Subject: Re: Structure field concatenation
Posted by davidf on Thu, 07 Sep 2000 14:31:54 GMT
View Forum Message <> Reply to Message

Ben Tupper (btupper@bigelow.org) writes:

> Martin Schultz wrote:
>
>>
>> If you want to
>> point to an undefined variable (well, who wants to do this anyway),
>>
>
> Hello,
>
> I hate to pipe up because I'm the gear-slipping dope that may have started this mess
> (although I'm learning much just listening.) I find utility in a pointer to an
> undefined variable useful when working with lists of things that a user can
> completely empty. (Like a base map with or without any number of overlays, or the
> datasets have not been loaded yet.) Isn't it analogous to a container object that is
> waiting for additions?

Maybe Ben *used* to be a "gear-slipping dope", but
that hardly describes him lately. Let's just say I've
been pushed over my knowledge horizon more times than
I like to admit in the past couple of months by his
thought-provoking questions. :-)

In any case, I'd like to provide more support for the
utility of pointers to undefined variables. Let me give
you a specific example: CW_FIELD.

If you set CW_FIELD up so that it will accept, say,
integer values, and you happen to leave the field
blank, then when you go and get the value in the field
it will return a 0 to you. What's wrong with that?, you
say.

What is wrong is that a 0 is a valid integer value.
So you go willy-nilly on with your code thinking that
you have got something decent. But suppose the number
were the X Size of an image. And now suppose you want
to Congrid your image into this size:

```
displayImage = Congrid(image, xsize, ysize)
```

This causes an error. But now the error message is
very strange (probably impossible to understand if

you don't have a lot of IDL experience) and is one
step (and many lines of code) removed from where the
error really occurred.

If CW_FIELD had returned an undefined variable (well,
what else would it be if the field was blank?), then
the error would have been something that is easily
understood. What is more, it is something that could
be easily checked for:

```
Widget_Control, fieldID, Get_Value=theValue
IF Size(theValue, /TName) EQ 'UNDEFINED' THEN $
   Message, 'Whoops. Field is blank. Try again.'
```

You could argue that you could as easily check to see
if the size is 0, and you would probably be right,
except in those cases where 0 is a valid value. Then
you are really out of luck. (Unless you decide, as
the authors of CW_FIELD did, that 0 is the default
value if the value is undefined. Dubious, at best.)

In any case, I find it *much* more useful to get
an undefined variable when the field is undefined,
so that is why FSC_INPUTFIELD, which is my CW_FIELD
replacement that looks editable on Windows machines,
uses pointers to store the value. If the field is
undefined, then the pointer points to an undefined
variable.

The lucky side effect of using a pointer to store
the value, is that I can also include the value
of the field in the event structure itself, which
is something CW_FIELD has never been able to do.
So I get two major benefits from using pointers.
Hard to argue with those economics. :-)

Cheers,

David

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