

微机操作系统系列丛书 (二)

Microsoft Windows for Pen Computing



Version 1
Designed to
work with
Windows 3.1

Programmer's Reference

超级笔驱动图形环境程序员参考手册

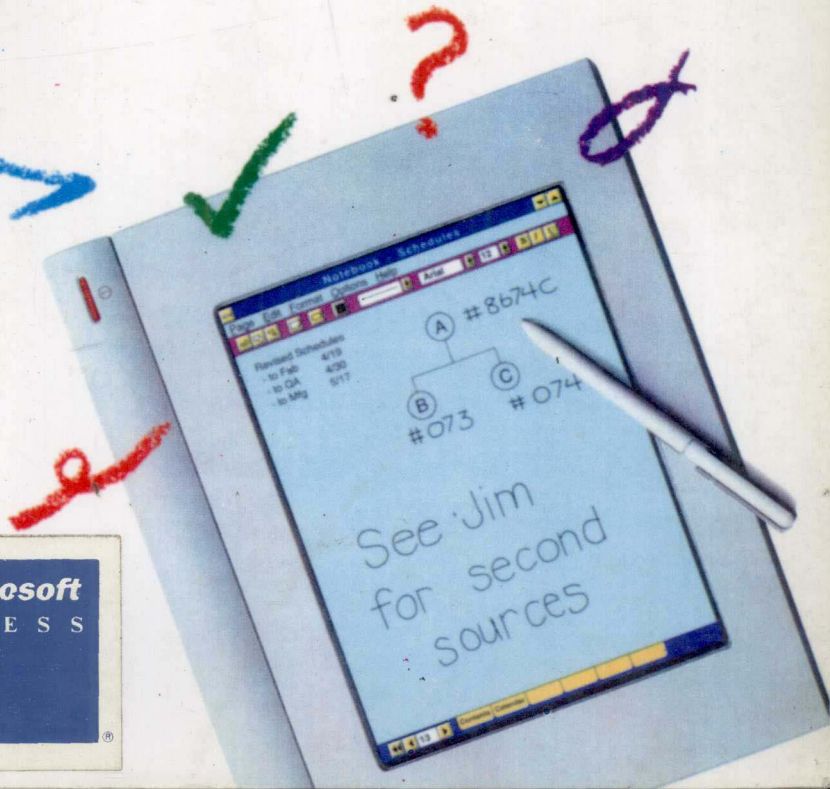
著

译

审校

工 王小潜

都 李淑兰



Microsoft Windows 3.1 for Pen Computing

超级笔驱动图形环境程序员参考手册

Terry A. Ward

著

Stephen M. Liffick

谢 工 王小潜

译

文 都 李淑兰

审校

学 苑 出 版 社

1994

(京)新登字 151 号

内 容 提 要

Windows for Pen Computing 是继 Windows 3.1 操作环境之后,Microsoft 公司推出的又一杰作。它是对 Windows 3.1 的扩充,包括一组动态链接库(DLL)和能够在 Windows 下进行笔输入并识别手写体的驱动程序。不管计算机是否为笔触式计算机,应用程序开发人员都可以增加笔 API 窗口类,还可享受其它笔 API 服务,并且可确保他们的程序能在运行 Windows 3.1 的计算机上一致地运行。Windows for Pen Computing 提供了更有效的安装 Windows 应用程序及硬件平台的基础。

本书重点讨论 Windows 下的笔输入环境,内容包括 Pen Computing 的安装、体系结构、对“纸”上“墨迹”的手写识别、墨迹管理、笔控件、笔 API、笔结构、笔消息和常量。为了加深理解,除了穿插典型的例子外,第五章给出了一个完整的笔应用程序范例。另外,附录中还说明了与笔有关的.INI 文件设置。由于笔输入和手写体识别是一个全新的领域,因此本书不可避免地涉及到一些全新的概念和稍显晦涩的描述;对初学者而言,这很正常——曲径通幽,您会渐入佳境;有了笔输入环境,键盘和鼠标迟早会成为历史。

需要本书者,请与北京 8721 信箱联系,邮政编码 100080,电话 2562329。

版 权 声 明

本书英文版由 Microsoft 出版社(Microsoft Press)出版,版权归 Microsoft 公司所有。本书中文版版权由 Microsoft 公司授予北京希望电脑公司和学苑出版社独家出版、发行。未经出版者书面许可,本书的任何部分不得以任何形式或任何手段复制或传播。

微机操作系统系列丛书(二)

Microsoft Windows 3.1 for Pen Computing

超级笔驱动图形环境程序员参考手册

原 著: Terry A. Ward Stephen M. Liffick
翻 译: 谢 工 王小潜
审 校: 文 都 李淑兰
责任编辑: 甄国宪
出版发行: 学苑出版社 邮政编码: 100036
社 址: 北京市海淀区万寿路西街 11 号
印 刷: 兰空印刷厂
开 本: 787×1092 1/16
印 张: 12.75 字数: 295 千字
印 数: 1~5000 册
版 次: 1994 年 5 月北京第 1 版第 1 次
ISBN 7-5077-0885-3/TP·27
本册定价: 49.00 元

学苑版图书印、装错误可随时退换

目 录

前言	1
组织	1
手册约定	1
附加文档	2
系统需求	2
启动	2
第一章 MS Windows for Pen Computing 入门	3
1.1 目的	3
1.2 什么是 Microsoft Windows for Pen Computing	3
1.3 什么是 Pen Computing SDK	5
1.4 怎样交付 Pen Computing	5
1.5 笔 API 的核心是什么	6
1.6 应用程序怎样支持笔	7
1.7 关于本手册	11
第二章 Pen Extensions 的结构	12
2.1 目标	12
2.2 组件	12
2.3 数据流	16
第三章 识别处理	24
3.1 概述	24
3.2 RC: 主要的数据结构	25
3.3 应用程序和识别程序	27
3.4 识别结果: RCRESULT 结构	32
第四章 管理笔应用程序中的墨迹	36
4.1 HPENDATA 数据类型和墨迹	36
4.2 墨迹函数	39
4.3 使用墨迹对象的共同方案	44
第五章 笔应用程序范例	46
5.1 PENAPP 应用程序概述	46
5.2 WinMain 和初始化函数	47
5.3 数据处理和显示函数	52
5.4 MainWndProc	52
5.5 InputWndProc	54
5.6 InfoWndProc 和 RawWndProc	57

第六章	使用笔控件和 ProcessWriting 函数	58
6.1	使用手写编辑笔控件	58
6.2	使用 Bedit(Boxed Handwriting Edit)控件	65
6.3	使用 ProcessWriting 函数	67
第七章	可取代的组件: 识别程序和字典	70
7.1	识别程序	70
7.2	字典	82
第八章	笔 API 概述	88
8.1	笔 API 分类	89
第九章	笔 API 参考指南	95
第十章	笔结构	152
第十一章	笔消息和常量	176
附录 A	初始化文件指南	193
A.1	.INI 文件的格式	193
A.2	修改 SYSTEM.INI 文件	194
A.3	修改 CONTROL.INI 文件	195
A.4	修改 PENWIN.INI 文件	196
A.5	修改 WIN.INI 文件	198

前 言

本手册描述 Microsoft Windows 图形环境中的 Pen Computing 软件开发工具包 (SDK)。书中包含下列内容:

- (1) 安装;
- (2) SDK 的组成和结构;
- (3) 笔应用程序设计接口(API)。

组 织

本手册包括下列各章及附录:

章	说 明
第一章	“Microsoft Windows Pen Computing入门”, 综述Pen Computing, Pen SDK及启动信息。
第二章	“Pen Extensions的结构”, 综述用来建立笔应用程序的扩展笔API结构。
第三章	“识别处理”, 综述在识别处理中使用的数据结构和方法。
第四章	“管理笔应用程序中的墨迹”, 综述在墨写处理中使用的数据结构和方法。
第五章	“笔应用程序范例”, 给出一个使用子窗口进行手写输入的范例应用程序。
第六章	“使用笔控件和ProcessWriting函数”, 给出一个使用笔控件的范例应用程序。本章还介绍 ProcessWriting 的功能。
第七章	“可取代的组件: 识别程序和字典”, 给出范例识别程序和字典动态链接库(DLL)。
第八章	“笔API概述”, 综述笔API。
第九章	“笔API参考指南”, 描述Pen Extensions的功能。
第十章	“笔结构”, 说明Pen Extensions的结构。
第十一章	“笔消息和常量”, 说明Pen Extensions消息和常量。
附录A	“初始化文件指南”, 用于Windows和Pen Extensions的.INI文件设置。

手册约定

本手册使用下列约定:

作 用	说 明
()	在语法说明中, 用括号括起一或多个要传给函数的参数。
:	程序范例中的垂直省略号说明这部分程序被省略了。
...	跟在一项之后的水平省略号说明可能有更多相同形式的项出现。
[[]]	在命令行或语法说明中用两个方括号括起可选字段或参数。
	垂直杠说明可选择它两边的任意一项; 在符号图中, 垂直杠指明可能的字符选择。
{ }	花括号说明用户必须选用括起的项之一。
大写字母	大写字母说明键名和键序列名, 例如CTRL+ALT+DEL。如果键名用逗号分隔而不是用加号分隔, 例如 ALT, F, 这时就必须依次而不是同时按下所列的键。

附加文档

本手册假定读者已基本熟悉 MS Windows 编程。下面列出的参考文献解释了 MS Windows 中的 Pen Computing 图形环境:

- (1) Microsoft Windows for Pen Computing文档: 这是笔计算系统的介绍和自学资料, 这些资料说明用户接口和 Windows 应用程序。
- (2) Microsoft Windows for Pen Computing联机帮助: 提供Windows用户文档的完整说明。
- (3) Microsoft Windows软件开发工具(SDK)文档或等价的文档说明: 提供有关 Microsoft Windows 图形环境的应用程序设计接口的信息。
- (4) Microsoft Windows设备驱动程序开发工具(DDK)文档或等价的文档说明: 描述 Microsoft Windows 设备驱动程序的应用程序设计接口。如果要开发笔或图形输入板驱动程序, 则要求了解这一方面的内容。

系统需求

利用下列软件和硬件可以开发笔应用程序:

- (1) IBM个人计算机或能兼容运行Microsoft Windows 3.1或后续版本的计算机。
- (2) 鼠标、图形输入板或Microsoft Windows for Pen Computing系统所支持的其它定位设备。
- (3) Microsoft Windows 3.1版SDK。
- (4) Microsoft Windows 3.1版设备驱动程序开发工具(DDK), 仅在建立笔、显示器、图形输入板或键盘驱动程序时才需要。
- (5) Microsoft C优化编译程序5.1版或更高版本, 或Microsoft Quick C for Windows 1.0 版或更高版本。
- (6) Microsoft宏汇编程序5.1版或更高版本 (仅在建立笔、显示器或键盘驱动程序时才需要)。

还可以使用由其它制造厂家 (如 Borland 国际公司) 生产的相应开发软件。

启动

可运行系统盘中 1 号盘上的 Windows 程序 SETUP.EXE 来安装此 SDK。

注意: 不要把此版本的 SDK 安装在任何旧版本上。必须新的目录中安装此 SDK。本书假定已把 SDK 安装在缺省的 PENSDK 目录中。

用户一定要阅读 1 号盘上的 README.TXT 文件。

第一章 MS Windows for Pen Computing 入门

本章提供有关 Microsoft Windows for Pen Computing 的背景信息。该章讨论了一般的 Pen Computing, Pen Computing 应用程序设计接口(API)以及基于笔的应用程序接口设计, 本章的内容是使用 Microsoft Windows for Pen Computing 系统进行笔计算的基础知识。

1.1 目的

自从 1988 年初, Microsoft Windows for Pen Computing 开发小组的目标就已确定为建立兼容的基于笔的操作环境。

基于笔的操作环境能够建立可扩充当前图形用户接口(GUI)技术以及在“笔和纸”水平上与用户交互作用的应用程序, 这是令人振奋的, 因为它相对于当前的键盘和鼠标来说更适合用户, 给用户造成的困难更少。人们对笔是很熟悉的, 也知道怎样使用, 因此, 如果用笔作为交互作用的手段, 那么用户和计算机之间的交互就更加自然了。

兼容的基于笔的操作环境可以不加修改地与已有的 Windows 应用程序进行交互。Microsoft Windows for Pen Computing 提供了更有效的安装 Microsoft Windows 应用程序和硬件平台的基础。借助 Windows 独立软件商(ISV)的代码基础和经验, Pen Extensions 拓展了 Microsoft Windows 的应用范围。

MS Windows for Pen Computing 展示了当前 Windows 产品的发展方向, 并提供了 ISV 感兴趣的工具, 利用它可建立以笔为基础的高级应用程序。

1.2 什么是 Microsoft Windows for Pen Computing

MS Windows for Pen Computing (也称 Pen Extensions) 是 MS Windows 3.1 操作环境的一系列扩充模块。Pen Extensions 包括一组动态链接库(DLL)和能够在 Microsoft Windows 下进行笔输入并识别手写体的驱动程序。Pen Extensions 的组成对普通 Windows 3.1 应用程序是透明的, 而且也适用于那些想提高功能的应用程序。

在一组全新的 API 中都得到笔服务(请参考后面讨论的“笔 API”), 每台运行 MS Windows 3.1 的计算机都可使用笔 API。

不管计算机是否为笔触式计算机, 也不管它是否安装了笔设备, 应用程序开发人员不仅可以增加笔 API 窗口类, 还可以使用其它笔 API 服务, 并且可确保他们的程序能在运行 MS Windows 3.1 的计算机上一致地运行。

如果使用这样的笔, 就可用笔 API 增强应用程序, 使得它能够应用高级的专用笔特性, 并自动借助笔控件类进行交互作用。如果没有这样的笔, 也不需要修改.EXE 文件, 因为它同样能够运行, 只是不能进行笔操作而已。

目前已经为 MS Windows 3.1 设计并建立起这样的 Pen Extensions 了(专用笔特性要求此版本或更高版本的 Windows)。

Pen Extensions 可细分为四大类, 下面几节将简要描述之。

1.2.1 设备驱动程序

欲使用 MS Windows for Pen Computing, 则需要安装笔驱动程序, 一般地还需使用调整过的标准显示器驱动程序。

1. 笔驱动程序

笔驱动程序是一个可安装的设备驱动程序, 它的主要作用是从数字化设备中取出数据供 Windows 系统使用。来自数字化设备的数据一般由(x, y)坐标组成, 它指定笔的位置, 还可包含压力和角度信息。应用程序可获得所有由笔驱动程序提供的信息, 并由此决定怎样使用这些信息。

笔驱动程序相对于鼠标驱动程序来说有三个重要区别。首先, 相对于鼠标驱动程序, 它要以更高的采样率和更高的分辨率报告数据。为了支持手写识别, 这是必要的。其次, 它还应该报告压力、角度、旋转或其它有关笔状态的信息。最后, 它可借助特有的 Pen Computing 接口来管理高数据率和密度, 以此避免大量无用的信息。

2. 显示驱动程序

高质量的笔用户接口要求能识别墨迹, 即把在当前笔位置上画线这种操作反映到屏幕上。为了提高性能并减少代码重复, Microsoft 公司已在 RC 管理程序中完成了识别墨迹的功能。

要支持墨写处理, 则需要 Windows 显示驱动程序和笔接口之间建立通信; 采用此方法可以在中断时间绘出墨迹。非常类似于在使用鼠标器时在屏幕上立即显示出鼠标光标, 通过 MS Windows for Pen Computing 和 Windows 显示驱动程序在中断时间的交互作用就能在使用笔时立即显示出墨迹。

1.2.2 识别程序

系统的这一部分(实际上把(x, y)点流转化为可识别的字符)其实是 DLL, 称为识别程序(recognizer)。Microsoft 识别程序能够识别出手写的 ANSI 字符集中的字符, 但是, 识别程序部件也可被三组识别系统所取代, 这些系统可以提高识别率, 识别不同的符号集并处理 Microsoft 识别程序还不能识别的情况。一些其它的生产厂家已经开始开发 Pen Computing 系统的识别程序。

除了标准字符之外, 识别程序还可以识别圆圈、方块、三角形、Kanji 符号、Gregg 缩写、数学符号、CAD/CAM 符号及其它符号和字符。为了识别圆圈、椭圆、矩形、正方形和直线, MS Windows for Pen Computing SDK 中还包含一个特殊的识别程序(SHAPEREC.DLL)。

1.2.3 Pen Computing DLL

PENWIN.DLL 是 Windows 3.1 系统中不由识别程序处理的所有专用笔部件的管理程序。PENWIN.DLL 是笔 API 的主要扩充部分，它是笔驱动程序、显示驱动程序、识别程序和 Windows 笔应用程序之间的媒介。

MS Windows 3.1 的零售版中包含 PENWIN.DLL，运行时，PENWIN.DLL 确定系统是否为笔触式系统并采取适当的操作。正因为 Windows 3.1 系统中有这种 DLL，才保证了开发人员总能对任何其它的笔 API 进行调节。

1.2.4 笔接口

在 MS Windows for Pen Computing 中有很多新的接口组件，它们使笔交互作用更容易进行，借助它们还能访问一个小的浮动式工具条（称为笔调色板）。这些新的接口组件包括训练程序、Gesture 编辑程序，屏幕键盘和书写窗口，还包括四个新的控件，它们提供了判别系统的专有用户、设置用户配置选项、确定屏幕方位的手段。在本手册后面还要详细讨论这些组件，它们为用户提供了基本的接口，使得用户能管理和修改它们的 Pen Computing 环境。

1.3 什么是 Pen Computing SDK

Microsoft Windows for Pen Computing SDK (Pen SDK) 是 MS Windows 3.1 版 SDK 的扩充。Pen SDK 包含库、INCLUDE 文件和数字采样源。Pen SDK 可使 Windows ISV 开发出基于笔的应用程序和定制的识别程序。

开发笔应用程序和识别程序也需要 MS Windows 3.1 版 SDK，Pen SDK 假定 Windows 3.1 SDK 中有现成的 INCLUDE 文件、库和工具。

任何能够编译用 C 编写的 Windows API 的开发工具都能够编译笔 API 和库，只要开发工具直接调用 Windows 3.1 函数，并与包含在 MS Windows 3.1 中的标准 API 库相链接。

1.4 怎样交付 Pen Computing

MS Windows for Pen Computing 是原设备制造商(OEM)的产品，这意味着开发者和最终用户只有在购买了相应的硬件时才需购买此产品，这些硬件包括可移植的笔触式计算机、笔记本计算机（带屏幕且像 Pen Computing 计算机一样便于携带）或数字化的图形输入板（用户桌面）。

要注意的是，前面提到的 Pen SDK 不包括 MS Windows for Pen Computing 运行组件。读者可在 Windows 3.1 下测试自己的笔应用程序，但是，只有在安装了笔硬件时才能完整地测试它们。

1.5 笔 API 的核心是什么

本节假设读者已熟悉标准 Microsoft Windows 程序设计约定，这里只给出简要介绍，在本书后面将进行详细讨论。

用 Windows 应用程序开发专用笔功能有七个主要元素：手写控件(hedit)、盒式编辑控件(bedit)、数据流、ProcessWriting 函数、Recognize 函数、RC 结构和 RCRESULT 结构。

1.5.1 手写编辑控件

手写编辑控件是编辑控件的取代品。手写编辑指常规的 Windows 编辑，但它接受手写输入，并能在消息传递接口中进行识别处理。

1.5.2 盒式编辑控件

盒式编辑控件完全是一种新的 Windows 控件，它能够实现盒式输入。下面是盒式编辑控件的例子，它展示出字符单元。有时把此控件称为 comb。

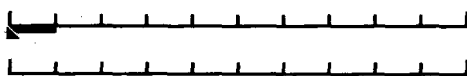


图 1.1 盒式编辑控件

由盒式编辑控件支持的消息集实质上是标准编辑控件消息集的超集，但它提供了识别处理和修改手段。这类控件是为笔设计的，但它还支持与带触发式键盘的计算机兼容的键盘接口。

1.5.3 数据流概述

Pen Extensions 中的数据流类似于常规 Windows 程序中的数据流。笔驱动程序把事件报告给 Windows，就好像它们是鼠标事件一样。应用程序把笔当作鼠标一样处理。

当用户把笔放到一个区域中或窗口中时开始操作，该操作所对应的 Windows 消息是 WM_LBUTTONDOWN，应用程序能够确定该区域是书写区。当应用程序确定笔正处于书写区中时，它就调用 ProcessWriting 或 Recognize 函数并等待结果，这两个函数激活墨写过程，处理墨迹识别，并把结果返回给应用程序。

有关数据流的更详细的讨论可参看第二章“Pen Extensions 的结构”。

1.5.4 ProcessWriting 函数

应用程序使用 ProcessWriting 函数请求所有要用的识别参数以及作为 WM_CHAR 返回的结果，这是为现有 Windows 应用程序添加手写功能的最快的方法（由于应用程序受到管理识别结果的复杂性的限制）。应用程序接收 WM_LBUTTONDOWN 消息，并通过调用 ProcessWriting 进行响应。WM_CHAR 被接收，且 ProcessWriting 返回之

后，处理过程就重新启动。

1.5.5 Recognize 函数

如果说 ProcessWriting 函数是为现有 Windows 应用程序添加手写功能的最容易的方法的话，那么 Recognize 函数就是最灵活的方法。调用程序可以完全控制识别处理，在 Recognize 返回之前，通过称为 WM_RCRESULT 的新消息返回结果。有必要对识别处理执行更精确的控制或实现一些高级功能的应用程序可以使用此函数取代 ProcessWriting。

1.5.6 RC 结构

RC 结构是笔 API 的主要控制数据结构，它由应用程序使用，以便在调用 Recognize 或 ProcessWriting 时调节识别处理。RC 控制像墨迹宽和颜色这样的参数，以及所使用的识别程序、产生结果的时间（用户根据它终止墨写）、所希望的输入类型（即数字或字符）及其它参数。

1.5.7 RCRESULT 结构

此结构用于把识别结果传给调用 Recognize 或 ProcessWriting 的应用程序。WM_RCRESULT 消息包含指向 RCRESULT 结构的指针，RCRESULT 包含识别程序认为用户最可能输入的东西、可能的选择和用户输入的实际墨写数据。

1.6 应用程序怎样支持笔

下面讨论的内容可以看作是增强笔应用程序功能和修改现有 Windows 应用程序的准则，根据这些准则可以知道要修改应用程序的什么地方（如果它是在基于笔的系统上运行的话）、在哪里特别要求支持笔以及通过与高级专用笔特性的协作把值加到哪里。

注意：没有什么能代替在集成显示 / 数字化设备上测试应用程序（如果没有实际的基于笔的计算机），以了解笔的强度和限制。特别地，要在这样的计算机上测试应用程序，以便确认安装了笔时它的工作情况。这类测试有助于了解哪些笔接口更加合适。

1.6.1 对笔的基本认识（笔能力的应用）

要对笔有一个基本的认识，这是必经的基础工作，为的是进一步了解基于笔的接口和笔的特性。对笔有基本的认识而所作的努力是最低限度的，这样生成的应用程序才能完全与面向桌面的应用程序兼容。

1. 利用 RegisterPenApp 记录应用程序

调用 RegisterPenApp 函数将产生由手写编辑窗口在应用程序中建立的所有编辑窗口的替代品，注意这包括组合框的编辑字段部分。这是认识笔的第一步。但是，当用手写编辑控件取代它们时，就不可能对标准编辑控件进行子分类。

2. 手写识别是困难的

手写识别存在的一个问题，就是很难使用户满意。应用程序要做的第一件事就是限制最大扩展可能的手写量。例如，如果能为组合框提供一个可接受的输入清单，就不妨给出该清单，根据此清单可得到近似 100% 的识别率。

类似地，对按钮的识别总是准确的。用户会发现，使用按钮和选择清单将提高应用程序的可用性，而不管是否用到笔。

3. 书写区域

在应用程序中提供书写区域是可行的，这样就允许文本和手势表示项。ProcessWriting 函数就是为此目的而提供的，它支持标准的编辑手势表示和文本项。在 Pen SDK 中，PENSDK\SAMPLES\PENPAD 目录中的 PAD 范例程序用到了此函数。

4. 电源管理

Windows 笔应用程序以及所有 MS Windows 3.1 应用程序都可在带有有限电池的便携式系统上运行。在 Windows 3.1 中，新的电源管理设施提供了检测 Windows 应用程序中空闲时间的手段，并在系统空闲时换上特殊的节能机制。但是，陷入作为主消息循环的 PeekMessage 循环中的应用程序从不会空闲。不调用 GetMessage 或 WritMessage 的应用程序将会过早地耗尽任何便携式计算机的电池，因为它们使 Windows 空闲检测机制失效。

5. 显示问题

Windows 笔应用程序要在单色显示器上运行，因为许多早期的基于笔的计算机使用单色显示器。设计位图和彩色图形时，应用程序设计者应该考虑到这一点。

在设计应用程序时，应该想到桌面配置很可能从一个执行文件改变为另一个执行文件，因为屏幕可能从肖像模式转换到风景模式，所以，Windows 程序应能动态地对每次执行确定显示方式。在设计工具条和对话框时还应特别注意，因为很容易使它们全部或部分不在屏幕上（当显示器处于肖像模式时）。

6. 避免只使用键盘

在应用程序中要小心创建依赖于键盘加速键的命令，例如，仅通过按 ESC 键将窗口恢复到正常大小。但是，如果用户没有键盘，就无法把窗口恢复到正常大小了，这时可用按钮或手势取代键盘上的键。这是工具条可为笔和鼠标用户提供又快又好的接口的又一例证。

7. 理解便携式平台

因为 Microsoft Windows 不断地开发更好的便携式平台，所以应用程序开发者应该严格遵守这样的平台大小和存储限制。

有许多策略可以保证这一点。一种是使用压缩文件格式来存储应用程序和数据，另一

种是打破应用程序的数据依赖性，把它分解成许多小块(DLL)。如果只需要应用程序的一个子功能模块，就可把这些小块“抛之脑后”。一般地，在有很大容量的机器上可以一次安装整个应用程序，但基于笔的计算机没有这样大的存储容量。

8. 其它要考虑的问题

任何时候都有理由要求遵循“文本出现在所写的地方”这一准则：用户在书写被接受之前不必进入字段或窗口，这是手写编辑类的特性之一。在应用程序中，任何书写区域的操作都应相同。

笔“很擅长”指定总的位置，如指示区域，选择菜单项，按下按钮等，但笔“不擅长”指定屏幕上的位置。要检验上述事实，可试着用笔在单个显示像素上点一下，这将影响用户对按钮大小、所按对象的句柄大小和字体大小等的判断。

1.6.2 增强型笔应用程序

“扩充笔识别”描述了提供特定笔功能的应用程序的质量。有许多元素有助于使 Pen Computing 更易进行，因为这些元素是以用户都熟悉的交互作用方式为基础的。以下将讨论这些问题。

1. 盒式编辑控件

对于从用户那里获取手写输入来说，盒式输入是最强有力的方法，并且只要有可能，就应在笔应用程序中使用它。盒式编辑控件的优点如下：

- (1) 识别程序有很好的分段和基线信息。
- (2) 当局限于盒时用户书写得更整齐。
- (3) 盒式控件接口对于笔输入特别自然和合适。

2. 墨写字段：把墨迹作为数据保留

用户输入墨迹并作为墨迹存储起来的能力是非常强大的。手写编辑和盒式编辑类都支持此功能，但还需要把墨写能力添加给现有的窗口类。通过把墨迹以模拟格式存储起来，就可以避免与手写识别有关的问题，并且用户还能够理解和管理包含在墨迹中的信息（实质上，这是电子设备作了记录）。

如果墨水是作为图形数据输入的，那么要在后面识别出此数据将很困难。在 MS Windows for Pen Computing 中，在手写编辑和盒式编辑控件中实现了延迟识别。笔 API 中的 RecognizeData 函数可使应用程序延迟识别它自己存储的墨迹。

应用程序开发人员不应该忽视在应用程序中提供墨写能力的重要性；利用笔 API 实现墨写功能是很容易的。把强有力的可用性很好的特性加到自己的应用程序中将是非常有益的。

3. 上下文

笔应用程序能够提供在识别处理开始时使用的与上下文有关的信息以及在处理结束之

后使用的与上下文有关的信息。通过提供上下文，应用程序可大幅度地提高它们的识别率。

例如，在识别开始之前，让识别程序知道将会出现的数据类型是很有用的。如果字段是数值型的，那么就应使编制的识别程序只要求数值型的数据；如果字段是字母型的，则应使编制的识别程序只要求字母型的数据。其它类型的上下文信息包括手势表示、Kanji符号和常规的字母数字字符。手写编辑和盒式编辑控件通过提供的上下文信息支持消息接口，并且当使用 ProcessWriting 和 Recognize 时，RC 结构提供相同的信息。

除了为识别程序提供预处理线索之外，MS Windows for Pen Computing 提供了字典路径，以检查待识别的笔输入是否在词条集合中，从而帮助识别处理。字典是一个 DLL，它与 PENWIN.DLL 通信，以帮助确定识别程序的猜测中哪个是最佳的。可将此功能放到用户应用程序中的适当位置，例如状态清单、应用程序的专用术语、用户名或关键字，也可以是任何其它可对识别处理进行核实的预期输入。

借助此方法，即使是写得很差的手写输入，也可通过把它与所期望的输入集进行比较而得到正确的解释。MS Windows for Pen Computing 包含了一个英语字典，以便为正常的英语输入提供此能力。

4. 好的图形用户界面

除了在本章前面说明“手写识别是困难的”之外，还有更加基本的 GUI 元素需要特别注意，以保证产品有好的笔交互功能，它们包括对象链接和嵌入(OLE)、定向操作、工具条的使用和通用的标准用户界面。

1.6.3 高级笔功能（以笔为中心的应用程序）

以笔为中心的功能可直接用在基于笔的平台上，它们通常很复杂，以致难以只用键盘和鼠标来管理。将注意力集中在实现基于笔的便携式计算机上的个人信息管理(PIM)应用程序就属于这一类。以笔为中心的特性和功能很多，下面只给出一般的思想。

1. 文本就在所写的地方

描述此特性需要用到几个术语，智能目标确定(smart targeting)就是其中之一，其思想是，如果用户把文本写在特定的地方，那么应用程序就能“明白”要把此文本放到何处，并且保证把它放到那里。此功能其实就像电子纸一样简单，亦即，可以在写文本的地方识别文本，并且把它放到用户以后要管理的目标中。

2. 注释

相对于墨写字段，注释层更倾向于围起所有字段或位置，而限于个别字段或位置。用户可以在应用程序窗口上“潦草”地书写并打印注释，选择、修改以及隐藏它们，这是一个非常复杂而又强有力的以笔为中心的功能。这种功能的一个例子就是注释字处理文档并随同文档文件把手写编辑传回给作者以便修订。

3. 特殊识别和外形识别

特殊的识别处理是另一个强有力的以笔为中心的功能，它可以把雕刻文字转换成特殊应用程序和上下文的输入。MS Windows for Pen Computing 中包含的外形识别程序就是这样一个识别程序。

能够重新构图（如将大致的圆和正方形重构成实际的圆和正方形）的绘图程序包是很有价值的。例如，CAD/CAM 应用程序中就有为识别所属应用领域中的专用符号而设计的特殊识别程序。

4. 笔和纸

笔和纸喻指从用户到应用程序的转换：应用程序将工作区域作为“纸”来对待，并作出进一步的决定和处理。

这类功能包括上下文相关的笔输入。我们考虑在应用程序的绘图区中输入潦草文字的例子。如果外形识别程序不能确定写的是什麼，那么就把它作为草书字保留起来。如果它能识别草书字，那么就快速将它与圆圈、正方形、三角形或直线作比较，看它像哪一个。如果把相同的雕刻文字输入到应用程序的书写区中，则将其转换成字母。

笔和纸的另一个功能称为数学纸(math paper)。如果把应用程序的区域设计成针对数学符号，那么当用户写出数学符号时，就应该知道怎样执行计算。例如，如果应用程序能够“明白”等号是手写输入的一部分，那么等号左边的输入就应当作为一个公式看待，而且必须提供答案。

这类应用程序的另一个共同特性是能够忽略光标。借助笔，用户可以把笔尖放在定位设备上，以指明它的位置，并由此减少对光标的依赖。在实际操作中，用户会发现这只是部分正确，但当视觉反馈从光标移到笔所指示的对象或区域的修改处时，它将变得既强有力又易于掌握。例如，一个对象的句柄只有在笔移到对象的边缘上时才出现。

1.7 关于本手册

本书后面的部分将对前面提到的问题作进一步的阐述，其中包括 MS Windows for Pen Computing 的详细背景知识和有关 RC 及 RCRESULT 结构的使用信息，此外，还包括在应用程序中增加墨迹支持的有关信息。与 MS Windows API 参考手册相结合，本书可作为在应用程序中使用笔 API 和 Pen Computing 功能的指南。

第二章 Pen Extensions 的结构

本章讨论 Microsoft Windows for Pen Computing 的结构和组件，并讨论这些元素是怎样与 Windows 3.1 系统交互作用的。假定读者已熟悉基本的 Windows 结构、程序设计技术和命名约定。

2.1 目 标

Pen Extensions 的目标是给独立软件开发者(ISV)提供建立运行于 Microsoft Windows 顶层上的笔应用程序的能力。功能强大的笔应用程序要求比简单的鼠标和键盘更多的东西。灵活的应用程序设计接口(API)必须允许 ISV 在设计它们的笔接口时完全自由。API 必须能够使“笔和纸”以自然而直观的方式与用户交互。

这些目标要求下列几点:

- (1) 笔必须以用户期望的方式使用。例如，当在图形输入板表面移动笔时必须留下墨迹，就像用笔在纸上写字一样。此外，笔接口应当是自然的，而且容易使用。
- (2) 用户必须能在单个笔操作中指定命令和位置，这称为手势表示(gesture)。手势表示的使用对于笔和计算机的结合来说是独一无二的，而且它给用户提供了某些比常规的笔和纸的交互作用更多的优点。
- (3) 应用程序必须能存取所有识别结果和后选识别结果，还必须能够在用户输入时处理墨迹。这种灵活性使应用程序能够以适合于任务的方式处理用户输入，即信息以墨迹的形式留在纸上，并被解释成文本字符或图形，或者为用户提供后选清单，以便获得正确的结果。
- (4) 应用程序应能建立和使用为数学符号、Gregg速写及其它专用的手写识别目的而定制的识别程序。笔 API 提供了传送单个用户输入给多个识别程序的机制，这样，如果一个识别程序不能确定它是正确的识别结果，那么另外一个就有可能确定。
- (5) 所安装的 Windows 应用程序应提供与笔交互作用的手段，它应采用简单的鼠标仿真和写窗口的形式，识别结果由此送给应用程序。此外，该特性应能在系统中所有书写区域中工作（至少利用最少的笔功能，而不是真实的笔和纸功能），不管应用程序是否为支持这样的笔而作了特殊修改。

2.2 组 件

图 2.1 说明了构成 MS Windows for Pen Computing 的各种组件间的关系以及 Microsoft Windows 和 Windows 应用程序间的关系。