精通

Windows API

——函数、接口、编程实例

范文庆　周彬彬　安靖　编著

从Windows应用程序开发工具讲起，引导读者快速入门

详细讲解API和关键结构的使用方法

提供156个Windows API使用实例

人民邮电出版社
POSTS & TELECOM PRESS

CD-ROM

精通

Windows API

——函数、接口、编程实例

范文庆　周彬彬　安靖　编著

人民邮电出版社
北京

图书在版编目（CIP）数据

精通Windows API：函数、接口、编程实例 / 范文庆，周彬彬，安靖编著. —北京：人民邮电出版社，2009.2
ISBN 978-7-115-19095-6

Ⅰ. 精… Ⅱ. ①范…②周…③安… Ⅲ. 窗口软件，Windows—软件接口—程序设计 Ⅳ. TP316.7

中国版本图书馆CIP数据核字（2008）第168772号

内 容 提 要

Windows API（Windows Application Program Interface，Windows 应用程序接口）是一系列函数、宏、数据类型、数据结构的集合，运行于 Windows 系统的应用程序，可以使用操作系统提供的接口来实现需要的功能。本书由浅入深、循序渐进地教授读者如何使用 Windows API 进行 Windows 应用程序开发。全书共 18 章，分为 3 个部分，第 1 部分（第 1 章～第 3 章）介绍 Windows 程序设计基础；第 2 部分（第 4 章～第 17 章）按照程序设计的各个方面进行划分，包括文件系统、内存管理、进程与线程、用户界面、Shell 程序开发、Windows GDI、Socket 网络通信驱动程序开发、安全机制等内容；第 3 部分（第 18 章）作为全书的总结和补充。

本书适合广大的 Windows 应用程序开发人员、Visual C++开发工程师、网络游戏开发人员、软件培训机构学员和高校学生阅读。

精通 Windows API——函数、接口、编程实例

◆ 编　　著　范文庆　周彬彬　安　靖
责任编辑　屈艳莲
执行编辑　黄　焱

◆ 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@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：35
字数：922 千字　　2009 年 2 月第 1 版
印数：1－4 000 册　　2009 年 2 月北京第 1 次印刷

ISBN 978-7-115-19095-6/TP

定价：69.00 元（附光盘）

读者服务热线：(010)67132692　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

关于 Windows API

Windows API（Windows Application Program Interface）是一系列函数、宏、数据类型、数据结构的集合。运行于 Windows 系统的应用程序可以使用操作系统提供的接口来实现应用程序需要的功能。Windows 开发平台是所有程序开发平台中非常重要的一种，是程序设计中一个重要的方面。本书介绍了 Windows 应用程序开发的各个方面。

本书的写作方法

程序设计是一门实践性很强的学科，本书在编写过程中十分注重与实际开发工作相联系，在讲解每一个编程要点时，都以实例为核心进行分析、演示，并把实际工程中开发人员可能遇到的问题以实例的方式向读者讲解。

与此同时，本书同样重视基础知识的讲解，重视原理的说明，对程序所使用的每一个 API 和每一个结构的使用方法都尽量详细说明。本书还在第 18 章详细介绍了 Windows 系统调用的内部过程。

解决开发人员实际编码过程中可能出现的问题也是本书的一个主要目的，因此本书使用了大量篇幅来讲解编译选项等最容易被忽略，但是却是最容易在程序设计中出现问题，也是常常使得程序员不知道如何解决的问题。

本书主要内容

本书共分 18 章，涵盖了 Windows 应用程序设计的基本内容。第 1、2、3 章是 Windows 程序设计的基础。第 4 章～第 17 章按程序设计的方面划分，每个方面一章。第 18 章是全书的总结和提高。

第 1 章主要讲解如何配置 Windows 应用程序的开发工具，从最简单的实例入手，使读者对 Windows 应用程序的开发有一个初步的认识。

第 2 章对 Windows API 作了简要的介绍，包括 Windows API 基本的功能分类、数据类型等。

第 3 章介绍了开发工具配置与使用，包括基本的编译链接工具 cl.exe、rc.exe、link.exe，Platform SDk 的使用，Makefile 的编写以及使用 WinDbg 的调试方法等。

第 4 章文件系统，主要介绍了磁盘和驱动器管理、文件和目录管理、内存映射文件等内容。

第 5 章内存管理，主要介绍了内存管理原理、堆原理、全局和局部内存管理、虚拟内存管理、内存操作与内存消息管理等内容。

第 6 章进程、线程和模块，主要介绍了进程管理，线程、纤程的创建、删除以及获取方法，获取进程状态信息，动态链接库的加载、释放和获取方法等内容。

第 7 章线程同步，主要介绍了线程同步的基本原理、同步对象等内容。

第 8 章服务，主要介绍了基本概念、服务程序的编写、对服务程序的控制和管理等内容。

第 9 章图形用户界面，主要介绍了字符界面程序，窗口、控件、菜单、对话框等界面元素的使用方法。

第 10 章系统信息的管理，主要介绍了系统基本信息、时间信息以及注册表信息的管理。

第 11 章进程间通信，主要介绍了邮槽、管道、剪贴板的创建与使用方法，数据复制消息的使用方法等内容。

第 12 章 Windows Shell 程序设计，主要介绍了 Windows Shell 的目录管理、文件协助、Shell 扩展等内容。

第 13 章 Windows GDI，主要介绍了使用 GDI 指定文字的字体、绘制线条、绘制图形、对位图的操作、坐标变换的方法等内容。

第 14 章网络通信与配置，主要介绍了 Socket 通信的相关内容以及 IP Helper 的使用方法。

第 15 章程序安装与设置，主要介绍了 cab 文件的创建、INF 文件的创建、安装程序 setup.exe 的编写、使用 msi 文件进行安装等内容。

第 16 章设备驱动管理与内核通信，主要介绍了与设备有关的 API 函数、如何对设备驱动进行控制、如何编写设备驱动程序等内容。

第 17 章用户、认证和对象安全，主要介绍了数据认证中的基本概念、安全机制程序示例以及用户的增加、删除、权限更改等内容。

第 18 章 Windows API 的内部原理，主要介绍了 Windows 系统中的对象封装、x86 平台程序函数调用、可执行程序结构与 API 函数接口内部机理等内容。

参与本书编写的人员

本书由范文庆、周彬彬、安靖负责编写并统编全书稿，另外感谢以下人员为本书所做的工作：张墨、郭永红、周瑜、王建伟、孙琼、田旭、范文庆、钟金鑫、王欣、张曦文、尚玉珊、张丛辉、王玮、刘超、张圣亮、李凡、马堃、徐路迎、赵国锋、孙颂武、汪荷君、孙明、林雪梅、黄惠英、刘雯等。

由于时间仓促，加之水平有限，书中不足之处在所难免，敬请读者批评指正。本书责任编辑的联系方式是 huangyan@ptpress.com.cn，欢迎来信交流。

编　者

2009 年 1 月

目　录

第 1 章 Windows 应用程序开发入门

如何开始 Windows 应用程序开发的学习呢？本书将从一个最简单的 Windows 应用程序入手，讲解 Windows 应用程序设计中的基本概念、编程工具的使用安装与使用方法。通过本章的学习，读者将会对 Windows 应用程序的设计有一个初步的、直观的认识。

1.1 第一个实例程序

本节将给出一个完整的实例程序，并对该程序源代码进行分析。通过本节的学习，读者可以了解 Windows 应用程序源代码的必要组成部分以及它们的组织结构。

1.1.1 start.exe

start.exe 是本书的第一个实例，创建一个简单的消息对话框。

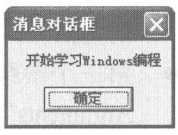

图 1-1 第一个实例程序

实例 1-1 第一个 Windows 应用程序 start.exe

打开本书配套光盘，在第一章实例代码目录中找到可执行文件 start.exe，运行 start.exe，会弹出如图 1-1 所示的消息对话框。

这就是一个最简单的 Windows 应用程序实例运行的效果，实现代码如下：

```
/* 头文件 */
#include <windows.h>
//连接时使用 User32.lib
#pragma comment (lib, "User32.lib")
/* ************************************
* WinMain
* 功能: Windows 应用程序示例
**************************************/
int WinMain(
            HINSTANCE hInstance,
            HINSTANCE hPrevInstance,
            LPSTR lpCmdLine,
            int nCmdShow
            )
```

```
{
    // 调用 API 函数 MessageBox
    MessageBox(NULL,
        TEXT("开始学习 Windows 编程"),
        TEXT("消息对话框"),
        MB_OK);
    return 0;
}
```

1.1.2 Windows API

实例 1-1 中，程序最核心的功能实现使用了 Windows API 的 MessageBox 函数。

就像 C 语言有一系列库函数一样，任何一种程序开发平台都会提供众多的接口函数供开发人员使用。Windows 应用程序的开发核心问题是如何使用 Windows SDK 所提供的 API。

所谓 API 是“Application Program Interface”的简写，中文含义为“应用程序接口”，它是一系列函数、宏、数据类型、数据结构的集合。运行于 Windows 系统的应用程序可以使用这些操作系统提供接口来完成应用程序需要的功能。

MessageBox 函数是 Windows 众多 API 中的一个，其的功能是弹出一个对话框。

1.1.3 程序入口函数

WinMain 函数是程序入口点，相当于 C 语言的 main 函数，其定义如下：

```
int WinMain(
          HINSTANCE hInstance,
          HINSTANCE hPrevInstance,
          LPSTR lpCmdLine,
          int nCmdShow
               );
```

WinMain 函数有 4 个参数：hInstance、hPrevInstance、lpCmdLine、nCmdShow。

- 第一个参数是 hInstance，数据类型是 HINSTANCE，此参数表示应用程序本次运行实例的句柄。
- 第二个参数是 hPrevInstance，数据类型也是 HINSTANCE，表示应用程序之前运行实例的句柄，但是在实际应用中，此参数始终为 NULL。
- 第三个参数 lpCmdLine 是运行时参数。如在 cmd 命令行中运行“start.exe Command Arguments”，那么 lpCmdLine 就指向字符串“Command Arguments”，作用与 C 语言中 main 函数的参数作用类似。数据类型 LPSTR 是 Windows API 中常用的字符串类型。
- 最后一个参数是 nCmdShow，数据类型是 int，表示应用程序窗口（如果有）的显示状态。

WinMain 函数的返回类型是 int 型的。在实例 1-1 中，使用了如下语句来返回。

```
return 0;
```

1.1.4 start.c 代码分析

现在分析 1.1.1 小节给出的实例 1-1 程序的源代码 start.c。

在代码的最开始是源代码文件信息的注释。

之后是预编译声明，在本实例中共有两条预编译声明。

```
#include <windows.h>
```

这一行代码指明了包括 Windows.h。Windows.h 是 Windows 应用程序开发中常用的头文件，在 Windows 应用程序开发中所使用的很多的数据类型、结构、API 接口函数都在 Windows.h 或 Windows.h 所包含的其他头文件中进行了声明，比如实例 1-1 所使用的 MessageBox 函数。如果不

声明包括 Windows.h，那么编译器将不能识别 MessageBox。

```
#pragma comment (lib, "User32.lib")
```

这行代码指明将源文件编译生成目标文件（.obj）后，将目标文件链接成可执行文件的过程中需要使用到 User32.lib，因为 User32.lib 中包含了 MessageBox 的链接信息。如果在链接程序时，调用链接器的参数中指明了链接时需要使用到 user32.lib，这一句程序代码可以省略（参见第 3 章）。

代码接下来是 WinMain 函数的定义。在 WinMain 函数中调用了 MessageBox 函数。

```
MessageBox(NULL,
        TEXT("开始学习 Windows 编程"),
        TEXT("消息对话框"),
        MB_OK);
```

MessageBox 函数的原型如下：

```
int MessageBox(
        HWND hWnd,
        LPCTSTR lpText,
        LPCTSTR lpCaption,
        UINT uType
);
```

第一个参数 hWnd 是消息框所属的窗口的句柄，可以设置为 NULL。

第二个参数 lpText 是字符串，表示消息框所显示的消息。

第三个参数 lpCaption 是字符串，表示消息框的标题。

第四个参数 uType 是消息框的类型，在实例 1-1 中设置为“MB_OK”，意思是消息框包含一个“确定”按钮。

在 MessageBox 函数调用之后，程序返回 0，结束 WinMain 函数，退出进程。

1.2 编译代码

本书使用 Visual Studio 和 Platform SDK 作为程序编译和连接的工具。各版本的 Visual Studio，包括 Visual Studio 6.0、Visual Studio 2003、Visual Studio 2005、Visual Studio 2008 都可以用于编译本书中的实例程序。

过去已经发行了的所有 Windows 版本，每一个版本都有一个 Platform SDK。Windows 系统为应用程序提供了很多的调用接口，如果要使用这些调用接口就需要用于 Platform SDK。Platform SDK 提供了开发 Windows 应用程序所必须的头文件、库文件等。Windows 应用程序的开发接口是 C\C++语言形式的。读者可以使用 Platform SDK for Windows XP SP2、Platform SDK for Windows Server 2003、Platform SDK for Windows Vista。

Visual C++ 2005 Express Edition（Visual C++速成版）是 Visual Studio 2005 的一个子版本。包括了编译和连接的全部工具 VC\VC++8.0，对编译本书的示例程序已经足够使用。

微软公司提供官方下载，下载地址为 http://msdn2.microsoft.com/zh-cn/express/default.aspx。各个版本的 Platform SDK 都也可以从微软公司的网站上下载到。

在本书中，将综合使用 Visual C++ Express Edition 2005、Visual C++ Express Editon 2008 以及 Microsoft Platform SDK XP SP2 版本、2003SP1 版本和 Vista 版本。

1.2.1 安装 Visual Studio

Visual Studio 包括速成版、专业版和团队系统版本（Visual Studio Team System）。速成版是免费的版本，专业版和团队系统版本是商业版。但是速成版的功能与专业版的功能并没有太大的区别。

如果读者使用 Visual C++速成版，需单独安装 Microsoft Platform SDK，并将其与 Visual Studio 集成在一起，才可以在本机进行 Windows 应用程序的开发。

Visual C++速成版的安装步骤如下。

（1）从微软公司的站点上下载中文版的 Visual C++速成版，也可以从本书配套光盘中获得。运行 vcsetup.exe。

（2）单击“下一步”按钮。

在这一步，读者需要选择需要安装的组件。推荐安装图形化的 IDE，IDE 将大大方便代码的编辑、编译等工作。不安装图形化 IDE，读者也可以使用命令行工具来编译程序。

（3）在安装的过程中，需要连接上 Internet。安装程序将下载 Windows Installer 3.1、NET Framework 和 Visual C++ 2005 速成版进行安装。

安装完成后，进行注册就可以使用了。在 Visual Studio 安装目录下的 bin 文件夹中，读者可以看到包括 cl.exe、nmake.exe、link.exe、lib.exe、rc.exe 等编译和链接工具。

安装了 Visual C++速成版后仅仅可以进行标准 C 程序和 C++程序的开发，但是还不能调用 Windows API，无法完成一个 Windows 应用程序所需的大部分功能，因为缺少 Platform SDK。

1.2.2 安装 Microsoft Platform SDK

SDK 是 Software Development Kit 的缩写，即“软件开发工具包”。Microsoft Platform SDK 是进行 Windows 应用程序的开发包。

每一个 Windows 发行版本都会有对应的 Platform SDK。SDK 中会使用与这个版本相一致的头文件和库文件等。每一个版本 Windows 中新增加的 API、数据结构中也会在其中体现。

下面以 Platform SDK for Windows Server 2003 R2 为例说明，其他版本的安装方法类似。如图 1-2 所示。

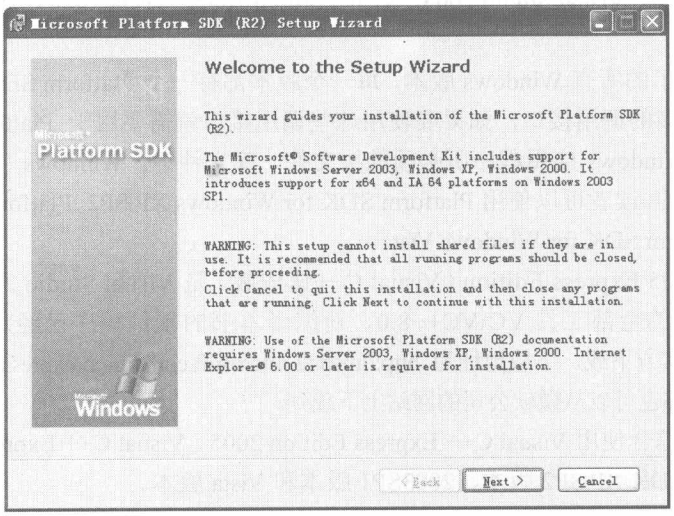

图 1-2 Platform SDK 安装界面

单击“Next”按钮，选择安装选项。

在一般情况下，选择典型安装可以满足使用需要。这里需要进行一些设置，所以选择

“Custom”，如图 1-3 所示。

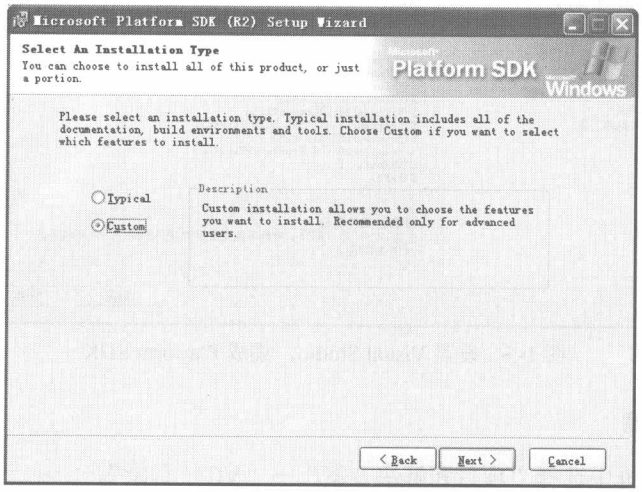

图 1-3　选择安装类型

在出现如图 1-4 所示的安装选择时，选择“Configuration Options”→“Register Environment Variables”选项，并选择完全安装。如果不选择此项，Visual Studio 工具将无法找到 SDK。

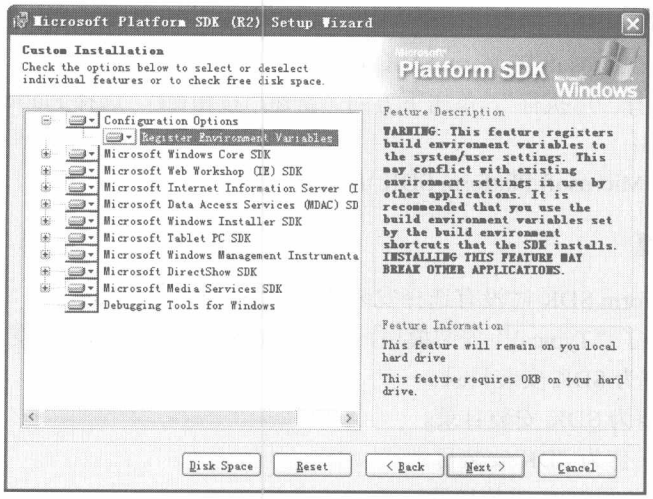

图 1-4　选择安装 Platform SDK 的功能

在安装完成后，可以从 Platform SDK 安装目录中找到若干头文件和库文件，如 Windows.h、Kernel32.lib 等文件，这些就是我们进行 Windows 应用程序开发所必不可少的支持。

1.2.3　集成 Microsoft Platform SDK 与 Visual C++速成版

在安装完成后，还需要对 Visual C++速成版进行设置，使编译链接工具可以找到 SDK。如果编译链接工具找不到 SDK，那么在源代码中的 API 调用将会产生编译链接错误。

在 Visual C++速成版中进行如下操作。在菜单中选择“工具”→“选项”，出现“选项”对话框，如图 1-5 所示。