

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

--

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

Page 1 of 4 ---- Generated from [comp.lang.idl-pvwave](#) archive

>
> On occasion, someone needs to do a quick, one-off, thing in IDL and they
> grab my one license.
>
> While they have the license, my batch job decides "It's time to run"
> (determined by an external source sending a mail message, I've got no
> control over when it happens). Of course, since the other user has the
> license, my job starts up in "timed demo" mode. If my job could finish
> in 7 minutes, I wouldn't care, but it takes several hours so I do care
> :-).
>
> Had my "mail initiated" job been run interactively, I'd be asked, "do
> you want to wait for a license" and I've I'd answered "YES" I'd be put
> "on hold" until the other person finished. Is there a way to achieve
> this "wait until the license is available" in a "batch job stream",
> perhaps with some command line option I'm not aware of and/or an
> environment variable?
>
> Thanks!
> Bruce
>
> PS this is IDL 5.2 running on Solaris 2.7.
> PPS Buying a second user license isn't an option :-(

Try expect, a package which allows you "programmed dialogue with interactive programs". You could expect "IDL> " and respond with "@batchfile.pro" or expect the 'wait for a license' warning and respond with "Y" then expect "IDL> " and respond with "@batchfile.pro". You would want to disable the timeout on the latter, since the time spent waiting is unknown.

Another option... always use idl as a specific user (e.g. idl_user). Everyone who wanted to run idl would run as this user, since the licensing scheme lets a given user run multiple copies of idl on a given screen. Your script would first switch to that user (easy if running as root), and anybody else who would want to use idl would be required to switch to that user. To enforce this, let idl_user own the binaries in \$IDL_DIR/bin/bin.solaris, and make them owner executeable only. idl_user need not have a home directory, since it's there for licensing permissions only.

A temptingly close easter egg is found in the binary: the option -autow. Doesn't seem to do what you wanted though -- too bad.

Good luck,

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|

Subject: Re: Too many users...

Posted by [Bruce Bowler](#) on Thu, 15 Jul 1999 07:00:00 GMT

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

"J.D. Smith" wrote:

>

> Bruce Bowler wrote:

>>

>> Suppose I have a node-locked 1 user license. 99.44% of the time, this
>> is fine for my needs (it's a fairly "dedicated" system, running "mostly"
>> one application in batch, most of the time).

>>

>> On occasion, someone needs to do a quick, one-off, thing in IDL and they
>> grab my one license.

>>

>> While they have the license, my batch job decides "It's time to run"
>> (determined by an external source sending a mail message, I've got no
>> control over when it happens). Of course, since the other user has the
>> license, my job starts up in "timed demo" mode. If my job could finish
>> in 7 minutes, I wouldn't care, but it takes several hours so I do care
>> :-).

>>

>> Had my "mail initiated" job been run interactively, I'd be asked, "do
>> you want to wait for a license" and I've I'd answered "YES" I'd be put
>> "on hold" until the other person finished. Is there a way to achieve
>> this "wait until the license is available" in a "batch job stream",
>> perhaps with some command line option I'm not aware of and/or an
>> environment variable?

>>

>> Thanks!

>> Bruce

>>

>> PS this is IDL 5.2 running on Solaris 2.7.

>> PPS Buying a second user license isn't an option :-)

>

> Try expect, a package which allows you "programmed dialogue with
> interactive programs". You could expect "IDL> " and respond with
> "@batchfile.pro" or expect the 'wait for a license' warning and respond
> with "Y" then expect "IDL> " and respond with "@batchfile.pro". You
> would want to disable the timeout on the latter, since the time spent

> waiting is unknown.

An interesting idea, but yet one more piece of software to maintain...

> Another option... always use idl as a specific user (e.g. idl_user).
> Everyone who wanted to run idl would run as this user, since the
> licensing scheme lets a given user run multiple copies of idl on a given
> screen. Your script would first switch to that user (easy if running as
> root), and anybody else who would want to use idl would be required to
> switch to that user. To enforce this, let idl_user own the binaries in
> \$IDL_DIR/bin/bin.solaris, and make them owner executeable only.
> idl_user need not have a home directory, since it's there for licensing
> permissions only.

While an intriguing idea, and probably "legal" within the "letter of the law", I'd say it violates the "spirit of the law". That's just my opinion though...

> A temptingly close easter egg is found in the binary: the option
> -autow. Doesn't seem to do what you wanted though -- too bad.

Hadn't thought of looking for easter eggs :-)

What I ended up with, following the email suggestion of an RSI employee (who shall remain nameless since I can't find his email address :-)) is to parse the output of `lmstat -A` in the shell script that I run. If that shows that the license is in use, I wait awhile and try again. The specific code I use (in case anyone else is interested) is

```
while (`/usr/local/rsi/idl/bin/lmstat -A | grep nodelocked | wc  
-m`)  
    sleep 600  
end
```

Bruce
