
Subject: Calling IMSL from IDL under Solaris
Posted by [cramblit](#) on Thu, 25 May 1995 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

I've put together a list of the things I needed to do in order to be able to call IMSL routines from C-code called from IDL, under the Solaris operating system on a Sun sparystation. Hopefully this will save someone else some effort.

1.) Since IDL and IMSL both use socket-based network code to manage their software licenses, and my IDL license file is in a non-standard location, I set the following environment path variable:

```
setenv LM_LICENSE_FILE $IDL_DIR/license/license.dat:/usr/local/flexlm/licenses/license.dat
```

where IDL_DIR is the directory in which IDL resides, and the second path is for the IMSL license file.

2.) I linked my C-routines into a shareable object library using

```
cc -i -Kpic -c file.c -x04
```

```
ld -i -dy -G -z defs -o lib.so files.o -L/usr/opt/lib -R/usr/opt/lib  
-limsl -lF77 -lM77 -lm -lsocket -lnsl  
-Y P,/opt/SUNWspro/SC2.0.1:/usr/ccs/lib:/usr/lib -lc
```

3.) Before calling an IMSL routine, I saved the disposition of the SIGALRM signal:

```
old_alarm=sigset(SIGALRM,new_alarm)
```

and restored it after calling IMSL:

```
old_alarm=sigset(SIGALRM,old_alarm)
```

This seems to prevent a curious SIGALRM signal from causing IDL to exit a few minutes after calling the IMSL routine. This suggests that the IMSL call is changing the handler of this signal and failing to restore it properly. (I have no idea how or why these signals are used by either product.)

Bob Cramblitt
cramblit@ecn.purdue.edu
