

CMP

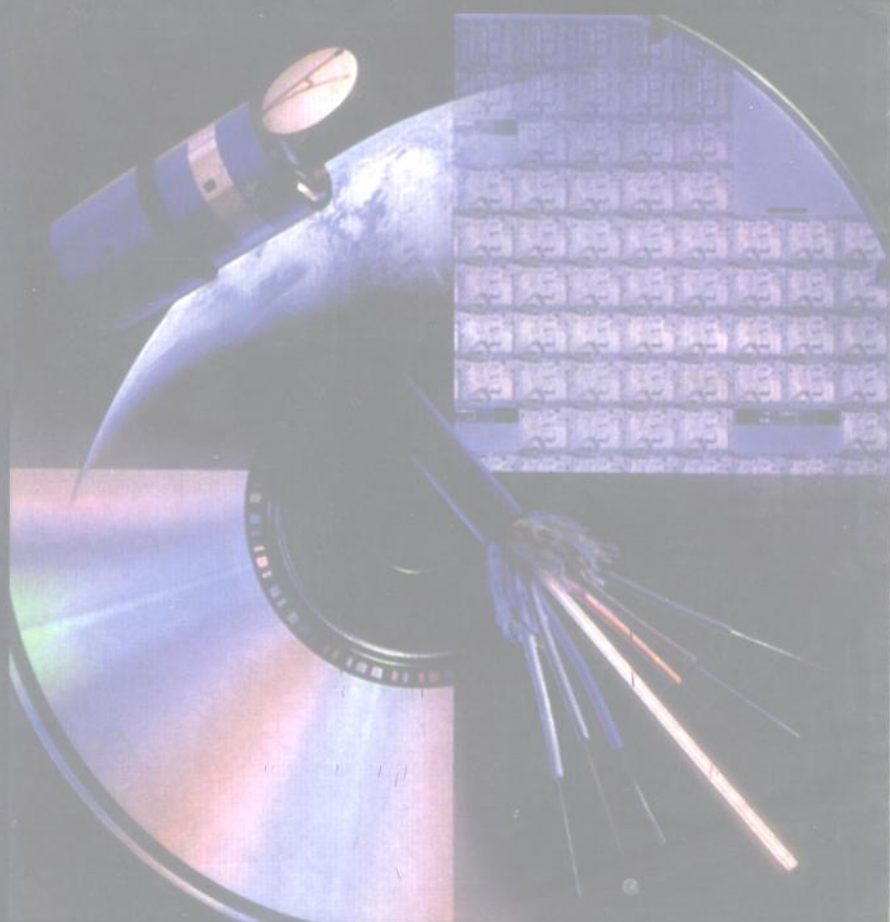
Windows 95 API How-To

Windows 95

API 开发人员指南

(美) Matthew Tells 著
Andrew Cooke

计算机软件开发
与程序设计
系列丛书



机械工业出版社



西蒙与舒斯特国际出版公司

17

405960

计算机软件开发与程序设计系列丛书

Windows 95 API 开发人员指南

(美) Matthew Telles 著
Andrew Cooke

吴冰 姚彦忠 等译

孙绍麟 审校



机械工业出版社
西蒙与舒斯特国际出版公司

JS146/04

本书以问答的形式组织起来 100 多个有关 Windows 95 API 使用技巧的专题。涉及到的范围包括：系统的信息，文件和目录，任务调度，图形设备接口，打印，应用程序间通信，多媒体，以及对话框、编辑控制、列表框、菜单等界面组件。介绍了一些窗口操作的实现，如：置前，置后，改变大小和移动位置等。同时还介绍了一些编程技巧以及如何完善应用程序。在每个专题中都有具体的例子程序的实现，既为读者示范了实现的步骤，同时还为读者提供了进一步开发的模板。

本书主要面向中高级开发人员，同时也考虑到了初学者，因此本书对不同层次的读者都是适用的。

Matthew Telles, Andrew Cooke: Windows 95 API How-to.

Authorized translation from the English edition Published by Waite Group Press.

Copyright 1996 by Waite Group Press.

All rights reserved. For sale in Mainland China only.

本书中文简体字版由机械工业出版社和美国西蒙与舒斯特国际出版公司合作出版，未经出版者书面许可，本书的任何部分不得以任何方式复制或抄袭。

本书封面贴有 Prentice Hall 防伪标签，无标签者不得销售。

版权所有，翻印必究。

本书版板登记号：图字：01-96-1224

图书在版编目 (CIP) 数据

Windows 95 API 开发人员指南 / (美) 特利斯 (Telles, M.), (美) 库克 (Cooke, A.) 著; 吴冰等译. —北京: 机械工业出版社, 1997. 6

(计算机软件开发与程序设计系列丛书)

书名原文: Windows 95 API How-to

ISBN 7-111-05463-6

I. W... I. ①特... ②库... ③吴... II. 软件接口, API-指南 N. TP334

中国版本图书馆 CIP 数据核字 (97) 第 05475 号

出版人: 马九荣 (北京百万庄南街 1 号 邮政编码 100037)

责任编辑: 何伟新

三河永和印刷有限公司印刷·新华书店北京发行所发行

1997 年 6 月第 1 版第 1 次印刷

787mm×1092mm 1/16·42.25 印张·1039 千字

0001-7000 册

定价: 74.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

译 者 序

Windows 95 的推出曾在世界各地引起了很大的轰动,它那强大的功能和新颖的用户界面赢得了广大用户的青睐。当前,大量的 16 位应用程序纷纷移植到这个新的 32 位平台上,而且更多的新软件正在开发之中。尽管在此 32 位平台上已有不少优秀的可视开发环境,如: Visual C++、Delphi、Visual Basic 等等。但是,在有些情况下直接使用 Windows 95 API 来开发应用程序可能会更高效、更实用。而且,有些功能在可视开发环境中是难以实现的,必须直接使用 API 函数来实现。因此,作为当前的 Windows 程序员,了解 Windows 95 API 是很有必要的。

本书是以一种非常容易查阅的问答形式组织起来的,详细介绍了 Windows 95 API 的使用技巧。涉及的范围非常广泛,不仅包括 Windows 95 中的新特征,而且还包括常用的 Windows 功能。因此本书不仅便于 Windows 程序员迅速转向新的平台,同时也适用于 Windows 编程的初学者。在本书的翻译过程中,由于译者的水平有限,错误和不妥之处难免,恳请广大读者指正。

参加本书翻译的有:孙绍麟老师翻译了前言,吴冰翻译了第 1、2、3、4、7、12、15 章,姚彦忠翻译了第 8、9、10、11、13、14 章,李跃宁翻译了第 5 章和第 6 章。孙绍麟老师最后审校了全书。译者对华章公司的编辑们所提供的帮助和支持表示感谢。

前 言

起初，Windows 应用编程接口（API）曾风靡一时。程序员对 API 称道“好极了！”，他们借助 API 开发了 Windows 程序，用户对 Windows 程序情不自禁地说：“真棒！”。由数以百计称为“函数”的魔块所组成的 API 似乎至高无上，随后，API 轰动了计算机界。

当 Windows 程序设计领域处于发展初期时，Windows 程序员可使用的编程工具唯有 API 函数。但这些函数难以理解，易于误用，还会导致出错。

随着可视编程环境的到来情况发生了巨变，程序员能采用拖放（drag and drop）方式来开发完美的全能应用程序，它们的界面友好、操作简便。Borland 和 Microsoft 创建的类库替代了 API 的神秘功能。其实，在当今应用程序的类库和定制控制是构筑在 Windows API 的基础上的。这些类库和函数加速了 Windows 应用编程的开发。程序员第一次把梦想做的事情变成了现实，而不必忧虑必须编制数以千行计的代码。这也导致了非常多的程序员在类库前面“故步自封”，对下层 API 函数的强大功能一无所知。

但当前，如果程序员要编写符合当代标准的程序，则没有比理解 API 的性能更重要。虽然类库和定制控制使开发应用程序容易得多，但它们只触及到 Microsoft 的 Windows 系统功能的皮毛。

本书的宗旨是指导用户去实现只利用 API 调用来完成开发应用程序的所有技巧。读者不仅会学到如何深入 Windows 的内部，而且将会鉴赏到这个图形用户界面的强大功能和灵活性。可以这样说，程序员只有真正理解 Windows API 的内含，才能成功地扩展那些类库和可视开发环境。

本书以烹饪书的编写方式展开。因而，读者可以根据当前应用程序开发的需要，随意跳到有关任务的专门章节去参阅所需的内容。

本书共分为十五章，每章集中讨论应用编程接口函数的一个独立部分。

第 1 章：获取系统信息。本章介绍有关确定正在运行的计算机处理器，正在使用的 Windows 版本及应用程序可用的内存有多少等的简单方法。另外，还将讨论找出按下哪些键和接通 NUMLOCK 键的编程小技巧。本章中还说明如何确定计算机的网络名（如果联网的话）及如何查找当前登录的用户名。最后，将考虑在 Windows 95 中新设立的注册表（Registry），并说明在应用程序中如何利用这一功能。

第 2 章：文件和目录。本章介绍如何处理在 Windows 95 下的严重错误，以及如何获取有关磁盘、驱动器和目录方面的信息。并将说明如何识别 CD 为音乐、照片或普通的数据光盘。另外，还介绍有关如何清除磁盘缓冲区和拷贝文件。

第 3 章：应用程序和任务控制。本章阐述包括查找当前运行在系统上的其他程序或任务，以及启动或终止任务所需步骤的一些技巧和提示。并将说明如何确保在给定时间内只运行应用程序的单一实例，如何实现后台处理，以及如何从应用程序中终止和重新启动 Windows 系统。

第 4 章：绘图和图形设备接口（GDI）。本章将阐述 GDI 的来龙去脉，以及如何利用 Win-

dows API 来描绘一个优美的世界。这里还将讨论装入、显示、缩放和旋转位图，以及利用鼠标来绘制“橡皮带线”。同时还介绍如何利用简单方法和新的 WinG 图形库来绘制图表和图形，甚至创作动画。

第 5 章：对话框。本章将讨论如何利用对话框作为应用程序的主窗口，如何处理在对话框内的快捷键，如何改变对话框中的字体和颜色及对话框中的单独控制。另外将说明如何通过显示和隐藏控制来激活和禁止对话框中控制及改变对话框的过程。Windows 95 的核心功能之一是广泛应用的对话框内属性单 (property sheets)，这里将说明如何能在应用程序中内含这些特性。

第 6 章：编辑控制。本章介绍有关如何实现只读编辑控制，以及如何能够获取口令，查找和替换文本，添加文本，撤消用户动作和确认输入。

第 7 章：列表框。本章介绍如何能在应用程序中使列表框达到最大程度的实用，如何在纵向和横向滚动列表框，如何在列表框内添加多列，以及如何创建自绘制的列表框。此外，将考察有关 Windows 95 的新的树 (Tree) 控制及说明如何在应用程序中有效地利用这种控制。

第 8 章：菜单。本章介绍如何能激活、禁止、添加和删除菜单项。并说明在菜单中如何选择项及如何改变核选框的外观。最后，将讨论有关应用程序窗口的系统菜单及如何修改它。

第 9 章：文档和编辑器。本章介绍如何自动创建新的 MDI 窗口，如何利用对话框作为 MDI 子窗口，以及如何能够利用简单的 Windows 95 API 命令来实现一个完整的文本编辑器的基本功能。在本章中还考察了在 Windows 95 中新富文本编辑 (Rich Text Edit) 控制及在应用程序中利用它时必须做什么。

第 10 章：打印。本章涉及读者感兴趣的几个功能，包括查找已安装了哪些打印机，并确定其性能。另外还介绍了如何将用户数据复写于文件中，以及如何查找任何特定打印机的可用字体。

第 11 章：应用程序之间的通信。本章涉及有关 DDE、OLE 和 Windows 剪贴板的讨论。对这些在 Windows 95 中较难的概念提供了广泛的例子。

第 12 章：音效和音乐。本章中介绍有关播放声音，读光盘信息及记录 MIDI 信息等。

第 13 章：有关窗口的应用。本章介绍编写屏幕保护程序的基础，还讨论了使用户窗口如何总是保持在顶部或底部，并如何移动和缩放它们。另外还介绍了如何装入其他应用程序中的图标信息，以及如何改变用户自己的程序图标。

第 14 章：程序设计的技巧。本章讨论有关确认指针，置版本信息于文件中，以及利用和装入动态连接库等。

第 15 章：完善应用程序。本章包括了有关利用状态条、工具条、在线帮助等信息。并介绍如何使程序图标动起来，以及如何在对话框或窗口的背景上显示位图。

本书面向的主要对象

本书宗旨是给习惯于继续使用可视开发环境的高级程序员作为媒介来参考。本书目的是为了解决在 Windows 应用程序的正规开发中所遇到的问题。如果开发人员正在使用 Visual C++、Visual Basic、或 Delphi 创建新的应用程序，而且要求超出 MFC、VB 控制及 Delphi 组件所提供范围的功能，那么本书将提供帮助。

此外，本书对于那些愿意进一步提高 Windows 编程知识的学生和业余爱好者也是一本优

秀的参考书。在此，给出的大量具体例子适合于各种不同层次的程序员所需的程序设计问题和方法。

最后，本书分成两种编码“风格”。第一种风格是在单纯的 C 环境下使用原始的 API 函数编程的例子，这种风格适合于那些习惯于使用没有一个与“实际”操作系统相分离的高层库的程序员。第二种风格是利用 Visual C++ 的 MFC 开发环境编程的例子，这些例子是针对那些要求 MFC 的强大功能但需要超出 Microsoft 所支持的功能范围的程序员。在本书中汇集了这两种不同的编程风格，为应用程序的开发提供了一个自由创造的理想天地。

405960

The Microsoft Windows95 开发人员指南	86.00
Visual Basic4 开发人员指南	73.00
Visual FoxPro3.0 开发指南	119.00
Visual Basic4.0API 程序设计	83.00
I/O 接口程序设计入门与应用	28.00
精通 VISUAL BASIC4.0 多媒体程序设计	15.00



Internet 实务系列丛书

快快乐乐畅游 Internet	34.00
WWW 文件设计-HTML 语言实务	29.00
Java/VRML 设计大全	32.00
Web 动态技术入门	23.00
Java 轻松上手	26.00
WWW 上站与建站实务	20.00
Mosaic 与 WWW	10.00
Archie-Internet 标题检索工具	10.00
Netscape Navigator Gold3.0 与网络资源搜寻	20.00
JavaScript 快速查询手册	12.00

北京华章图文信息有限公司

地址：北京市百万庄南街 14 号

邮政编码：100037

电话：68326444 传真：68326444

E-mail: huazhang@public3.bta.net.cn

户名：北京华章图文信息有限公司

帐号：014-144738-13-47

开户行：北京工商银行百万庄分理处信息院代办处

增值税号：110102101444065

目 录

前言

译者序

第 1 章 获取系统信息

- 1.1 确定 Windows 的当前版本 2
- 1.2 获取有关显示器、鼠标及系统的配置信息 3
- 1.3 确定计算机的处理器类型 7
- 1.4 确定多少系统内存是可用的 8
- 1.5 获取系统上的用户注册信息 9
- 1.6 获取计算机的网络名 12
- 1.7 找出计算机上当前登录入网的用户 15
- 1.8 找出键盘上当前按下的键 18
- 1.9 使用 Windows 95 的注册表存取信息 22
- 1.10 找出程序或 DLL 的版本号 30

第 2 章 文件和目录

- 2.1 查找 Windows 目录和 System 目录 36
- 2.2 显示 I/O 错误信息 41
- 2.3 检测驱动器中是否有盘 50
- 2.4 简单地实现文件拷贝 56

第 3 章 应用程序和任务控制

- 3.1 找出系统上正在运行的任务 66
- 3.2 激活另一任务 73
- 3.3 关闭其他的应用程序 78
- 3.4 找出应用程序的执行文件名 83
- 3.5 确保应用程序同时只能运行一个实例 85
- 3.6 在后台运行其他任务 88
- 3.7 启动另外的程序并等待其运行结束 94
- 3.8 从应用程序中终止和重新启动 Windows 98

第 4 章 绘图和图形设备接口

- 4.1 显示 256 色的位图 104
- 4.2 改变位图中的颜色 116
- 4.3 旋转位图 128
- 4.4 随鼠标拖动绘制“橡皮带线” 144

- 4.5 捕捉窗口或部分屏幕 155
- 4.6 生成动画 165
- 4.7 实现位图的拖放 178

第 5 章 对话框

- 5.1 利用对话框作为应用程序的主窗口 191
- 5.2 改变对话框中的字体 197
- 5.3 改变对话框中控制的字体 203
- 5.4 改变对话框的背景颜色 208
- 5.5 激活和禁止对话框中的控制 211
- 5.6 改变对话框中显示的控制 218
- 5.7 在对话框中使用属性单 221
- 5.8 在对话框中的按钮上绘制图象 225
- 5.9 在执行另一操作的同时显示“进度”对话框 228

第 6 章 编辑控制

- 6.1 使编辑控制只读 236
- 6.2 利用编辑控制获取口令 243
- 6.3 改变编辑控制的背景颜色 249
- 6.4 在编辑控制中替换文本 256
- 6.5 给编辑控制添加撤消功能 262
- 6.6 确认编辑控制中的输入有效 269
- 6.7 利用剪贴板进行剪切和粘贴操作 277

第 7 章 列表框

- 7.1 随同列表项存储信息 287
- 7.2 捕捉列表框中的双击 295
- 7.3 通过拖放在列表框中移动列表项 298
- 7.4 滚动列表框 305
- 7.5 实现宽列表的水平滚动 308
- 7.6 在列表框中右对齐数字 311
- 7.7 实现自绘制列表框 318
- 7.8 在列表框中存放更多的列表项 324
- 7.9 实现层次列表(树) 344

第 8 章 菜单

- 8.1 激活和禁止菜单项 356
- 8.2 添加和删除菜单项 364
- 8.3 为菜单项添加核选标记 374

8.4	利用自己定义的核选标记	382	12.1	播放音效	555
8.5	为系统菜单添加选项	397	12.2	读取 CD 中的曲目信息	560
8.6	设计点击鼠标右键后弹出的菜单	403	12.3	播放 CD 音乐	565
第 9 章 文档和编辑器			12.4	播放 MIDI 音乐	574
9.1	自动打开 MDI 子窗口	416	第 13 章 有关窗口的应用		
9.2	在 MDI 窗口中利用无模式对话框	419	13.1	编写屏幕保护程序	584
9.3	随 MDI 子窗口调整列表框或编辑控制 的大小	422	13.2	创建无标题条的窗口	590
9.4	创建简单的文件浏览器	426	13.3	保持一个窗口在所有其他窗口的 上面	593
9.5	查找和替换文本	430	13.4	把一个窗口移到所有其他窗口的 下面	595
9.6	显示大于 64KB 的文件	435	13.5	改变鼠标光标的形状	596
9.7	改变编辑控制中插入光标的类型	458	13.6	改变应用程序窗口极小化时的 图标	599
9.8	改变插入光标的闪烁速度	461	13.7	装入另一应用程序的图标	601
第 10 章 打印			13.8	移动和缩放应用程序中的窗口	605
10.1	确定打印机的性能	465	第 14 章 程序设计的技巧		
10.2	确定当前打印机的页面大小和 方向	469	14.1	确定指针是否有效	610
10.3	利用通用打印机对话框	472	14.2	确定字符串是否有效	612
10.4	打印到文件	477	14.3	在应用程序中放置版本信息	614
10.5	确定可用的打印机字体	479	14.4	编写动态连接库	617
10.6	确定打印队列的状态	482	14.5	利用动态连接库	622
第 11 章 应用程序之间的通信			14.6	创建各种分辨率下都能显示的 应用程序	624
11.1	支持从系统程序拖放到应用程序	487	第 15 章 完善应用程序		
11.2	支持从应用程序拖放到另一 应用程序	490	15.1	实现上下文相关帮助	627
11.3	查看剪贴板的内容	496	15.2	创建状态条	631
11.4	利用剪贴板进行剪切、拷贝和 粘贴	499	15.3	创建工具条	634
11.5	编写动态数据交换客户程序	502	15.4	实现在运行时修改工具条	637
11.6	编写动态数据交换服务程序	506	15.5	从在线帮助中显示范例	642
11.7	在动态数据交换中支持系统主题	512	15.6	在应用程序启动时显示 About 框	647
11.8	使文件对象的链接和嵌入 (OLE) 兼容	547	15.7	显示扉屏 (splash screen)	653
11.9	创建 OLE 服务程序对象	552	15.8	确定应用程序的图标	660
第 12 章 音效和音乐			15.9	显示作为窗口或对话框背景的 位图	662

第 1 章 获取系统信息

设计一个应用程序，运行在各种版本的 Windows 操作系统上，与简单地设计数据表格和菜单相比，程序员需要考虑更多方面的问题。现在的用户都希望应用程序能自动地检测运行环境，比如自动测试运行的 Windows 版本，自动测试屏幕的大小以便将窗口和数据表格置于屏幕的中央，以及自动获取某些网络信息用于应用程序中。更重要的是，应用程序要能默认完成某些操作，而不需要用户的介入。如果如此的话，用户将不需要为每一个应用程序配置键盘，或为每一个应用程序确定 CAPSLOCK 键和 NUMLOCK 键是开状态还是闭状态。

应用程序的所有这些小调整的实现，以及要确保用户操作错误及设置错误不会引起混乱和灾难，这些都是程序员的责任。

1. 如何确定 Windows 的当前版本

如果应用程序使用的某些特性依赖于运行的 Windows 版本，便需要确定用户机器上安装的 Windows 版本。在本节中将介绍如何确定用户的操作系统是 Windows 3. x，Windows 95，还是 Windows NT。

2. 如何获取有关显示器、鼠标及系统的配置信息

确定这些信息是非常重要的：用户是否使用鼠标（因为键盘是很难实现双击的），使用多少磁盘驱动器及什么类型，显示器的大小。所有这些信息都可利用 Windows API 函数得到。在本节中将介绍如何查看磁盘驱动器的数目及其可能的类型，如何确定是否安装了鼠标及用户是否互换了左右键。最后，将介绍如何获取屏幕的大小及如何使窗口或对话框居中。

3. 如何确定计算机的处理器类型

确定用户计算机的处理器类型是很有用的，可以将此信息保存起来，以便于测试程序存取，或用于桌面帮助信息中。假如用户需要各种应用程序信息时，只需简单地按一下键，便将弹出一个对话框，显示用户需要的所有信息。这是使应用程序界面友好的绝妙途径。

4. 如何确定多少系统内存是可用的

在设计一个使用大量系统内存的应用程序时，最好在其运行前能检查是否有足够的资源。为了防止严重的内存和磁盘操作崩溃，最好不断确认程序是否用光了系统内存或其他系统资源。在本节中将介绍如何确定可用的系统内存数，以防止内存和磁盘操作崩溃。

5. 如何获取系统上的用户注册信息

专业版的程序一般能自动检测 Windows 软件的原注册用户。任何人都能取到此信息，但需要巧用 Windows API 函数。在本节中将介绍如何从 Windows 中获取此信息，同时介绍如何巧取其他程序的资源以用于自己的应用程序中。

6. 如何获取计算机的网络名

在任何网络环境中，最好能知道所使用的计算机的网络名。可以使用 Windows API 函数来获得此信息，用于显示或用于用户登录系统。在本节中将介绍如何获取计算机的当前网络名。

7. 如何找出计算机上当前登录入网的用户

一旦获取了计算机网络名，自然也想获取当前登录的用户名。此信息可用于显示，或用在应用程序中定制一些基于用户名的选项。这样用户便可简单地登入计算机，使所有的程序都按自己的需求来运行。这些都可使用 API 函数来实现。

8. 如何找出键盘上当前按下的键

假如用户要滚动复杂的文档，如果应用程序并不检查键 ↓ 按下的同时用户是否按下键 CTRL 或 SHIFT，则显然是非常不方便的。而且，你可以想象假如用户同时按下几个键而应用程序却少记录了一个，这样的应用程序用户能使用吗？在本节中将介绍如何检测键盘上所有键的状态。

9. 如何使用 Windows 95 的注册表存取信息

如果你习惯于使用 Windows 3.x API 编程，那么就可能对初始化文件 (.INI) 进行过大量的操作。这些文件很容易被破坏，或是因为用户使用了文本编辑器编辑，或是因为被其他的应用程序覆盖。Windows 95 通过引入注册表的思想解决了此问题。应用程序利用 Windows 注册表函数来维护此文件。在本节中将介绍如何在注册表中添加新项，以及如何从此文件中读出已存在的项。

10 如何找出程序或 DLL 的版本号

迟早，所有的程序员都会被过时文件的问题所困扰。有时，此问题只是烦人，应用程序不能按要求运行。有时则是灾难性的：破坏数据，内存操作崩溃或死机。在本节中将介绍如何获取文件的版本号，及如何把此信息放入应用程序中以便在运行时进行确认。

表 1-1 列出了在本章中使用的 Windows 95 API 函数。

表 1-1 在第 1 章中使用的 Windows 95 API 函数

GetVersionEx	VerQueryValue	GetFileVersionInfoSize	GetFileVersionInfo
GetSystemInfo	GetDriveType	GetSystemMetrics	
RegOpenKeyEx	GlobalMemoryStatus	RegOpenKey	
WNetGetConnection	RegEnumValue	RegCloseKey	
RegEnumKey	WNetGetUser	GetKeyboardState	
RegCreateKey	RegQueryValue	RegSetValue	

1.1 确定 Windows 的当前版本

问题

有时，程序员需要确定哪一个版本的 Windows 正在运行，以便根据版本号而采取相应的步骤。

方法

已编入文档的 Windows API 函数 GetVersionEx 可以用来确定哪一个版本的 Microsoft Windows 操作系统正在运行。Windows NT 表示为 3.51，指当前发行的 Windows NT 3.51；Windows 95 表示为 Windows 4.0。

步骤

按照下列步骤实现一个例子程序。运行此例子程序，从菜单 SysInfo 中选择菜单项 Windows Version，将弹出一个对话框，显示 Windows 的版本号。

实现例子程序的具体步骤如下：

1. 在 Visual C++ 中，利用 AppWizard 创建新的项目文件，并命名此项目文件为 CH11.MAK。

2. 进入 AppStudio，在菜单 IDR_MAINFRAME 中添加新的菜单 SysInfo。在 SysInfo 中添加新的菜单项 &Windows Version，其对话框 ID 为 ID_WIN_VERSION。

3. 启动 ClassWizard。从下拉列表中选择对象 CMainFrame，从对象列表中选择 ID_WIN_VERSION，选择消息 COMMAND 以定义消息处理。点击按钮 Add Function，键入函数名 OnWinVersion，然后选择按钮 Edit Code。

4. 在 OnWinVersion 中，输入下列代码：

```
void CMainFrame:: OnWinVersion ()
{
    char buffer [80];
    OSVERSIONINFO osinfo;

    // Get version info

    osinfo.dwOSVersionInfoSize = sizeof (OSVERSIONINFO);
    GetVersionEx (&osinfo);

    // Break into the major and minor versions of Windows

    sprintf (buffer, "Version %ld. %ld", osinfo.dwMajorVersion, osinfo.dwMinorVersion);

    MessageBox (buffer, "Windows", MB_ICONINFORMATION | MB_OK);
}

```

用法

Windows API 函数 GetVersionEx 返回一个包含版本信息的结构给应用程序。在本节中，我们只对结构中的域 dwMajorVersion 和 dwMinorVersion 感兴趣，这两个域包含 DWORD 值（即双整数）。

应该注意，调用函数 GetVersionEx 前必须先先在域 dwOSVersionInfoSize 中设置结构的大小，这是为了将来结构的改变而不影响已有的代码，以后新增的所有域都将附加在结构的末尾，而函数将只返回程序员所要求的部分。

注释

对当前所支持的 Windows 95 版本，函数 GetVersionEx 的返回值为 4.0。对当前版本的 Windows NT，其返回值为 3.51。函数 GetVersionEx 是专门的 32 位函数，不能用在 Windows 3.x 中。要得到 Windows 3.x 的版本号，应该使用 16 位的函数 GetVersion。

1.2 获取有关显示器、鼠标及系统的配置信息

问题

有时，程序员希望能够确定用户目前正在使用的是什么配置，此信息应该包括驱动器的数目和类型，以及用户是否使用鼠标，当然，最好还能够确定用户使用的显示器的类型以及显示器的分辨率。

方法

确定所有的信息需要好几步。首先可以调用 Windows API 函数 `GetDriveType` 来确定机器安装了哪些驱动器以及它们是什么类型（硬盘，CD-ROM，软盘，或网络驱动器）。

使用 Windows API 函数 `GetSystemMetrics` 可以获取信息的第二部分（是否有鼠标以及显示器的大小）。此函数返回给程序员大量的信息，在本节中我们只使用其中的一部分。

步骤

按照下列步骤实现一个例子程序。运行此例子程序，选择菜单 `SysInfo` 以及菜单项 `Drives`，将弹出一个对话框（如图 1-1 所示）。对话框包含所有可能的驱动器符（A~Z）及对应驱动器的类型。类型是下列中的一个：

- 不确定的：此驱动器可能在系统中不存在。
- 可移动驱动器：通常为软盘驱动器。
- 固定驱动器：通常为硬盘驱动器或 CD-ROM。
- 网络驱动器或 CD-ROM：通常为网络驱动器，但某些 CD-ROM 驱动器也归为此类。

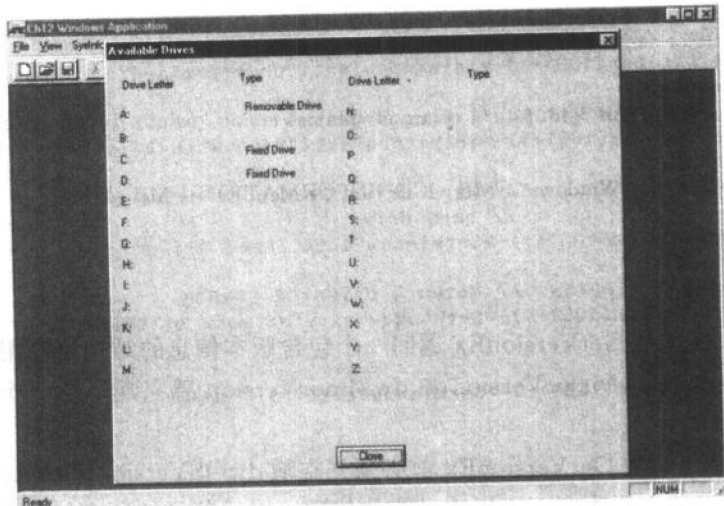


图 1-1 系统可用的驱动器

实现例子程序的具体步骤如下：

1. 在 Visual C++ 中，利用 AppWizard 创建新的项目文件，并命名此项目文件为 CH12.MAK。利用 AppStudio，创建新的菜单 `SysInfo`。在菜单 `SysInfo` 中添加菜单项 `Drives`，标识符为 `ID_DRIVE_INFO`。

2. 在 AppStudio 中，创建新的对话框 `IDD_DIALOG1`。添加 26 个静态文本域，标题为驱动器符（A:，B:，C: 等等）。与其对齐，添加另外 26 个静态文本域。与驱动器符同顺序，ID 分别赋为 1001，1002 等等。

3. 在 AppStudio 中，选择 Class Wizard，然后选择 New Class 来创建新的对话框类。新类

命名为 CDriveDlg，并接受所有其他的缺省值。为消息 WM_INITDIALOG 添加一个函数（命名为 OnInitDialog）。

4. 在刚创建的函数 OnInitDialog 中，添加下列代码：

```

BOOL CDriveDlg:: OnInitDialog ()
{
    CDialog:: OnInitDialog ();
    char buffer [10];

    for ( int i=0; i<26; ++i ) {
        sprintf (buffer, "%c: \\", 'A' + i );
        WORD ret = GetDriveType (buffer);
        switch ( ret ) {
            case 1: // This indicates this is beyond the last drive defined.
                break;
            case 0: // Non-determined drive type
                GetDlgItem (1001+i) ->SetWindowText ("Undetermined");
                break;
            case DRIVE_REMOVABLE: // Floppy drives
                GetDlgItem (1001+i) ->SetWindowText ("Removable Drive");
                break;
            case DRIVE_FIXED: // Hard drive OR mapped drive on hard-drive
                GetDlgItem (1001+i) ->SetWindowText ("Fixed Drive");
                break;
            case DRIVE_REMOTE: // Network drive OR CD-ROM
                GetDlgItem (1001+i) ->SetWindowText ("Network or CDROM");
                break;
            default:
                GetDlgItem (1001+i) ->SetWindowText ("");
                break;
        }
    }

    return TRUE; // return TRUE unless you set the focus to a control
}

```

5. 进入 ClassWizard，从下拉列表中选择对象 CMainFrame，从对象列表中选择对象 ID_DRIVE_INFO，从消息列表中选择消息 COMMAND。点击按钮 Add Function，新函数命名为 OnDriveInfo，在 CMainFrame 的函数 OnDriveInfo 中添加下列代码：

```

void CMainFrame:: OnDriveInfo ()
{

```

```

CDriveDlg dlg;

dlg.DoModal ();
}

```

6. 在文件 MAINFRM.CPP 顶部的 include 文件列表中添加下列 include 文件行:

```
#include "drivedlg.h"
```

7. 编译并运行此例子程序。

用法

API 函数 `GetDriveType` 通常用来确定安装的驱动器类型。尽管此函数可能会被某些 CD-ROM 驱动器、网络驱动器及软件模拟磁盘驱动器所“欺骗”，但仍是一个确定当前系统上哪一个驱动器符可用的理想方法。这些驱动器接着就可以用在应用程序中。

函数 `GetSystemMetrics` 通常用来获取多种有关系统及其性能的信息。在本节中只使用此 API 函数获取的某些有用的信息。程序员可以使用此函数来将窗口置于屏幕的中央,同样也可使对话框和显示的信息居中。

注释

使用 Delphi 也可完成同样的任务。按照下列步骤实现一个例子程序,用来获取有关鼠标及显示器的信息。运行此例子程序,将显示出一个包含有关信息的表单。在例子程序中还使用了函数 `GetSystemMetrics` 来将表单置于屏幕的中央,此函数为获取屏幕大小的有效途径。

按照下列步骤实现此例子程序:

1. 创建新的项目文件 SYSINFO.DPR。在表单中添加三个文本域,标题分别为 `Mouse Present`、`Screen Width` 和 `Screen Height`。添加另外三个文本域,命名为 `MousePresent`、`ScreenWidth` 和 `ScreenHeight`。

2. 双击表单区中的任何地方(除了前面定义的任一文本域外),在表单的方法 `FormCreate` 中添加下列代码:

```

procedure TForm1.FormCreate (Sender: TObject);
Var
  screen_height, screen_width: Integer;
  ht, wt: Integer;
begin
  screen_width := GetSystemMetrics (SM_CXSCREEN);
  screen_height := GetSystemMetrics (SM_CYSCREEN);

  form1.top := (screen_height div 2) - (form1.height div 2);
  form1.left := (screen_width div 2) - (form1.width div 2);
  if (GetSystemMetrics (SM_MOUSEPRESENT) <> 0) then
  begin
    MousePresent.Caption := 'Yes';

```



```

end
else
    MousePresent.Caption := 'No';
    ScreenWidth.Caption := IntToStr (screen_width);
    ScreenHeight.Caption := IntToStr (screen_height);
end;

```

3. 运行此例子程序。显示在屏幕上的表单如图 1-2 所示。

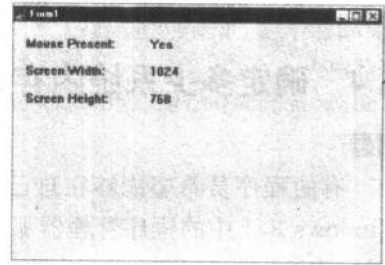


图 1-2 包含屏幕大小和鼠标安装信息并被居中的表单

1.3 确定计算机的处理器类型

问题

有的程序员希望知道运行自己程序的机器安装的是什么处理器，然后就此信息用于桌面帮助信息，或只是用来确定自己的应用程序能否在此机器上运行。

方法

可以使用 Windows API 函数 `GetSystemInfo` 来获取处理器信息。此函数返回多种信息，其中就包括处理器的类型。此函数识别当前的这个（或这些）（因为 Windows NT 可以运行在多处理机上）处理器为 386、486 或 Pentium。由于 Windows NT 和 Windows 95 不能运行在 80286 或更低型号的处理器上，所以此函数不识别这些处理器型号。

步骤

按照下列步骤实现一个例子程序。运行此例子程序，选择菜单 `SysInfo`，选择菜单项 `Processor Type`。将弹出一个消息框，显示当前机器上安装的处理器类型。消息框为下列选择之一：80386，80486，Pentium。

实现例子程序的具体步骤如下：

1. 在 Visual C++ 中，利用 AppWizard 创建新的项目文件，并命名此项目文件为 CH13.MAK。将菜单 `SysInfo` 添加到主菜单中，在菜单 `SysInfo` 中添加新菜单项 `&Processor Type`，ID 为 `ID_PROCESSOR`。

2. 打开 ClassWizard。选择对象 `CMainFrame`，选择对象 ID 为 `ID_PROCESSOR`，选择消息 ID 为 `COMMAND`。选取按钮 `Add Function`，命名方法为 `OnProcessor`。

3. 选择按钮 `Edit Code`，在方法 `OnProcessor` 中输入下列代码：

```

void CMainFrame:: OnProcessor ()
{
    SYSTEM_INFO sinfo;

    GetSystemInfo (&sinfo);
    if ( sinfo.dwProcessorType == PROCESSOR_INTEL_386 )
        MessageBox ( "80386 Processor", "Information", MB_ICONINFORMATION | MB_OK );
    if ( sinfo.dwProcessorType == PROCESSOR_INTEL_486 )
        MessageBox ( "80486 Processor", "Information", MB_ICONINFORMATION | MB_OK );
}

```