

第一部分 使用多媒体

第1章 用 Visual Basic 开发多媒体

Visual Basic 是微软公司一个功能强大的可视编程开发环境,由于它简单易用,又有利用微软最新操作系统(特别是 Microsoft Windows 95)的能力,因而很快成为计算机工业中开发多媒体应用程序的首选环境之一。

使用 Visual Basic 4.0,可在相同代码基础上开发 16 位和 32 位应用程序,这不但有助于 Windows NT 和 Windows 95 平台上的 32 位应用程序迅速占领市场,同时也有助于开发 Windows 3.x 和 Windows for Workgroups 平台上的应用程序。

多媒体开发要求集成多种功能,根据应用程序的要求,这可能包括:音频、数字视频、视频采集、叠加视频、动画、图象、外设、CD-ROM、以及许多其他的软件和硬件技术。这就要求你的目标平台(操作系统)提供访问和集成这些技术的功能,要求你的开发环境有使用这些功能的能力。Visual Basic 4.0 具有在最新的、功能强大的 Windows 操作系统上运行的能力,同时提供调用像多媒体、普通控件和网络等这些操作系统功能的友好用户界面。

本章着眼于 Microsoft 操作系统和 Visual Basic 4.0 提供的多媒体服务,提供了在应用程序中建立和添加多媒体功能的一个概述。通过本章的学习,你将了解:

- 哪些类型的多媒体元素可用
- Video for Windows 功能
- 高级 Windows 多媒体服务
- Visual Basic 可使用的音频服务
- 如何使用 OLE 和 VBX 控件增加扩展功能
- 多媒体数据库功能

1.1 多媒体元素

对开发者来说,在 Windows 平台上有多种多媒体元素可用,多媒体元素由音频、数字视频、模拟视频、动画、图象和文本组成,这些元素代表了 Visual Basic 程序员可用的六种基本多媒体类型。

1.1.1 音频

当一个多媒体程序涉及到音频时,最典型的类型是波形音频(Waveform audio),其他可用的音频类型有 CD Audio 和 MIDI(Musical Instrument Digital Interface)。

1.1.1.1 波形音频

波形音频是存储于计算机硬盘文件中的电子数字声音。通常情况下,波形音频通过连于PC机上的音频卡将音频源接到音频卡上的音频输入端口,当录音命令发送到音频卡时,音频输入被数字化并存储到计算机的硬盘,图1.1显示了这些连接。

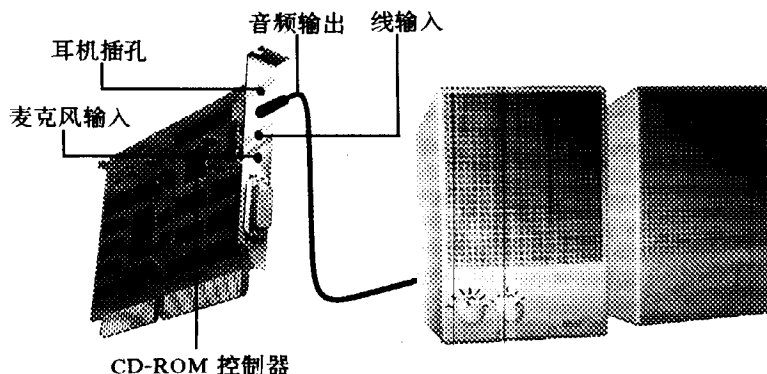


图 1.1 PC 音频卡的典型连接

根据音频卡的功能和应用程序的需要,波形音频可以以不同形式数字化。当音频卡第一次出现在市场上时,是8位声卡。位数表示在音频采集过程中存储的采样点的个数。现在,所有的音频卡制造厂都生产16位音频卡,使得16位音频卡成为最流行和最常见产品。

注: 8位音频卡只能采集8位音频,而16位音频卡可设置为采集8位或16位音频。

数字音频的采样率对用户来说是最重要的变量。采样率指音频采集每秒的采样数,在音频质量和存储空间两种需求之间存在许多权衡因素,对于数字音频,高采样率意味每秒钟采集更多的样点,因而可获得更高质量音频记录,但却需要更大的存储空间。采样率以千赫计算,表1.1列出了当前应用程序中常用的采样率。

表 1.1 常用音频采样率

采样率	描述
11.025khz	常用于语音
22.05khz	提供高质量的音乐和语音混合效果
44.1khz	CD音质的采样,提供最高质量的音乐效果

波形音频可以以单声道或双声道模式数字化。单声道模式等效于调谐(AM)广播,只模拟一个输入声源;双声道模式等效于调频(FM)广播,使用两个音频输入,双声道模式比单声道模式需要更多的存储空间。

1.1.1.2 MIDI

MIDI(乐器数字接口)是一种在计算机文件中存储乐器音调的方法,当播放MIDI文件时,模拟发出该乐器的声音。MIDI文件的播放也可以是通过位于音频卡上的MIDI端口控制诸如电子琴和合成器之类的乐器,这些乐器将接收的音调作为输入。MIDI文件通过内部

或外部的音序器播放出来,音序器解释以音调形式存储于 MIDI 文件的控制事件,并将控制事件传送到内部或外部的合成器。

在 MIDI 文件中没有存储实际的音频采样而是存储乐器事件,因此 MIDI 音频较波形音频需要更小的存储空间和更低的数据传输速率,这使 MIDI 音频成为应用程序背景音乐的理想选择。MIDI 文件只能包含乐器指令事件或音序,而不能包含语音。

在 Windows 平台上使用 MIDI 服务时,消息从 MIDI 文件直接送到 MIDI 映射程序(mapper)。MIDI 映射程序是 Windows 多媒体扩展的一部分,它根据 MIDI 映象设置中的存储值,负责将 MIDI 输出消息送到 MIDI 输出设备;MIDI 映射程序可改变消息的目的通道和输出设备,同时也可更改程序修改的消息、音量值和关键值。

MIDI 映象的设置可由用户定义和由 Windows 控制面板中的 MIDI 映射程序 MIDI mapper 的支程序控制,MIDI 映象设置不能通过应用程序改变。

1.1.1.3 CD Audio

CD Audio 常被称为红皮书音频标准(red book audio),即通过 PC 机上 CD-ROM 驱动器可控制并播放的 CD 音乐。CD-ROM 驱动器不仅允许用户查看如硬盘目录之类的非音乐 CD,而且还可识别红皮书音频标准中定义的 CD 格式(音频 CD),并使 PC 机软件如同家用立体声 CD 播放器一样控制 CD。CD 驱动器的音频输出可连接到音频卡的输入端口,这样,从扩音器听到的音频就如同波形音频一样;也可将 CD-ROM 驱动器的音频输出直接连到扩音器上。通常,多媒体 PC 机或多媒体升级套件中,CD-ROM 驱动器从内部直接连到音频卡,因而不需外部连接。

CD Audio 代表高品质的数字音频信息,但所需的存储空间最大,典型的存储比例为每秒 176KB,每一张 CD-ROM 只能存储 73 分钟的 CD Audio。

1.1.2 数字视频

数字视频为多媒体开发者和最终用户开创了一个新领域。数字视频是经视频采集板数字化并存储于计算机硬盘的视频图象,一般的播放频率从 15fps(frame per second: 每秒帧数)到 30fps,播放速率与计算机硬件、压缩技术和视频使用的窗口大小直接相关。

1.1.2.1 Video for Windows 和 QuickTime

对于 Windows 操作系统,微软公司的 Video for Windows(VFW)是标准文件格式。Video for Windows 是同 Windows 3.1 一起发布的,它在 Windows 平台上首次提供视频和音频同步,其他具有竞争力的标准有苹果计算机公司的 QuickTime 技术、MPEG(Motion Photography Experts Group,动感图片专家组)视频、DVI(Digital Video Interactive,交互式数字视频)。Video for Windows 和 QuickTime 文件格式使用其自己的 codec(压缩解压缩驱动程序)进行数字视频的压缩和解压缩。

Codec 为采集和播放数字视频提供了巨大的灵活性。当你买数字采集卡时,它可能有内置的视频压缩技术,Intel 公司 Smart Video Recorder 就是这种类型的视频采集卡。Smart Video Recorder 直接将 Intel 公司的 Indeo 压缩技术内置于卡上的电路中,这能使采集的视频直接压缩到文件中,压缩是由采集卡上的硬件完成的。

如果你的视频采集卡不支持视频压缩,在采集视频之前需选定采集格式。采集格式定义

象素深度和使用的格式,在视频被采集后,选择压缩程序并用所选的视频压缩技术存储文件。

播放数字视频时,播放系统必须具备压缩程序,Video for Windows 和 QuickTime 播放系统都要检测采集视频的 codec,调入之后在播放过程中将视频解压。

一些视频采集卡支持硬件辅助的采集视频播放。通常,这些卡支持他们自己的采集格式。当视频以这种格式采集时,播放驱动程序检测在播放过程中是否拥有采集卡,如果具备采集卡,则视频通过采集卡上的硬件帧缓冲区播放,这时播放的速度快,分辨率高,硬件协助缩放图象。如果播放系统不具备视频采集卡或视频缓冲区已被占用,视频则像其他任何数字视频文件一样通过软件解压在 VGA 显示器上播放。带有 MCT(Media Control Technology,媒体控制技术)和 Auravision 芯片的采集卡支持此技术。

由于 Video for Windows 和 QuickTime 的 codec 是基于软件的,因此数字视频的播放和采集与硬件相关。CPU、视频卡和采集卡的速度越快,拥有的内存越大,则播放和采集速度越快。高速系统同时还可在不影响播放的情况下增大视频帧的尺寸。

1.1.2.2 MPEG 和 DVI

MPEG 和 DVI 都是基于文件的数字视频格式,与 Video for Windows 和 QuickTime 不同的是,它们是基于硬件的格式,也就是说,播放 MPEG 和 DVI 需要特定的视频播放卡。MPEG 和 DVI 的播放效果要比基于软件的强得多。MPEG 和 DVI 可以以 30fps 的速率全屏幕播放,而且带有音频效果。这两种方式效果更好,当然价格也更高。

由于需要特定的视频采集硬件,生成 MPEG 视频的价格非常昂贵。此外生成 MPEG 文件需要脱机压缩,目前 MPEG 视频不能在 PC 机平台上采集。生成高质量 DVI 视频同样非常昂贵,但与 MPEG 不同的是,DVI 视频可以在 PC 机平台上采集。常见的 DVI 采集卡有 IBM 公司的 ActionMedia II。ActionMedia II 卡能够将采集到的视频信息转化为可播放的 DVI 格式。在 PC 机上采集 DVI 文件可生成高质量画面,但是要想获得更高质量的 DVI 文件,它与 MPEG 一样需要脱机压缩。

由于 DVI 和 MPEG 的价格较高,所以高档应用程序才使用。基于硬件和基于软件的压缩格式之间的差距正迅速缩小,带有奔腾处理器的计算机能够以 30fps 的速率播放全屏幕的 Video for Windows 文件,这在几个月以前还是不可能的事。在选择基于硬件还是基于软件的压缩技术之前,先仔细估算一下你的应用程序的需求。

1.1.2.3 数字视频采集

当微软公司推出 Video for Windows 时,它不仅成为 Windows 平台上数字视频文件格式的标准,而且是最流行的。目前,其他具有竞争力的标准只有 IBM 公司的 ActionMedia II 视频采集卡,它可采集 DVI 视频。Apple 公司的 QuickTime 和 MPEG 视频目前尚不能在 Windows 环境下采集,QuickTime 视频在 Macintosh 平台上采集,而 MPEG 视频则通过特殊视频采集硬件脱机采集。

Video for Windows 由两部分组成:Video for Windows runtime 和 Video for Windows SDK。Video for Windows runtime 是一系列可免费散发的 DLL 和 MCI 驱动程序,它提供播放 Video for Windows 文件的能力。Video for Windows SDK 包含帮助开发者生成视频应用程序和采集视频信息的一系列的可编程 API 和工具。

Video for Windows SDK 是允许第三方软件开发商编写辅助采集数字视频的工具,例如 Lenel Systems International 开发的 MediaRecorder Toolkit。MediaRecorder Toolkit 具有强大的可并入应用程序的采集控件,此采集控件允许采集调色板、数字视频、静态图象,还允许裁剪图象。

为了获得更大的灵活性和控制,Visual Basic 程序员可以直接面向 Video for Windows capture API 编写应用程序。该 API 提供了最好的控制和最大的灵活性,但同时也是可用的最低级 API。若要在 Visual Basic 中使用该 API,必须要通过一段由 C 语言编写的 DLL 代码。

对于 Windows NT 平台,目前有 32 位的 Video for Windows SDK,这是一个可通过 32 位 Visual Basic 访问的低级 API,它同样需要 C 语言编写的 DLL 才能运行。Video for Windows SDK 和其他数字视频采集工具将在第 11 章“数字视频播放的使用”中进一步讨论。

另外还有一些厂商,他们制作的视频采集应用程序可用于脱机采集、编辑数字视频,然后可将视频合并到你的应用程序中。

1.1.3 模拟视频

模拟视频可使应用程序在可缩放窗口不丢帧地以 30fps 播放全动作视频。与数字视频不同,其信息来源不是文件而是诸如激光影碟、摄象机、和 VCR 之类的外部设备。模拟视频利用两种不同的技术将视频展现于桌面。它利用叠加视频卡和外部设备播放视频信号。

叠加视频卡提供外部视频源,叠加视频卡如果不具有内置 VGA 功能,则需要通过 VGA 卡上的功能连接器连到 VGA 卡。用户从 VGA 显示中选定特定颜色作为关键颜色(key color),多数叠加卡选择品红作为默认关键颜色,当显示叠加视频窗口时,背景用关键色填充,外部视频源就显示在屏幕显示关键色之处(亦即叠加视频窗口)。这就是所谓双平面技术(two-plane technology),VGA 面由 VGA 适配器提供,而叠加面由叠加视频卡提供。

外部设备如摄象机、VCR 和激光影碟给叠加视频卡提供视频源,一些叠加视频卡具有 TV 谐调器,因而不需要外部信号源。多数应用程序中使用的外设具有可编程接口,这个端口通常是串行口。一个应用程序,如紧急救护训练程序,可能有不同的救护序列存于激光影碟,在训练过程中,应用程序给学员播放视频片段,然后根据不同情况给出测试。在第 9 章“叠加视频外设的使用”中,你将学会利用模拟视频生成应用程序而不用了解每一外设的内部情况。

1.1.4 动画

动画就是播放一系列形成连续动作的图形图象,同时还应带有同步的音频。有两种类型的动画:cast-based 和 frame-based。

Cast-based 动画,也称为对象动画,即演示中的每一个对象都是有其模式、大小、形状、颜色和速度的独立元素。演示脚本控制对象在每一帧动画中的位置和速度。

Frame-based 动画(帧动画)由一系列快速连续播放的帧画面构成,即屏幕中一帧帧连续变化的画面形成动画。帧动画类似于数字视频技术,由于每帧代表在某一指定时间片中播放的实际画面,所以每一帧可作为独立的单元进行编辑。通常的卡通动画就是帧动画。

与此相比,cast-based 动画是完全基于时间的,没有可单独编辑的独立帧,每一帧是决定

动画中每个元素的位置和作用的时间片,也就是只有一些决定某一时刻对象的位置和动作的信息而没有图片。cast-based 动画是由显示所有成员之间关系的著作软件编辑生成的,然后再对每一元素进行单独编辑。MacroMedia Director 就是一个生成 cast-based 动画的应用程序。

用于 cast-based 动画的技术有时也叫做精灵动画(sprite animation),可用在应用程序中生成动画效果,例如,可用它产生一行文本从屏幕穿过的效果。

1.1.5 图象

多媒体编程的大部分工作是制作图形或图象。图象有两种基本描述格式:位图和矢量,表 1.2 列出了常用的位图和矢量文件格式。

表 1.2 常用 Windows 位图和矢量文件格式

文件格式	文件类型	扩展名
Windows Bitmap	位图	.BMP, .DIB
Tagged Interchanged File Format	位图	.TIF
Graphics Interchange Format	位图	.GIF
Kodak Photo CD	位图	.PCD
Targa	位图	.TGA
Icon	位图	.ICO
Cursor	位图	.CUR
PaintBrush	位图	.PCX
Microsoft Run Length Endoded(RLE)	位图	.RLE
GEM	位图	.IMG
Joint Photography Experts Group(JPEG)	位图	.JPG
Windows Metafile	矢量	.WMF
AutoCAD	矢量	.DXF

位图图象中每一象素由红、绿、蓝(RGB)三色构成,这些象素可代表不同深度,如每象素 8 位、每象素 16 位或每象素 24 位。如果象素深度是每象素 8 位,RGB 三色映象到文件中所包含的调色板,RGB 三色的组合和调色板映射构成图象。

矢量文件不包含象素信息,而是由一系列画图指令构成,这被称为图元文件(Metafile)。当读取文件时,画图指令被解释后送到 Windows GDI 来执行。通常矢量文件占用空间比位图文件少,常用在建筑设计或工程应用的程序中,如 AutoCAD。

图象在多媒体编程中非常重要。现有许多从一幅图象移到另一幅图象的切换技术,这种技术只适用于位图图象,见第 6 章“建立切换效果”。位图常用作应用程序和演示的背景。

1.1.6 文本

多媒体的引人入胜之处应归功于数字视频和音频,而文本对于多媒体程序员来说是非常重要的,但它却是经常被忽略的元素。虽然一幅图片能表达千言万语,但一幅有少量注解的图画比一幅单独的图画更容易被理解和认识。

考察一下 Windows 应用程序中的工具栏,工具栏之所以流行是因为将文字表述转换为

图形或图标描述。最初的想法是工具栏中位图图像构成的按钮会给用户直观的感觉,但近来给工具栏增加工具提示(当鼠标划过按钮时会有文本弹出)的趋势在增加。显然文本仍然具有很强的生命力。

文本元素可以有多种使用方式和显示效果,可以在屏幕上移动,也可以在位图图像上透明显示,在第10章“动画”中将解释此技术以及其他将文本用于应用程序的技术。

1.2 高级多媒体服务

Windows 为开发环境提供了许多高级多媒体服务,这些服务的形式是媒体控制接口(Media Control Interface, MCI)和音频 API。Visual Basic 用控件(control)提供访问服务,即以简易方法来实现这些服务。

1.2.1 媒体控制接口

媒体控制接口(MCI)为应用程序与不同的多媒体设备、外设和驱动程序之间的通信提供了一个标准的、与设备无关的方法。MCI 是在开发 Windows 3.0 Multimedia Extension 时建立的,随着 Windows 3.1 的问世,MCI 已成为 Windows 操作系统的固有部分。

MCI 设计为可与不同的设备类型通信,这些包括波形音频、MIDI 音频、CD Audio、数字视频、叠加视频卡、VCR、激光影碟、MPEG 视频等各种不同的设备类型。由于 MCI 功能强大而且灵活,它已是 Visual Basic 程序员最完善的 API 集。

MCI 与特定的 MCI 驱动程序通信,这些 MCI 驱动程序解释应用程序发出的 MCI 命令,并将用户命令翻译为执行指令所必需的与设备相关的命令。

MCI 有两种与设备通信的方式:字符串界面和命令界面。字符串界面向 MCI 驱动程序发送文本命令,这些命令被解释并翻译为消息命令后再发送给 MCI 驱动程序。字符串界面适用于脚本语言或用一系列 MCI 命令快速操作设备的情况。

虽然命令(或消息)界面执行起来有些麻烦,但因为不需进行字符串的分析和查找,它能提供较快的执行速度。命令界面对每一命令类型有相应的结构,这些结构由相应的命令、标志和信息构成并发送给设备驱动程序,返回信息也具有同样的结构。

1.2.2 高级音频服务

Windows API 为播放音频文件提供了几个高级 API。这些 API 使用非常方便,使用者不需具备有关 Windows 体系结构或多媒体编程的知识。然而,使用上的简便却使程序在控制播放音频上缺乏灵活性。根据你的需要这些 API 将满足程序在音频播放方面的需要。

Windows 为音频播放提供的 API 有两个函数:MessageBeep 和 SndPlaySound,它们都是单行函数,与 WIN.INI 或播放波形音频文件的注册数据库项有关,这些数据库项将音频文件映射到诸如 ApplicationClose 或应用程序定义项这样的系统事件。

1.3 低级音频服务

Windows 为 Visual Basic 程序员提供了一些低级音频服务,这些服务控制波形音频文件

和 MIDI 文件的播放,并控制音频卡的辅助输出。尽管所有的 API 都能被 Visual Basic 环境调用,但仍存在一些对 Visual Basic 的限制,禁止某些音频播放策略的使用。

低级波形音频 API 可用于定制并控制播放 MIDI 文件和波形音频文件的缓冲区。高级音频 API 和 MCI 界面使用的缓冲区可能不能满足你的应用程序的需要,因而在建立分布式或网络应用程序中定制缓冲区以防止音频中断是一个重要任务。低级 API 也是提供建立音频编辑、特殊效果和音频操作应用程序的最好界面。

1.3.1 低级波形音频

低级波形音频命令可控制播放、音量、播放速度和快速定位,如果应用程序需要控制这些元素,则必须选择低级音频服务。所幸的是,这些 API 的使用相当简易直观。

如果没有用 C 语言这样的外部语言编写的 DLL 代码的帮助,则用低级命令播放音频将会非常困难,这是因为使用低级 API 时 Windows 操作系统中的音频播放引擎需要回调 DLL 中的函数。

低级命令加以扩展可以做到不需回调函数就能播放较短的波形文件。但 MCI 和高级音频服务已为你提供了播放功能,因而无须这么做。

1.3.2 低级 MIDI 音频

低级 MIDI 音频播放与波形音频可归为一类。低级 MIDI 音频播放并未给开发者提供更多的控制,控制仅限于音量、播放速度、快速定位等。仅当 MIDI 卡的 MCI 驱动程序缺少应用程序需要的功能(如控制音量)时,才使用低级 MIDI 音频功能。

1.3.3 辅助音频设备

Windows 操作系统定义可编程控制的辅助音频设备。辅助设备常被看作多媒体系统的一部分,事实上,可控制的辅助设备就是你 PC 机音频卡的一部分,如果你的音频卡提供此通道且 CD-ROM 已与之连接,它们就是线输入和 CD-ROM 音频输入线。

辅助 API 用来控制每一输入的音量设置。如果你的应用程序将 CD Audio 或外部音频输入音频卡,这是控制音量设置的惟一方式,所以辅助 API 是非常重要的。

1.4 Visual Basic 扩展:为 Visual Basic 添加多媒体功能

对于开发多媒体应用程序来说,Visual Basic 是一个功能强大的环境,它使用方便,易于掌握,且与 Windows 操作系统结合紧密。Visual Basic 最大的特点之一就是其可扩展性。Visual Basic 开发环境有任何平台、任何环境上的附加(add-on)软件模块组成的巨大的工具箱。

Visual Basic 开发环境的扩展可以通过附加工具和库进行,如 Visual Basic custom controls(VBX)、OLE custom controls(OCX)、OLE 自动服务、动态链接库(DLL),也可通过建立用户自己的 Visual Basic 自动部件扩展。分布于 Visual Basic 环境中的各种类型的 OLE 控件,只须打开便可使用。

这些附加部件实际上可提供像 C 这样的低级开发环境所能开发或访问的任何功能。

1.4.1 DLL 支持

DLL(动态链接库)是一个例程库,它可被许多应用程序调用而不需将 DLL 代码连接到调用它的应用程序。与 C 语言或 Pascal 语言中的静态库不同,静态库中的所有代码均被链接到客户应用程序,而 DLL 代码只有在运行时才链接到应用程序。

Windows 和大多数 Windows 应用程序就是建立在可复用的 DLL 上的。使用 DLL 可减小可执行文件(.EXE)的大小并在程序运行时减少物理空间的占用。Visual Basic 与多数 Windows 开发环境一样,提供对外部 DLL 的支持。当调用 Windows SDK 时 Visual Basic 就使用 DLL。Windows 就是建立在这种共享代码概念的基础上的。

DLL 可用于所有 Windows 应用程序,与开发应用程序的语言无关,这是因为 Windows 提供了 DLL 库中函数的语言定义,确保参数以固定的方式传送给例程,并且在函数退出时以相同的方式清除堆栈——无论 DLL 是用 Pascal、C 还是其他什么语言写的。

惟一的例外是用 C++ 类的方法,导出的 C++ 类接口除 C++ 应用程序外,不容易从应用程序中访问,这是由于在导出的方法时编译器对方法的实际名字进行了破坏。

有许多媒体工具厂商出售可附加的 DLL,其功能有切换、动画、图象处理、视频采集等等。对于 Visual Basic 程序员来说,Windows SDK 是使用最广泛的 DLL 资源。所有用于 MCI、高级和低级音频例子的函数均来源于 MMSYSTEM.DLL,它是 Windows 3.1 系统的一部分,对于 Windows 95 和 Windows NT,该 DLL 被称为 WINMM.DLL。

1.4.2 Visual Basic 控件(VBX)

当 Visual Basic 首次发行时,它为软件开发厂商建立了一个新工业——软件组件工业。这并非微软公司的远见卓识所至,而是因为 Visual Basic 是第一个可以使用方便的附加控件——Visual Basic 控件(VBX)——的商业化程序。

许多软件厂商就是建立在 Visual Basic 控件基础上的。VBX 为在 Visual Basic 环境中快速而简便地增加编程功能提供了一种简单的方法,对于多媒体程序员来说,这更是不可多得的好事。目前市场上的 VBX 包括数据库控件、通信控件、网络控件,当然还有多媒体控件。

目前的 VBX 有图象显示、图象操作、切换效果、倾斜、叠加视频控件,还有许多其他多媒体控件。VBX 是扩展 Visual Basic 开发环境功能的最快最有效的方式。

然而,VBX 的投资前途并不光明。这是由于 VBX 是根据 16 位环境设计的,它的体系结构不能扩展到 32 位环境。VBX 将由一个新技术——OLE 自定义控件(OCX)——所代替。Visual Basic 4.0 仍继续支持 VBX 控件,但只是在 16 位环境。

注: 微软公司提供了 VBX 到 OLE 控件的转换工具,它可很顺利地将一种体系结构转换为另一种。此工具将加快从 VBX 控件到 OLE 控件的转换。

1.4.3 OLE 自动服务器

OLE 自动服务器(Automation Server)是微软公司提供给 Windows 操作系统的最先进的技术之一。OLE 2.0 于 1993 年面市时就具有自动特性。自动 OLE 定义了一个接口,通过此接口无论什么编程环境都可使应用程序成为可编程的。Microsoft Word 就是一个可编程

的应用程序。例如,开发者可通过 Word 自动 OLE 接口在自己的应用程序中调用 Word 的各种功能,如拼写检查功能。

我们注意到在软件开发工业中,所有开发厂商想要解决的关键问题就是建立与应用程序无关的软件,也就是说代码无须更改就可被一次又一次地使用;当函数被调用时,带有预定义函数的软件对象返回可预测的结果。软件对象生成时其内部实现并不重要,关键是其接口和预期结果。

VBX 和 DLL 是生成复用软件的最初尝试。自动 OLE 是生成真正可复用软件部件的第一步,自动对象是面向对象的部件,它与开发环境无关。

自动 OLE 为程序员提供对象(objects),这些对象由属性(properties)和方法(methods)组成。属性表示数据变量——应用程序用对象的载体管理服务器对象的形式和行为;方法表示对象完成的动作。

自动对象包含在服务器中,OLE 自动服务器的类型有两种:处理服务器(in-process server)和本地服务器(local server),处理服务器是 DLL,而本地服务器是 EXE。在不久的将来分布式服务器(distributed server)将作为第三种类型的服务器问世,它是运行在另一台计算机上的 EXE 文件,而目前所有的服务器都在同一台计算机上运行。

Visual Basic 是 OLE 载体(container),载体是能嵌入或控制 OLE 服务器并能控制自动服务器的程序。Visual Basic 语言通过 Object 变量类型提供支持自动 OLE 的结构,Visual Basic 函数 CreateObject 用于建立一个自动服务器对象的实例。

当前市场上已出现了一些多媒体 OLE 自动服务器,其中最强大的自动服务器之一是 Lenel Systems 公司的 MediaDeveloper,MediaDeveloper 是为图象、音频、数字视频、模拟视频、动画对象提供一个统一接口的自动服务器。还有一些 OLE 自动服务器开发厂商提供了开发应用程序的其他类型多媒体服务器,最新的 OLE 自动服务器开发厂商列表可参见微软公司 CompuServe 上的 WINOBJ 论坛。

1.4.4 用 Visual Basic 建立 OLE 自动服务器

Visual Basic 4.0 提供了可合并入用户自己的多媒体应用程序的强大功能——建立你自己的 OLE 自动服务器的能力。在第 12 章“OLE 自定义控件的构造与使用”中,你将学习如何应用 Visual Basic 建立一个多媒体 OLE 服务器应用程序。

通过建立你自己的自动服务器,你可利用你的应用程序所具有的功能,并可将这些功能提供给其他的程序,例如,如果你的多媒体应用程序读取特殊的文件格式(如 JPEG 图象文件),你可提供读取 JPEG 文件的方法并返回一个 Windows 位图句柄,其他程序也可使用此方法读取它们所不支持的图象文件。

建立 OLE 自动服务器的功能为生成应用程序开辟了一条新的道路,开发者现在可以将此功能用于自己的以及其他的具有多媒体功能的应用程序。

1.5 新一代产品:OLE 自定义控件(OCX)

Visual Basic 4.0 引入了对新一代组件软件——OLE 自定义控件(OCX)——的支持。OCX 提供与 VBX 类似的功能,但它建立在一个全新的、为今天以及未来操作系统设计的体

系结构之上的。

OLE 控件体系结构设计为可适应 16 位和 32 位环境,这确保你今天使用的控件能继续在明天使用。OLE 控件基于两个 OLE 关键部件:自动和内置(in-place)编辑。OLE 控件通过自动方式访问控件的属性和方法。

与 VBX 不同,如果控制提供控件菜单的话,当 OLE 控件在程序运行过程中被激活时,程序菜单可被更改并被并入控件菜单;如果未提供控件菜单,则程序功能不变。内置编辑特性允许在使用控制过程中调用控件特定的项,例如,在一个使用数字视频的实时应用程序中,会有视频采集控件,当此控件被激活时,程序菜单与视频采集菜单合并以控制数字视频的采集。

1.6 Visual Basic 的多媒体控件

Visual Basic 提供了一些内置的多媒体控件——MCI、图形和文本控件,它们辅助多媒体应用程序的开发。Visual Basic 所包含的控件开始进行多媒体开发的一个简单的方法,这无需投入大量的资金和时间。

注: 在早期版本的 Visual Basic 中,MCI 控件、图形控件和文本控件属于 VBX 控件,在 Visual Basic 4.0 中已升级为 OLE 控件。

1.6.1 MCI 控件

MCI 控件为开发多媒体遇到的所有 MCI 设备类型提供了一个基本接口。有为各种命令如 Play, Stop 和 Pause 提供接口的用户可选按钮,此接口可作成可见的(例如,在应用程序运行时为最终用户提供的控件按钮)也可作成不可见的,无论控件是否可见,开发者都可编写代码来操作控件。

1.6.2 图象控件

图象(Image)控件用于显示基本 Windows 图象类型。图象控件可读取的图象文件格式如表 1.3 所列。

表 1.3 Visual Basic 图象控件所支持的图象文件格式

图象文件格式	文件扩展名
Windows Bitmap	.BMP, .DIB
Windows Metafile	.WMF
Windows Icon	.ICO

图象控件可显示以普通模式或缩放模式装载的文件,也可用来建立到其他程序的 DDE 连接,同时支持 OLE 2.0 拖放功能。图象控件不是作为 OLE 自定义控件实现的。

1.6.3 标志控件

标志(Label)控件与图象控件相同,也不是作为 OLE 自定义控件实现的。标志控件用来

显示文本,当标志控件用于透明背景时,文字可显示于其他控件(如图片控件)之上,产生图象字幕或叠加视频的效果。

1.6.4 图片剪贴控件

图片剪贴(Picture Clip)控件是一个用于为快速显示而存储位图文件的先进控件,在创建动画或当有大量位图文件需要存储时特别有用。图片剪切控件的通常用法是存储按钮的所有不同状态(普通状态、按下状态、失效状态),并使用 GraphicCell 属性选择合适的位图在图片框或图象控件下显示它。

图片剪切控件在运行时不可见。特别是从一个位图中建立一个图象数组时可定义图片剪切控件的行和列。图象的引用则是通过定义图片框控件或图象控件的 GraphicCell 属性进行的。Visual Basic 4.0 有一个示例程序,该程序是图片剪切控件用法的演示,它位于 SAMPLES\PICCLIP 子目录下,文件名为 REDTOP.VBP。

1.6.5 图片框控件

图片框(Picture Box)控件与图象控件类似,它可读取与图象控件相同的文件格式(参见表 1.3)。

图片框控件仅显示图片框内的图象,如果图象小于图片框则图象显示于图片框控件的左上角;如果图象大于图片框,则图象从图片框左上角开始逐点画下,图片框之外的图象部分不可见。这就是图象裁剪技术。图片框控件具有的自动大小属性可自动调整控件至图象的大小。

1.7 多媒体数据库开发

与 Visual Basic 4.0 集成在一起的是功能强大的 JET Database Engine。JET 引擎是使多媒体数据库应用程序生成屏幕快照的 OLE 自动对象集。JET 引擎是与 Microsoft Access 2.0 一起发布的一个同样的关系型数据库。JET 引擎的使用接口可以连接到不同的桌面数据库格式,也可使用 ODBC(开放式数据库连接)驱动程序访问客户机/服务器数据库。内部格式为 MS Access,其他可用的桌面数据库有 Paradox,dBASE III,dBASE IV,FoxPro 和 Btrieve;文本文件和电子表格也可使用。ODBC 连接提供对诸如 SQL Server,Oracle,Sybase,Informix 数据库服务器这样的 SQL 客户机/服务器数据库的访问。

正如在以后章节可见到的,开发多媒体应用程序时有两种在数据库系统中访问和存储多媒体数据的方法:基于文件(file-based)的方法或 BLOB(Binary Large Object,二进制大对象)方法。

基于文件的方法用于多媒体数据存于硬盘或 LAN 上的情况。文件名和路径存于数据库中,访问文件时,检索文件定义并送到多媒体控制例程,此方法执行直接而简便,易于实现。所有数据库系统均可用——包括文本文件和电子表格——且文件可用于外部程序。

当所有多媒体数据存于数据库中的某个表格时,使用 BLOB 方法。此系统将多媒体数据看作为字母数字式的数据,所有相关的记录数据存储在一起。使用 BLOB 字段,所有数据都通过数据库表格存储在一起,不必担心文件被删除或移动。

文件方法和 BLOB 方法各有利弊,在执行你的多媒体数据库应用程序前应先仔细考虑。

以下为使用文件多媒体数据库的优缺点:

优点:

- 实现简单,多媒体字段为标准的文本字段;
- 支持任何数据库类型包括文本文件和电子表格;
- 多媒体文件可与外部程序共享;
- 存储和检索多媒体数据速度快。

缺点:

- 不改变所有的数据库记录就很难将多媒体数据移动;
- 导致数据装载遍布各处(本地硬盘,LAN,远程 PC 机);
- 导致不可共享或不能访问数据,例如,网络用户中不同的驱动器映射可导致无效的文件位置。

以下为多媒体应用程序使用 BLOB 的优点和缺点:

优点:

- 所有数据集中存储;
- 移动数据库文件简单;
- 数据可被所有用户访问和共享。

缺点:

- 在不从数据库应用程序输出的情况下,数据不能被外部程序共享;
- 存储和检索速度慢;
- 只可用于有 BLOB 支持的数据库系统。

在第 14 章“采用 JET 引擎的多媒体数据库”中有对每种方法的详细描述。

1.8 超文本和热区

超文本(hypertext)和热区(hot spots)是相似的技术,用于把文本或图象区域链接到其他文本或图象。它的作用是允许用户以自己定义的方式在应用程序中导航。此技术的典型例子是 Windows Help Engine。当引用 Windows 帮助文件时,用户可根据自己的意向查找主题。

Visual Basic 访问 Windows 操作环境中的工具,如 JET Engine、Windows API 和 OLE 结构存储,它们在其应用程序中提供建立超级文本和热区的方法。这些技术在建立训练和信息站应用程序时是最必需的,第 7 章“超文本和热区”将详细介绍这些技术。

1.9 相关内容

Visual Basic 程序员可使用内置于 Windows 平台和包含于 Visual Basic 开发环境中的各

种多媒体功能,学会这些功能并掌握它们在应用程序中的正确使用方法可使你方便地建立多媒体应用程序,当多媒体编程的兴奋到来时会带给你乐趣和享受。

- 要了解所有多媒体设备类型的 Windows MCI 编程接口,见第 3 章“媒体控制接口 (MCI)”,该章向你详尽介绍 Windows 如何通过使用 MCI 提供多媒体功能。
- 要了解如何控制图片和文本来产生切换效果、热区和超文本功能,参见第 5 章“图形和调色板”。
- 要学习在多媒体开发中如何建立并使用 OLE 自定义控件,见第 9 章“叠加视频外设的使用”。该章检查 OLE 自定义控件开发工具以演示如何建立你自己的控件,也讨论了关于如何在 Visual Basic 4.0 中建立 OLE 自动对象。
- 要学习建立多媒体数据库应用程序的技术,见第 12 章“OLE 自定义控件的构造和使用”。该章有对 JET 引擎和使用 BLOB 字段的概述。

第2章 Windows 多媒体开发

“多媒体”这个词对不同的人有不同的含义。它通常的含义是用文本以外的媒体给他人传送消息,这里的媒体可以是动画、图象、音频或视频等。因为要将多媒体传送给观众,必须将其集成到一个应用程序或产品中。通常要求程序员能够开发多媒体功能并将其包含于应用程序中,但这种方法非常耗时且效率很低,如果将多媒体能力置于操作系统中就会使这种工作变得容易多了。

将多媒体作为操作系统的一个标准的甚至是核心的组件是多年来已经普及于 Apple 平台上的一个概念,直到 1991 年底,微软公司才将多媒体作为 Windows 的核心组件。随着多媒体能力的引入,计算机领域已经能够开发许多过去难以想象的应用程序。

了解操作系统的能力和局限性对于开发和引用适时而健壮的应用程序是非常重要的,这点在多媒体世界尤为重要。

本章对 Windows 多媒体能力设计进行详细讨论,从 Windows 3.0 的多媒体到 Windows 95,对于每种平台上提供的服务都给出了一个较为全面且完整的发展过程综述。

通过本章的学习,你将了解:

- Windows 多媒体接口的设计原理和方法。
- 适用于 Windows 95 的多媒体类型及其与 Windows 3.1 和 Windows NT 的比较。
- 什么是 Windows 媒体控制接口(MCI)及其工作原理。

2.1 将多媒体能力集成到 Windows 中

Windows 3.0 的问世成为 PC 机操作系统革命的一个开端。Windows 3.0 是第一个有直观图形界面的通用 PC 机操作系统,其图形界面和多任务能力使开发一类新型的应用程序成为可能,这类应用程序可为用户提供先进的图象、演示和画图功能。

当 Windows 3.0 出现于计算机屏幕上时,Windows 3.0 SDK 尚无内置的多媒体能力。Windows SDK 为开发者提供了 Windows 操作系统功能的可编程接口。在 1991 年底,微软公司发行了 Windows 多媒体扩展(Multimedia Extensions)和 Windows 多媒体开发工具(Multimedia Development Kit),多媒体开发工具中的 API(即多媒体扩展),是 Windows 3.0 编程环境中的特别工具集。多媒体扩展为应用程序开发者提供了开发多媒体应用程序的高级和低级服务。多媒体扩展提供了一种访问多媒体 PC 机扩展功能的途径。

曾作为独立产品销售的多媒体扩展在 Windows 3.1 中成为中心组件。Windows 3.1 SDK 包含所有曾作为 Windows 多媒体开发工具一部分的 API 及高级和低级服务,多媒体第一次成为 PC 机平台上操作系统的中心组件而不是附加部分,这是多媒体发展过程中的重要事件。多媒体成为操作系统的中心组件的意义在于:多媒体应用程序成为主流,运行多媒体应用程序不再对操作系统有特殊的要求。

多媒体扩展已集成到 Windows 3.1 以后发行的所有 Windows 版本中。Windows 3.1 和 Windows for Workgroups 是具有多任务能力的 16 位操作系统,新一代 Windows——Windows NT 和 Windows 95 是具有抢先式多任务能力的 32 位操作系统。

建立 32 位抢先式多任务环境中运行的真 32 位应用程序的能力会促使一类新型的多媒体应用程序的产生,这些应用程序包括视频会议、视频点播(Video On Demand)、连续媒体服务、多媒体培训和信息站等。所有这些应用程序需要在局域网(LAN, Local Area Network)、广域网(WAN, Wide Area Network)或电话线上传输视频和音频,多任务环境能确保这些应用程序的运行时间,而不中断多媒体传输。

由于多媒体服务已集成到操作系统中,多媒体服务需要达到下列要求;

- 可扩展性(Extensibility):容易添加未来的对操作系统的增强功能并向下兼容;不需改变操作系统的体系结构就可以对其增加新的多媒体技术。
- 与设备无关(Device independence):所有 PC 机多媒体硬件和外设可由同一命令集激活;为应用程序开发者和最终用户提供的多媒体硬件应为即插即用型。

下面先讨论 Windows 体系结构在 Windows 3.0 到 Windows 3.1 的 16 位操作系统中达到这些设计要求的过程,然后再考察 Windows NT 和 Windows 95 中使用的先进的多媒体技术。

2.2 16 位 Windows 多媒体体系结构

当微软的设计工程师在 Windows 上增加多媒体扩展时,他们需满足多媒体子系统设计目标的需要,并确保与当前 Windows 体系结构的兼容性。Windows 多媒体系统的设计师通过在 Windows 上增加一个新的 DLL——MMSYSTEM.DLL 达到了特定的设计要求,Windows 3.1 发行时,该 DLL 成为 Windows 的一个标准组件。MMSYSTEM.DLL 在系统设备驱动程序和应用程序代码间建立了一个转换层,Windows 多媒体系统提供的所有多媒体应用程序的程序接口(API)包含于 MMSYSTEM.DLL 中。

MMSYSTEM.DLL 还提供所需的到驱动程序的运行时链接,MMSYSTEM 中的 API 动态装载所需的设备驱动程序。这种运行时链接是任何多媒体操作系统成功的关键,应用程序不要求链接到特定的硬件设备驱动程序,这就避免应用程序只能在链接到它的设备驱动程序上运行。

运行时链接是通过严格定义的、一致的设备驱动程序接口实现的。微软公司定义了一个可安装的设备驱动程序接口,此接口是一个被现有多数多媒体硬件生产厂商认可的一致接口。遵循此接口定义,硬件厂商可确保多媒体应用程序能够发挥其硬件的优势,同时,软件厂商可确保应用程序与设备无关。这个可安装的驱动程序接口的定义在 Windows 多媒体开发工具(MDK, Multimedia Development Kit)中。

图 2.1 说明了 Windows 和多媒体子系统之间的关系。

MMSYSTEM.DLL 是应用程序代码和设备驱动程序代码之间的一个组件。MMSYSTEM.DLL API 包含 MCI API 和低级多媒体支持函数,这些支持函数包括对以下设备的低级支持:

- 波形音频
- MIDI 音频
- 辅助设备
- 低级影片播放器支持
- 多媒体定时器服务
- 游戏杆支持

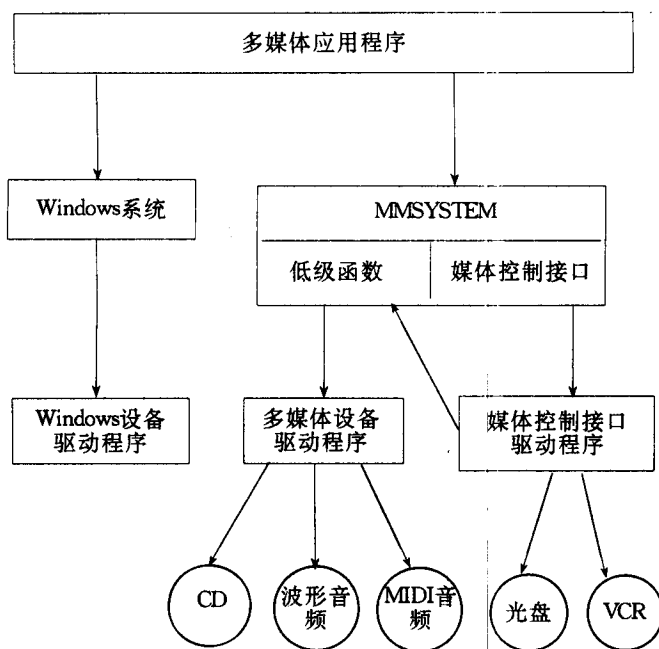


图 2.1 Windows 系统多媒体系统结构

当应用程序访问 Windows 多媒体服务时, MMSYSTEM 将函数调用翻译为适当的设备驱动程序调用, 图 2.2 说明了此点。

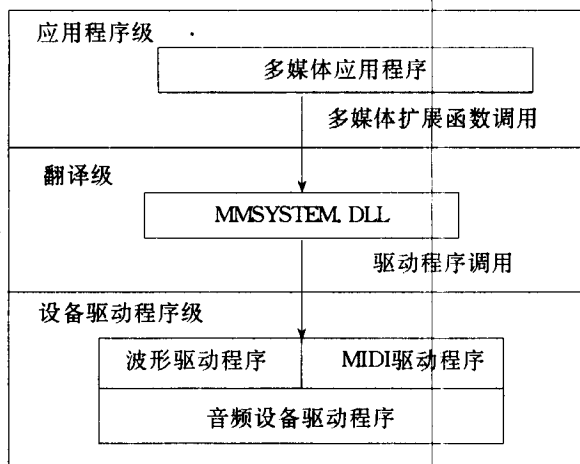


图 2.2 多媒体 API 传送给设备驱动程序的过程