Subject: Re: Pointer Behavior Objects Vs Plain routines?
Posted by Pavel A. Romashkin on Wed, 11 Sep 2002 17:19:42 GMT
View Forum Message <> Reply to Message

savoie@nsidc.org wrote:
>
> *Morning coffee hits* Aha, but I'm doing is defining what it points to _the
> first time_ with a copy. This changes the copy, making it a valid pointer,
> leaving my original pointer alone. Duh.
>
> But if the pointer is already valid, I should be able to dereference the copy
> and store whatever I want, blissfully ignorant of what it was pointing at
> thanks to the magic of IDL pointers.

Precisely. If passed pointer is not valid, the copy means nothing, too,
and is not returned back to the caller. If it is valid, changing the
copy will change the original. You need no objects to illustrate this.
Make a tiny change to your code:

```
*******
PRO CHANGEPTR, passedPtr
on_error, 1 ; Needed because will err on null ptr
  *passedPtr =  'This is weird?'
END
*******
IDL> a = {a:ptr_new(), b:ptr_new(/alloc)}
IDL> changeptr, a.a
% Unable to dereference NULL pointer: PASSEDPTR.
% Error occurred at: CHANGEPTR           7
%                   $MAIN$
% Execution halted at: $MAIN$
IDL> changeptr, a.b
IDL> print, *a.a
% Unable to dereference NULL pointer: <POINTER  (<NullPointer>)>.
% Execution halted at: $MAIN$
IDL> print, *a.b
This is weird?
```

Works just as you described.
Cheers,
Pavel