
Subject: Re: running IDL as a 'sleeping' process
Posted by [Russell\[1\]](#) on Wed, 04 Jan 2012 21:14:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Jan 4, 10:21 am, Paulo Penteado <pp.pente...@gmail.com> wrote:

> On Jan 3, 5:48 pm, desertdryad <dry...@gmail.com> wrote:

>

>> Is this sort of daily repeating process in the realm of possibility
>> for IDL to do? Should I be thinking of some sort of 'wrapper' program
>> that calls IDL to do this? I know just enough C/C++ to get in
>> trouble :) but not much of anything like Java and the like. Comments
>> and suggestions welcome. Thanks!

>

> One simple and IDL-only (thus platform-independent) way to do it is
> use IDL's wait procedure. The program can do its thing, then call
> systime() to find out how long to sleep, then call wait to sleep for
> the desired time.

This is true, but this will mean you have an IDL prompt sitting there with no access to it. It will essentially be "stuck" on the wait statement for a very long time, and you may not want it to start at some preset time. Instead, a (perhaps more elegant) way along these lines, would be to use the widget_timer functionality. What amounts to the wait command with a long wait time derived from systime() would then be enveloped in an event handler that would trigger at some defined time. That could be simple (such as every 24 hours trigger), or you could test the existence of some auxiliary file (say every 30s or so) and when the file exists, then trigger the main code. This is probably a lot more work than you signed up for, but really not that bad once you get the hang of it. The virtue of this is that you get the IDL prompt back, and this thing essentially runs "in the background".

Russell
