

# It Talks, It Sings, It's Multimedia Windows

Tom Yager

**W**indows 3.1 seems like much more than a point release. But one crucial piece of it—multimedia support—hit the streets several months before its commercial release. The Windows Multimedia Extensions are part of the Multimedia PC specification (see "The Multimedia PC: High-Powered Sight and Sound on Your Desk," February BYTE). With version 3.1, the software side of MPC becomes a standard part of Windows.

What the extensions buy you depends largely on your configuration. The most accessible addition is audio. If you have an audio I/O card like Creative Labs' Sound Blaster Pro, the new routines and applications in version 3.1 will let you record and play digitized sound. If your audio card supports MIDI, or you have a component MIDI device that's compatible with Roland's MPU-401, version 3.1 will let you record and play MIDI data. The Multimedia Extensions are, not surprisingly, extensible, so in addition to audio and MIDI, they support the following media types: CD audio, videodisc, video overlay, and animation. Each of these types (except animation) requires additional hardware and, in most cases, drivers supplied by vendors.

In a way, animation requires additional hardware, too: a Mac, which is

now the only way to create an MPC-compatible animation. Support for full-motion (digital) video (using the "digitalvideo" media type) is expected shortly, and there are also reserved (but not supported) media types called dat, scanner, vcr, and other. I don't know what an "other" does, but the rest of the names are self-explanatory.

Some of the media types are covered by their own low-level programming calls. For some operations, such as MIDI recording, it helps to have the most direct control possible. But the heart of the multimedia programming support is the Media Control Interface. This is a well thought-out, high-level function-call interface to a command interpreter. An MCI command string includes a command, a target device, and arguments. The interpreter routes the command to the appropriate handler for the specified device.

The result is readable code and straightforward addition of new media types to existing applications. The string interface also works well with alternative Windows programming environments like ToolBook. Those of us who work in C need Microsoft's Multimedia Development Kit, which requires a CD-ROM drive and the Windows Software Development Kit.

MCI's command structure is pat-

terned after a tape transport. Commands supported by most devices include open, play, and pause. The first two are all that's required to get most media types to play or display their output. Here is one that plays a digital audio file called WILLIAM.WAV:

```
open \sounds\william.wav  
type waveaudio alias bill  
play bill
```

The alias simply gives you an easier way to reference the open file. Playing a MIDI file, an animation, a track on an audio CD, or any of the other supported media types is no more difficult than this, and digital video and other future types will follow the same command format.

If you're not a programmer, the fact that MCI is so easy to write helps you as well. MPC-compatible software will become available rapidly because MCI is such a cinch to learn and use. Microsoft probably hopes you'll see all the multimedia stuff in version 3.1 and run out and buy an MPC upgrade kit. Maybe that's not such a bad idea.

---

*Tom Yager is a BYTE technical editor and director of the Multimedia Lab. He can be contacted on BIX as "tyager."*

---

workstations with 2 MB or more. Version 3.1 will run in standard mode in the presence of DOS 5.0's EMM386 memory manager, something that version 3.0 refused to do. But users who have favored standard mode even on high-end hardware might want to reconsider that policy in light of version 3.1's enhanced mode.

The virtual memory manager has been overhauled, and disk I/O benefits from a new 32-bit virtual device driver that works with Western Digital and compatible hard drive controllers. A performance boost, not mode-dependent, comes from the new Smartdrive. Caching both reads and writes, it keeps my disk noticeably quieter than its version 3.0 counterpart.

## Keep on Keeping On

As a DOS multitasker, version 3.0's reach often exceeded its grasp—it encouraged you to try things that either wouldn't work or wouldn't work reliably. This wasn't surprising, given the sandy DOS foundation on which Windows builds its airy superstructure of virtual machines (VMs).

What has pleasantly surprised me about version 3.1 is that, although the basic architecture hasn't really changed, DOS multitasking does work a whole lot better. Preliminary CPU and screen I/O benchmarks peg version 3.1 VMs well ahead of their version 3.0 counterparts. And the VMs are much more stable. Incidentally, version 3.1 can handle 9600-bps serial communi-

cations, even in the background.

Matching some of OS/2 2.0's features, windowed VMs now provide DOS mouse support and offer a variety of alternative fonts. An .INI file remembers the size and location of windowed DOS programs.

Windows now traps the three-fingered salute. Press Ctrl-Alt-Del, and version 3.1 prompts you to continue the current DOS (or Windows) task, kill it, or actually reboot. If you pull the plug on a DOS task in this situation, Windows shuts it down without the scary termination message.

According to Microsoft, Windows does a sanity check to see if DOS seems healthy. Even so, a rogue DOS task can subtly alter low memory and bring down the whole