

# Re: Enhanced PLAY.CMD with play list support

Source: <http://sci.tech-archive.net/Archive/sci.astro/2004-11/0364.html>

---

**From:** Michael Baldwin Bruce (*mbbruce\_at\_mighty.co.za*)

**Date:** 11/06/04

Date: 5 Nov 2004 19:07:17 -0800

bruce aka tholen@antispam.ham the troll wrote in message  
news:<kcxid.63356\$K13.36017@twister.socal.rr.com>...

> *Marty writes:*

>

> >>>>>> *Peter Weilbacher wrote:*

>

> >>>>>> *Michael DeBusk wrote:*

>

> >>>>>>> *Marty wrote:*

>

> >>>>>> *The most important aspect of the script for me was the resource leak  
> >>>>>> fix. Something inside of Timidity\_MCD does not get cleaned up correctly  
> >>>>>> without the call to mciRxExit(). Strangely, missing this call doesn't  
> >>>>>> seem to cause a problem with any other codecs that I've found, just  
> >>>>>> Timidity. I guess the other codecs are smarter about cleaning up after  
> >>>>>> sloppy processes.*

>

> >>>>>> *I've probably encountered that problem. There have been times when  
> >>>>>> PLAY.CMD would fail, while other audio file playing utilities would  
> >>>>>> continue to work, including some Windows-based ones. But I have also  
> >>>>>> encountered one instance in which PLAY.CMD worked from inside my editing  
> >>>>>> program but failed as a standalone utility in another windowed command  
> >>>>>> prompt session. Haven't been able to figure that one out.*

>

> >>>>> *Did you get any messages from PLAY about a "REXX Alias" or something to  
> >>>>> that effect?*

>

> >>>>> *Unfortunately, I cannot reproduce the problem on demand, so I have to  
> >>>>> rely on memory for the error message, but it was something like  
> >>>>> "mciRxInit routine not found".*

>

> >>>> *I've seen this type of thing before with other DLLs (most prominently  
> >>>> REXXUTIL). The call to SysLoadFuncs returns a success, but REXX later  
> >>>> claims that the functions which were allegedly just loaded are not  
> >>>> available. I think this is a separate issue from the ones that I  
> >>>> addressed with my changes.*

>

> >> *A separate issue involving a bug in REXX?*

>  
> > *I believe so, since I've seen it outside of the context of multimedia.*  
>  
> >>>> *Instead of maintaining handles with binary data and the*  
> >>>> *like, in REXX you create a name or "alias" to associate with your MCI*  
> >>>> *device. Every multimedia operation you do in REXX is done against this*  
> >>>> *alias so that the state is maintained by the MMPM libraries, but your*  
> >>>> *context is maintained.*  
> >>>>  
> >>>> *Unfortunately, PLAY.CMD hard-codes a REXX alias name, so that if you*  
> >>>> *attempt to run two of them at the same time, the same alias is used and*  
> >>>> *the context is applied from your original running PLAY.CMD. Likewise,*  
> >>>> *if you fail to close out and free the REXX alias (due to killing the CMD*  
> >>>> *process, for example), its state continues to be maintained by the*  
> >>>> *system. The alias is global. You have to free up the REXX alias before*  
> >>>> *it can be used again.*  
>  
> >>>> *Well, that seems consistent with one occasion in which I was repeatedly*  
> >>>> *playing a very short audio clip, pressing the key to repeat the play*  
> >>>> *very rapidly, such that it seems possible the system started up the next*  
> >>>> *PLAY before completely cleaning up after the previous PLAY.*  
>  
> >>>>> *My version of PLAY checks for this condition and performs the free if*  
> >>>>> *needed.*  
>  
> >>>>> *I'm not sure under what conditions my audio subsystem supports the*  
> >>>>> *simultaneous playing of two audio streams. I'm using the SoundMax*  
> >>>>> *audio built in to the Intel 850 motherboard (ADI 1885). I know that*  
> >>>>> *if I'm playing a .wav file, then fire up Mahjongg, it complains about*  
> >>>>> *the device being busy, even though I've disabled sounds in Mahjongg.*  
>  
> >>>>> *Even if your audio drivers are capable of audio stream mixing, an*  
> >>>>> *application can still open the device in exclusive mode. What are you*  
> >>>>> *using to play the wave file?*  
>  
> >>> *I've tried a variety of different utilities over the years. I probably*  
> >>> *use PLAY.CMD more often than anything else. I like the simplicity of*  
> >>> *the command line interface.*  
>  
> >>>> *I don't think the system will default to*  
> >>>> *opening it exclusively if you double-click a wave file or use PLAY.*  
>  
> >>> *Well, Mahjongg complains that the device is locked when PLAY is*  
> >>> *being used to play a .wav file. When it's finished playing, no*  
> >>> *such message occurs. Maybe Mahjongg wants exclusive access and*  
> >>> *refuses to share, but then it shouldn't say that the device is*  
> >>> *locked when it really isn't (if that is indeed the case).*  
>  
> >> *Usually when a program requests exclusive access, as long as no other*  
> >> *applications have the device open exclusively, it will be able to rip*  
> >> *the device away from programs that open it in shared mode.*

>  
> *And I have encountered situations in which music being played was*  
> *abruptly stopped while some other application took over, and then*  
> *resumed when that other application was done, but I don't remember*  
> *which applications were involved. In addition to PLAY.CMD, I have*  
> *also utilized z! for .mp3 files and Internet audio streams.*

those apps probably don't like Barnes either, bruce.

>  
> >>>> *On the other hand, the driver came with WarpMix, so it looks like it*  
> >>>> *should be able to mix at least a few audio sources.*  
>  
> >>> *I think WarpMix is tool used to set volume levels of various audio*  
> >>> *channels (CD, line in, etc.), correct?*  
>  
> >> *Yes.*  
>  
> >>> *That's different from stream*  
> >>> *"mixing" where different audio streams are combined, in software or*  
> >>> *hardware in some cases, into one stream that is played. The former*  
> >>> *process is an analog one and only happens among a fixed set of channels.*  
> >>> *The latter is digital and can take place over a variable number of*  
> >>> *audio streams.*  
>  
> >> *Odd to use "mix" in the name if it doesn't do any mixing.*  
>  
> > *It does mixing of a sort, but I certainly never liked the fact that the*  
> > *same word describes two very different scenarios. It can be quite*  
> > *confusing.*  
>  
> >>>>>> *And I've*  
> >>>>>> *also seen cases where the first invocation of PLAY.CMD would return an*  
> >>>>>> *error message but a second invocation would cause the entire windowed*  
> >>>>>> *command prompt session to die.*  
>  
> >>>>> *This one sounds like the MM libraries ran out of resources somewhere or*  
> >>>>> *were left in an unstable state. This can be caused by not cleaning up*  
> >>>>> *properly. I've seen this happen on my own system but haven't been able*  
> >>>>> *to nail down the exact spark that triggered it. Since modifying*  
> >>>>> *PLAY.CMD, I haven't seen this happen, but I'm not convinced that the*  
> >>>>> *problem is gone yet.*  
>  
> >>>>>> *I'm curious as to how much digging others did to try and find the*  
> >>>>>> *kinds of TIMEFMT supported by PLAY.CMD. I know it supports "samples"*  
> >>>>>> *and "ms" (for milliseconds), but the default units are something like*  
> >>>>>> *15 samples (for 44.1 kHz audio). Haven't dug enough to understand*  
> >>>>>> *the origin of that choice. Any other time formats supported?*  
>  
> >>>>> *15 samples / 44100 samples/sec = .00034 secs of audio*  
>

sci.astro: Re: Enhanced PLAY.CMD with play list support

> >>>> *Seems like a strange choice indeed. It is approximately 1/100th of a*  
> >>>> *normal process time slice (32ms). That's a bit of a stretch, though.*  
>  
> >>>> *Sufficiently odd that I never use the default (other than to*  
> >>>> *investigate it), which means I always have to use the TIMEFMT keyword,*  
> >>>> *which is a bit annoying. On the other hand, my editing program works*  
> >>>> *with the audio samples in memory, so to play an edited clip using*  
> >>>> *PLAY.CMD, I need to write it to disk, so as long as I write just the*  
> >>>> *clip I want to hear, I don't have to bother with the FROM, TO, and*  
> >>>> *TIMEFMT keywords. Would be nicer to simply play a clip of audio*  
> >>>> *samples directly from memory and thus avoid hitting the disk every*  
> >>>> *time I want to repeat a short clip, but then I'd have to learn an*  
> >>>> *entirely new part of the multimedia system. One of the editing*  
> >>>> *tools plays a short clip whose end points I can move in 0.01 sec*  
> >>>> *increments using the cursor keys; it has to write a new .wav file*  
> >>>> *every time I press the key so as to utilize PLAY.CMD.*  
>  
> >>>> *The learning curve for the multimedia subsystem proper can be quite*  
> >>>> *steep. It would allow you the ability to play from memory, volume*  
> >>>> *control, and data format independence (you could use codecs to read and*  
> >>>> *write your raw audio data), but getting there is tricky.*  
>  
> >>>> *Reading .wav files is sufficiently simple that I can avoid the overhead*  
> >>>> *of a codec. On the other hand, handling every possible data type is a*  
> >>>> *headache for the source code. There are digital audio units becoming*  
> >>>> *available that can record 24-bit samples. Judging from the recording*  
> >>>> *time available, they're not wasting space by padding the samples to*  
> >>>> *32-bit. If I wanted to handle files like that, I'd need to rewrite*  
> >>>> *the code to read three 8-bit samples, because there is no native 24-bit*  
> >>>> *data type.*  
>  
> >>>> *I may be mistaken, but I believe some sound cards support 24-bit samples*  
> >>>> *in hardware (which could be why such WAV files might be appearing). I*  
> >>>> *don't know if the OS/2 codec supports such a thing. I'd imagine not.*  
>  
> >>>> *I expect to be finding out over the next few months if reviews of the*  
> >>>> *Edirol R-1 coax me into buying one.*

lol!!!! Wanker!

>  
> >>>> *But allowing the use of a codec could open up other possibilities, such*  
> >>>> *as MP3 or Ogg-Vorbis.*  
>  
> >>>> *How about extracting the audio stream from a video DVD?*

Which does about as much as pissing on your keyboard, bruce.

>  
> >>>> *In what language is your main project written?*  
>

sci.astro: Re: Enhanced PLAY.CMD with play list support

> >> *I rely heavily on a library of routines that I've written over the last  
> >> two decades, largely in Fortran. The Fourier transform is an example.  
> >> A lot of the "magic" my editor can perform (fixing wrong notes, for  
> >> example) uses the Fourier transform.*  
>  
> > *Understandable to hold on to such code. Certainly not the easiest thing  
> > to code accurately in C.*  
>  
> >>> *I ask because I'm  
> >>> trudging my way through the same territory right now in one of my C  
> >>> projects, and maybe I can provide you with a simplified code snippet to  
> >>> get the system going for you. I have DART and raw MMPM code from past  
> >>> projects, but they were muddled down in the mechanics of synchronization  
> >>> with a video stream, so they don't serve as good examples.*  
>  
> >> *They might make good examples to the WarpVision people. The only time  
> >> I tried WarpVision, I never could get decent synchronization with the  
> >> video, even with the adjustment keys provided.*  
>  
> > *I think the problem that Warpvision addresses is a problem of accuracy  
> > within the header information, as much as its own timing accuracy. Some  
> > video clips just plain have the wrong information for the playback speed  
> > of audio versus video. Other players can adjust usually by sync'ing up  
> > on "key" frames that occur periodically in the playback. At least the  
> > sync problem with emulation (the source of my code) is one where the  
> > sound has to be \*created\* at the right rate. The issue gets muddled  
> > when there are skipped frames, but it's generally an easier problem  
> > because it's more controllable.*  
>  
> >>> *Also, they  
> >>> only dealt in pumping raw data out to the sound card. My new code will  
> >>> need to be a bit more robust (being format-independent, for example).*  
>  
> >> *But API calls are API calls, regardless of the language used to call  
> >> them.*  
>  
> > *True. Translation from one language to another is relatively trivial  
> > when writing code based largely on API calls.*

irrelevant and illogical, bruce.