Subject: Re: Porting to VM
Posted by David Fanning on Thu, 11 Dec 2003 03:19:59 GMT
View Forum Message <> Reply to Message

JD Smith writes:

> I've been trying to get a large application working in the new IDLVM, and
> have run into trouble: apparently, in all calls like:
>
> XManager, /NO_BLOCK
>
> the NO_BLOCK is ignored, and the call blocks anyway. I suppose this makes
> sense, since IDL's non-blocking functionality is provided by the
> command-line, which isn't there in the VM. Why would you care about
> blocking if there's no command line to interact with? For my purposes,
> the distinction between blocking and non-blocking is whether calls which
> invoke XManager return immediately, or wait until the managed widget
> dies. This isn't mentioned in the "Restrictions" on the VM FAQ page.
> Anyone managed to deal with this issue?

In SAVE files I've made recently, I've had to add
JUST_REG keywords to all the XMANAGER calls before the
last one, which will block and get everything else
going.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Porting to VM
Posted by JD Smith on Thu, 11 Dec 2003 16:55:44 GMT
View Forum Message <> Reply to Message

On Wed, 10 Dec 2003 20:19:59 -0700, David Fanning wrote:

> JD Smith writes:
>
>> I've been trying to get a large application working in the new IDLVM,
>> and have run into trouble: apparently, in all calls like:
>>

>> XManager, /NO_BLOCK
>>
>> the NO_BLOCK is ignored, and the call blocks anyway. I suppose this
>> makes sense, since IDL's non-blocking functionality is provided by the
>> command-line, which isn't there in the VM. Why would you care about
>> blocking if there's no command line to interact with? For my purposes,
>> the distinction between blocking and non-blocking is whether calls
>> which invoke XManager return immediately, or wait until the managed
>> widget dies. This isn't mentioned in the "Restrictions" on the VM FAQ
>> page. Anyone managed to deal with this issue?
>
> In SAVE files I've made recently, I've had to add JUST_REG keywords to
> all the XMANAGER calls before the last one, which will block and get
> everything else going.

I heard from an RSI engineer who directed me to a mention of this
issue in the manual, hidden in a note under XManager:

    NO_BLOCK is ignored by IDL Runtime. If a main procedure uses
    XMANAGER with the NO_BLOCK keyword set, IDL Runtime defers
    subsequent processing of the commands following the XMANAGER
    call until the widget associated with the call to XMANAGER is
    destroyed.

What this doesn't say is that *subsequent*, deeper calls to
XManager,/NO_BLOCK will be respected; it's on the "main-level" call
which otherwise would have yielded to a non-existent active command
line which is ignored. The moral is, for full VM/Runtime portability,
don't rely on anything which happens *after* the very first call (or,
more correctly, any top-level call) to XManager executing right away.
Hopefully this can get added to the IDLVM FAQ.

JD

Subject: Re: Porting to VM
Posted by JD Smith on Thu, 11 Dec 2003 19:31:56 GMT
View Forum Message <> Reply to Message

On Thu, 11 Dec 2003 09:55:44 -0700, JD Smith wrote:

> On Wed, 10 Dec 2003 20:19:59 -0700, David Fanning wrote:
>
>> JD Smith writes:
>>
>>> I've been trying to get a large application working in the new IDLVM,
>>> and have run into trouble: apparently, in all calls like:
>>>

>>> XManager, /NO_BLOCK
>>>
>>> the NO_BLOCK is ignored, and the call blocks anyway. I suppose this
>>> makes sense, since IDL's non-blocking functionality is provided by the
>>> command-line, which isn't there in the VM. Why would you care about
>>> blocking if there's no command line to interact with? For my
>>> purposes, the distinction between blocking and non-blocking is whether
>>> calls which invoke XManager return immediately, or wait until the
>>> managed widget dies. This isn't mentioned in the "Restrictions" on
>>> the VM FAQ page. Anyone managed to deal with this issue?
>>
>> In SAVE files I've made recently, I've had to add JUST_REG keywords to
>> all the XMANAGER calls before the last one, which will block and get
>> everything else going.
>
> I heard from an RSI engineer who directed me to a mention of this issue
> in the manual, hidden in a note under XManager:
>
> NO_BLOCK is ignored by IDL Runtime. If a main procedure uses
> XMANAGER with the NO_BLOCK keyword set, IDL Runtime defers
> subsequent processing of the commands following the XMANAGER
> call until the widget associated with the call to XMANAGER is
> destroyed.
>
> What this doesn't say is that *subsequent*, deeper calls to
> XManager,/NO_BLOCK will be respected; it's on the "main-level" call
> which otherwise would have yielded to a non-existent active command line
> which is ignored. The moral is, for full VM/Runtime portability, don't
> rely on anything which happens *after* the very first call (or, more
> correctly, any top-level call) to XManager executing right away.
> Hopefully this can get added to the IDLVM FAQ.

As I have discovered, there is a corollary to this statement. While the top-level calls to XManager in the VM always block, it appears nested lower-level calls *never* block, even if you omit the NO_BLOCK keyword (or explicitly set it to NO_BLOCK=0); IDL runtime/VM essentially ignore NO_BLOCK, in either state.

This means, to get blocking in the VM, you'll have to use the /MODAL keyword with GROUP_LEADER in the widget_base of the widget being displayed. If specifying a GROUP_LEADER won't work for you (because you don't have one), you can use the obsolete /MODAL keyword to XManager, but you'll have to endure warnings about the use of obsolete features causing global decline, hair loss, and infirmity of spirit.
Another option is making a hidden (unrealized) base as a group leader.
Ugly, yes, but supported.

JD

Subject: Re: Porting to VM
Posted by David Fanning on Thu, 11 Dec 2003 19:53:13 GMT
View Forum Message <> Reply to Message

JD Smith writes:

> Another option is making a hidden (unrealized) base as a group leader.
> Ugly, yes, but supported.

The real problem with this solution, on Windows anyway, is that your modal widget doesn't get a little window icon of its own in the task bar down below where the application icons are stored. So, if your modal window happens to get *behind* one of the other application windows (happens more often than you would think!) the chances of you finding it are almost non-existent. Most people just swear at the non-responding application for two minutes, then hit Control-Alt-Delete. :-)

Cheers,

David

P.S. Let's just say I've noticed a lot of RSI programmers are fond of this solution. :-)

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155