

Windows NT 3.1 编程技术

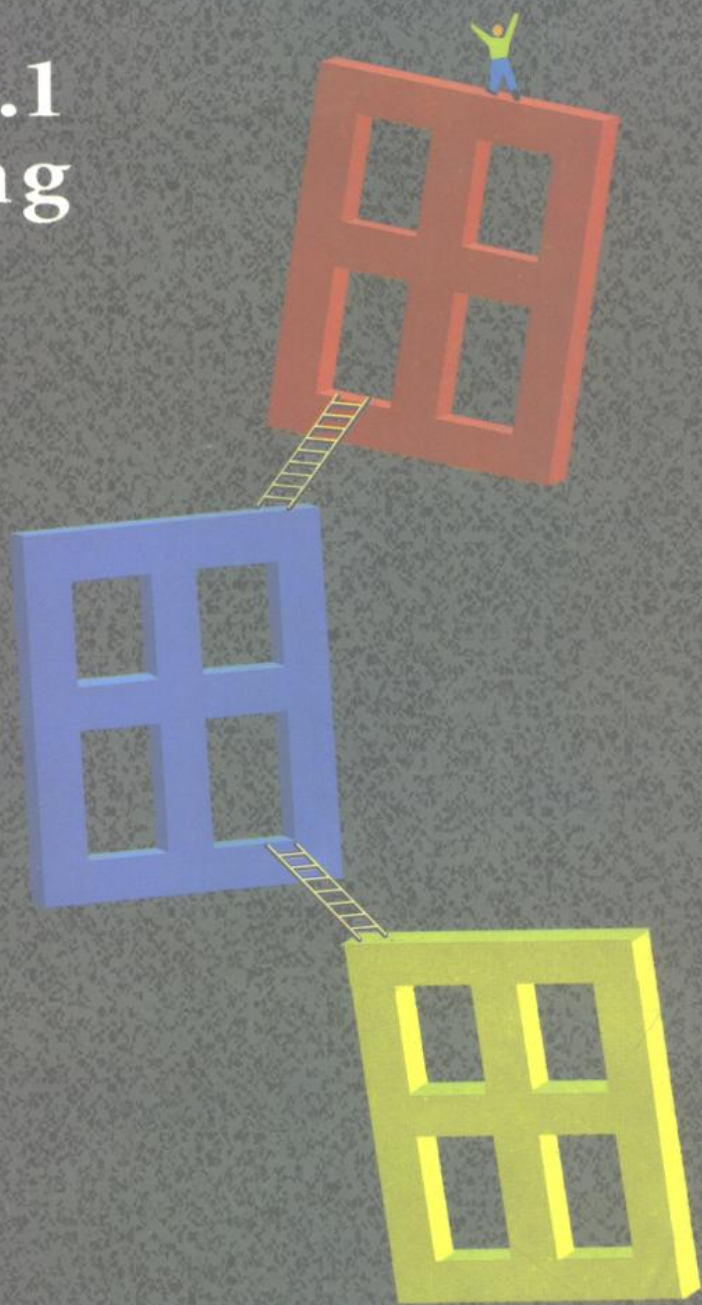
Windows NT 3.1
Programming



**Ben
Ezzell**

李成辉 周长缨 译
刘锦德 校

電子工業出版社





Windows NT 3.1 Programming

Ben Ezzell

Windows NT 3.1 编程技术

李成辉 周长缨 译
刘锦德 校

电子工业出版社

(京) 新登字055号

内 容 提 要

本书简明地介绍了美国微软 (microsoft) 公司最新推出的 Windows NT 操作系统的编程技术和环境。全书共分四部分, 其中第一部分介绍Windows NT如何运作及所提供的编程工具, 第二部分考察WindowsNT应用编程时可运用的资源及有关资源编辑器的类型, 第三部分探讨Windows的图形操作问题, 第四部分则介绍应用程序之间各种数据交换方法。书中还列举了200多个源程序, 用来说明Windows NT的编程原理, 这些源程序复制在一张软盘上, 供读者选购。

本书适用于计算机软件开发人员, 计算机专业技术人员及计算机专业的大学生和研究生。



Copyright© 1993 by Ziff-Davis Press. All rights reserved.

Ziff-Davis Press and ZD Press are trademarks of Ziff Communications Company.

本书英文版由美国Ziff-Davis Press出版, Ziff-Davis Press已将中文版独家版权授予北京美迪亚电子信息有限公司。未经许可, 不得以任何形式和手段复制或抄袭本书内容。

Windows NT 3.1 编程技术

李成辉 周长缨 译

刘锦德 校

责任编辑 陈平

*

电子工业出版社出版 (北京市万寿路)

电子工业出版社发行 各地新华书店经销

北京美迪亚电子信息有限公司排版

北京市顺义县天竺颖华印刷厂印刷

开本: 787×1092毫米 1/16 印张: 31.375 字数: 700千字

1993年12月第1版 1993年12月第1次印刷

印数: 4000册 定价: 49.50元

ISBN 7-5053-2185-4/TP·559

出版说明

计算机科学技术日新月异，为了引进国外最新计算机技术，提高我国计算机应用与开发的水平，中国电子工业出版社与美国万国图文有限公司合资兴办的北京美迪亚电子信息有限公司取得了美国 Ziff-Davis Press 的独家版权代理。Ziff-Davis Press 授权本公司通过电子工业出版社等出版机构全权负责在中国大陆出版该公司的中文版和英文版图书。现在与广大读者见面的是最近推出的第一批图书。今后我们还将陆续推出 Ziff-Davis Press 的最新计算机图书和软件，为广大读者提供更好的服务，传递更多的信息。

美国 Ziff-Davis Press 是全美最大的计算机出版商之一，它出版的书籍、杂志和光盘，主办的展览和会议，提供的咨询和网络服务，形成了整个行业潮流的主导。我们优选翻译出版的图书是 Ziff-Davis Press 的最新计算机图书，并在出版过程中直接采用了该公司提供的电子排版文件，从而大大缩短了图书的出版时间，从根本上弥补了以往翻译版图书要落后原版书较长的“时差”现象，这在电子技术日新月异的时代具有深远的意义。

北京美迪亚电子信息有限公司
1993年10月

目 录

序言	1
----------	---

第一部分 Windows NT 编程技术初步

第一章 Windows NT 环境	7
1.1 硬件环境	8
1.2 软件（系统）环境	9
1.2.1 磁盘文件系统——16位或32位	9
1.2.2 路径与目录要求	9
1.3 Windows NT 编程工具	12
1.3.1 Borland C 的安装	13
1.3.2 Microsoft的WIN 32软件开发包	13
1.4 小结	16
第二章 Windows NT 应用的编程技术	17
2.1 WinHello： 一个入门性的Windows程序	17
2.1.1 WinMain 过程（函数）	18
2.1.2 创建应用程序窗口	21
2.1.3 消息处理循环	23
2.1.4 消息以及事件驱动方式下的编程	24
2.1.5 .DEF（定义）文件	29
2.2 Windows NT 应用样板程序： Template.C	30
2.2.1 Template.I 文件	31
2.2.2 Windows.H 头文件	31
2.2.3 Template.H 头文件	32
2.2.4 Template.RC 资源描述文件	32
2.3 Windows 编程规则	34
2.3.1 常量类型与标识符	35

2.3.2	数据类型	36
2.3.3	Windows结构	36
2.3.4	句柄标识符	36
2.4	小结	37
2.5	WinHello 程序清单	37
2.6	Template 程序清单	40
第三章	从DOS到Windows NT	48
3.1	在Windows NT 下执行DOS应用程序	49
3.2	未知情况下的设计	49
3.2.1	PainText 程序	50
3.3	处理WM_PAINT 消息	50
3.3.1	使窗口区域无效	51
3.3.2	响应WM_PAINT 消息	52
3.4	控制图形文本显示	53
3.5	Windows字体制式和度量	54
3.5.1	计算字符大小	56
3.5.2	窗口坐标和边界限制	56
3.6	将文本输出到窗口	57
3.6.1	Textout 函数	58
3.6.2	滚动杆与消息	59
3.6.3	确定并改变窗口大小	63
3.7	小结	64
3.8	PainText 程序清单	64
第四章	从Windows 3.x到Windows NT	70
4.1	Windows 3.x与Windows NT 的语法差异	70
4.1.1	消息及其附属参数	73
4.1.2	应用程序类的有关值	74
4.1.3	子窗口和对话框函数	75
4.1.4	文件操作	77
4.1.5	字符串操作	78
4.1.6	过时的函数：对ANSI/OEM 字符的支持	79
4.1.7	可移植的API	79
4.2	Microsoft与Borland编译器的语法差异	80
4.3	小结	82
第五章	键盘、光标和字符	83

5.1	Windows NT 键盘驱动程序	83
5.1.1	对多国语言的支持	85
5.1.2	键盘事件消息	85
5.2	KeyCodes演示程序	88
5.3	虚拟键盘	90
5.3.1	GetKeyState/GetAsyncKeyState函数	94
5.3.2	响应键盘消息	94
5.3.3	WM_CHAR 字符事件消息	95
5.4	文本输入处理	96
5.4.1	脱字符 (caret) 和光标 (cursor)	96
5.4.2	光标函数	96
5.4.3	定位光标	98
5.4.4	WM_CHAR 消息处理	101
5.4.5	产生事件消息	103
5.5	小结	105
5.6	KeyCodes程序清单	105
5.7	Editor程序清单	110
第六章	在Windows中使用鼠标	116
6.1	GENUS MUSMECHANO 的改进	116
6.2	检测是否已安装鼠标	117
6.3	鼠标游标	117
6.4	鼠标操作和事件	118
6.4.1	Windows中的鼠标事件	118
6.4.2	鼠标事件消息	118
6.5	Mouse1实例: 跟踪鼠标	121
6.6	Mouse2实例: 鼠标游标的形状	122
6.6.1	ChildWndProc 函数中的游标操作	124
6.7	Mouse3实例: 击键测试	126
6.8	小结	126
6.9	Mouse1实例的程序清单	127
6.10	Mouse2实例的程序清单	129
6.11	Mouse3实例的程序清单	134
第七章	子窗口和控制器	140
7.1	子窗口控制器	140
7.2	按钮类型	141
7.2.1	下按按钮形式	142

7.2.2	选择框形式	142
7.2.3	收音机按钮形式	142
7.2.4	特殊的控制器/调节器	143
7.3	按钮操作	144
7.3.1	创建按钮用的CreateWindow 函数	144
7.3.2	按钮控制器的通信：一种双向通道	146
7.3.3	向控制器发消息	147
7.3.4	查询控制器状态	148
7.3.5	按钮（窗口）标签	149
7.4	小结	149
7.5	Button1 程序清单	149
7.6	Button2 程序清单	155

第二部分 Windows NT 应用程序资源

第八章	应用程序资源	164
8.1	资源类型	165
8.1.1	加速键： 键盘资源	165
8.1.2	位图、图标和游标： 图像资源	165
8.1.3	对话框	165
8.1.4	菜单	166
8.1.5	字符串资源	166
8.2	文件和文件类型	166
8.2.1	资源链接	167
8.2.2	动态链接库	167
8.2.3	头文件	167
8.3	Resource Workshop 管理程序	167
8.3.1	打开项目文件	168
8.3.2	Resource Workshop 的Preferences对话框	169
8.3.3	增加新的资源元素	169
8.3.4	编辑现有的资源元素	170
8.3.5	Memory选项	171
8.3.6	标识符	171
8.3.7	Edit选项	172
8.3.8	View选项	173
8.3.9	Window 选项	174
8.4	小结	174

第九章 位图、图标、游标和字体	175
9.1 位图编辑器的选择	175
9.1.1 Resource Workshop 的图像编辑器	176
9.1.2 Microsoft的图像编辑器	177
9.2 图标资源	178
9.2.1 Fileview程序中的两个图标	178
9.3 游标资源	178
9.4 定制字体	179
9.5 小结	182
第十章 两种对话框编辑器	183
10.1 Microsoft SDK对话框编辑器	183
10.2 Borland Resource Workshop 对话框编辑器	184
10.3 对话框风格	184
10.3.1 对话框(窗口)类型	186
10.3.2 边框风格	186
10.3.3 Dialog Style选项	187
10.3.4 其它的风格选项	187
10.4 对话框控制器元素	188
10.4.1 下按按钮	188
10.4.2 收音机按钮	189
10.4.3 选择框	189
10.5 面向文本的域	189
10.5.1 编辑输入域	190
10.5.2 静态文本域	190
10.5.3 列表框	191
10.5.4 组合框	191
10.6 其它对话框控制器	192
10.7 Resource Workshop 工具操作器	193
10.8 控制器的对齐、定位和确定大小	195
10.9 FileView应用程序中的三个对话框实例	196
10.9.1 About对话框	197
10.9.2 File Type对话框	197
10.9.3 File Selection对话框	198
10.10 小结	198
第十一章 创建菜单资源	200

11.1	Resource Workshop 菜单编辑器	200
11.1.1	菜单编辑器中的菜单	202
11.1.2	菜单大小和限制	203
11.1.3	定制菜单项	203
11.1.4	设置菜单项ID	204
11.1.5	设置菜单项类型	204
11.1.6	菜单中断符	204
11.1.7	Initial state 对话框选项	204
11.2	菜单描述文件	205
11.3	小结	207
第十二章	加速键、字符串和头文件	208
12.1	加速键资源	208
12.1.1	定义加速键组合	209
12.1.2	定义加速键描述文件	209
12.1.3	Resource Workshop 加速键编辑器	210
12.2	编辑字符串资源	211
12.2.1	定义字符串	211
12.3	构造字符串表	211
12.3.1	Resource Workshop 的字符串表编辑器	212
12.4	头文件	213
12.4.1	Identifiers对话框中的其它选项	215
12.5	小结	216
第十三章	FileView应用程序——集中演示所有的资源类型	217
13.1	在WinMain 中装载资源	217
13.1.1	变量的初始化	218
13.1.2	载入键盘加速键	219
13.2	建立对话框	219
13.2.1	初始化对话框	221
13.2.2	读对话框内容	222
13.3	小结	224
13.4	FileView程序清单	224
第十四章	消息对话框	234
14.1	实例一： 消息对话框	234
14.1.1	消息框图标	235
14.1.2	消息框按钮	235

14.1.3 消息框形态选择	236
14.1.4 其它标志	237
14.1.5 消息框返回的消息	237
14.2 实例二: Borland的可定制控制按钮	237
14.3 装载BWCC库	239
14.4 小结	240
14.5 MsgBoxes 程序清单	240
14.6 MsgBox-2 程序清单	245

第三部分 Windows NT 图形功能

第十五章 Windows NT 的图形设备界面 (GDI)	254
15.1 访问设备场境	255
15.1.1 PAINTSTRUCT 结构	255
15.1.2 其它设备场境访问途径	256
15.2 获取信息场境	256
15.2.1 设备场境信息	257
15.3 映射方式	261
15.3.1 设置和获取映射方式	263
15.3.2 坐标变换: 视区与窗口	265
15.3.3 关于映射方式的演示实例	267
15.4 小结	268
15.5 Display程序清单	268
15.6 Modes程序清单	277
第十六章 颜色、调色板和绘图模式	287
16.1 确定彩色性能	288
16.1.1 Windows颜色与设备颜色	289
16.1.2 颜色定义	289
16.1.3 搅合色	290
16.2 定制刷子	291
16.2.1 UnrealizeObject函数	292
16.2.2 消除刷子或其它对象	293
16.3 颜色和绘图模式	293
16.4 小结	295
16.5 Color1程序清单	295
16.6 Color2程序清单	298

16.7	PenDraw程序清单	303
第十七章	绘图工具	309
17.1	图形的形状和工具	309
17.1.1	标准形状	310
17.1.2	逻辑画笔	310
17.1.3	逻辑刷	311
17.2	建立图形	312
17.2.1	矩形	313
17.2.2	椭圆	313
17.2.3	弧、弦和饼图	314
17.3	商业图形	316
17.3.1	BarGraph实例程序	317
17.3.2	PieGraph实例程序	318
17.4	多边形	320
17.4.1	多边形填充方式	320
17.4.2	PenDraw3实例程序	321
17.5	小结	322
17.6	PenDraw2程序清单	322
17.7	BarGraph程序清单	330
17.8	PieGraph程序清单	335
17.9	PenDraw3程序清单	339
第十八章	刷子、位图、DIB和BitBlt	344
18.1	位图刷：类型 I	344
18.2	位图刷：类型 II	345
18.2.1	PenDraw4实例程序	347
18.3	预定义的位图	347
18.3.1	旧式的位图	348
18.3.2	SetBitmapBits和GetBitmapBits函数	349
18.3.3	单色位图	350
18.3.4	彩色位图	351
18.4	与设备无关的位图 (DIB)	352
18.4.1	DIB头	352
18.4.2	BITMAPINFOHEADER 结构	352
18.4.3	DIB位图的颜色表	353
18.4.4	DIB位图图像	354
18.4.5	OS/2位图	355

18.4.6	有关位图尺寸的函数	356
18.5	建立和使用DIB位图	356
18.5.1	第一步: 提供全局的实例句柄	357
18.5.2	第二步: 定义DrawBitmap函数	357
18.5.3	第三步: 建立设备场境	358
18.5.4	第四步: 位图数据和坐标映射	359
18.5.5	第五步: 图像数据的传送	359
18.5.6	第六步: 清理工作	360
18.5.7	StretchBlt 函数	360
18.5.8	PenDraw5: DrawBitmap操作	362
18.6	小结	363
18.7	PenDraw4程序清单	363
18.8	PenDraw5程序清单	367
第十九章	字型和风格	378
19.1	文本输出特征	378
19.1.1	SetTextAlign 函数	378
19.1.2	ExtTextOut 函数	380
19.1.3	TabbedTextOut 函数	381
19.1.4	GrayString 函数	381
19.1.5	DrawText 函数	382
19.2	设备场境更改	383
19.3	字体和字型	384
19.3.1	字型简史	385
19.3.2	图形字体	386
19.3.3	使用逻辑字体进行字符字体选择	388
19.3.4	Fonts 演示程序	393
19.4	小结	397
19.5	Fonts程序清单	398

第四部分 应用程序之间的信息交换

第二十章	元文件操作	411
20.1	记录元文件	412
20.2	重放元文件	414
20.3	以磁盘文件形式存贮元文件	415
20.3.1	生成临时文件	416

20.3.2	删除临时文件	417
20.3.3	访问临时元文件	418
20.4	元文件的结构	418
20.5	关于元文件的注意事项	421
20.5.1	存贮和恢复设备场境	422
20.6	小结	423
20.7	PenDraw6程序清单	423
第二十一章	裁剪板数据传递方法	428
21.1	裁剪板观察程序与真正的裁剪板	428
21.2	裁剪板操作	429
21.2.1	缺陷与限制	429
21.3	裁剪板数据格式	430
21.3.1	特殊用途格式	431
21.4	访问裁剪板	432
21.4.1	检查数据项目的可用性	434
21.4.2	提取裁剪板中数据	435
21.5	ClipBd演示程序： 裁剪板操作实例	435
21.5.1	用裁剪板传送文本	436
21.5.2	用裁剪板传送位图	437
21.5.3	用裁剪板传送元文件	439
21.6	其它裁剪板格式	443
21.6.1	私有的裁剪格式	443
21.6.2	延迟提交	444
21.6.3	特殊环境下的消息	444
21.6.4	由所有者负责显示的裁剪板数据	445
21.6.5	用户定义的专有格式	446
21.7	小结	446
21.8	ClipBd程序清单	447
第二十二章	动态数据交换 (DDE)	457
22.1	动态数据交换入门	457
22.1.1	DDE的术语和特点	458
22.1.2	DDE基本概念	459
22.2	DDE管理库	459
22.3	DDE_Demo应用程序	460
22.3.1	DDE应用程序的初始化： WinMain 函数	461
22.3.2	DdeCallback回调函数	464

22.3.3 WndProc 函数中的DDE元素	466
22.3.4 PostTransaction 函数	469
22.4 小结	470
22.5 DDE_Demo程序清单	470
附录: 程序源代码文件	480

序 言

初次接触Windows NT，它既令人感到熟悉，但同时又使人察觉与Windows 3.x有着明显的不同。Windows NT提供着与Windows过去版本同样的操作和特征（这些曾使Windows变得非常流行），并且还能运行已存在的Windows 3.1的应用程序。然而，由于Windows NT是一个真正的操作系统，而不只是运行在DOS的限制下的一个外壳，所以它能够充分发挥最新的CPU80386和80486的长处并使用户获得好处。Windows NT与Windows 3.x两者的差异不仅是外表上的，实际上Windows NT中有着许多重要变化（用户对这些常毫无所见），它们对Windows的操作提供了非常重大的改善。

读者（程序员），在能够为Windows NT重新编写自己的应用程序或设计新的应用程序之前，首先必须了解Windows NT这一新的操作系统是如何工作的，以及它提供着什么样的工具和选择。这是本书第一部分所涉及的主要内容。第一章介绍Windows NT的概貌，以及为创建Windows NT应用程序所需要的编译器和工具。

在第二章中，我们将看到Windows NT与Windows 3.1在应用程序源码上的一些差异，其中包括结构、类型、类和消息方面的不同。随后，为了读者学习之便，本章还给出了一个样板应用程序，它将用于本书以后的演示程序中，并以Template.I为名的include文件的形式出现。

在第三章中，内容转向如何显示信息的基本技巧。首先是有关文本信息的显示，其中也包括了字体和字体制式的介绍。

第四章的主题是介绍Windows 3.x和Windows NT的差异，主要以早期Windows版本的应用程序转变成新系统（Windows NT）的形式来反映。虽然，某些差异从外观上来看是微小的，甚至只是风格上的；但是，在这两个系统之间以及为两者所编写的应用程序的源代码方面仍然存在明显不同，源代码的不一致必将导致应用程序的运行有本质上的差别。所以，在为Windows NT开发新的应用程序和转换老版本的应用程序之前，必须掌握一些基本知识。此外，本章还将考察一下对于Windows NT版本，Microsoft C编译器与Borland C编译器之间的差异。

在第五章中，主题又回向Windows NT的实际编程。主要介绍键盘输入以及虚拟键码。在此，将看到一个简单的编辑器程序，用以展示虚拟键码是如何产生和被解释的。

当然，对于Windows（任何版本）这样的图形系统，鼠标与键盘一样重要，甚至比键盘更为重要。所以，在第六章将介绍鼠标的移动消息，游标光标以及其它键盘消息，同时还附有一些演示程序以阐明这些消息是怎样起作用的。

在第七章中，将通过介绍子窗口控制器和控制器按钮，把主题转向图形控制器

单元。

本书的第二部分介绍应用程序的资源以及创建资源的资源编辑器。在每一章内，将展示如何创建各种资源，这些资源将用于第十三章的FileView应用程序中。

第八章将考察资源的类型和资源的文件，并将介绍与Borland的32位编译器一起发行的Resource Workshop。

第九章介绍位图资源，包括位图图像、图标、游标和字体资源。同时，还将考察位图编辑器。

进入第十章，主题转换成资源对话框，其中包含了Microsoft SDK和Borland Resource Workshop的对话框编辑器。本章内容还涉及到对话框的风格、控制、大小、定位对话框元素以及对话框创建工具。

第十一章介绍菜单资源。其中将展示如何运用Resource Workshop的菜单编辑器或通过直接创建菜单描述文件来生成菜单资源。

第十二章的内容仍围绕着资源问题，其中介绍了加速键、字符串资源和字符串表，并以介绍头文件和资源记忆法作为结束。

在第十三章中，综合了前面章节所讨论的各类资源，编写出一个演示程序FileView。在其中，使用并展示出所有这些资源的特征以及前面其它章节所介绍的知识。该演示程序中还介绍了一些基础性的内容——例如目录和文件操作——这些在前面章节中未作过介绍。

第十四章继续讨论应用程序的资源。其中将介绍一些资源，它们是通过使用现成的Windows消息框和选择Borland的Windows定制控制器，在运行中以消息框的形式创建的。

第三部分介绍Windows的图形操作。首先在第十五章中介绍Windows的设备场境和映射模式。第十六章继续这一主题，介绍颜色和调色板，并展示了主要的ROP绘图模式。

第十七章的内容转向图形工具。其中首先介绍了绘图和填图的基本工具和笔型，在此的绘图、填图对象包含曲线和多边形。随后展示了一系列简单的商业用图。

第十八章中，继续讨论绘图工具，其中介绍了位图刷和填充方式。此外，还考察了预定义的位图图案以及与设备无关的位图(DIB)和位图文件结构。

第十九章，主题回向文本显示。该章将较其前更详尽地介绍文本显示技巧，其中包括了字体选择、对齐方式、字型，以及如何修改设备场境来控制文本的显示。

第四部分讨论应用程序之间的数据交换的种种方法。第二十章介绍具有元文件操作形式的另一种绘图操作方式，它允许一绘图操作被记录下来，并被当前应用程序所重显；或者使用载剪板的特性（在第二十一章中讨论）传递给其它应用程序，由它们加以重显。

载剪板的数据传递是第二十一章的主题。首先，将载剪板观察者与真实的载剪板（它是Windows内一个可共享的资源）作了比较。其后，还介绍如何动用载剪板，载剪板的标准数据格式和定制数据格式。

最后，在第二十二章中，讨论了一个更复杂但更有效的数据传递方式——动态