



聂江武 杜 娜 周 峰 编著

中文版

Visual C++.NET 2005

**基础与实践教程**



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

# Visual C++.NET 2005 中文版

## 基础与实践教程

聂江武 杜 娜 周 峰 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

## 内 容 简 介

Visual C++ .NET具有复杂的应用程序框架以及对于.NET的支持, 本书着眼于最新的.NET版本Visual Studio 2005, 由浅入深地介绍Visual C++ .NET编程技术。

本书第1章、第2章着重介绍.NET基本框架、Visual Studio 2005新增功能及Visual Studio 2005的集成开发环境。第3章、第4章详细介绍Visual C++ 2005编程中的数据部分和运算部分, 以及面向对象编程技术。第5章完整介绍了.NET框架下MFC应用程序开发技巧。第6章至第9章着重介绍实际应用程序开发, 包括网络编程、数据库设计、图形图像编程, 并在最后给出一个综合开发案例, 提高读者的.NET综合开发能力。

本书采用理论与实例结合、相互渗透、逐步引导的方法, 通过实例剖析技术的具体应用, 使读者能十分容易地入门并逐步精通Visual C++ 的开发方法。

本书可作为初学者的入门教程, 更适用于有一定编程基础的读者, 通过对本书的学习, 读者可以迅速提高自己的编程水平, 达到实际商业开发的要求。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有, 侵权必究。

### 图书在版编目(CIP)数据

Visual C++ .NET 2005中文版基础与实践教程/聂江武, 杜娜, 周峰编著. —北京: 电子工业出版社, 2007.9

ISBN 978-7-121-04680-3

I. V… II. ①聂…②杜…③周… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2007)第100053号

责任编辑: 徐云鹏

特约编辑: 卢国俊

印 刷: 北京天竺颖华印刷厂

装 订: 三河市金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

北京市海淀区翠微东里甲2号 邮编: 100036

开 本: 787×1092 1/16 印张: 29.625 字数: 750千字

印 次: 2007年9月第1次印刷

定 价: 45.00元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至zlts@phei.com.cn, 盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线: (010) 88258888。

# 前言

本书旨在为所有需要了解**Visual C++ 2005**应用程序的设计、实现、部署和运行的读者做实用性介绍。

我们编写本书的目的十分明确。在过去的5年里，我们为各种各样的业界人士和开发技术人员培训.NET技术，还参与了几个大型.NET技术应用程序的设计、实现和部署。在此期间，我们发现还没有一本书能够连贯紧凑地从整体的角度来介绍**Visual C++ 2005**技术。**Visual C++ 2005**一方面是C++实现，提供功能强大的集成开发环境；另一方面，它提供了一些C++类库，其中最具影响力的当然是MFC。随着**Visual C++**的发展，MFC已经不是**Visual C++**唯一的类库了，其他还有ATL、STL等。此外，**Visual C++**还融合了大量的开发工具，特别是与Internet相关的开发支持。目前很多图书只有理论介绍，缺乏实际应用方面的内容。这就意味着如果想要全面学习并掌握最新的**Visual C++**编程技术的各个方面，就必须参加课程培训、买很多包含大量重复信息的图书或翻阅大型工具书。

而本书解决了上述问题，它提供了对**Visual C++.NET**从简单到高级技术的各方面的介绍，不仅有理论知识方面的介绍，还把我们多年的实际商业开发技巧和经验融合其中，使读者能够掌握一般实际项目中所需要的基本知识。

本书共分9章：

## 第1章 Visual C++ 2005编程

本章介绍了.NET基本框架、**Visual C++ 2005**的新增功能以及**Visual C++ 2005**编程环境的熟悉，为后面的学习打下坚实的基础。

## 第2章 Visual Studio 2005集成开发环境

本章详细介绍了**Visual Studio 2005**集成开发环境的组成部分，以及程序开发的完整过程及相应的技巧。

## 第3章 Visual C++ 2005编程基础

本章讲解了**Visual C++ 2005**中的数据类型、运算符、程序结构、数组、字符串、指针等程序设计基础，还就异常处理做了详细介绍。

## 第4章 面向对象编程

本章重点讲解面向对象编程的基本概念。

## 第5章 使用.NET框架

本章讲解了.NET框架以及利用.NET框架下的MFC进行C++程序设计。重点通过示例介绍了Windows程序设计。

## 第6章 Visual C++ 2005网络编程

本章重点讲解Winsocket编程、MFC套接字编程以及WinInet编程。主要通过两个网络编程的例子来让大家更好地学习网络编程。

## 第7章 访问数据库：ADO.NET与OLE DB

本章就数据库访问介绍了ADO.NET与OLE DB的相关知识，旨在通过这些介绍帮助读者掌握数据库的相关操作。

## 第8章 Visual C++.NET的图形图像编程

本章重点讲解了图形图像编程的基本概念和基本操作，并通过一些典型的例子详细展示了图形图像编程的魅力。

## 第9章 综合实例 魔方游戏

本章在综合前面的Visual C++.NET编程知识的基础上，通过对基于.NET框架的魔方游戏系统案例的讨论，使读者能够熟练掌握并使用.NET框架进行程序开发。

本书在编写过程中得到了青岛大学国际学院在技术上的大力支持，以下人员对本书的编写提出过宝贵意见并参与了本书的部分资料搜集工作，他们是孙更新、孙宁、王荣芳、李德路、李岩、周科峰、陈勇、高云、于凯、王春玲、姚国发，感谢北京美迪亚电子信息有限公司各位老师的帮助和指导。

由于时间仓促，加之水平有限，书中的缺点和不足之处在所难免，敬请读者批评指正。

---

为方便读者阅读，若需要本书配套资料，请登录“华信教育资源网”（<http://www.hxedu.com.cn>），在“资源下载”频道的“图书资源”栏目下载。

# 目 录

|                                           |            |
|-------------------------------------------|------------|
| <b>第1章 Visual C++ 2005编程 .....</b>        | <b>1</b>   |
| 1.1 .NET基本框架 .....                        | 1          |
| 1.2 公共语言运行时 (CLR) .....                   | 2          |
| 1.3 Visual C++ 2005的新特点 .....             | 3          |
| 1.4 学习Windows编程 .....                     | 7          |
| 1.5 Visual C++ 2005的第一个应用程序 .....         | 9          |
| 本章小结 .....                                | 13         |
| <b>第2章 Visual Studio 2005集成开发环境 .....</b> | <b>14</b>  |
| 2.1 什么是集成开发环境 .....                       | 14         |
| 2.2 集成开发环境IDE的使用 .....                    | 15         |
| 本章小结 .....                                | 30         |
| <b>第3章 Visual C++ 2005编程基础 .....</b>      | <b>31</b>  |
| 3.1 Visual C++ 2005中的数据、变量和运算 .....       | 31         |
| 3.2 选择结构和循环结构 .....                       | 57         |
| 3.3 数组、字符串和指针 .....                       | 83         |
| 3.4 程序结构 .....                            | 104        |
| 本章小结 .....                                | 137        |
| <b>第4章 面向对象编程 .....</b>                   | <b>139</b> |
| 4.1 类与类对象 .....                           | 139        |
| 4.2 类的构造函数 .....                          | 149        |
| 4.3 类的析构函数 .....                          | 156        |
| 4.4 类的私有成员 .....                          | 161        |
| 4.5 this指针 .....                          | 167        |
| 4.6 类的常对象 .....                           | 170        |
| 4.7 类对象数组 .....                           | 173        |
| 4.8 类的静态成员 .....                          | 174        |
| 本章小结 .....                                | 180        |

|                                         |            |
|-----------------------------------------|------------|
| <b>第5章 使用.NET框架 .....</b>               | <b>182</b> |
| 5.1 .NET框架下的MFC .....                   | 182        |
| 5.2 Windows窗体 .....                     | 195        |
| 5.3 对话框和控件 .....                        | 215        |
| 5.4 图形化输出 .....                         | 222        |
| 本章小结 .....                              | 229        |
| <b>第6章 Visual C++ 2005网络编程 .....</b>    | <b>230</b> |
| 6.1 Winsocket编程 .....                   | 230        |
| 6.2 MFC套接字编程 .....                      | 257        |
| 6.3 MFC WinInet编程 .....                 | 291        |
| 本章小结 .....                              | 298        |
| <b>第7章 访问数据库: ADO.NET与OLE DB .....</b>  | <b>300</b> |
| 7.1 ADO.NET技术基础 .....                   | 300        |
| 7.2 ADO.NET对象模型 .....                   | 304        |
| 7.3 OLE数据库模板 .....                      | 311        |
| 本章小结 .....                              | 349        |
| <b>第8章 Visual C++ .NET的图形图像编程 .....</b> | <b>350</b> |
| 8.1 数字图像的基本概念 .....                     | 351        |
| 8.2 设备相关位图 (DDB) 与设备无关位图 (DIB) .....    | 352        |
| 8.3 图像的特殊效果显示 .....                     | 361        |
| 8.4 图像几何变换 .....                        | 375        |
| 8.5 图像处理 .....                          | 382        |
| 8.6 三维图形绘制与OpenGL .....                 | 401        |
| 本章小结 .....                              | 428        |
| <b>第9章 综合实例: 魔方游戏 .....</b>             | <b>430</b> |
| 9.1 游戏实例功能介绍 .....                      | 430        |
| 9.2 主要功能及相关代码分析 .....                   | 431        |
| 9.3 具体代码实现 .....                        | 442        |
| 本章小结 .....                              | 468        |

# 第 1 章

## Visual C++ 2005编程

### 本章导读

本章将主要介绍用C++编写应用程序的一些本质概念。通过一个向导让你很快熟悉Visual C++ 2005集成开发环境（IDE）。IDE在操作上非常直观方便，在学习的过程中，你将能够自己轻松学会大部分内容。熟练使用IDE最好的方法是通过对简单程序的创建、编译以及执行的整个过程的处理。

### 重点提示

本章应重点掌握以下内容：

- Visual C++ 2005的基本组件是什么？
- .NET框架是由哪些部分组成的？它们各自能带来哪些好处？
- 解决方案和工程分别是什么？怎样创建它们？
- 关于控制台应用程序。
- 怎样创建并编辑一个程序？
- 怎样编译、链接和执行一个控制台程序？
- 怎样创建和执行一个Windows程序？



### 1.1 .NET基本框架

在大家已经决定使用.NET框架作为自己的开发平台的时候，我们首先需要确定希望创建何种类型的应用程序或组件。假设我们已经完成了这些次要的细节，所有的设计都做好了，规格说明也完成了，一切准备就绪。

正如微软其他.NET开发产品一样，.NET框架也是Visual C++ 2005的中心概念。.NET技术是微软公司推出的一个全新概念，“它代表了一个集合、一个环境和一个可以作为平台支持下一代Internet的可编程结构。”

.NET框架由两个中心元素组成：一个是公共语言运行时（CLR，即Common Language Runtime），应用程序在这里执行；另一个是一组.NET框架类库，这些.NET框架类库为应用程序在CLR运行时提供必要的函数性支持，不管是用哪种编程语言写的程序，C++也好，C#也好，只要是任何支持.NET框架的语言都可以使用.NET类库。

这样就可以通过Visual C++ 2005来开发两种不同的基本的C++应用程序。你可以编写

应用程序按照原本的方式在你的机器上执行，这种应用程序被称为原生C++程序。你可以在被ISO/ANSI语言标准定义版本的C++中编写原生C++程序。你同样也可以在被称为C++/CLI的扩展的C++版本中编写应用程序，这样的程序可以在公共语言运行时CLR控制下运行，这些程序将被称做CLR程序或者C++/CLI程序。

Microsoft .NET框架是一个平台，它提供了一个多语言编程环境，同时集成了应用程序和服务，为Internet规模应用程序的部署和操作提供了高度的灵活性。.NET框架主要由以下几个主要部分组成：.NET支持的语言、公共语言执行环境（Common Language Runtime）、统一类库的分层集合和ASP.NET。.NET的最终目标就是让用户在任何地方、任何时间，以及利用任何设备都能访问所需的信息、文件和程序。

.NET开发平台包括.NET框架和.NET开发工具等组成部分，如图1-1所示。.NET框架（Framework）是整个开发平台的基础，包括公共语言运行库和框架类库。.NET开发工具包括Visual Studio.NET集成开发环境和.NET编程语言。其中，Visual Studio.NET集成开发环境用来开发和测试应用程序。.NET编程语言包括Visual Basic、Visual C++和新的Visual C#等用来创建运行在公共语言运行库（CLR）上的应用程序。



图1-1 .NET基本框架

.NET框架严格来说不能算是Visual C++ 2005的部分，而更应该看做是Windows操作系统的组成部分，它使建立应用软件和Web服务变得更容易。.NET框架在代码的可靠性和安全性方面带来了重要的优势，同样在代码集成方面也有更强的能力，可以使你的C++代码集成到20多种其他的编程语言写的代码中去，只要这些代码也是基于.NET框架即可。基于.NET框架还有另外一个优势就是运行开销很小，只不过在主要的环境中你是觉察不到的。

## 1.2 公共语言运行时（CLR）

公共语言运行时CLR为更多的包括Visual Basic、C#、C++等高级语言编写的程序提供标准的执行环境。CLR规范现在已被欧洲计算机制造商协会（ECMA）标准化为公共语言基础结构，ECMA-335（CLI）标准介绍了一种独立于语言的运行时引擎，它能够将没有生命力的元数据转换成自汇编的、强大且类型安全的软件系统。填充这些软件系统的“托管组件”可以用各种不同的计算机语言编写，而且仍然能够很好地进行交互，并具有很高的保真度。这

种合作精神的秘密就在于CLI的共享运行时基础结构，它不仅包括标准化的元数据和中间语言表示等低层功能，还包括普遍可用的类库等与生产效率有关的高层功能。为了使用现有代码库，CLI标准还尽可能详细地描述了非托管软件如何能够与托管组件安全共存，使它们可以无缝地共享计算资源并共同承担责任。

现在让我们来看看共享源代码CLI实现可能令人感兴趣的几个方面：

- 在提高可移植性和使代码库更便于访问方面，此代码与用于Microsoft商业CLR实现的代码在实现方面有着显著差别。但是，如果你对以下内容的细节感兴趣，那么此软件的发布肯定会吸引你的注意：虚拟对象系统、对象架构的设计方法、JIT编译器和内存回收器的工作原理，或者Microsoft Visual Studio的工作原理等。

- 如果你是一位教师，喜欢将此代码库作为示例开发相关主题的课件；或者你是一位研究人员，所从事的研究或原型设计能够从基于此源代码的构建中获益，那么该软件的发布将会令你感到欣喜。例如，使用以程序为导向的优化方法（例如数据和代码布局或代码压缩）的用户会有效利用运行时的反射功能。另外，编译器的内部教程可能会引用C#、Microsoft JScript或运行时JIT编译器作为示例。可能的主题还有很多很多。

- 最后，如果你正在考虑实现自己的CLI版本，或者需要进一步了解编译器和工具针对通用中间语言或基本类库的方法，此软件将是不可或缺的指南。共享源代码CLI提供了完整的版本和自由的许可条款，这对于每个人来说都是一个珍贵的实例和测试案例。

CLI本质上可以说是一种特定的虚拟机环境，它使我们可以用不同的高级编程语言编写应用程序，却可以不加修改地在不同的系统环境下执行。CLI为虚拟机规范了一个标准的中间语言，这样高级语言编写的源程序就可以直接编译了。在.NET框架下，这个中间语言就是指MSIL（Microsoft Intermediate Language）。



## 1.3 Visual C++ 2005的新特点

在 Visual C++ 2005中，C++语言已得到扩展，可支持开发在有垃圾回收的虚拟机上运行的组件和应用程序。这些添加的内容已获得C++/CLI标准协会的批准。Visual C++ 2005支持面向Microsoft实现的垃圾回收虚拟机.NET Framework CLR（公共语言运行库），但它也可用于面向符合CLI标准的任何平台。

现在，.NET编程的Visual C++ 2005中支持的C++语法要比Microsoft Visual C++ 2003中介绍的C++托管扩展支持的语法自然得多。虽然仍支持C++托管扩展，但它已被否决而多用新的语法。有关更多信息，请参见[ms-help://MS.VSCC.v80/MS.MSDN.v80/MS.VisualStudio.v80.chs/dv\\_vclang/html/1e400ee6-3ac9-4910-a608-9d3d5993e423.htm](http://ms-help://MS.VSCC.v80/MS.MSDN.v80/MS.VisualStudio.v80.chs/dv_vclang/html/1e400ee6-3ac9-4910-a608-9d3d5993e423.htm)。

### 1. 常规库更新

在混合模式下，在托管构造函数之前调用本机构造函数。对于单个可执行文件，将在任何托管全局对象的构造函数之前调用所有本机全局对象的构造函数，而不考虑本机和托管对象是否存在于相同的编译单元中。

因此，依赖于在本机构造函数之前调用的托管构造函数的应用程序可能会以错误的方式运行。如果全局对象之间存在依赖项，则应确保仅存在从托管对象到本机对象的依赖项。确

保某个对象为托管对象或本机对象的最佳方式是将其放入以该方式生成的编译单元中。

从Visual C++ 2005开始, 标准C++库和ATL库、MFC库以及CRT库都是带有清单的并行共享程序集。编译器和链接器设置现在默认为使用mt.exe生成清单。这将确保所有Visual C++可执行文件都具有引用它们与ATL、MFC和CRT有依赖关系的清单。

Visual C++ 2005支持将标准的C++库以及ATL、MFC和CRT库用做使用/clr编译的混合程序集。当代码中混合有本机代码和基于MSIL的\_\_clrcall代码时, 这些混合库允许使用它们所有的现有函数。

Visual C++ 2005中的库支持更改解决了存在于Visual C++ .NET和Visual C++ .NET 2003的混合DLL中的“加载程序锁定”问题。但仍然有一个限制, 即DllMain代码决不可以访问CLR。

在Visual C++ 2005中, pure模式编译 (/clr:pure) 不支持ATL和MFC库。

## 2. MFC和ATL的新功能

### (1) ATL

新的全局函数: AtlComPtrAssign AtlComQIPtrAssign

新的符号: ATL MIXED

新的方法: CArray::RelocateElements Sids::NetworkService

已否决的函数: 由于安全方面的问题, 某些ATL函数现在已被否决, 标记为\_ATL\_INSECURE\_DEPRECATED。可以使用\_ATL\_SECURE\_NO\_DEPRECATED来取消在使用这些函数时生成的编译器警告。

已否决的全局函数: 以下函数在Visual C++ .NET 2002及更高版本中已过时, 但考虑到与Visual C++以前版本所创建项目的向后兼容性, 仍然可用, 如AtlModuleRegisterTypeLib AtlModuleUnregisterTypeLib、AtlModuleRegisterServer、AtlModuleUnregisterServer、AtlModuleUnregisterServerEx。

### (2) ATL Server

已将/nowchar\_t添加到sproxy.exe。

已否决的函数: 已将\_set\_security\_error\_handler从CRT移除, 所以应当不再使用AtlsSecErrHandlerFunc。

### (3) MFC

MFC Windows窗体支持: 已将Windows窗体支持添加到MFC。使用MFC Windows窗体支持类, 可以在MFC应用程序内承载Windows窗体控件, 就如同MFC对话框或视图内的ActiveX控件。此外, 可以将Windows窗体的窗体承载为MFC对话框。

随Windows XP一起提供的GDI+: GDI+ 包括在Windows XP中, 可用做Windows NT 4.0 SP6、Windows 2000、Windows 98和Windows Me的可再发行组件。要下载最新的可再发行组件, 请参见<http://www.microsoft.com/msdownload/platformsdk/sdkupdate/psdkredist.htm>。

新的方法: CDC::SetGraphicsMode、CDC::GetGraphicsMode、CDC::SetWorldTransform、CDC::ModifyWorldTransform、CDC::GetWorldTransform、CStdioFile::Seek。

已否决的方法: AfxOleTypeMatchGuid、CArchive::IsByteSwapping、CInternetConnection::QueryOption、CInternetFile::QueryOption。由于安全方面的问题, 某些MFC函数现在已被否

决, 标记为 `_AFX_INSECURE_DEPRECATED`。

### 3. CRT新功能

(1) 添加了若干函数的安全版本。这些函数以更好的方式处理错误, 并强制对缓冲区实施更严格的控制, 以避免常见的安全缺陷。新的安全版本由 `_s` 后缀标识。

(2) 现有安全性较低的许多函数版本已被否决。若要禁用否决警告, 请定义 `_CRT_SECURE_NO_DEPRECATED`。

(3) 许多现有函数现在会验证其参数, 并在传递了无效参数时调用无效参数处理程序。

(4) 现在, 许多现有函数在过去没有设置 `errno` 的地方对其进行了设置。

(5) 添加了具有 `integer` 类型的 `typedef errno_t`。每当函数返回类型或参数处理来自 `errno` 的错误代码时, 就会使用 `errno_t`。 `errno_t` 替换 `errno`。

(6) 与区域设置相关的函数现在有了新的版本, 这些版本将区域设置作为参数, 而不是使用当前的区域设置。这些新的函数具有 `_l` 后缀。添加了若干新函数, 以使用区域设置对象。新的函数包括 `_get_current_locale`、`_create_locale` 和 `_free_locale`。

(7) 添加了新函数, 以支持对文件句柄的锁定和取消锁定。

(8) `_spawn` 系列函数不会像在早期版本中那样, 在成功时将 `errno` 重置为零。

(9) 提供了可用于指定参数使用顺序的 `printf` 系列函数版本。请参见 `printf_p` Positional Parameters、`_cprintf_p`、`_cprintf_p_l`、`_cwprintf_p`、`_cwprintf_p_l`、`_printf_p`、`_printf_p_l`、`_wprintf_p`、`_wprintf_p_l`、`_sprintf_p`、`_sprintf_p_l`、`_swprintf_p`、`_swprintf_p_l`、`_fprintf_p`、`_fprintf_p_l`、`_fwprintf_p`、`_fwprintf_p_l`、`_vprintf_p`、`_vprintf_p_l`、`_vwprintf_p`、`_vwprintf_p_l`、`_vsprintf_p`、`_vsprintf_p_l`、`_vswprintf_p`、`_vswprintf_p_l` 和 `_vfprintf_p`、`_vfprintf_p_l`、`_vfwprintf_p`、`_vfwprintf_p_l`。

(10) 现在支持 Unicode 文本格式。函数 `_open` 支持 `_O_TEXTW`、`_O_UTF8` 和 `_O_UTF16` 属性。 `fopen` 函数支持指定 Unicode 格式的 “`ccs=ENCODING`” 方法。

(11) 现在提供了一个用托管代码 (MSIL) 生成的 CRT 库的新版本, 可在用 `/clr` (公共语言运行库编译) 选项进行编译时使用。

(12) `_fileinfo` 已被移除。

(13) `time_t` 的默认大小现在为 64 位, 这样就可以将 `time_t` 和几个时间函数的范围扩展到 3000 年。

(14) CRT 现在支持按每个线程设置区域设置, 添加了 `_configthreadlocale` 函数以支持此功能。

(15) 添加了 `_statusfp2` 和 `_control87_2` 函数, 以允许对 x87 和 SSE2 浮点处理器上的浮点控制字进行访问和控制。

(16) 添加了 `_mkgmtime` 和 `_mkgmtime64` 函数, 以支持将时间 (`struct tm`) 转换为格林威治标准时间 (GMT)。

(17) 对 `swprintf` 和 `vswprintf` 进行了更改以更好地符合标准。

(18) 新的头文件 `INTRIN.H` 提供了某些内部函数的原型。

(19) `fopen` 函数现在具有一个 `N` 属性。

(20) `_open` 函数现在具有一个 `_O_NOINHERIT` 属性。

(21) 溢出时, `atoi`函数现在返回`INT_MAX`并将`errno`设置为`ERANGE`。早期版本中未定义溢出行为。

(22) `printf`系列函数支持根据ANSI C99标准, 使用格式类型说明符“%a”和“%A”实现的十六进制浮点输出。

(23) `printf`系列现在支持“ll” (`long long`) 大小的前缀。有关更多信息, 请参见Size and Distance Specification。

(24) 已将`_controlfp`函数进行了优化, 以提供更好的性能。

(25) 添加了某些函数的调试版本。有关更多信息, 请参见`_strdup_dbg`、`_wcsdup_dbg`、`_tempnam_dbg`、`_wtempnam_dbg`、`_getcwd_dbg`、`_wgetcwd_dbg`、`_getdcwd_dbg`、`_wgetdcwd_dbg`和`_fullpath_dbg`、`_wfullpath_dbg`。

(26) 添加了`chgsignl`和`cpysignl` (`long double`版本)。

(27) 将`_locale_t`类型添加到了类型表。

(28) 添加了新的宏`_countof Macro`用于计算数组中的元素个数。

(29) 在每个函数主题中, 添加了一节有关.NET Framework等效项的内容。

(30) 现在, 几个字符串函数可以选择在输出缓冲区太小时截断字符串, 而不是失败; 请参见`_TRUNCATE`。

(31) `_set_se_translator`现在需要使用/EHa编译器选项。

(32) 在/Za下 (对于C代码) 以及手动设置`__STDC__`时 (对于C++代码), `fpos_t`现在变为`__int64`。以前它是一个结构。

(33) `_CRT_DISABLE_PERFCRIT_LOCKS`可以提高单线程程序的I/O性能。

(34) 在对符合ISO C++名称的优选中, 已将POSIX名称否决 (例如, 使用`_getch`而不是`getch`)。

(35) 新的链接选项.obj文件可用于pure模式。

(36) `_realloc`组合了`realloc`和`calloc`的功能。

#### 4. 标准C++库的新增功能

(1) 标准C++库现支持对迭代器的错误使用进行检测。

(2) `for each`关键字用于处理STL集合。

#### 5. 新的C++支持库

支持用C++托管编程的新库: C++ Support Library、功能`auto_gcroot`、`auto_handle`、`lock`、`call_in_appdomain Function`和`com::ptr`。

#### 6. Visual C++代码DOM

(1) `CppCodeGenerator`提供生成符合Visual C++ 2005中CLR功能的C++代码的函数。

(2) `CppCodeGenerator 7`提供生成符合C++托管扩展语法 (Visual C++ 2005之前的CLR编程语法) 的C++代码的函数。



## 1.4 学习Windows编程

在Windows下执行的交互式应用程序通常来说有两个基本的方面：首先需要创建图形用户界面（GUI）供用户交互；还有就是要有能处理这些交互的应用程序函数。Visual C++ 2005为开发Windows应用程序的这两个方面提供了大量的辅助功能。在本章的后面你将会发现，你根本不用写一行代码就可以创建一个带有图形用户界面GUI的Windows程序，所有的这些创建GUI的基本代码都由Visual C++ 2005自动产生了。当然，还可以根据你自己的需要对这些基本代码进行扩展和修改，以满足更复杂的程序设计需要。正因为这样，你首先要学习原本的C++和C++/CLI版本的语言（这时候不考虑Windows编程），当使用C++比较顺手的时候，就可以尝试用C++和C++/CLI来开发更丰富完整的Windows应用程序。这意味着在你学习C++的时候，就要在仅仅是命令行的形式进行输入/输出的方式下进行C++程序设计。当你已经很习惯这种相当有限的输入/输出方式的时候，就能够集中精力解决C++语言工作的具体细节以及避免图形用户界面GUI的创建和控制之类的固有的复杂性。在你已经很熟悉C++的时候，就会发现可以很轻松自然地将C++应用到Windows应用程序开发中去。

### 1.4.1 学习C++

Visual C++ 2005完全支持由两个独立标准定义的两个版本的C++：

- ISO/ANSI C++标准是为执行普通的应用程序——非托管的C++而设计的。这种版本的C++在大多数的计算机平台都能得到支持。

- C++/CLI标准是专门为编写CLR程序而设计的，它是对ISO/ANSI C++的一种扩展。

在后面的章节中将会学习到C++语言，因为C++/CLI是对ISO/ANSI C++的一种扩展，我们首先学习ISO/ANSI C++的一些内容，然后再加上C++/CLI特点介绍。

用C++/CLI来编写程序可以让你充分利用.NET框架性能方面的很多优势，有些事情是没办法通过ISO/ANSI C++来实现。虽然C++/CLI是对ISO/ANSI C++的一种扩展，但是要让程序能够在CLR上运行就意味着你的程序必须遵循CLR程序的要求。有关这方面的一个例子就是ISO/ANSI C++提供的方便的动态内存分配和释放就不能被CLR很好地兼容，如果想用CLR机制来管理内存，就必须使用C++/CLI类库，而不是ISO/ANSI C++类库。

### 1.4.2 控制台应用程序

同开发Windows应用程序一样，Visual C++ 2005也允许编辑、编译、测试C++程序，并不需要Windows程序的一些装备。也就是说应用程序实质上是基于字符和命令行的程序。在Visual C++ 2005里面我们把这些程序称为控制台应用程序，因为你是在字符状态下通过键盘和屏幕来同这些程序打交道。

看起来好像编写控制台应用程序跟Windows应用程序没有什么关系，但是当你开始学习C++（在学习Windows编程之前一定要学）的时候，这是非常必要的。即使一个很简单的Windows程序也有大量的代码，在学习C++的点点滴滴的时候不用被Windows的这种复杂性所困扰，这一点很重要。在刚开始的时候熟悉一下C++是如何工作的，在你写大量的重量级的Windows程序之前，最好是花一定的时间在轻量级的控制台应用程序上。

当在学习C++的时候,你就可以集中精力于语言特点上而不必关心你所操作的环境。你写控制台应用程序的时候就只有文本接口,但这对于理解C++完全足够了,因为定义这种语言的时候就没有涉及图形处理能力。自然地,在后面当你开始在一般的C++应用程序中或是在CLR的Windows程序中使用MFC的时候,本书将提供大量的图形用户界面编程。

这里有两种截然不同的控制台应用程序可以使用:一种是Win32控制台应用程序,是基于ISO/ANSI C++标准的;另外一种是CLR控制台应用程序,它是用C++/CLI标准工作的。

### 1.4.3 Windows编程概念

我们这里的Windows编程方法就是使用Visual C++ 2005提供的所有工具。Visual C++ 2005提供的项目创建工具可以为各种各样应用程序(也包括基本的Windows程序)自动生成基本的代码框架。创建一个项目就是所有应用程序或是用Visual C++ 2005开发组件的一个开始,要了解这些项目是如何工作的,就要看看创建一些例子的机制,包括后面要讲到的Windows编程的一些纲要性的内容。

一个Windows程序有着与在命令行下执行的控制台程序不同的结果类型,它更复杂。在控制台程序中,你可以从键盘上输入,可以直接通过命令行的形式输出,而一个Windows程序要访问计算机的输入/输出设备,只能通过Windows操作系统提供的函数进行,不允许直接访问硬件资源。因为在Windows下可能在同一时间几个程序都是活动的,那Windows就要确定当前的输入(例如鼠标按键或是键盘的按键操作)是给哪个程序的,哪个程序与这个信号相对应。这样Windows操作系统首要的就是要控制同用户的所有交互通信。

同样,在Windows应用程序和用户之间的交互接口上还存在一个问题,就是用户会在任何可能的时候给出各种各样的不同的输入。用户可能选择任何数量的菜单选项,单击工具栏按钮,或者在Windows应用程序中的任何位置单击鼠标。一个设计优良的Windows应用程序就必须随时准备着处理任何时候可能发生的各种可能类型的输入,因为事先根本就无法知道

将要出现什么类型的输入。这些用户行为在第一个实例里面被操作系统捕获,并被Windows视为事件。来自于应用程序用户接口的事件就会导致一段专门的程序代码被执行。用户活动的次序决定了程序执行的过程是什么样子。用这种方式工作的程序被称为事件驱动程序,这和传统的过程程序有所不同,传统的过程程序仅仅只有一种执行次序。过程程序的输入被程序代码所控制并且只能在程序允许的时候才能出现,而Windows程序主要是由代码段来响应用户行为导致的或是Windows自身导致的事件。这种程序结构如图1-2所示。

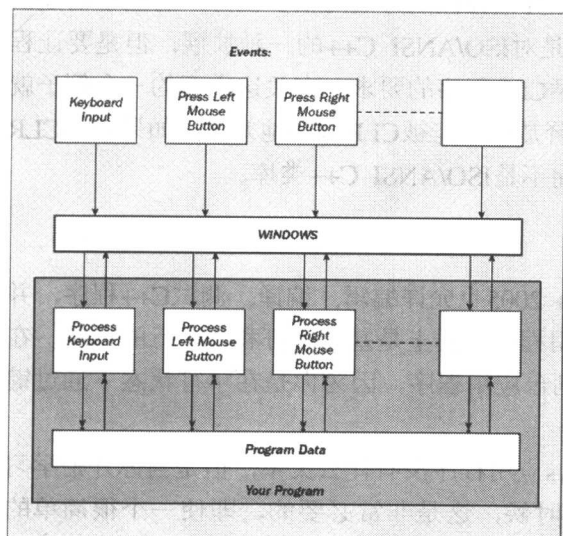


图1-2 Windows程序结构



## 1.5 Visual C++ 2005 的第一个应用程序

### 1. 创建第一个Visual C++ 2005可执行程序

从程序中打开Microsoft Visual Studio 2005，将会显示一个空白的窗口，如图1-3所示。

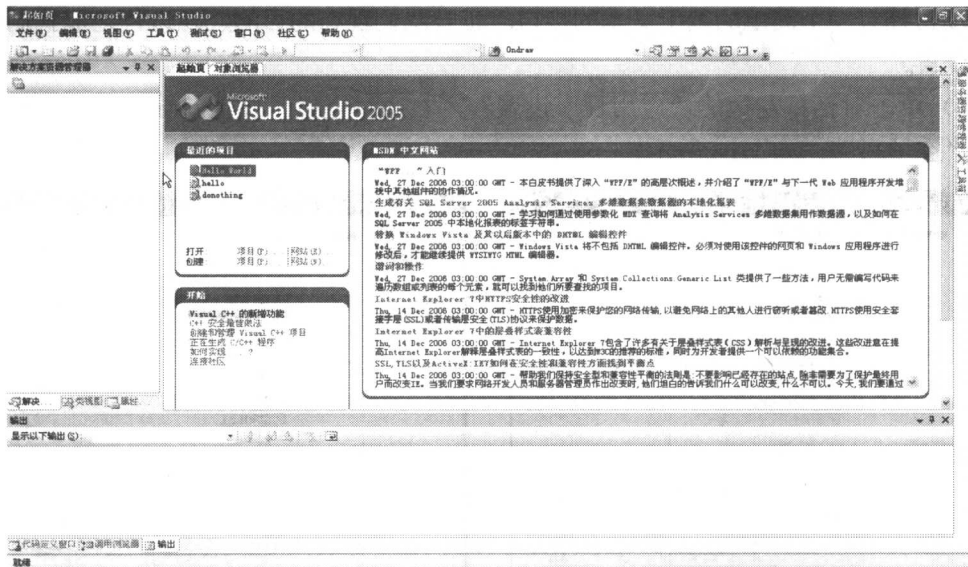


图1-3 Visual Studio 2005开始界面

这个窗口就是强大的Visual Studio .NET集成开发环境（IDE）。它包含了你将要用来创建功能丰富、简单好用的应用程序所需的所有工具。

### 2. 创建一个工程

首要的是为Hello World程序创建一个项目。通过以下的步骤来创建这个项目：

(1) 在文件菜单下单击“新建►项目”（或者可以按下Ctrl+Shift+N组合键）。如图1-4所示就是“新建项目”对话框。

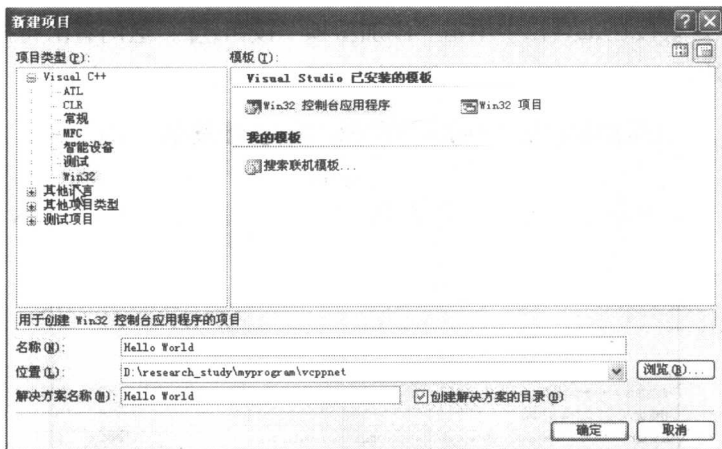


图1-4 “新建项目”界面

(2) 在“项目类型”里面选择 Visual C++ 中的 Win32，然后在右边的模板类型里面选择“Win32 控制台应用程序”。然后在“名称”框中写上项目名称：Hello World。

(3) 在“位置”框中为你的新项目选择好路径：直接在组合框中输入路径，或者通过“浏览”按钮找到正确的路径。

(4) 路径输入完毕后，单击“确定”开始 Win32 应用程序向导，如图 1-5 所示。

该项目拥有一个简单的基于文本的接口，正合我们现在的目的。在进行之前我们还是要做一些设置。

(5) 单击对话框左边的“应用程序设置”，右边对应显示设置向导，在“附加选项”中选择“空项目”，如图 1-6 所示。

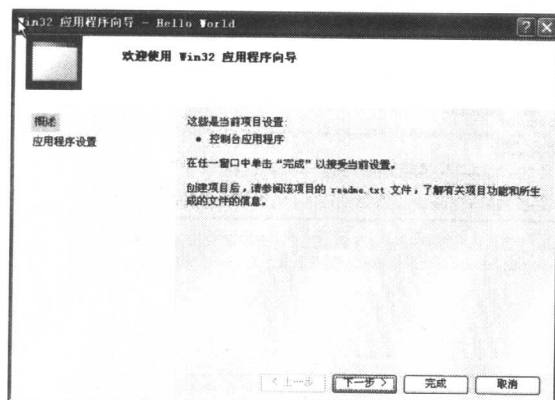


图1-5 应用程序向导

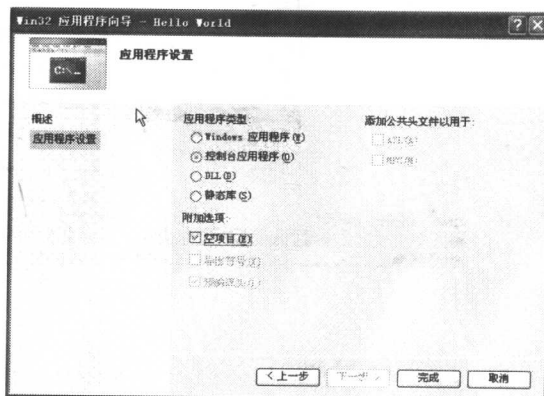


图1-6 应用程序向导

(6) 单击“完成”来创建项目。

这个向导将为控制台项目正确初始化所有的编译器设置。

### 3. 在项目中添加C++源文件

一个空项目本身没有什么特别令人兴奋的地方，让我们向项目里面添加点新的 C++ 源文件。正如通常在 Visual Studio .NET 中所见，你将有很多方法来做这同一件事。

(1) 要么是在解决方案浏览器中右键单击 Hello World 图标，单击“添加”，然后单击“新建项”；要么是直接在工具栏中单击“添加新项”按钮 。这两者都将打开“添加新项”对话框，如图 1-7 所示。

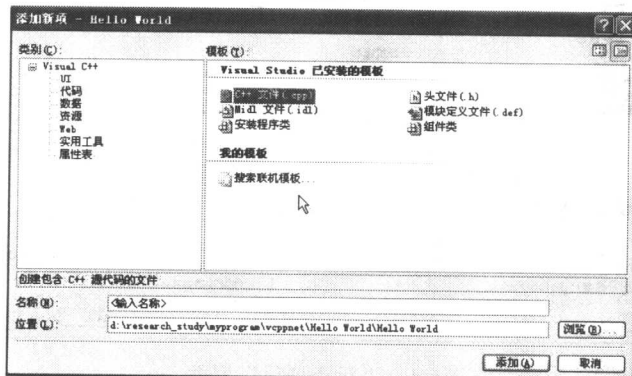


图1-7 添加新项