中出现如图 1-20 所示的结果时，表示该 Form 项目已成功生成，并弹出窗体，如图 1-21（左）所示。在编辑框输入“Hello”，单击“Say”按钮，弹出消息对话框，如图 1-21（右）所示。关闭消息对话框和“初试 Form 项目”窗口，回到开发环境界面

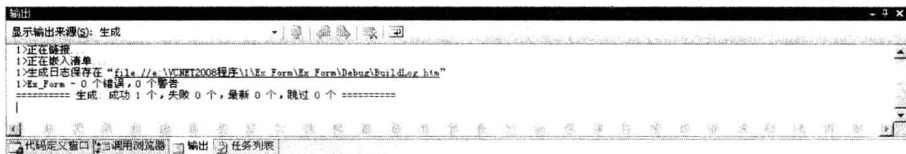


图 1-20 生成的过程

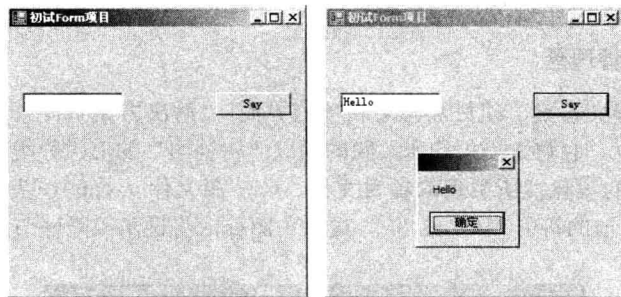


图 1-21 Ex_Form 运行结果

在输入程序代码时，为了更好地发挥 IntelliSense（智能感知）的作用，通常在编写代码时有意识地添加如 this 这样的指针（以后会讨论），以便能让 IntelliSense 根据上下文自动弹出相应的成员等内容，从而可快速输入代码。例如，图 1-22 就是一个例证。



图 1-22 使用 IntelliSense