
Subject: Non-blocking socket

Posted by [Helder Marchetto](#) on Tue, 05 Nov 2013 22:50:53 GMT

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

Hi,

well, working on sockets is not my thing, but I can't get around the fact that IDL "waits" for an answer and this is not the case elsewhere.

Let me explain...

I open a socket this way:

```
SOCKET, Unit, IP, Port, $  
    CONNECT_TIMEOUT=Connect_Timeout, $  
    READ_TIMEOUT=Read_Timeout, $  
    WRITE_TIMEOUT=self.Write_Timeout, $  
    Error=RetError, /GET_LUN
```

The IP is a local one and points to another software on the pc (Win 7-64 bits) that acts as socket server. The timeouts are set to 5, 1 and 1 seconds for connection, read and write, respectively.

Now when I attempt to read a simple string (a device name) in a loop of 100, it takes roughly 100 seconds.

To read I use something like:

```
Value = "  
WRITEU, Unit, 'cmd '+STRTRIM(I,2)  
READF, Unit, Value
```

I could not get around this by changing the timeouts or removing them (even worst!). So, what I did is take some time to learn to do this in another language and find out if the socket server is slowing me down or the way the socket is implemented in IDL. Well I tried this in python and after quite some pains and fiddling with a language I don't know about, I managed to get the job done in 0.007 seconds (1.5 min in IDL).

Now I know very well, that the problem is in the way the socket command is implemented... is there any way to get around this without "waiting" in IDL and read directly byte/char data in while loops?

Maybe not an elegant solution, but that's how I see it done elsewhere.

Many thanks,
Helder
