
Subject: Re: Reading Wav files

Posted by [Mike Alport](#) on Thu, 22 Aug 2002 08:21:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks guys - both your suggestions have worked. When I was first playing around with read_wav, I must have inadvertently made it a variable. An explicit ".compile read_wav" then seems override that earlier definition. I have now also learnt about STRICTARR.

Thanks again,

Mike

PS - now I just need to play .wav files from IDL - I see that Matt Feinstein asked the same question a few days ago wrt IDL_Tools.

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2002.08.21.20.24.41.192394.2937@as.arizona.edu...

> On Wed, 21 Aug 2002 09:08:35 -0700, Mike Alport wrote:

>

>> This seems a trivial problem: the IDL 5.5 documentation gives the
>> following syntax for reading sound .wav files: Result = READ_WAV (
>> Filename [, Rate]) So, when I try it, I get: IDL> Result = READ_WAV(
>> "ramsay0150.wav") % Variable is undefined: READ_WAV.

>>

>> Have they switched the calling procedure name on us? Thanks, Mike Alport

>

> This is probably related to the ()/[] degeneracy in IDL array indexing.

> I can replicate your error:

>

> IDL> result = read_wav('/usr/share/licq/sounds/fun/Auth.wav')

> % Compiled module: READ_WAV.

> IDL> read_wav=2

> IDL> result = read_wav('/usr/share/licq/sounds/fun/Auth.wav')

> IDL> help,result

> RESULT BYTE = Array[5400]

>

> No problem thus far.. but what if I start IDL again with:

>

> IDL> read_wav=1

> IDL> r=temporary(read_wav)

> IDL> help

> % At \$MAIN\$

> I INT = 167

> R INT = 1

> READ_WAV UNDEFINED = <Undefined>

>

> Uh, oh, an "undefined variable" READ_WAV exists. You can imagine what
> will happen now:

>
> IDL> result = read_wav('/usr/share/licq/sounds/fun/Auth.wav')
> % Variable is undefined: READ_WAV.
> % Execution halted at: \$MAIN\$
>
> IDL thinks you're trying to index an undefined variable! The problem
> is, variables sometimes exists as "undefined" zombies, even after you
> think they're gone. In this state, that can still "shadow" or overlap
> with function calls. A few options for solving the problem:
>
> 1. Find out where the undefined variable READ_WAV is getting created,
> and eliminate that.
>
> 2. Use "compile_opt STRICTARR" at the beginning of your procedure, or
> at the main level:
>
> IDL> compile_opt STRICTARR
> IDL> result = read_wav('/usr/share/licq/sounds/fun/Auth.wav')
> % Compiled module: READ_WAV.
>
> 3. Use "forward_function READ_WAV" to declare it a function.
>
> 4. Clamor for RSI to make STRICTARR the default (though duck as Craig
> throws an eraser at your head.)
>
> You could also just put "compile_opt STRICTARR" in your startup file,
> and I think it then applies to all routines and interactive commands
> (anybody know for sure?).
>
> Good luck,
>
> JD
