Subject: Re: EXECUTE limits?
Posted by Thomas A. McGlynn on Mon, 21 Dec 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Alex Schuster wrote:
>
> Wayne Landsman wrote:
>
>> In practice, I find the V5.0 EXECUTE limits the same as they were in V4.0.
>>
>> In the example below, I try to use EXECUTE to define an anonymous
>> structure at run-time. (Note that I can't use the CREATE_STRUCT
>> function, because CREATE_STRUCT requires that one know the number of tags
>> beforehand.) This works fine so long as the EXECUTE string does not
>> exceed a certain length -- which seems to vary from machine to machine.
>> Otherwise I get a "Program code area full" error message.
>
> I experienced that such long statements work in compiled code only, not
> at the command line or via EXECUTE. BTW, this worked even better in IDL
> 4.
>
> But you can use CREATE_STRUCT this way:
>
> IDL> d = { name:0 }
> IDL> for i = 0, 99 do d = create_struct(d, 'name' + strtrim(i,2), 0)
>
> Alex
> --

The problem with the approach you give is that you need to have a variable
of the type you wish the structure element to be. E.g., if I know that
...the 54th element is to be a fltarr(10,10,10) and the 55th is a double...
where the types are defined dynamically then I need to do something
like:

```
for i=0, number_of_fields
   strng = "fld = "+typestring(i)
   execute(strng)
   d = create_struct(d, name(i), fld)
endfor
```

where I've collected the names and the type definitions into character
arrays. So I have to do an execute as well as a create_struct for
every line.

I've also been shown that IDL has a problem with structure definitions
of the form:

```
str = {a1: 0,a2:0, ... a996:0}
```

for very large structures -- although such structures can
be built piecemeal as you describe above.

The solution I've adopted (and sent to Wayne) is try to optimize the number
of calls to execute and create_struct so that we have:

```
names = ['name1', 'name2', ...]
types = ['0', 'fltarr(10)', 'double(10,10)', '1.0'...]

str = mrd_struct(names, types)
```

and in MRD_STRUCT we do something like:

```
for(each field) do begin
  while (length of strng fits in EXECUTE buffer) do begin
    strng = strng + names(i)+":"+types(i)
  endwhile
  execute("substr = {"+strng+"}")
  total_str = create_struct(total_str, substr)
endfor
execute("substr = {"+strng+"}")
total_str = create_struct(total_str, substr)
```

This retains a lot more flexibility than the IDL standard create_struct
but minimizes the number of calls to execute and create_struct.

My testing seems to indicate that it's reasonably efficient up to
a thousand or so fields, but it does slow up substantially then -- though
that may be that IDL has trouble with the very large structures.

Regards,
Tom McGlynn
tam@silk.gsfc.nasa.gov
