Subject: Re: Socket server in > IDL 6.3
Posted by rtowler on Fri, 27 Jul 2007 10:55:09 GMT
View Forum Message <> Reply to Message

On Jul 27, 12:23 am, Robbie <ret...> wrote:
> I've been mucking about with the undocumented feature of SOCKET, /
> LISTEN in IDL.
>
> Copying the synopsis from a previous post last year
<snip>
>
> I've found that FILE_POLL_INPUT is absolutely essential in setting up
> a socket server. FILE_POLL_INPUT
> has difficulties when you mix listening and connected LUNs, although
> you can check each separately by using the TIMEOUT=0 keyword.
>
> I've come across a difficult bug. I want to read at least 100K chunks
> of data over the incoming stream. READU produces a runtime error if
> the operation blocks. The server seems to work for a while, and then
> it suddenly stops and produces the "OPERATION BLOCKS" error. If I exit
> from IDL and come back in, the server seems to work ok again. When the
> error occurs the (FSTAT(lun)).TRANSFER_COUNT is set to something like
> 630k and this is the exact same number every time the error occurs
> regardless of the size of the chunks of data.
>
> Has anyone else had this experience?
>
> Robbie
>
> http://barnett.id.au/idl

Have you tried Randall Frank's TCP socket library? It's part of his idl_tools package available from Ronn Kling's website. It implements both client and server sockets and you can easily compile it separately if you don't want to package the whole library with your application. FWIW, I modified it for UDP sockets and it works great.

-Rick

Subject: Re: Socket server in > IDL 6.3
Posted by Robbie on Sat, 28 Jul 2007 02:08:18 GMT
View Forum Message <> Reply to Message

Thanks for the reply,

I've had a brief look at it a few times. I think that I'm conditioned to the use of writeu and readu and I'm a little hesitant about sendvar

and recvvar.

I've been persisting object states as structures. I was using readu/ writeu to implement a messaging protocol for this.

I was looking at writing an initial implementation of my server in IDL and then porting it Python once I had a stable messaging protocol.

I'm actually planning on writing a server which does object persistence with the backing store as a mixture of the server filesystem and a database.

Robbie
