Subject: openr and /get_lun
Posted by Craig Markwardt on Fri, 14 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I have noticed that the use of /GET_LUN and ERROR keywords to openr is
not as helpful as I would have hoped. Do other have this experience?
The problem is that when an error occurs, it is hard to know whether
the file unit was "gotten" or not.

For example:

```
pro test1
  openr, unit, filename, /get_lun, error=err
  free_lun, unit
end
```

If there was an error, then it is possible that UNIT was never set,
and is hence undefined. FREE_LUN doesn't take undefined variables.

If there is error checking to do, I don't know exactly what it should
be. So I find myself explicitly doing this instead:

```
pro test2
  get_lun, unit
  openr, unit, filename, error=err
  free_lun, unit
end
```

Comments?

Craig

--
 --- --------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 --- --------------

Subject: Re: openr and /get_lun
Posted by John-David T. Smith on Mon, 17 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

"Robert S. Mallozzi" wrote:
>
> In article <38FB4B75.936477C5@astro.cornell.edu>,
>   "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>> "Robert S. Mallozzi" wrote:

>>>
>>> I sure wish we had a boolean datatype - the mistake of
>>> using something like "IF (NOT error) THEN" is one that
>>> is really a pain to find, although it certainly makes
>>> your code much more readable.
>>
>> We don't need a boolean data type... we need IF to examine not just the first
>> bit of the value, but the whole thing, and use C's 0=false, anything else =true
>> paradigm. Here's hoping.
>>
>> if NOT 2 then print,"this isn't right!"
>
>
> This would certainly break backward compatibility - there
> has to be someone, somewhere that relies on the fact that in
> IDL, odd = true and even = false ! I feel as you do that this
> was a design mistake made a long time ago, in a programmer's
> mind far, far away...

Anyone who thinks 2 is false deserves to have his programs broken.

Subject: Re: openr and /get_lun
Posted by mallors on Mon, 17 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

In article <38FB4B75.936477C5@astro.cornell.edu>,
 "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
> "Robert S. Mallozzi" wrote:
>>
>> I sure wish we had a boolean datatype - the mistake of
>> using something like "IF (NOT error) THEN" is one that
>> is really a pain to find, although it certainly makes
>> your code much more readable.
>
> We don't need a boolean data type... we need IF to examine not just the first
> bit of the value, but the whole thing, and use C's 0=false, anything else =true
> paradigm. Here's hoping.
>
> if NOT 2 then print,"this isn't right!"

This would certainly break backward compatibility - there
has to be someone, somewhere that relies on the fact that in
IDL, odd = true and even = false ! I feel as you do that this
was a design mistake made a long time ago, in a programmer's
mind far, far away...

Regards,

-bob

--
```
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~
Robert S. Mallozzi                                   256-544-0887
                                                Mail Code SD 50
http://gammaray.msfc.nasa.gov/          Marshall Space Flight Center
http://cspar.uah.edu/~mallozzir/             Huntsville, AL 35812
 ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ ~~~~~~~~~~
```

Subject: Re: openr and /get_lun
Posted by mallors on Mon, 17 Apr 2000 07:00:00 GMT

View Forum Message <> Reply to Message

In article <on66tglhxl.fsf@cow.physics.wisc.edu>,
 Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:
>
> I totally agree that the error condition must be handled. What I was
> getting at is that OPENR, ..., /GET_LUN does two things: allocate a
> LUN, and open a file. If you get an ERROR condition back, it's
> impossible to know which of these two things failed. In fact in the
> above example you gave, the unit FL may be undefined, so FREE_LUN will
> fail.

You can tell which of these things failed based on the
name of the error that is returned in the !ERROR_STATE
structure. The documentation states that although the
error code number (a negative integer) can change across IDL
sessions, the error name cannot.

```
  ;== The program TEST.PRO tries to open many files,
  ;== each with a different unit number, and stops
  ;== if there is an error

  IDL> .RUN TEST
  % Stop encountered:  $MAIN$
  IDL> PRINT, error
       -234
  IDL> HELP, !ERROR_STATE.NAME
  <Expression>    STRING    = 'IDL_M_FILE_NOLUNLEFT'
  IDL> PRINT, !ERROR_STATE.MSG
  OPENW: All available logical units are currently in use.
  IDL>
```

```
 IDL> CLOSE, /ALL
 IDL>
 IDL> OPENR, fl, 'nofile', /GET_LUN, ERROR = error
 IDL> PRINT, error
      -222
 IDL> PRINT, !ERROR_STATE.NAME
 <Expression>    STRING    = 'IDL_M_CNTOPNFIL'
 IDL> PRINT, !ERROR_STATE.MSG
 OPENR: Error opening file. Unit: 100, File: nofile
```

Regards,

-bob

--
~~~~~~ ~~~~~~~~~
Robert S. Mallozzi   256-544-0887
                                        Mail Code SD 50
http://gammaray.msfc.nasa.gov/   Marshall Space Flight Center
http://cspar.uah.edu/~mallozzir/   Huntsville, AL 35812
~~~~~~ ~~~~~~~~~

Subject: Re: openr and /get_lun
Posted by Craig Markwardt on Mon, 17 Apr 2000 07:00:00 GMT

View Forum Message <> Reply to Message

mallors@ips1.msfc.nasa.gov (Robert S. Mallozzi) writes:

```
>  In article <onitxkz7p7.fsf@cow.physics.wisc.edu>,
>   Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:
>>
>>  I have noticed that the use of /GET_LUN and ERROR keywords to openr is
>>  not as helpful as I would have hoped.  Do other have this experience?
>>  The problem is that when an error occurs, it is hard to know whether
>>  the file unit was "gotten" or not.
...
>
>
>  I guess I never thought about it too much, because if
>  there is an error with the OPEN, then I should handle
>  it somehow:
>
>     OPENR, fl, 'nofile', /GET_LUN, ERROR = error
>     IF (error NE 0) THEN BEGIN
>        MESSAGE, /CONTINUE, 'Could not open file.'
>        RETURN
```

```
>     ENDIF
>       .
>       .
>       .
>     FREE_LUN, fl
>
>
```

> Otherwise, if you don't want to handle the error, you can just
> free the unit number conditionally, as I am sure you know:

```
>
>     IF (error EQ 0) THEN FREE_LUN, fl
```

I totally agree that the error condition must be handled. What I was
getting at is that OPENR, ..., /GET_LUN does two things: allocate a
LUN, and open a file. If you get an ERROR condition back, it's
impossible to know which of these two things failed. In fact in the
above example you gave, the unit FL may be undefined, so FREE_LUN will
fail.

David suggests using N_ELEMENTS(FL) to see if it's defined. That
works, but only if that's the first time I use FL, something I didn't
point out in my original example.

As the error checking got more detailed, I realized that it's easier
and takes less code to decouple the GET_LUN from the OPEN. Hence,

```
  GET_LUN, fl
  OPENR, fl, file, ERROR=err
  IF error NE 0 then <Handle error>
  FREE_LUN, fl
```

is guaranteed to work since FL is always defined.

> I sure wish we had a boolean datatype - the mistake of
> using something like "IF (NOT error) THEN" is one that
> is really a pain to find, although it certainly makes
> your code much more readable.

I agree with you there. OR, do we need boolean operators instead?
For example, a BNOT operator which takes the "logical" NOT instead of
the bitwise NOT,

```
  NOT 0   -> 255      BNOT 0 -> 1
  NOT 1   -> 254      BNOT 1 -> 0
```

Craig

--

```
 ------------------------------------------------------------ --------------
Craig B. Markwardt, Ph.D.         EMAIL:    craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 ------------------------------------------------------------ --------------
```

Subject: Re: openr and /get_lun
Posted by John-David T. Smith on Mon, 17 Apr 2000 07:00:00 GMT

View Forum Message <> Reply to Message

```
"Robert S. Mallozzi" wrote:
>
>  In article <onitxkz7p7.fsf@cow.physics.wisc.edu>,
>         Craig Markwardt <craigmnet@cow.physics.wisc.edu> writes:
>>
>>  I have noticed that the use of /GET_LUN and ERROR keywords to openr is
>>  not as helpful as I would have hoped.  Do other have this experience?
>>  The problem is that when an error occurs, it is hard to know whether
>>  the file unit was "gotten" or not.
>>
>>  For example:
>>
>>  pro test1
>>    openr, unit, filename, /get_lun, error=err
>>    free_lun, unit
>>  end
>>
>>  If there was an error, then it is possible that UNIT was never set,
>>  and is hence undefined.  FREE_LUN doesn't take undefined variables.
>
>  I guess I never thought about it too much, because if
>  there is an error with the OPEN, then I should handle
>  it somehow:
>
>     OPENR, fl, 'nofile', /GET_LUN, ERROR = error
>     IF (error NE 0) THEN BEGIN
>        MESSAGE, /CONTINUE, 'Could not open file.'
>        RETURN
>     ENDIF
>       .
>       .
>       .
>     FREE_LUN, fl
>
>  Otherwise, if you don't want to handle the error, you can just
>  free the unit number conditionally, as I am sure you know:
>
>     IF (error EQ 0) THEN FREE_LUN, fl
```

>
>
> I sure wish we had a boolean datatype - the mistake of
> using something like "IF (NOT error) THEN" is one that
> is really a pain to find, although it certainly makes
> your code much more readable.

We don't need a boolean data type... we need IF to examine not just the first bit of the value, but the whole thing, and use C's 0=false, anything else =true paradigm. Here's hoping.

if NOT 2 then print,"this isn't right!"

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: openr and /get_lun
Posted by Joseph B. Gurman on Thu, 20 Apr 2000 07:00:00 GMT
View Forum Message <> Reply to Message

In article <8dfqbo$ep1$2@hammer.msfc.nasa.gov>,
mallors@ips1.msfc.nasa.gov (Robert S. Mallozzi) wrote:

> In article <38FB4B75.936477C5@astro.cornell.edu>,
> "J.D. Smith" <jdsmith@astro.cornell.edu> writes:
>> "Robert S. Mallozzi" wrote:
>>>
>>> I sure wish we had a boolean datatype - the mistake of
>>> using something like "IF (NOT error) THEN" is one that
>>> is really a pain to find, although it certainly makes
>>> your code much more readable.
>>
>> We don't need a boolean data type... we need IF to examine not just the
>> first
>> bit of the value, but the whole thing, and use C's 0=false, anything
>> else =true
>> paradigm. Here's hoping.
>>
>> if NOT 2 then print,"this isn't right!"
>
>

> This would certainly break backward compatibility - there
> has to be someone, somewhere that relies on the fact that in
> IDL, odd = true and even = false ! I feel as you do that this
> was a design mistake made a long time ago, in a programmer's
> mind far, far away...
>
> Regards,
>
> -bob

One man's mistake is another's feature (or something like that).

The "low bit 0 = false, low bit = 1 true" convention is from VMS (way back in the pre-Alpha days, even.... what did they call those things, VAXen? VAXes?), with the more significant bits yielding addition information on the specific error or (in the case of oddness) warning, &c.

No doubt due to operant conditioning programming VAX system services, I find this convention more useful than C's convention.

Chacun a son error convention....

Joe Gurman

--
Joseph B. Gurman / NASA Goddard Space Flight Center / Solar Physics Branch /
Greenbelt MD 20771 / work: gurman@gsfc.nasa.gov /other: gurman@ari.net

Government employees are still not allowed to hold opinions while at work,
so any opinions expressed herein must be someone else's.