Subject: Re: IDL: from Sun to MsWindows
Posted by thompson on Thu, 03 Jun 1993 18:14:25 GMT
View Forum Message <> Reply to Message

knipp@ipi.uni-hannover.de (K. Knipp) writes:

> Hi,
> I've got two questions concerning the adaption of IDL-routines written on SunOs
> for MsWindows:

> - does anyone have a list of general hints to this problem, like
> ; avoid system-calls
> ; short (8 characters) routine-names
> ; ... ?

You can have routine names that are longer than eight characters. You just make the DOS name of the file the first eight characters of the routine. It is better not to have two or more routines which start with the first eight characters--however, you can deal with this too by simply concatenating the procedures with names that share the first eight characters into a single file. (If one procedure calls another, and their names start with the same eight characters, then the procedure being called should be first in the concatenated file). I will post a procedure file (CONCAT4DOS.PRO) that will automatically concatenate procedure files into a form that can then be simply copied to MsWindows/DOS machine.

I haven't had any luck so far adapting software that works through spawning OS commands, but maybe that will be easier in a future release of IDL for Windows.

Another thing to avoid is software that using all three mouse buttons (for example the routine ZOOM in the standard IDL user's library). Some PCs seem to be capable of supporting this, and some don't. It does appear, however, that one always has the left (!ERR=1) and right (!ERR=4) buttons available.

> - how can use non-printable characters under MsWindows, p.e.:
> ; print,'\007' ; (bell)
> ; print,'\x08' ; (move cursor left) ?
> MsWindows just prints ths string on the screen.

I haven't worried about ringing the bell, but I have come up with a solution for software that manipulates screen output. For example, consider a routine that continuously prints out the position of the cursor to the screen. (RDPIX in the standard IDL user's library is an example of such a routine. On Unix and VMS workstations this is accomplished with formatted output statements. However, this doesn't seem to work with IDL for Windows. The workaround I came up, and which I'm happy with, is to use a text widget instead of writing directly to the output log. I will also post an example (CRS.PRO) of how this

is done. (Note that in this example widgets are used without ever calling XMANAGER.)

Bill Thompson
