Microsoft

Win32 应用程序设计接口

——参考手册（上卷）

［美］ Microsoft 公司

彭木昌 王秀清 胡克雅 许博义 译

郑 方 钟向群 文 极 校

清 华 大 学 出 版 社

目　录

（上卷）

目　录

（上卷）

第 0 章

概　述

0.1　Win32 API 概述

自 1985 年，Microsoft Windows 图形环境首次公布以来，它已成为领导潮流的个人计算机图形系统。1990 年 5 月发行的 Microsoft Windows 3.0 版本，通过在保护模式下运行应用程序，使得开发更复杂的应用程序成为可能，从而打破了 Microsoft MS—DOS 操作系统 640K 内存的束缚，起到了里程碑的作用。这一技术上的突破引来了大量的应用程序，对 Windows 环境在市场上取得巨大成功起到了决定性的作用。从下表给出的应用程序的销售数量上不难看出其优越性（见图 1）。

	12/91 Installed base	Forecast[1] annual run rate 1992	YTD 1991[2] application volume
Windows 3.0	7.9M	9.2M	$711M
MS-DOS	96.0M	24.3M	$2，148M
Macintosh	6.5M	2.2M	$457M
PC UNIX	1.0M	.4M	n/a
OS/2	1.2M	.7M	$29M

图 1　市场分析

来源：1）IDC，1991 年 10 月；

2）Software Publishing Assoc.，1991 年 9 月

在 1990 年 5 月至 1991 年 10 月期间，全世界有 700 万以上的个人计算机用户拥有 3.0 版 Windows 软件的许可证。据 International Data Corporation（国际数据公司）估计，在 1992 年间还会增加 920 万用户。另外，又有 7 万多 Microsoft Windows 3.0 版本的软件开发工具包走向市场。这清楚地表明在接下来的 12 到 18 个月中，有可能出现大量的应用程序。截止到 1991 年秋天，市场上已有 4 900 多个 Windows 的应用程序。

基于这些成就和众多独立的软件开发者所取得的成功，Microsoft 正在延伸和扩展 Windows 环境，以便使 Windows 的应用程序能在更大范围的计算平台上运行，即从电池供电的便携机到高级 RISC 工作站以及多处理器服务器等。

我们正在扩展 Windows，使它成为真正 32 位，并添加另外的操作系统服务。用笔计算的 Microsoft Windows（Micorsoft Windows for Pen Computing）和带有多媒体扩展的 Microsoft Windows（Microsoft Windows with Multimedia Extensions）还将充分利用新的硬件技术的优点。

Windows 的现状

现在，许多人认为 Windows 只是在他们熟知的已使用多年的 MS—DOS 操作系统之

上增加了图形功能。这种观念给最终用户对 Windows 的升级带来了恐惧感。然而事实上，Windows 并不受 MS－DOS 的限制。

Windows 是一个完整的操作系统，它代替了 MS－DOS 的某些特征并具备领先于 MS－DOS 的许多优点。Windows 3.0 版本不使用 MS－DOS 屏幕或键盘 I/O，也不使用 MS－DOS 存储器管理，甚至用新的 Windows 专用设备驱动程序以取代 MS－DOS 的文件 I/O。Windows 3.0 版本增强模式能处理 32 位的设备驱动程序而不会受相形见绌的 640K 的 MS－DOS 的束缚。这些驱动程序通过 Windows 可以与那些同样不受限于 MS－DOS 的应用程序之间对话。

能与 MS－DOS 一起工作的优点就在于这可以维护由 MS－DOS 的长生命周期（用“计算机年”表示）所带来的成果。Windows 能同 MS－DOS TSR、MS－DOS 设备驱动程序一起工作。当然，它也可以运行 MS－DOS 应用程序。Windows 的以后版本将可继续在 MS－DOS 上使用。

Windows 的结构

自 1981 年 IBM PC 问世以来，个人计算机已经在性能和配置上变得越来越多样化。这种多样化将在接下来的几年中随着基于 RISC 处理器和多处理器系统的个人计算机的问世而进一步加剧。

这些不同的系统对操作系统自然会有不同的要求。例如，一个电池供电的便携机既需要小体积的内存和硬盘以减轻重量、减少费用，还需要电池管理功能以延长电池寿命。相反，网络服务器和特定任务的台式机（mission－critical desktop）需要完善的安全性以保证数据的完整性。而基于 RISC 的系统需要操作系统和应用程序两方面的可移植性。

一些销售商觉得不同领域的硬件需要完全不同的、带有互不兼容的应用程序的操作系统。他们售出不同的操作系统分别用于个人计算机、工作站、服务器，以及未来的基于笔的系统。其每个操作系统需要单独的、互不兼容的应用程序。这些不同平台之间的连接具有一定的复杂性。

Microsoft 努力寻求一种简单得多的解决方法。我们正发展 Windows，使之实现多重形和全兼容性。针对不同的硬件，将优化选择以实现不同的 Windows。用户在 Windows 开发和 Windows 应用程序上的投资将得到保护。Windows 应用程序将可在各种硬件上运行，包括从笔记本式的笔系统，到特定任务的台式机系统，再到多处理器和基于 RISC 的系统。

Microsoft Windows 正在发展成为一个完整的操作系统结构，它通过支持不同的操作模式来满足多种要求。现在，Windows 有三种模式：实模式、标准模式和增强模式。实模式提供与原先的 Microsoft Windows 版本的兼容；标准模式最适用于 80286 处理器并可访问该芯片所支持的所有 16MB 的存储器；增强模式通过支持多重的 MS－DOS 应用程序，并通过一种称作页面调度的技术，来利用 80386 和 80486 处理器的优点，提供了对机器中存在的物理地址以外的存储空间进行访问的应用程序。这三种模式同时支持 MS－DOS 和 Windows 应用程序。

在 Windows 3.0 版本成功的基础上，Microsoft 在 1992 年初公布 3.1 版本。3.1 版本

采纳了用户的反馈意见。它包括 TrueType，这是一种先进的可缩放字体技术，它改进了性能，引入了一种新近设计的文件管理程序，提高了网络的连接性，改进了系统的可靠性。3.1 版本支持 Windows 标准模式和增强模式。

同样在 1991 年间，Microsoft 通过提供对声音、动画和访问 CD－ROM 的扩展，即所谓的“Windows with Multimeda Extensions”来改进 Windows 标准模式和增强模式。同时，我们还将公布一种针对于剪贴板和笔式计算（clipboard and pen－style computing）的操作环境，即所谓的“Microsoft Windows for Pen Computing”。

Windows NT

1992 年，Microsoft 准备推出一种称为 Windows NT（New Technology）的新产品。Windows NT 建立在 32 位操作系统内核之上。Windows NT 将提供健壮性强的用户环境，该环境针对特定任务的应用程序、高级台式机平台和一可移植、可扩充的服务器环境（见图 2）。Windows NT 还将把 Windows 转化成一个 Microsoft LAN Manager 服务器平台，这样就在现有的由 LAN Manager 所支持的 OS/2、UNIX 和 VMS 的平台上增加了第四个服务器平台。

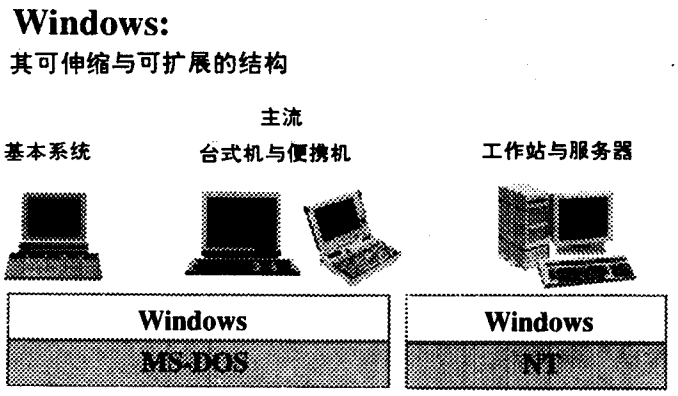

图 2 单用户界面与单道程序设计模型

Windows NT 不需要 MS－DOS 的功能。但是，它与大面积安装的 MS－DOS 和 Windows 应用程序兼容。除了与这些现有的应用程序兼容以外，Windows NT 还含有一些 90 年代及以后的高级台式机和服务器市场所要求的特征。

为支持大量的服务器应用程序，Windows NT 支持对称的多处理器，该处理器在基于线程级的处理器上对称地分布任务。这项设计最大程度地优选了多处理器系统中的每一个处理器并简化了多处理器应用程序的开发。

网络服务程序以及许多重要任务的应用程序需要安全性。为满足这一需求，Windows NT 已被设计成一个可靠的操作系统。Microsoft 与美国联邦政府一起，保证 Windows NT 具有“C2 级”安全性。而且今后的 Windows NT 的内部设计可提高到“B 级”安全性。

Windows NT 是 1991 年 4 月由 Microsoft、Compaq Computer Corp、Digital Equipment Corp、MIPS Computer Systems、Santa Cruz Operation 等宣布的高级计算环境 ACE 的关键组成部分。现在它已包括了 200 多个成员。

ACE 的第一步目标是要为基于微处理器的系统在其升级的时候提供一个开放性的、基于标准化的高级计算环境。ACE 第一步完全支持两个平台：基于 386/486 的 PC 和基于 MIPS RISC 的系统。作为跨越这两个环境的可移植的操作系统，Windows NT 是 ACE 标准的决定性因素。有了 Windows NT，现有的 MS－DOS 和 Windows 程序不加修改就可以在基于 MIPS 的计算机上运行。

作为对这些先进性的补充，Windows NT 基于内核的设计可以看作是与不同的操作系统环境兼容的核心。这一内核的设计提供了 Windows NT 与 MS－DOS 和 Windows 应用程序的兼容性，也为 Windows NT 奠定了基石，使其可以支持 Microsoft 开发的将作为工具包的 OS/2 和 POSIX 应用程序接口。同时，这项设计还允许 Windows NT 支持用新的 Win32 应用程序设计接口编写的应用程序。

Win32 API

迄今为止，开发者和最终用户已在 Windows 程序设计和 Windows 应用程序上进行了大量的投资。所开发的大多数应用程序都能在 16 位 80286 处理器和 32 位 80386 和 80486 系统中运行。由 Windows 3.0 支持的 16 位 API 编程，尽管性能很好，但由于其固有的存储器 16 位结构的限制，编码必须被分成不超过 64K（65536 字节）的段。这使得编程更加困难，而且 80486 和基于 RISC 的系统的高性能不能得以充分发挥。

Win32 API（Win32）的设计使得 Windows 16 位 API（Windows 16）向 32 位的过渡尽可能容易。Win32 API 在语法上只有很小的变化，API 的命名与 Windows 16 相同，语义也相同，消息序号也相同。事实上，完全可以保存独立的源码，它既可被编译成 16 位程序，也可被编译成 32 位程序。必要的变动将在后面的“32 位 Windows 操作系统可移植程序设计的考虑”中详述。

尽管 Win32 API 与 Windows 16 API 极其兼容，但它还具备了一些有效的新特征。这包括使用独立的寻址空间的抢先多任务进程（preemptively multitasked process）、抢先线程（preemptively thread）、信号灯（semaphore）、共享存储器（shared memory、有名管道（named pipe）、邮箱槽（mailslot）和存储器映射文件 I/O（memory－mapped file I/O）。图形设备接口（GDI）中改进了 Bezier 曲线（Bezier curve，译作“贝泽曲线”，一般均不译——译者注。）、路径（path）和变换（transform）。

MS－DOS Windows 和 Windows NT 都支持 Win32 API。1992 年期间，Win32 API 将首先从 Windows NT 产品中获得。1993 年，将把它加到 MS－DOS Windows 上。为 Win32 API 编写的程序的二进制码既可在 Windows NT 上也可在 MS－DOS Windows 上运行。MS－DOS Windows 和 Windows NT 支持 Win32 的所有特征，包括抢先多任务。Win32 程序将在 x86 和 MIPS 处理器间达到完全的源码级兼容。1992 年上半年将发行 Win32 API 软件开发工具包。

另外，Microsoft Languages 正在开发一个 Windows 的扩展产品，它在 Windows 3.1

上实现了 Win32 API 的一个兼容子集。这项产品对所有 Windows 3.1 特征提供了一个 32 位程序设计环境，但不包括在全部 Win32 API 中出现的高级功能，如抢先多任务（preemptive multitasking）。针对这一子集所写的程序将不必改动就可作为 Win32 应用程序在 Windows NT 和 MS—DOS Windows 上运行。有关该产品其他的信息将在 1992 年上半年公布。

下面着重叙述 Win32 API 的一些关键特征。

内核：基本操作系统

Windows NT 和 MS—DOS Windows 上的 Win32 API 均提供抢先的、基于线程的多任务。它还能在独立的寻址空间内运行所有的 Win32 和 MS—DOS 应用程序，以使得这些程序间不会互相干扰。

Win32 API 的可移植性设计超出了 80386、80486 处理器，特别是它可移植到 RISC 结构上。所有这些处理器具有不同的特征，但有同样的 32 位寻址和页面虚拟存储区结构。页面虚拟存储较段式虚拟存储系统在实现和运行方面更加有效。Win32 中的存储管理是安全的，这是因为操作系统在不同的存储页面上放置不同的存储对象，并允许一个应用程序来控制对存储对象的访问权（读、写、读/写、执行）。

有了很大的 32 位寻址空间，操作系统可以方便、有效地优化 I/O 文件。这是因为进程视该文件为非常大的存储对象并随机地访问它。操作系统通过页面故障（page faulting）能检测文件的读取权并将其数据读入，也能检测何时一个共享文件被写然后写出其数据。有了可配置进程的访问权限和物理存储页的稀疏分配，即使访问模式完全不可预测，进程仍能非常有效地实现数据存取。

GDI 的改进：Bezier 曲线、路径和变换

作为 Windows 3.0 和 3.1 版本的绘画 API 的 GDI 为应用程序提供一个有用的，与设备无关的绘图集。随着输出设备日趋复杂，绘图需要也更加复杂化。因而，GDI 也需做进一步的改进。

某些 3.0 和 3.1 版本的 Windows 应用程序需要使用 Windows 环境下的早期低级的绘图原语来实现高级图形功能。尽管这种能力已使应用程序销售商们在拓展 Windows GDI 方面有了一定的灵活性，但仍不能使他们充分地利用打印机和显示技术，应用程序开发者不得不为自己的图形显示（诸如 Bezier 曲线和路径）编写自己的算法。有了 Win32 API，开发者可以调用新的高级图形特征，这些新特征充分利用了先进的输出硬件中内部固有的绘图功能。在 Win32 下，显示 Bezier 曲线可以由制图设施或输出设备来处理，而在这些设施和设备中已实现了 Bezier 优化。

Win32 GDI 是一套完整的通用绘图包。Bezier 曲线是一组通用曲线，从中还可以得到一条直线。该函数通过与 PolyBézier（玻利—贝泽）功能的结合，使得绘出任意连续直线和曲线的组合成为可能。

Win32 增加了一个 Path API，这使得应用程序能够容易、有效地管理复杂的形状，这些复杂的形状可以包括任意直线、弧、椭圆和 Bezier 曲线的组合。一条路径可以通过调

用 BeginPath 函数开始，接下来调用绘画图元以定义该路径的形状和大小。调用 EndPath 函数可以结束这条路径。这样应用程序就可以通过绘图、剪贴、填充和变换这些定义过的形状。

Win32 Transform API 可以把用户绘制的虚拟二维表面映射到二维输出表面上。这一 API 与首先在 Windows 3.1 版本中实现的 TrueType 字体技术相组合，使其能绘出真正的与设备无关的图形，而系统能把该图形映射到显示表面上（包括位图和字体的旋转以及图元文件）。

开窗系统与系统类

Windows 开窗系统（windowing system）最显著的改进是刻意使得每个窗口消息队列与系统消息队列之间“去同步”(desynchronization)。这一改进防止了循环的应用程序所发生的错误，即在这些程序停止处理自己的消息时，会阻塞计算机系统的整个用户界面，从而导致其他应用程序无法运行。

从最终用户的角度看，“去同步”意味着当一个应用程序忙碌时，它们还可以做其他事情。例如，如果一个字处理程序正在忙于打印一份 100 页的文件时，用户可以激活另一个应用程序窗口或使用任务管理程序来开始另一个应用程序的工作。这样就有效地减少了用户等待屏幕空闲的时间。

消息队列的“去同步”与 Windows 3.0 和 3.1 版本的消息模型完全兼容。消息排列顺序是一样的。如果 WM_xyz 位于 WM_abc 之后，则它仍位于其后。这一兼容性是必要的，因为在 Win32 系统中，现有的 Windows 应用程序是在 Win32 消息系统上运行的。由于消息只是简单地从 32 位栈拷贝到 16 位栈然后送到应用程序中，所以，消息顺序不能改变。

网络的扩展

API 在某特定领域的每一次新的标准化，都会使编写新的应用程序变得更容易一些。由于网络层次的多样性，从网络接口卡和协议栈，到大量的网络进程间通信（IPC）机制，对当今的开发者来说，网络大概是最棘手的界面了。Win32 包括标准网络 API，它可以代替那些网络提供者原先不得不提供的 API。Win32 将提供与 Windows 3.0 版本所提供的 WinNet API 相仿的驱动型（driver—style）界面，这样第三方销售商就可以把他们的网络服务器接到 Windows 开放式的结构中去。

新的 32 位 WinNet API 定义了文件、打印、有名管道、邮箱槽、服务器浏览和机器配置。这就是说，应用程序能依赖于一致的程序设计界面而与基本网络无关。即使没有网络，API 也能正常使用且能返回恰当的错误代码。

Win32 API 包括点对点间（peer—to—peer）的有名管道、邮箱槽和 API，从而以允许远程过程调用（RPC）编译程序。通过 Win32，邮箱服务器销售商能在有名管道和异步通信的基础上建立起一个消息服务系统，而这些有名管道和异步通信是可以在来自不同网络销售商的网络操作系统、协议栈或网卡上运行的。

与 Windows 16 位 API 的兼容性

Windows 3.0 和 3.1 版本应用程序能在 MS－DOS Windows 和 Windows NT 上运行。为了保证与版本 3.0 和 3.1 兼容，所有的 Windows 16 位应用程序将在一个寻址空间内作为一个进程运行。它们相互间无优先可言，但是可优先于系统的其他进程，这反映了 Windows 3.0 版本增强模式下的特性。Windows 16 位应用程序会遇到 Win32 API 而没有“中间层”变换，也没有任何状态映射或消息重排。

Windows 的可执行程序也可在基于 RISC 的 Windows NT 机器上运行（见图 3）。这一平台需要卓越的性能，因为尽管一些编码会遇到 80286 仿真技术，但所有的 Windows 调用将被直接映射到 Windows NT 软件上。

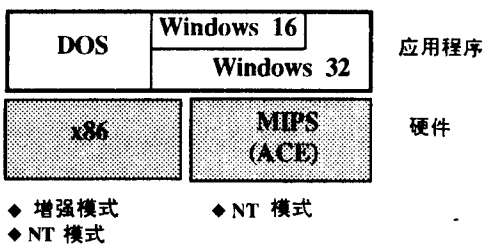

图 3 Windows 平台

Win32 API：Windows 的未来

现在，数百万计的人们都热衷于使用 Windows 3.0 版本。公司和独立的软件销售商们是 Windows 的主要支持者也是 Windows 应用程序的主要投资者。

为保护这项投资，Microsoft 公司正将 Windows 发展成完整的结构。通过不同的实现方法，Microsoft Windows 将广泛地运行于不同类型的硬件上，包括从基于笔的笔记本式微机到多处理器和 RISC 系统。

Windows NT 和 MS－DOS Windows 未来的版本将支持 Win32 API。这些 API 的设计简化了 Windows 应用程序从 16 位到 32 位的过渡，而且也使得开发新的 Win32 应用程序变得简单。它们具有许多优越的新特征，可以用来产生新一代强有力的 Windows 应用程序。

另外，在 Microsoft 的开发下，Win32 API 将被用作 Windows 未来版本的基石。这项常被称作“信息随手可得”的技术，将使得使用个人计算机更容易并可以为 Windows 用户提供优越的新功能。

0.2 Win32 操作系统可移植程序设计的考虑

摘要

预先颁布的 Microsoft Win32 软件开发工具包（SDK）将很快就可以使用了。在 Win-

dows 3.1 的β测试期间，开发者可以更改应用程序源码，作相应的改动使之成为健壮的 Windows 3.1 应用程序，同时也就可以准备将这种应用程序转换到 Win32 API 所提供的全 32 位环境中去。这里的内容不是介绍怎样编制 32 位源码，而是强调一些差异，这些差异有益于修改现有的 Windows 3.1 和以后的 Win32 应用程序源码。

Microsoft Win32 API 的目标

创设 Win32 应用程序设计接口（Win32 API）的主要目标是以下 6 个方面：

1. 为现有的 Windows 应用程序提供一条向 32 位过渡的途径；
2. 使 Windows 应用程序移植到 Win32 时尽可能容易；
3. 产生有效的映射层，以便在 Win32 系统上运行 Windows 3.x 二进制文件；
4. 支持单一的源码库以产生 Windows 3.x 和 Win32 二进制文件；
5. 为 Windows NT 和 MS—DOS Windows 的未来版本提供一致的 Win32 API；
6. 在简单地拓展现有的 Windows API 以保持兼容的同时，为先进的操作系统特征，诸如抢先多任务、IPC 机制、复杂的存储器管理和图形功能等，增加新的 API。

为达到上述目标，Microsoft 从现有的 Windows 3.1 API 中开发出了 Win32 API，它不允许数据类型、功能和结构任意更名。乍看起来，无论是从最终用户的角度还是对源码进行迅速检查的角度出发，一个 Win32 应用程序与现有的 Windows 3.0 或 3.1（以后记作 Windows 3.x）应用程序会难以区分的。而与 Windows 3.x API 的类似程序不同，一个真正的 Win32 应用程序，能更充分地利用大的线性内存分配、后台任务与计算的多线程特性，并借助于有名管道充分利用近程与远程的 IPC 机制，同时其他在§0.1“Win32 API 概述”中所详细介绍的众多特性，也可以更充分地得以利用。

Win32 API 首先产生于单处理机和多处理机 386 和 486 系统以及新的基于 RISC 的系统的 Windows NT 中。未来的 MS—DOS Windows 版本还将支持 Win32 API。Windows NT 和 MS—DOS Windows 的未来版本，包括线性寻址空间、线程和抢先多任务，都支持 Win32 的全部特征。在 MS—DOS Windows 或 Windows NT 上运行的 Win32 应用程序将与 Intel 386/486 处理器在二进制级兼容，与 RISC 处理器上的 Windows NT 在源码级兼容。

本部分集中论述 Windows 应用程序移植性的两个方面：

1. 开发者可采取的步骤。工作于 Windows 3.1 应用程序的开发人员，最好是使这些应用程序可与 Windows NT 达到二进制兼容。
2. 开发者可利用的创建 Windows 代码的技术。这种代码可移植性更强，并且到可以利用 Win32 软件开发工具包的时候创建应用程序的 Win32 版本会更容易。

二进制兼容性

Win32 系统可以运行现有的 Windows 3.x 应用程序，且带有通过动态数据交换（DDE）的相互操作性、对象连接与嵌入（OLE）、图元文件，以及其他 Windows 3.x 应用程序和真正的 Win32 应用程序的剪贴板。Windows 3.x 应用程序和 Win32 应用程序将在同一显示器上紧挨着出现而不是在分开的屏幕上运行。如果开发人员遵循以下规则，

Windows 3. x 应用程序将与 Windows NT 完全兼容：

- 保证 Windows 3. x 应用程序运行于标准/增强模式下；
- 使用公开发行的 Windows 3. x API、消息和结构；
- 不要直接修改 WIN. INI，使用一个简表串（profile string）API（如 WriteProfileString）；
- 使用 QUERYESCSUPPORT 以确定是否已实现了特定打印机驱动程序的换码序列。

在 Windows NT 上运行 Windows 3. x 二进制程序的能力不仅仅限于 386 和 486 系统，还可在基于 RISC 的 Windows NT 系统上运行。这是通过高性能的 PC 仿真程序和同样有效的映射层（mapping layer）技术来实现的。这些技术可用于“无缝地”集成那些运行于 386 和 486 系统中的 Windows NT 上的 Windows 3. x 应用程序。

设计需求

过去通过映射层技术来允许 Windows 应用程序在 OS/2 上运行。为了使基于 Windows 应用程序可以在 OS/2 上运行，以往的解决方法例如 OS/2 的 Windows 库（WLO）等均需要特殊的运行库和 DLL，这样 ISV 必须将 WLO 映射层 DLL 同其应用程序一道发行，以至于产品的配置和安装复杂化了。该方法在 Win32 系统下是不允许的。

Windows 3. x 版本应用程序开发人员不必为在 Win32 系统上获得二进制兼容性和卓越的性能而重新编译源码，使用特殊的运行库、开发方法或用特殊的工具，就可以生成兼容的可执行文件。

这种能够运行 Windows 3. x 二进制程序的能力使得最终用户可以升级到 Win32 系统并继续使用现有的 Windows 3. x 应用程序，如同使用本机 Win32 应用程序一样。这就保护了在现有 Windows 3. x 应用程序上的投资，并可以让用户在 Win32 应用程序公布后把 Windows 3. x 应用程序升级到 Win32 应用程序。本机 Win32 应用程序将受益于高性能的线性 32 位寻址和大量增加的数据处理。

Microsoft 公司鼓励 Windows 开发人员在所颁布版本上通过 Windows NT 的 β 测试程序来验证他们的一系列 Windows 3. x 产品，以保证二进制兼容性的彻底性与有效性。

Win32 所支持的特征

下面列出了 Win32 系统所支持的许多 Windows 3. x 特征。这可以说明现有的 Windows 应用程序可与未来的 Win32 系统二进制级兼容，而无需开发人员多做什么工作。它还说明 Win32 系统完全支持复杂窗口设计、图形显示和 Windows 3. x 二进制码的对低级操作系统的依赖性。

完全支持（无需改动）的主要用户界面特征包括下面这几个例子：

- 多文档界面消息和缺省的消息处理
- 资源文件（例如对话框、菜单、加速键表和用户定义的资源）
- DDE 消息和 DDE 管理库（DDEML）API
- 与 Windows 兼容的 OLE

· 图元文件
· 剪贴板数据交换

完全支持的主要图形界面特征包括：

· TrueType 与 TrueType API
· 用现有格式表示的 Windows 3. x 图标与光标
· 位图（BMP）及设备无关位图（DIB）
· 通过本机 Win32 打印机驱动程序进行打印

基本系统功能包括以下各项：

· IPC 的共享内存
· 支持 MS—DOS 有名管道的 NetBIOS 和 Microsoft LAN Manager
· MS—DOS 5. 0 界面（用 DOS3Call 或 INT21 调用）

实现二进制兼容的方法

Win32 API 将使用一个注册数据库（registration database）来维护所有的系统和应用程序配置信息。诸如 WIN. INI 之类的文件将不再存在于文件系统中，而代之以把调用简表 API（profile API，例如 GetProfileString）放入数据库中。所以，应用程序不应该直接通过 I/O 文件方式去创建或修改 *. INI 文件。Windows 3. x 简表 API 将用于管理所有简表信息。应使用简表 API 或 Windows 3. 1 注册数据库 API 来修改创建私有安装文件的安装程序。

Windows NT 发行了一套与 Windows 3. x 类似的打印机驱动程序。它是通过共享打印机微驱动程序而实现的。然而，为利用诸如 Postscript 打印机之类的设备的高级打印功能，Windows NT 还应包括本机 Win32 打印机驱动程序。Windows 应用程序应该永远在使用任何扩充的打印机驱动程序换码序列之前使用 QUERYESCSUPPORT。应用程序不应该假定某类打印设备（如 LaserJet 或 Postscript）能保证提供特殊的驱动程序，而应通过询问其支持能力以确保不会受其后的 Windows 3. x 或 Win32 打印机驱动程序更新的影响。

应用程序必须与 Windows 3. x 标准或增强模式兼容。Win32 系统不支持 Windows 实模式。应用程序应只使用发行的 Windows 3. x API、消息和结构。

可移植性编码技术

随着 Windows 3. 1 的公布，许多应用程序也随之更新，增多了诸如对 OLE 等特征的支持以及对 TrueType 的利用。由于 Windows 程序设计人员已仔细核对了他们的应用程序源码，所以，现在正是为以后准备程序之契机。未来版本将提供更强有力的新特征的 32 位环境。

现在的讨论主要集中在影响现有的 Windows 源码移植到 Win32 的一些重要问题上。尽管下表看上去似乎冗长、过于详尽，但所有的建议对生成高效的 Windows 3. 1 应用程序非常有用。另外，应用程序将更具可移植性，也更容易在 Win32 软件开发工具包 SDK 发行后，产生本机 Win32 应用程序。