
Subject: Re: Aynchronous IO and Widgets
Posted by [rivers](#) on Thu, 21 Dec 1995 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <4b653h\$58p@mercurius.lu.se>, Struan Gray <struan.gray@sljus.lu.se> writes:

> I want to use IDL to control instruments in the lab but also want to avoid
> inflicting long waiting times on my users. I've read the manual and played
> with some simple widget building and recognise that IDL is not meant to be
> threaded, and in particular wants each routine to finish before starting the
> next one.
>
> The problem is I want to collect data from rather slow devices which
> buffer a few kB before sending it to the data-taking computer. I'm sure I
> could bodge something together using a seperate program and files on disk or
> ramdisk, but this would be slow and klunky on my platform and anyway I'd
> prefer to have something fully integrated.
>
> What I had in mind was to use a stub widget which would use CALL_EXTERNAL
> to call an asynchronous I/O routine written in C whose completion routine
> would then insert an event into the IDL event queue, telling the instrument
> control base widget that the data was ready to collect. The main program
> could get on with plotting and analysing earlier data until the transfer was
> complete.

I have not yet tried the stub widget method, inserting events into the IDL event queue, although it is on my list of things to do. However, you can emulate this behavior without too much trouble.

- Make one of your base widgets generate timer events to enter an IDL "polling" routine. This routine polls the C code (via CALL_EXTERNAL) several times per second to see if the device is ready. The C code need not actually poll the device if asynchronous I/O is supported - it can simply maintain a flag indicating that the device has asynchronously reported that data are available.
- If there is data ready your timer event routine can read it directly, or it can send an event to another IDL widget informing it that there is data ready.

I have used this method for many projects of exactly the type you describe.

Mark Rivers	(312) 702-2279 (office)
CARS	(312) 702-9951 (secretary)
Univ. of Chicago	(312) 702-5454 (FAX)
5640 S. Ellis Ave.	(708) 922-0499 (home)
Chicago, IL 60637	rivers@cars3.uchicago.edu (Internet)
