



.NET 平台研究与开发丛书

# Visual C++ .NET 核心编程与开发实例

■ 刘洪波 刘洪波 刘洪波 编著

本书附盘可从本馆主页 <http://lib.szu.edu.cn/>  
上由“馆藏检索”该书详细信息后下载，  
也可到视听部复制

人民邮电出版社

## 图书在版编目 (CIP) 数据

Visual C++.NET 核心编程与开发实例/前沿软件研究室编著. —北京: 人民邮电出版社, 2001.10

(.NET 平台研究与开发丛书)

ISBN 7-115-09693-7

I. V... II. 前... III. C 语言—程序设计 IV. TP312

中国版本图书馆 CIP 数据核字 (2001) 第 066744 号

.NET 平台研究与开发丛书

### Visual C++ .NET 核心编程与开发实例

- ◆ 编 著 前沿软件研究室 刘 涛 陈鹏远  
责任编辑 杨 璐
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子函件 315@ ppth.com.cn  
网址 <http://www.ppth.com.cn>  
读者热线:010-67129212 010-67129211(传真)  
北京汉魂图文设计有限公司制作  
北京顺义向阳胶印厂印刷  
新华书店总店北京发行所经销
- ◆ 开本:787×1092 1/16  
印张:34  
字数:825 千字 2001 年 10 月第 1 版  
印数:1-5 000 册 2001 年 10 月北京第 1 次印刷

ISBN 7-115-09693-7/TP·2496

定价:58.00 元(附光盘)

本书如有印装质量问题,请与本社联系 电话:(010)67129223

# 第 1 章 Visual C++开发环境和 C++

进入 21 世纪,随着 CPU 处理速度的飞快增长和 Windows 操作系统的逐渐升级,面向对象的可视化编程技术越来越受到人们的重视,而微软公司推出的 Visual Studio 开发工具更为人们所青睐。

在微软公司推出 Windows 2000 操作系统后,为了充分利用该操作系统的优异性能,微软公司对 Visual Studio 工具进行了更深层次的开发,在 2000 年的 7 月,推出了一个全新的开发平台 Visual Studio.NET。在此平台上,包括 Visual C++、Visual Basic、Visual FoxPro 和 C# 等开发工具。

## 1.1 Visual Studio.NET 的特点

Visual Studio.NET 提供了最快捷的方式用来建立企业 Web 应用软件。使用 Visual Studio.NET 开发工具,能够进行可视化的设计,调试和配置端到端的解决方案,以用来解决所遇到的商业问题。通过开发服务器端的 Web 服务器组件,并且使这些组件通过 XML 和 HTTP 服务被其他开发者或者用户接受,就可以获得更多的商业机会。在非常完整的集成的 IDE 开发环境中,用户可以选择最适合的语言。

Visual Studio.NET 的开发环境是高度集成、共享式的开发环境。Web 框架可以用来可视化地建立交互式的 HTML 页面。Web 服务用来创建和销毁中间层次的运行在 HTTP 上面的对象。通过服务器浏览器,可以迅速地访问服务器的组件,包括消息队列和数据库。可视化组件设计可以可视化地创建服务器端的组件。集成、共享的环境可以增加软件的创建效率,使用户非常容易建立编程语言相互交叉的解决方案。可扩展和自动操作功能使用户能定制所需要的 IDE 开发环境。

这套开发软件为 C++语言的扩充提供了一个便捷的方法,使得现有的 C++代码可以在 .NET 平台上运行。特别是在 ATL 服务器和 COM 组件两个方面有较大的扩充。对于 ATL 服务器,能够使用熟悉的 ATL 编程模板对 Web 应用程序进行灵活的开发。对于属性化编程,可以使用少量的代码迅速地建立 COM 组件。下面具体介绍该软件的主要特点。

### 1.1.1 C++的编译器、连接器和 C++语言

在编译器方面,新增加了一些选项,如 /CLR、/Fx、/GH、/GL、/GS、/noHRESULT、/RTC、/showIncludes、/WL、/Wp64、/Y-、/Zc:forScope、/Zc:wchar\_t。从编译器中删除了 /Gi 和 /GD

两个选项。/O1 和/O2 使用/GF 代替/Gf, /Ox、/O1 和/O2 使用/b2 代替/Ob1。/QIfist 已经被存入档案。/w 允许指定特殊警告的行为。/Wall 允许将所有警告都设置成开启状态, 包括那些默认状态是关闭状态的警告。此外, 还引入了一个新的关键字\_unaligned, 另外还有一些关键字涉及到 C++ 的扩充。

在连接器方面, 增加了一些新的链接选项, 如/ALLOWBIND、/ASSEMBLYMODULE、/ASSEMBLYRESOURCE、/IDLOUT、/IGNOREIDL、/LTCG/MIDL、/NOASSEMBLY、/PDBSTRIPPED、/TLBID、/TLBOUT、/TSAWARE。CVPACK 的功能被取消。/DEBUGTYPE、/GPSIZE 和/WARN 选项被删除, /LINK50COMPAT 也被删除。原先 Visual C++ 6.0 中动态链接库的推迟加载辅助函数也有些变化。还有一些其他的更改, 读者可以参考帮助文件。

C++ 语言方面, 引入了一些新的关键字和数据类型。关键字有\_alignof、\_declspec (align)、\_declspec (deprecated)、\_declspec (noinline)、\_if\_exists、\_if\_not\_exists、\_interface、\_ptr64、\_super、\_pragma、\_sealed 等。template 关键字允许带参数。数据类型有\_m64、\_m128、\_m128i 和\_m128d, 这些都有特殊的用途, 还有 wchar\_t 数据类型。Visual C++ 现在支持非数字项 (称为 NAN 或 nan) 的比较, 非数字项已经成为一个任意数据类型。

### 1.1.2 属性化编程

属性化编程的特性是使用少量的代码迅速地建立 COM 组件。在开发环境中, 又新增加了一些属性。设计这些属性是为了给用户提供更有效更快捷的方式, 以简化 Visual C++ 中的 COM 编程。这些属性就像 C++ 关键字一样, 被使用在源文件中, 并且被编译器解释。这些属性还可以改变代码的行为, 甚至插入一些额外的代码来完成一些基本的工作, 例如 ActiveX 控件的执行、类的创建或者格式化数据库命令。

### 1.1.3 类库

在 Visual C++.NET 中, 类库增加了很多新的特点, 发生了一些新的变化, 主要的类库有 Active Template Library (ATL)、ATL Server、C Run-Time Library、Microsoft Foundation Classes (MFC)、OLE DB Templates、Shared Classes、Standard C++ Library 和 Old iostream Library。下面分别介绍。

ATL 活动模板库。ICollectionOnSTLImpl、IEnumOnSTLImpl、CcomEnumImpl、CComEnumOnSTL 和 CComEnum 支持 collections 和 enumerators。Cadapt、Utility classes、IaxWinAmbientDispatch、IaxWinHostWindow、CAxWindow2 支持 ActiveX 控件。此外, ATL 中还增强了 CComBSTR 和 CcomVariant 类的功能, 增加了新宏 OBJECT\_ENTRY\_NON\_CREATEABLE\_EX。

CcomCoClass::CreateInstance 简化了对象创建。\_ATL\_DEBUG\_INTERFACES 改良了调试支持。注册表支持 REG\_MULTI\_SZ 值。Cacl、Cdacl、Csac1、CsecurityAttributes、CsecurityDesc、Csid、CtokenGroups 和 CtokenPrivileges 是新增加的安全类。CsimpleArray、CsimpleMap、CcomSafeArray 和 CcomSafeArrayBound 是新增加的数组管理类。CcomCurrency 是通用类。

ATL 服务器是一套本来就有的类库, 开发者可以使用这套类库创建 Web 应用程序、Web

服务器程序和其他服务器程序。其中许多类都可以用在客户端应用程序或者组件中。ATL 服务器是 Visual C++ 的新特性。

在 C Run-Time 库中，增加了一些错误检查函数，可以使用 `_resetstkoflw` 对堆栈溢出进行恢复。还增加了一些新的函数：`_ctime64`、`_wfindnext64`、`_gmtime64`、`_wstat64`、`_wctime64`、`_fstat64`、`_localtime64`、`_time64`、`_findfirst64`、`_ftime64`、`_mktime64`、`_utime64`、`_wfindfirst64`、`_futime64`、`_stat64`、`_wutime64` 和 `_findnext64`，它们对日期的支持超过 2038 年。可以使用 `_CrtSetDbgFlag` 设置调试堆检查。`_CrtReportBlockType` 用来测定 block 块和 subblock 子块的类型。还增加了数学常量。`_cwprintf`、`_wprintf`、`_putws` 和 `_wtof` 是新的宽字符函数，`_strtoi64`、`_strtoui64`、`_wctoi64` 和 `_westoui64` 是新的 `_int64` 版本的串函数，`_getwch`、`_ungetwch()`、`cgetws`、`_getwche` 和 `_wcscanf` 是新的 Unicode 版本的控制台函数，`_scprintf`、`_scwprintf`、`_vscprintf` 和 `_vscwprintf` 是串函数，可以计算那些格式化输出的字符串的个数。`_set_security_error_handler` 用来注册安全错误处理，并且还有一些数据队列函数。

MFC 类库有了很大的变化。对于消息处理函数的返回类型和参数类型，MFC 提供了严格的检查，这种新特性通过出错信息来标识那些潜在的不安全的消息处理。例如，在 Visual C++ 以前的版本中，对于 `ON_MESSAGE` 或者 `ON_REGISTERED_MESSAGE` 消息函数，开发者使用 `void` 返回类型代替原有的 `LRESULT` 返回类型，并且编译器编译不会出现错误提示。而在 Visual C++.NET 中，这种潜在的类型不匹配错误会被捕捉，并且编译器会用一个出错信息来标识这种不匹配错误，开发者通过使用 `LRESULT` 作为返回类型并重新编译，可以修复这种潜在的错误。

MFC 中也新增添了一些函数，支持超过 2038 年的日期，如 `CTimeSpan::GetDays64`、`CTimeSpan::GetTotalHours64` 和 `CTimeSpan::GetHours64` 等。

MFC 中还增加了新类：`CHtmlEditCtrl`、`CHtmlEditView` 和 `CHtmlEditDoc` 用于 DHTML 的编辑，`CDHtmlDialog` 和 `CMultiPageDHtmlDialog` 用于 DHTML 的对话框制作，`ChttpArg` 和 `ChttpArgList` 使得 ISAPI 能够解析存在争议的列表，`COleControlSite`、`COleControlContainer` 和 `COccManager` 支持非 Windows 控件，在 MFC 应用程序中支持并增强了使用网页形式的帮助，`CprintDialogEx` 支持 Windows 2000 打印特性。

DAO 的支持也有所改变，Visual C++ MFC AppWizard 和 MFC DLL AppWizard 不再支持 DAO 数据库工程（即 DAO database projects），但是仍然能够通过 Add Class wizard 来添加一些支持 DAO 的类，微软公司推荐在新的工程中使用 OLE DB 或者 ODBC，而在维持现存的工程中使用 DAO。

OLE DB Templates（OLE DB 模板）中，新增添的使用者类是 `CDBAsynchBase`、`CDBAsynchInterface`、`CDBAsynchNotify`、`CDBAsynchObject`、`CDynamicAccessor`、`CDynamicParameterAccessor`、`CDynamicStringAccessor`、`CDynamicStringAccessorA`、`CDynamicStringAccessorW` 和 `CXMLAccessor`。新增加了两个宏，`CXMLAccessor` 宏用来在数据库中按照名字（name）约束特定的栏，`BLOB*` 用来约束二进制型的大数据对象（BLOB）。新增了 `IDBSchemaRowsetImpl`、`IErrorRecordsImpl`、`IFakeRowsetNotifyCP`、`IRowsetChangeImpl`、`IRowsetCreatorImpl`、`IRowsetNotifyCP`、`IRowsetNotifyImpl`、`IRowsetLocateImpl` 和 `IRowsetUpdateImpl`。增添了新的 64 位数据类型，在几个别的 OLE DB 模板类中要求使用新的 64 位数据类型代替旧的 32 位数据类型，读者可以参考 `atldb.h`、`atldbcli.h` 和 `atldbsch.h` 文

件来获取现在所使用的参数类型的具体信息。

Shared Classes（共享类）中，许多新的和修改后的类可以被应用在任何 C++ 程序中。

Standard C++ Library（标准 C++ 库）中很多类也被更新，如 hash\_map 类和 hash\_set 类，同时增强了类库的功能，支持可选择的例外处理，增强了代码的格式化程度，提高了可读性，还增强了进程之间的 DLL 支持和多线程支持等。

Old Iostream Library（旧的流输入输出库）在 Visual C++ 中的使用被减少，如果使用流函数的话，可能每次都会得到 Warning（警告），因为一些旧的函数在新版本中被去掉了，所以建议用户使用时一定要特别谨慎。

## 1.2 Visual C++.NET 的软、硬件配置

我们安装 Visual C++.NET 之前，必须安装 Visual Studio.NET 开发平台，因为 Visual C++.NET 的安装是包含在 Visual Studio.NET 的安装过程中的。因此，Visual C++.NET 的安装过程就是 Visual Studio.NET 的安装过程。Visual C++.NET 运行所需要的软、硬件配置同 Visual Studio.NET 一样，应满足以下要求（仅供参考）：

- 使用 486 或更高级处理器（推荐使用 Intel Pentium 处理器）。
- Windows 98、Windows NT 4 或 Windows 2000 操作系统，建议使用 Windows 2000 操作系统。
- 32MB 内存，建议 64MB 或更多内存。
- 380MB 硬盘空间，如果完全安装需要 1.3GB 的硬盘空间。
- CD-ROM 驱动器、键盘和鼠标。
- 显示器。

在安装过程中，建议先卸载 Office 2000，否则可能会造成严重冲突。如果只需要使用 ASP.NET（ASP+）的平台，则用户只需安装 CD3 中的 Framework SDK 即可。安装时还请注意，Windows 操作系统的语言和 Visual C++.NET 的语言应保持一致。

## 1.3 Visual Studio.NET 的安装

在我们使用 Visual C++ 之前，必须先安装 Visual Studio.NET 开发平台。由于 Visual Studio.NET 前所未有的捆绑了许多微软公司的产品组件，所以在真正安装之前，首先应更新安装所需的组件，这些组件都包含在安装盘中，更新是由安装程序自动完成的，用户不必过多考虑组件问题。当组件更新完毕，就可以启动 Visual Studio.NET 的安装程序。在安装过程中，还可以根据用户的需要对安装组件进行取舍。

在 Windows 2000 系统中安装 Visual Studio.NET，其过程大致分为两个部分。

第一部分，Windows 组件的安装：

（1）开启电脑进入 Windows 2000 操作系统。

（2）在光驱中先放入 CD1（第 1 张安装盘），光盘自动运行后，会出现如图 1.1 所示的

安装开始界面，开始安装。

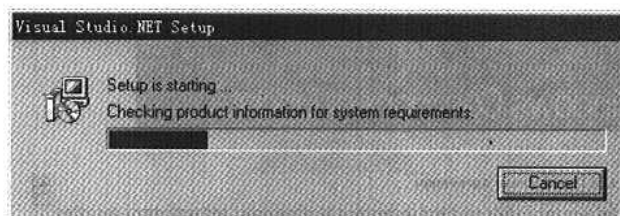


图 1.1 安装开始界面

(3) 弹出 Visual Studio.NET Setup 安装界面，如图 1.2 所示，选择第一个选项【Windows Component Update】(更新 Windows 系统组件)。

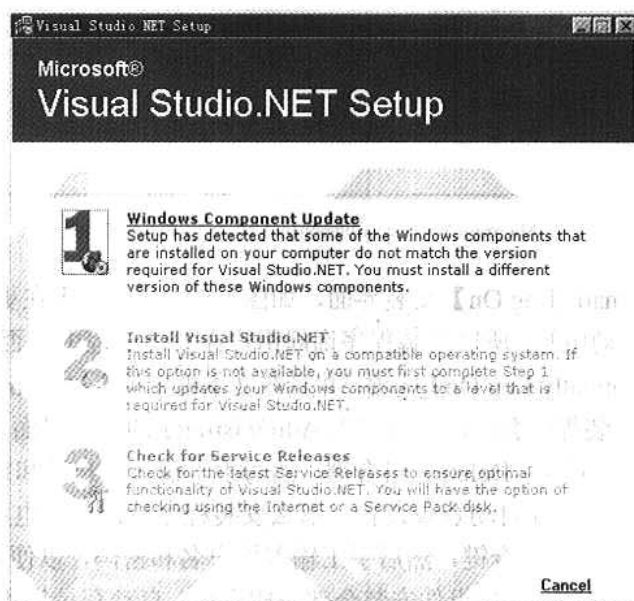


图 1.2 Visual Studio.NET Setup 安装界面

(4) 弹出【Insert Disk】对话框，如图 1.3 所示，要求更新组件，把 CD3 (第 3 张安装盘) 放入光驱，然后单击【OK】按钮。

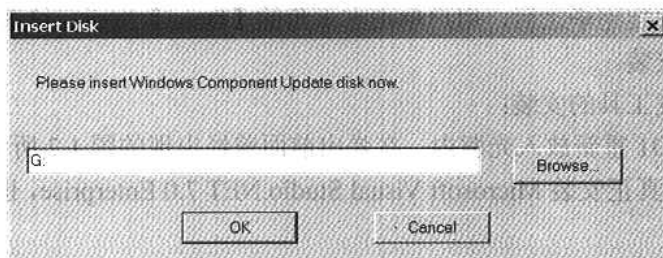


图 1.3 【Insert Disk】对话框

(5) 等待片刻后出现【Microsoft Windows Component Update 1.0 Setup】安装界面，如

图 1.4 所示，用户选择【I accept the Agreement】选项，然后单击【Continue】按钮。

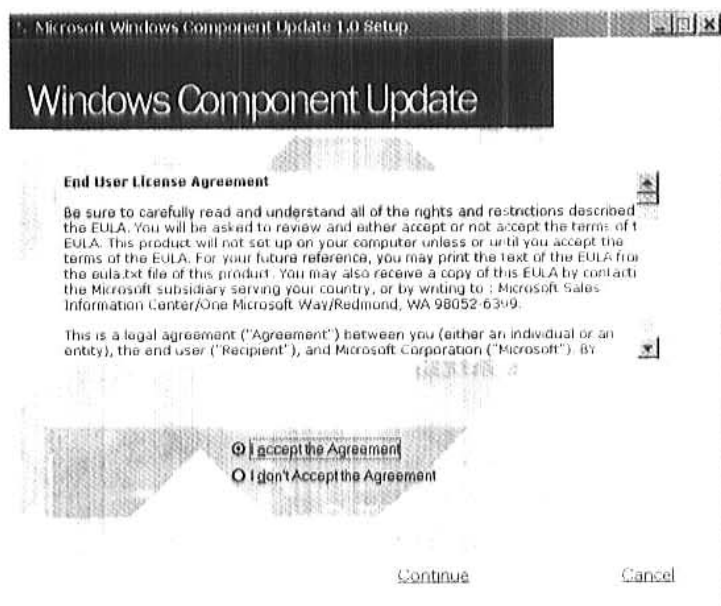


图 1.4 Microsoft Windows Component Update 1.0 Setup 安装界面

(6) 弹出【Automatic Log On】安装界面，如图 1.5 所示。由于在更新 Windows 组件过程中，需要反复重新启动电脑，所以安装程序询问用户在安装过程中是否采用自动登录设置。如果用户选择【Automatically Log On】选项，并且在文本框中输入 Administrator 用户的密码，则在重启电脑之后，安装程序会自动直接进入 Administrator 的用户界面，继续安装，不再需要用户按下【Ctrl】+【Alt】+【Delete】组合键，然后手工输入用户名和密码登录到 Windows 界面系统；如果用户没有选择自动登录设置，那么安装程序每次重启电脑后，需要用户按下【Ctrl】+【Alt】+【Delete】组合键，然后手工输入用户名和密码，以便登录到 Administrator 的操作界面，然后继续安装。一旦组件安装全部完成之后，自动登录设置会自动取消。下面，用户选择是否采用自动登录后，就可以单击【Continue】按钮，继续安装。

(7) 出现【Install Progress】安装进度界面，如图 1.6 所示。从这里可以看到，共需要更新 9 个 Windows 组件。在这些组件的更新过程中，安装程序需要重新启动电脑几次，如果用户想终止安装过程，可以单击【Cancel】按钮。

这些组件全部安装完毕之后，用户单击右下角的【Done】按钮，结束组件安装，然后可以进行第二部分的安装。

第二部分，开发工具的安装：

(1) 用户将 CDI 重新放入光驱内，在弹出前面曾经出现的图 1.2 所示的窗口后，这时，选择第二选项，开始真正安装 Microsoft Visual Studio.NET 7.0 Enterprise，也就是开始了 Visual C++.NET 的安装。

(2) 接着，弹出【Setup Start Page】安装界面，如图 1.7 所示。在该对话框中用户必须选择【I accept the Agreement】单选钮，输入序列号和用户名，然后单击【Continue】按钮，继续安装。读者也可以参考图 1.7 左侧的安装提示，进行安装。



(3) 接着弹出【Setup Options Page】安装界面, 如图 1.8 所示, 让用户根据自己的需要选择安装的项目, 这里选择【Visual C++】选项。在图 1.8 右侧上面选择安装路径, 下面显示的是磁盘空间的一些信息, 然后单击下面的【Install Now!】按钮, 继续安装。

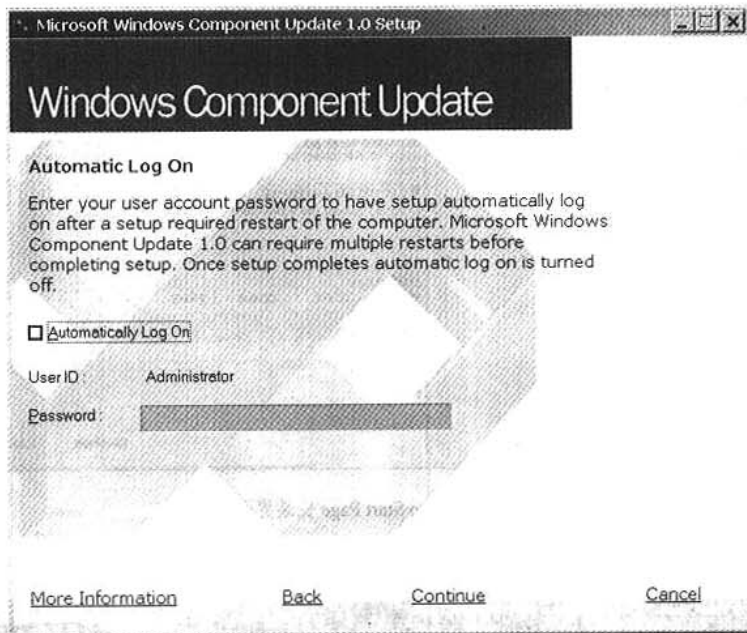


图 1.5 Automatic Log On 安装界面

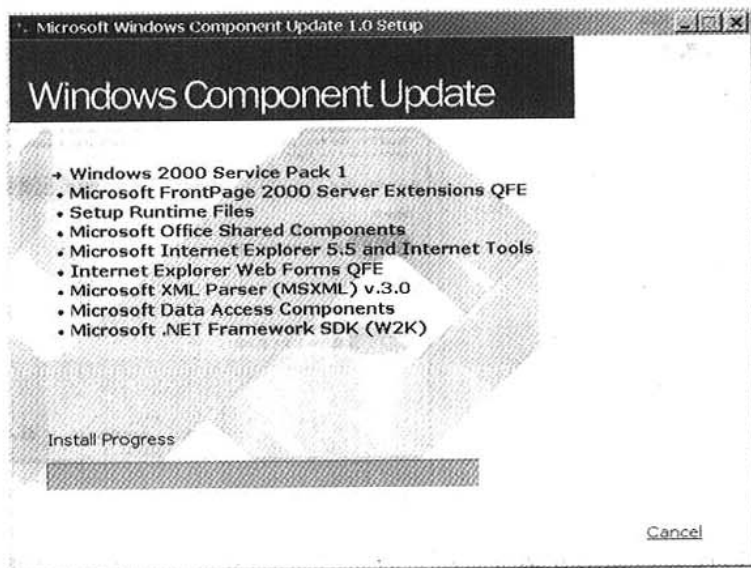


图 1.6 Install Progress 安装进度界面

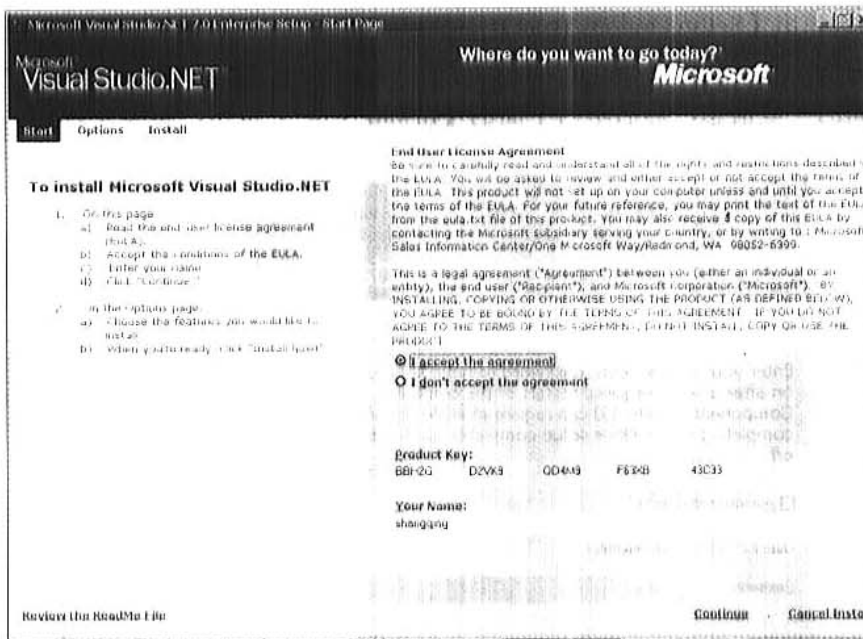


图 1.7 Setup Start Page 安装界面

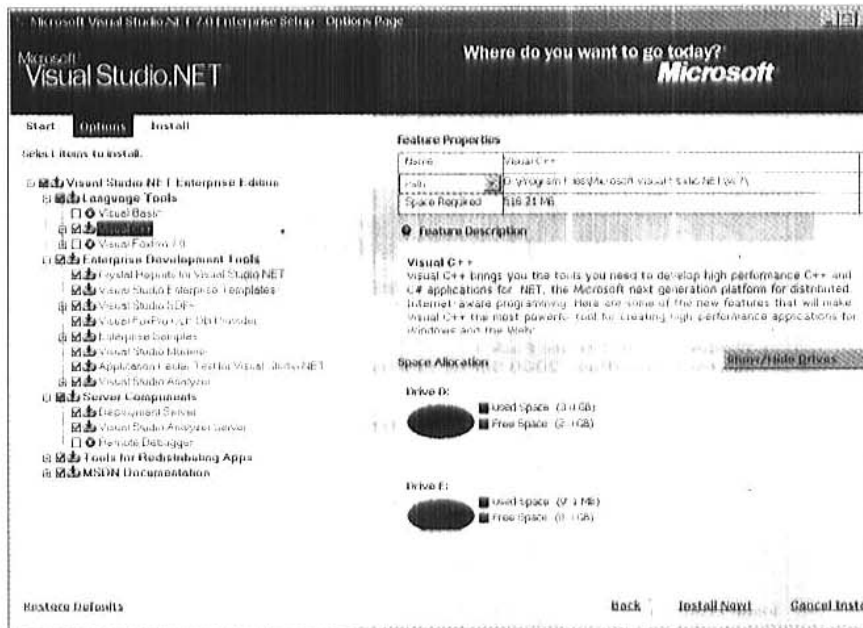


图 1.8 Setup Options Page 安装界面

(4) 在安装过程中会弹出安装进度指示界面，如图 1.9 所示。用户如果想终止安装过程，可以单击右下方的【Cancel Install】按钮。在安装完 CD1 之后，安装程序会自动提示用户插入 CD2 光盘，继续进行安装。在安装过程中，用户可以阅读一下图 1.9 中间部分的内容，它讲述了 Visual Studio.NET 的一些新特性。

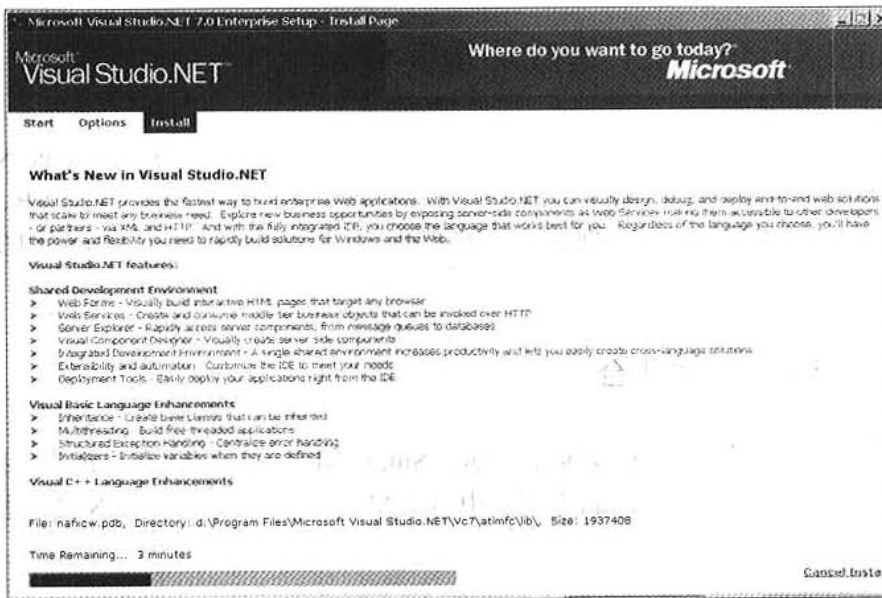


图 1.9 安装进度指示界面

(5) 安装完成之后, 会出现如图 1.10 所示的安装完成界面, 用户可以单击【View installation log】按钮来查看安装过程的记录, 还可以单击【Done】按钮来结束安装画面。这时, 不必重新启动电脑, 用户就可以直接使用 Visual Studio.NET, 进行 Visual C++ 的编程实践了。

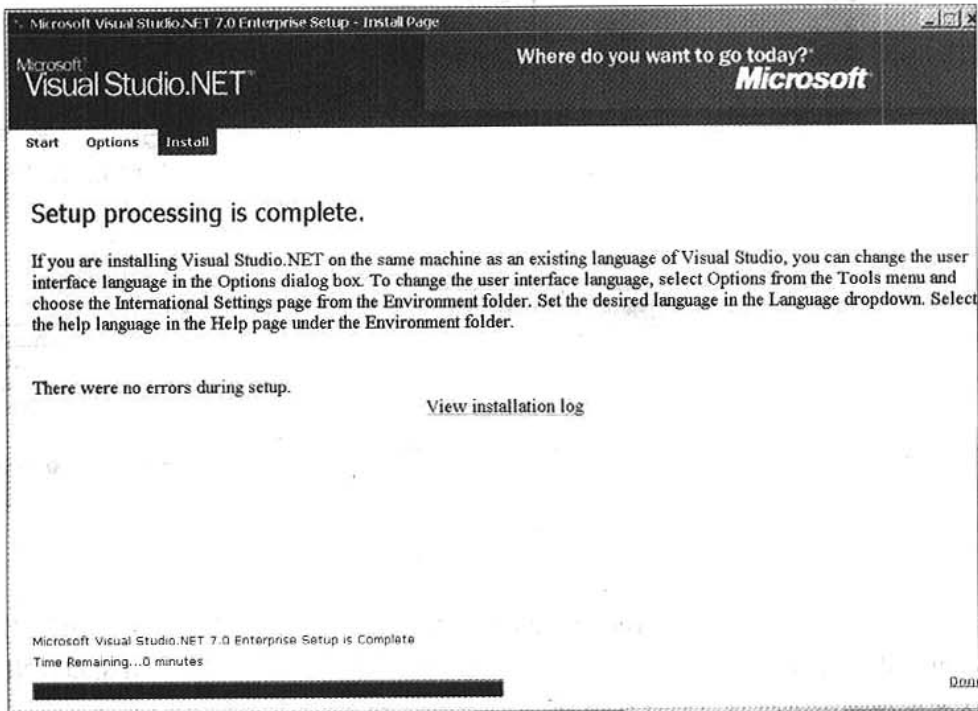


图 1.10 安装完成界面

## 1.4 Visual Studio.NET 集成环境简介

对于将要使用 Visual C++.NET 进行项目开发的人员来说,了解 Visual Studio.NET 的集成开发环境显得非常重要。新的.NET 开发平台和以往开发平台有很大的不同,VB、VC 和 C# 等项目都可以在.NET 平台下进行创建、编译和运行。本节将对 Visual Studio.NET 的集成开发环境的主要部分作简要的介绍。

### 1.4.1 Visual Studio.NET 平台

首先,用户已经按照前面的步骤将 Visual Studio.NET 安装到操作系统中。接着,单击【Start】按钮,从弹出的【Program】菜单中启动 Microsoft Visual Studio.NET,进入.NET 开发平台,如图 1.11 所示。

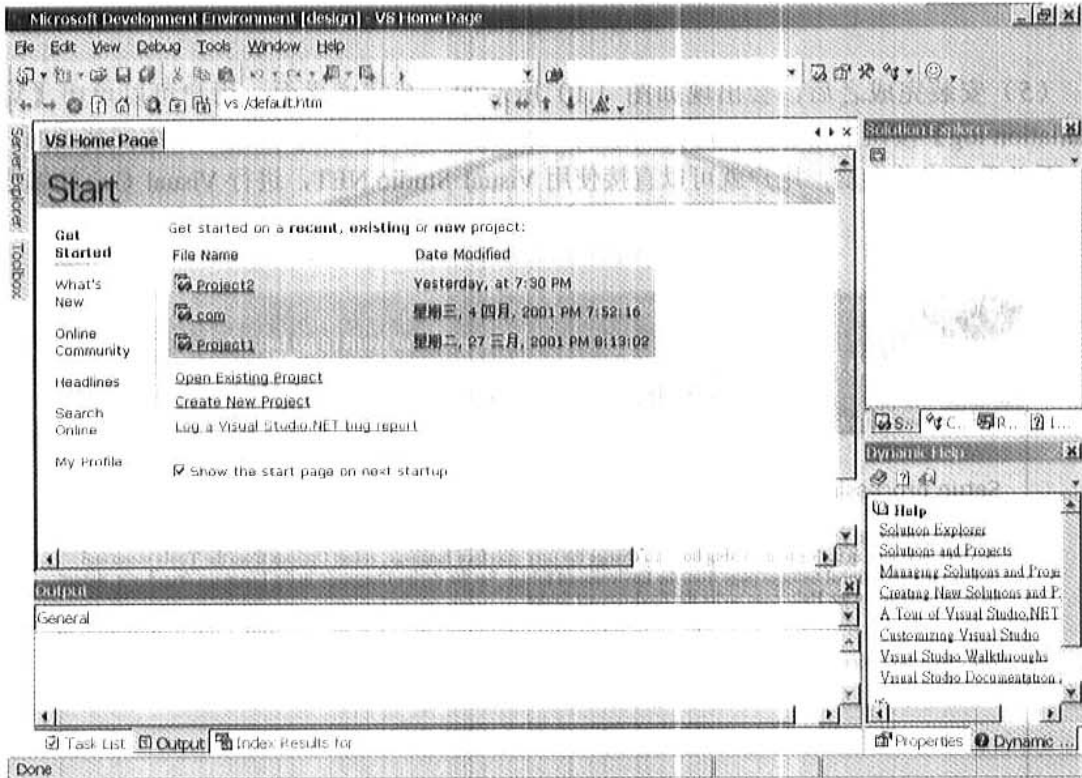


图 1.11 Microsoft Visual Studio.NET 开发平台

Visual Studio.NET 的界面和 Visual Studio 6.0 的界面有很大的不同,最主要的区别在于,Visual Studio.NET 把 IE 浏览器嵌入到了其开发平台中,可以直接通过 Visual Studio.NET 浏览网页,这和单独使用浏览器没有什么分别。

屏幕最上端是标题栏,用于显示应用程序名和所打开的文件名。标题栏的下面是菜单栏

和工具栏。工具栏的下面是平时我们所看到的 IE 浏览器中的一些工具按钮。在 Visual Studio.NET 主界面的中心位置,是我们以后会经常用到的主窗口,刚启动时会显示一个“VS Home Page”的页面,页面中会有一个大大的“Start”字样,这个主窗口可是说是一个多文档窗口,还兼作浏览器窗口,代码编辑和网页浏览都要在其中进行。主窗口的左侧有两个弹出式菜单,【Server Explorer】和【Toolbox】。主窗口右侧有两个主要窗口,上面是工作区窗口,包括 Solution Explore、Class View 和 Resource 等,下面显示的是属性页的内容(Properties)。在主窗口的下面是调试输出窗口。上面的这些窗口,均允许用户用鼠标拖动到界面的任何位置,还可以任意关闭和打开,方便了用户定制个人所需要的开发环境。对于一些新增窗口和主要窗口的介绍,请读者参考下面的内容。

## 1.4.2 Visual Studio.NET 的主窗口

Visual Studio.NET 的主窗口功能非常强大,可以进行新工程的创建,可以打开已经存在的工程,可以进行代码编辑,还可以浏览网页,如图 1.12 所示。

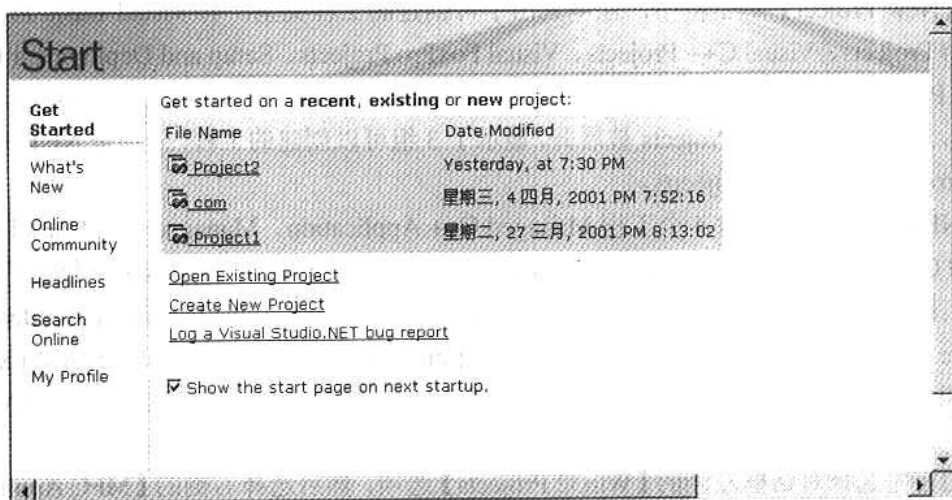


图 1.12 主窗口

在主窗口的左侧,有一些功能选项,使用户可以开始工作,了解新特性,进行在线交流等,我们首先用鼠标单击【Get Started】选项,看一下这个选项中的功能。

进入 Get Started 选项页面后,我们看到了一些和网页中类似的超级链接。点击这些超级链接,我们可以打开已经存在的工程或者最近所打开过的工程,使用【Open Existing Project】打开一个已经存在的工程,使用【Create New Project】创建一个新的工程,使用【Log a Visual Studio.NET bug report】可以链接到微软公司的站点,得到最新的关于 Visual Studio.NET 中的 bug 的信息。下面还有一个单选框,询问用户是否在下次启动 Visual Studio.NET 时,显示这幅 Start 页面。由于上面的这些功能选项现在以类似网页中的超级链接的方式实现,所以正体现了.NET(网络)的含义,使得以前不熟悉 Visual Studio 而熟悉上网的用户也能够很快掌握 Visual Studio.NET 中的操作方式。下面我们首先单击【Create New Project】选项,将弹出如

图 1.13 所示的【New Project】对话框。

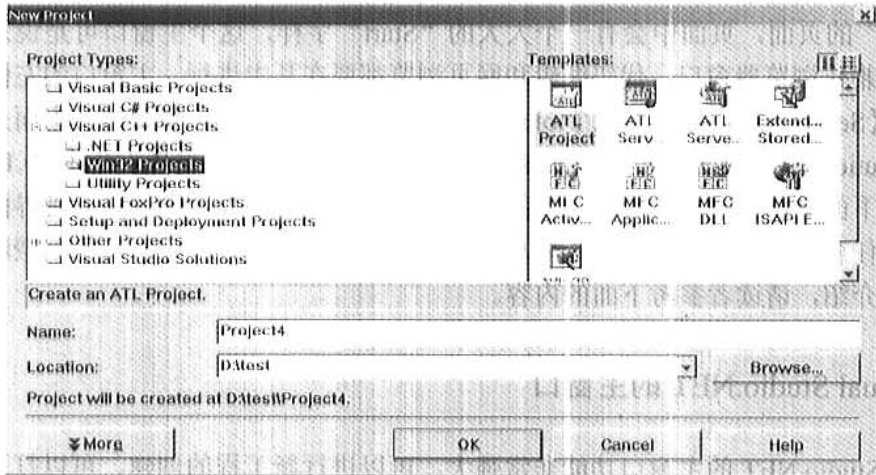


图 1.13 【New Project】对话框

在【New Project】对话框中，左侧列出了可创建的工程类型，有 Visual Basic Projects、Visual C# Projects、Visual C++ Projects、Visual FoxPro Projects、Setup and Deployment Projects 和 Other Projects，另外还有一个 Visual Studio Solutions。我们单击 Visual C++ Projects 前面的“+”号，这时 Visual C++ Projects 被展开，显示了 3 组可以创建的工程，分别是 .NET Projects、Win32 Projects 和 Utility Projects。

在 .NET Projects 中包含 4 个工程 Managed C++ Application、Managed C++ Class Library、Managed C++ Empty Project 和 Managed C++ Web Service，在 Win32 Projects 中包含了 9 个可创建的工程 ATL Project、ATL Server Project、ATL Server Web Service Project、Extended Stored Procedure Project、MFC ActiveX Control、MFC Application、MFC DLL、MFC ISAPI Extension Project 和 Win32 Project，在 Utility Projects 中包含两个工程 Custom AppWizard 和 Makefile Project。

我们首先选择对话框左侧的【Win32 Projects】选项，然后选择右侧的【MFC Application】选项，再在对话框下面输入名字，并且选择路径，单击【OK】按钮，会弹出如图 1.14 所示的【MFC Application Wizard】对话框，进入创建应用程序的向导。

在【MFC Application Wizard】对话框的左侧，【Overview】选项显示了工程的一些默认设置，如 Multiple document interface（多文档界面）、No database support（不支持数据库）和 No compound document support（不支持复合文档）。如果这时用户单击【Finish】按钮，向导将会按照这些默认设置创建一个新的工程。如果用户想更改这些默认设置，必须选择该对话框的其他选项。如在图 1.14 中，选择【Application Type】选项，则进入图 1.15 所示的【Application Type】对话框。

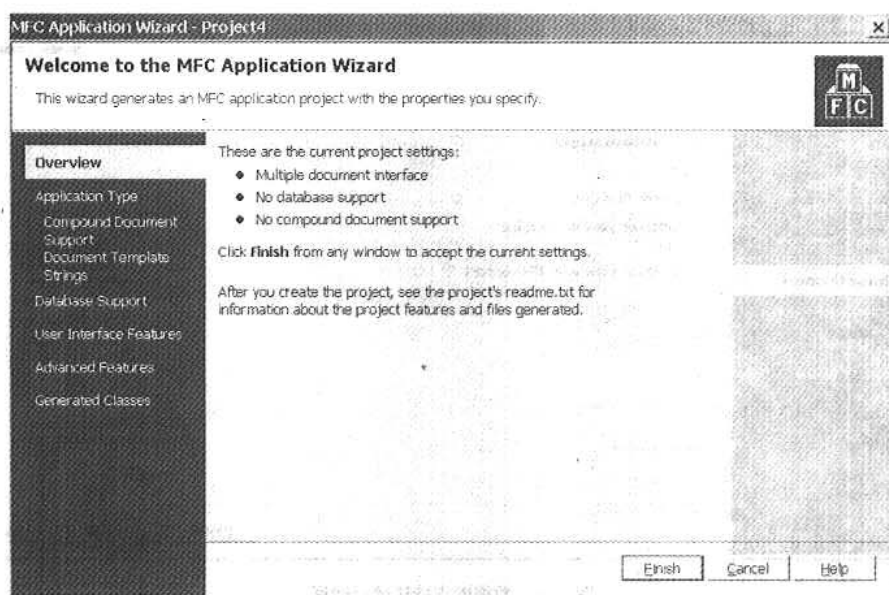


图 1.14 【MFC Application Wizard】对话框

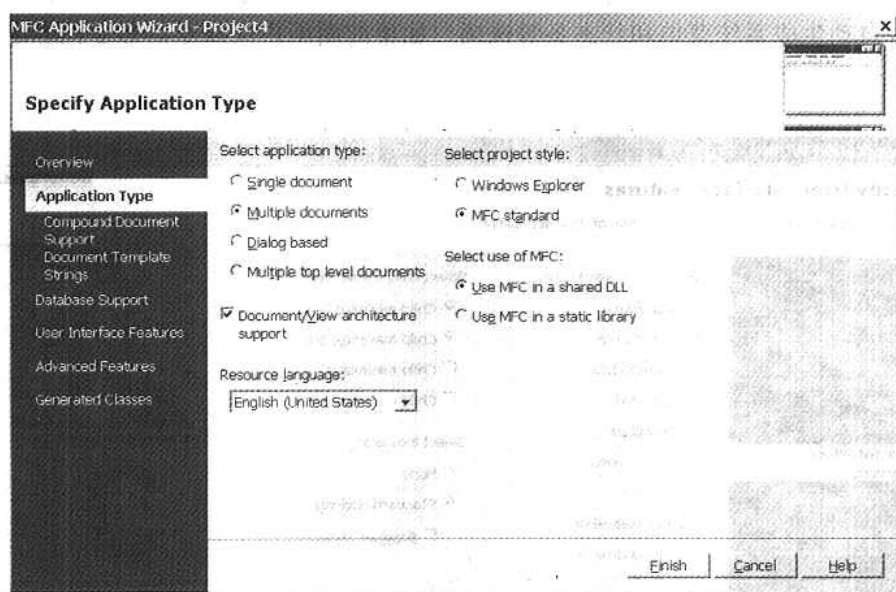


图 1.15 【Application Type】对话框

在【Application Type】对话框中，可以选择单文档、多文档或基于对话框的应用程序，可以选择工程的风格是 Windows Explorer 还是 MFC standard（MFC 标准风格），还可以选择使用 MFC 类库的方法。在 Application Type 下面还有两个子选项，即 Compound Document Support（用于指定支持何种类型的复合文档）和 Document Template Strings（用于指定文档模板串），可以更进一步对应用程序类型进行设置。单击【Application Type】下面的【Database Support】选项，会进入数据库支持选项对话框，如图 1.16 所示，这里我们不再详细叙述。

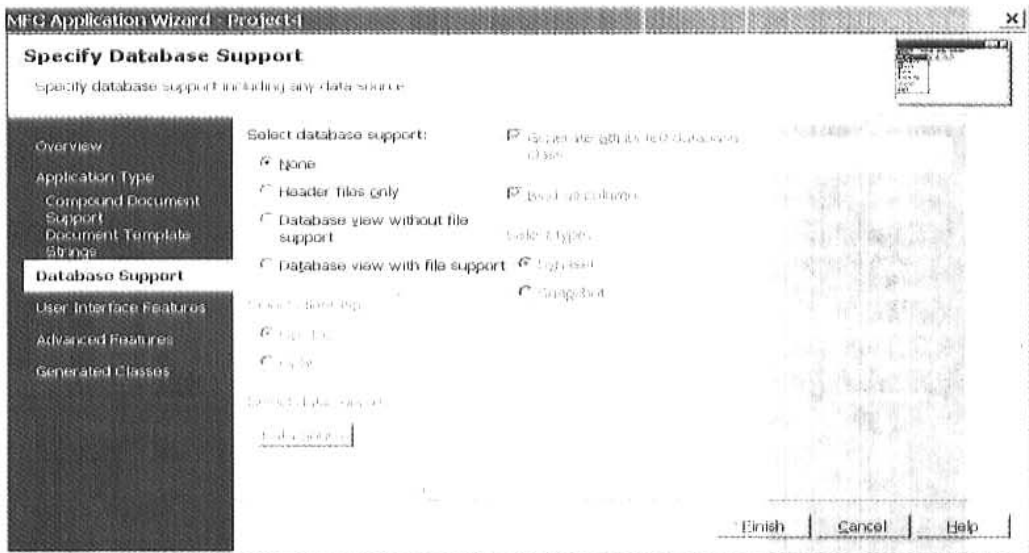


图 1.16 数据库支持选项对话框

接着单击【User Interface Features】选项，进入【User Interface Features】对话框，如图 1.17 所示，在其中可以设置所创建的应用程序的外观。用户可以设置是否支持最大化或者最小化，是否自动生成系统菜单和状态条等选项。如果前面用户选择支持多文档界面，这里还允许用户对子框架的一些风格进行设置。

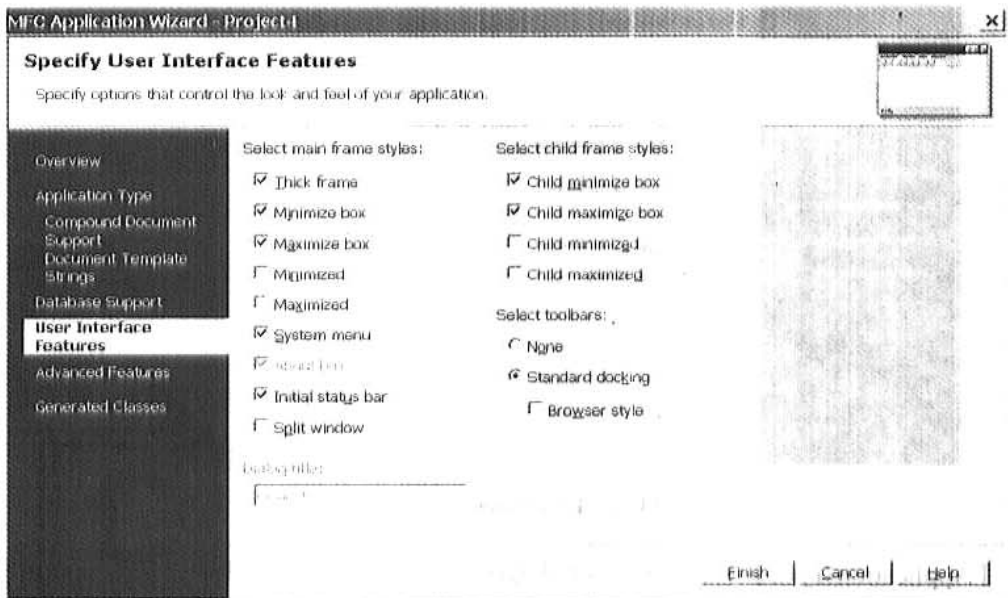


图 1.17 【User Interface Features】对话框

在【Advanced Features】对话框中，如图 1.18 所示，可以设置一些高级内容，包括打印、打印预览、自动化和 ActiveX 控件等。【Generated Classes】对话框的内容如图 1.19 所示。



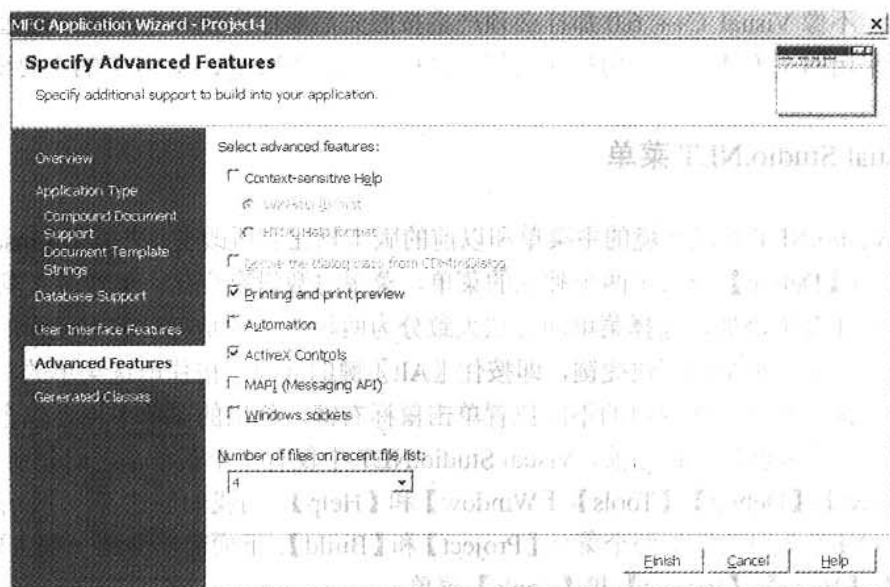


图 1.18 【Advanced Features】对话框

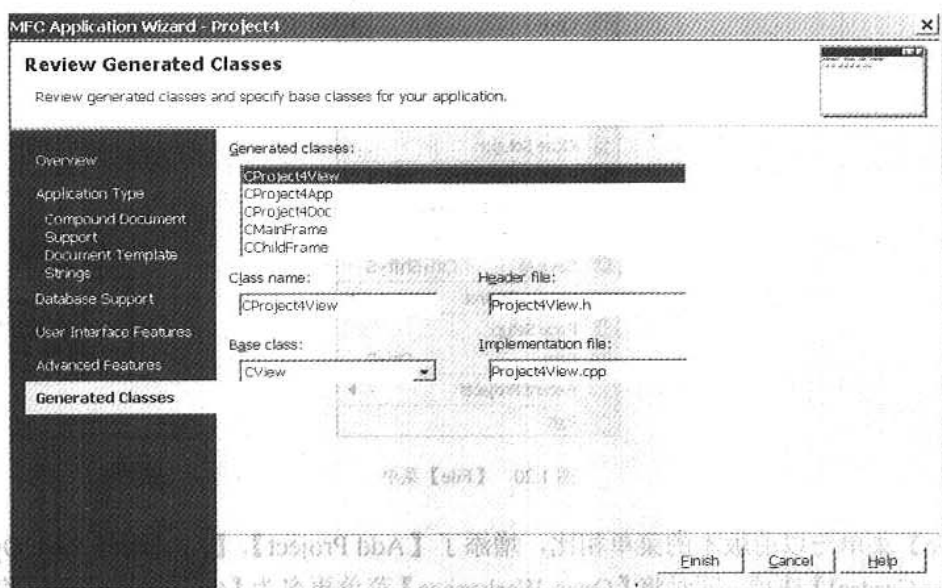


图 1.19 【Generated Classes】对话框

在这里，用户可以看到将由向导自动生成的全部类，还可以为应用程序指定基类。在以上这些操作步骤都完成之后，单击【Finish】按钮，向导会自动为我们生成符合我们要求的新的工程。

熟悉这些操作步骤之后，我们可以看到，仅仅通过主窗口就可以为我们创建新工程，并且操作方式就像在网页中点击超级链接一样。Visual C++.NET 把以前 6.0 版本中创建 MFC Application 的 6 个步骤全部放在一个向导对话框中，其操作方式也有了很大的不同。在 Visual C++.NET 中，用户完全可以随时通过选择不同的选项，对应用程序的类型、风格和基类等分