Subject: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by Christian Oehreneder on Thu, 07 Sep 1995 07:00:00 GMT
View Forum Message <> Reply to Message

MY PROBLEM AGAIN:

> Schalom!
>
> I want to store variables of different type together in structure. The
> type of each structure elemet is defined DURING RUNTIME.
> How can I create a structure, whos tag types and tag number are defined
> at runtime?
>
> Thanks in advance
> Chris
>

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by Juergen Paff on Thu, 07 Sep 1995 07:00:00 GMT
View Forum Message <> Reply to Message

"Christian Oehreneder (PHOTO-Dissertand)" <coehrene> wrote:

>> Schalom!
>>
Dto.
>> I want to store variables of different type together in structure. The
>> type of each structure elemet is defined DURING RUNTIME.
>> How can I create a structure, whos tag types and tag number are defined
>> at runtime?
>>
>> Thanks in advance
>> Chris
>>
>
I'm not shure I understood completly what you're up for, but here are some
remarks that may help:
(You didn't mention what you're using, Wave or IDL. But I assume the hints
below are valis for both systems. They at least hold for Wave)

1) You cannot redifine a struct that has a name. Fortunately, there are
unnamed structures. They can be defined at will, f.i.

```
my_struct={ , tagname1: value_of_type_tagtype1, $
            tagname2: value_of_type_tagtype2, $
            tagname3: value_of_type_tagtype3, $
            ; and so on.
```

```
        }
```
2) As you learn (name and) type only at run time, you might want to use the command

return_status=execute(string_argument)

string_argument has to be a string that conains a command that is valid at run time. Consider the following situation:
The type of tag1 is stored in string type1 , and the name in string name1 , and analogous for tag2, tag3. Consider further that values of the appropriate time are stored in variables with names value1 , value2 , and value3 .
Then you could use the code:

```
status=execute( $
      '"my_struct={ , '+name1+': '+value1+', '+ $
                  name2+': '+value2+', '+ $
                  name3+': '+value3+' }' "
```

At run time, this would fill in the contents of the string variables, and then execute the resulting command.
If you don't know the values yet, you could use a default initialization:

```
      '"my_struct={ , '+name1+': '+type1+'(0), '+ $
                '+name2+': '+type2+'(0), '+ $
                '+name3+': '+type3+'(0) }' "
```

Note: The above initialization with '0' may cause problems if type is, f. i., complex. If things like this can happen, you need to catch them with some ifs.
Okay, looks a little messy. However, this 'execute' command is incredibly helpful. You generate the code you need at runtime by glueing together the appropriate strings.
Note: There may be some errors in the above code, I couldn't test it.
Hope you got the gist.

In case I misunderstood you completely: plse send me mail.
Good luck Juergen

Subject: Re: Defining structuretype during runtime?? QUESTION!!
Posted by Juergen Paff on Fri, 08 Sep 1995 07:00:00 GMT
View Forum Message <> Reply to Message

Juergen Paff <paff@paff.slip.ethz.ch> wrote:
> Note: The above initialization with '0' may cause problems if type is, f. i.,
> complex. If things like this can happen, you need to catch them with
> some ifs.

Ooops: Sorry, delete the last 2 sentences. That was nonsense.
Juergen

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by sjt on Fri, 08 Sep 1995 07:00:00 GMT
View Forum Message <> Reply to Message

Christian Oehreneder (PHOTO-Dissertand) (coehrene) wrote:
: MY PROBLEM AGAIN:

: >Schalom!
: >
: >I want to store variables of different type together in structure. The
: >type of each structure elemet is defined DURING RUNTIME.
: >How can I create a structure, whos tag types and tag number are defined
: >at runtime?
: >
: >Thanks in advance
: > Chris
: >

One way is to use handles. I do this for a structure that has to contain
a variable length array of structures; it's not pretty but it works:

```
e.g.
pdefs= {graff_define, $
 .
 .
 nsets:0, $
 data: handle_create(), $
 .
 .
}

dtmpl= {graff_data, $
 xydata: handle_create(), $
 .
 .
}

handle_value, dtmpl.xydata, fltarr(2,2), /set
handle_value, pdefs.data, dtmpl, /set, no_copy
```

When you want to use the data structure, you need to use handle_value to
recover it and then replace it after use (no_copy is a good deal quicker
if you are using the structures a lot and certainly more memory efficient)

--

 +----------------------+--------------------------------- --+---------+
James Tappin,	School of Physics & Space Research	O__
sjt@star.sr.bham.ac.uk	University of Birmingham	-- \/`
Ph: 0121-414-6462. Fax: 0121-414-3722	

+---------- --+---------+

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by Jackel on Fri, 08 Sep 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <42n2sj$dhk@news.tuwien.ac.at> "Christian Oehreneder
(PHOTO-Dissertand)" <coehrene> writes:

> MY PROBLEM AGAIN:

>> Schalom!
>>
>> I want to store variables of different type together in structure. The
>> type of each structure elemet is defined DURING RUNTIME.
>> How can I create a structure, whos tag types and tag number are defined
>> at runtime?
>>
>> Thanks in advance
>>   Chris
>>

Have you considered using the intrinsic function "CREATE_STRUCT"?
Note that this will make named or anonymous structures.

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by Thomas A McGlynn on Mon, 11 Sep 1995 07:00:00 GMT
View Forum Message <> Reply to Message

jackel@canlon.physics.uwo.ca (Brian Jackel) wrote:
> In article <42n2sj$dhk@news.tuwien.ac.at> "Christian Oehreneder
> (PHOTO-Dissertand)" <coehrene> writes:
>
>> MY PROBLEM AGAIN:
>
>>> Schalom!
>>>
>>> I want to store variables of different type together in structure. The
>>> type of each structure elemet is defined DURING RUNTIME.
>>> How can I create a structure, whos tag types and tag number are defined
>>> at runtime?
>>>
>>> Thanks in advance
>>>   Chris
>>>

>
> Have you considered using the intrinsic function "CREATE_STRUCT"?
> Note that this will make named or anonymous structures.

There's a problem using CREATE_STRUCT if you don't know the number of elements in the structure in advance. It requires a separate argument for each value. I've built and seen other versions of functions that return structures where the both the names and values can be passed in as arrays. This gives the greatest flexibility in defining structures on the fly. E.g., one can call:

```
str = mrd_struct(['a','b','c'],['0.', '1L', 'intarr(30,30)'])
```

to create a structure with three elements, a scalar real, a scalar long and a 30x30 integer array. Using create_struct directly you've got problems if you don't know the number of elements in the structure.

You can implement this internally using the IDL create_struct to build the structure by concatenation, the execute function (for short structures), or by dynamically compiling functions defining the appropriate structure.

Good luck.

Tom McGlynn
Goddard Space Flight Center
mcglynn@grossc.gsfc.nasa.gov

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by hahn on Sun, 05 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Juergen Paff <paff@paff.slip.ethz.ch> wrote:
[...snip...]
> 2) As you learn (name and) type only at run time, you might want to use the
> command

> return_status=execute(string_argument)

> string_argument has to be a string that conains a command that is valid at
> run time. Consider the following situation:
[...snip...]
Ahem, but look at this warning included in the help for execute:
*** begin of quote ***
"Warning

"Do not use EXECUTE to create new variables inside procedures or functions. All

variables used by the string that is executed should be
referenced or defined when the program unit is originally compiled. New
variables cannot be created after a procedure or function is
compiled. Attempts to create new variables result in the error "Program data
area full".
*** end of quote ***

Maybe the IDL procedure Create_Struct is sufficient for what is
intended.

Hope this helps,
Norbert Hahn

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by R. Bauer on Mon, 06 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Norbert Hahn wrote:
>
> Juergen Paff <paff@paff.slip.ethz.ch> wrote:
> [...snip...]
>> 2) As you learn (name and) type only at run time, you might want to use the
>> command
>
>> return_status=execute(string_argument)
>
>> string_argument has to be a string that conains a command that is valid at
>> run time. Consider the following situation:
> [...snip...]
> Ahem, but look at this warning included in the help for execute:
> *** begin of quote ***
> "Warning
>
> "Do not use EXECUTE to create new variables inside procedures or functions. All
> variables used by the string that is executed should be
> referenced or defined when the program unit is originally compiled. New
> variables cannot be created after a procedure or function is
> compiled. Attempts to create new variables result in the error "Program data
> area full".
> *** end of quote ***

That's true, but you can undefine variables before whith
 a=n_elements(unknown)

help,unknown
UNKOWN   UNDEFINED = <Undefined>

Undefined variables could be used by execute, since idl3.6.1 !

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

Subject: Re: Defining structuretype during runtime?? QUESTION!! (no reply)
Posted by mgs on Mon, 06 Oct 1997 07:00:00 GMT
View Forum Message <> Reply to Message

In article <618ta0$l4m$3@sun27.hrz.th-darmstadt.de>,
hahn@hrz.tu-darmstadt.de (Norbert Hahn) wrote:

> Juergen Paff <paff@paff.slip.ethz.ch> wrote:
...
>> return_status=execute(string_argument)
>
>> string_argument has to be a string that conains a command that is valid at
>> run time. Consider the following situation:
> [...snip...]
> Ahem, but look at this warning included in the help for execute:
> *** begin of quote ***
> "Warning
>
> "Do not use EXECUTE to create new variables inside procedures or
functions. All
> variables used by the string that is executed should be
> referenced or defined when the program unit is originally compiled. New
> variables cannot be created after a procedure or function is
> compiled. Attempts to create new variables result in the error "Program data
> area full".
> *** end of quote ***

I think this is an IDL 4.0 limitation. Of course, I just removed IDL 4.0
from my system an hour ago. From the IDL 5.0 "New IDL Language Features"
section:

Limits Removed
The following limits that existed in previous versions of IDL have been removed:
...
There are no limits on new variables added by EXECUTE and RESTORE.
...

What version did Norbert's quote come from? I checked as many references to Execute as I could find in a few minutes, but didn't find one with the statement Norbert quoted.

--

Mike Schienle Interactive Visuals
mgs@sd.cybernex.net http://ww2.sd.cybernex.net/~mgs/
