
Subject: Re: IDL_IDLBridge and SHMMAP - shared memory not found

Posted by [Jim Pendleton](#) on Tue, 06 Mar 2012 14:37:41 GMT

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

On Tuesday, March 6, 2012 2:41:28 AM UTC-7, Percy Pugwash wrote:

> I'm trying to do some processing in "in the background" using the IDL_IDLBridge object. I want to avoid making unnecessary copies of all the data, and also gain the speed benefits mentioned here: <https://groups.google.com/d/topic/comp.lang.idl-pvwave/fRXH-IDbdKc/discussion>

>

> So, I am using the shmmap and shmvar functionality to share the data.

>

> shmmap,size=size(image),get_name=shmname

> var = shmvar(shmname)

> var[0,0] = image

>

> The IDL_IDLBridge object seems to be working OK

>

> bridge->setvar,'nlevels',nlevels

> bridge->setvar,'shmname',shmname

> bridge->execute,'makePyramid,nlevels,shmname=shmname'

>

> The problem seems to be that the shmmaped memory does not seem to be visible from the child IDL process. When I get run the execute method of bridge as above, I get the following error:

>

> IDL_IDLBRIDGE Error: SHMVAR: Shared Memory Segment not found: IDL_SHM_700_0

>

> However, I can see using help,/shared_memory (while the IDE is paused in debug mode) that the sought-after shared memory segment does indeed exist.

>

> IDL> help,/shared_memory

> IDL_SHM_700_0 BYTE = <WindowsAnonymous(IDL_SHM_700_0), Offset(0), Refcnt(0)>
Array[5120, 5696]

>

> Why can't the child process see the shared memory?

>

> I am using 64-bit IDL on Windows XP.

>

> P

One key here, depending on your platform and the type of shared memory you're creating, is the not-so-obvious `os_handle` keyword to `shmmap`. Once you find a pattern of `os_handle` and `segmentname` parameters that works for you, you tend not to have to think about it again.

In IDL process 1:

```
pro shared1
```

```
shmmap, 'firstsegment', dimension = 5, template = {widget_draw}, $
```

```
os_handle = 'percy'
```

```
v = shmvar('firstsegment')
v[0].modifiers = 1
help,/str,v[0]
end
```

In IDL process 2:

```
pro shared2
shmmap, template = {widget_draw}, get_name = gn, $
  os_handle = 'percy', dimension = 5
pv = ptr_new(shmvar(gn))
help,/str>(*pv)[0]
print, gn
end
```

Jim P.

Subject: Re: IDL_IDLBridge and SHMMAP - shared memory not found
Posted by [Percy Pugwash](#) on Mon, 12 Mar 2012 17:22:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tuesday, 6 March 2012 14:37:41 UTC, jimmylee...@gmail.com wrote:
> On Tuesday, March 6, 2012 2:41:28 AM UTC-7, Percy Pugwash wrote:
>> I'm trying to do some processing in "in the background" using the IDL_IDLBridge object. I want to avoid making unnecessary copies of all the data, and also gain the speed benefits mentioned here: <https://groups.google.com/d/topic/comp.lang.idl-pvwave/fRXH-IDbdKc/discussion>
>>
>> So, I am using the shmmap and shmvar functionality to share the data.
>>
>> shmmap,size=size(image),get_name=shmname
>> var = shmvar(shmname)
>> var[0,0] = image
>>
>> The IDL_IDLBridge object seems to be working OK
>>
>> bridge->setvar,'nlevels',nlevels
>> bridge->setvar,'shmname',shmname
>> bridge->execute,'makePyramid,nlevels,shmname=shmname'
>>
>> The problem seems to be that the shmmaped memory does not seem to be visible from the child IDL process. When I get run the execute method of bridge as above, I get the following error:
>>
>> IDL_IDLBRIDGE Error: SHMVAR: Shared Memory Segment not found: IDL_SHM_700_0
>>
>> However, I can see using help,/shared_memory (while the IDE is paused in debug mode) that the sought-after shared memory segment does indeed exist.
>>

```

>> IDL> help,/shared_memory
>> IDL_SHM_700_0 BYTE = <WindowsAnonymous(IDL_SHM_700_0), Offset(0), Refcnt(0)>
Array[5120, 5696]
>>
>> Why can't the child process see the shared memory?
>>
>> I am using 64-bit IDL on Windows XP.
>>
>> P
>
> One key here, depending on your platform and the type of shared memory you're creating, is
the not-so-obvious os_handle keyword to shmmap. Once you find a pattern of os_handle and
segmentname parameters that works for you, you tend not to have to think about it again.
>
> In IDL process 1:
>
> pro shared1
> shmmap, 'firstsegment', dimension = 5, template = {widget_draw}, $
>   os_handle = 'percy'
> v = shmvar('firstsegment')
> v[0].modifiers = 1
> help,/str,v[0]
> end
>
> In IDL process 2:
>
> pro shared2
> shmmap, template = {widget_draw}, get_name = gn, $
>   os_handle = 'percy', dimension = 5
> pv = ptr_new(shmvar(gn))
> help,/str>(*pv)[0]
> print, gn
> end
>
> Jim P.

```

Thanks Jim. Turned out my problem was much more fundamental however; I had thought that memory mapped with SHMMAP would be visible to child processes using only SHMVAR. When I realised I need to use SHMMAP in both the parent and child processes (thanks to your example), it worked a charm.

P