Subject: EXECUTE limits?
Posted by landsman on Sun, 20 Dec 1998 08:00:00 GMT
View Forum Message <> Reply to Message

When IDL V5.0 was released, one of the new features was supposedly the removal of language limits, e.g. "an unlimited number of variables is now accepted, and users can freely add new variable names with the EXECUTE statement."

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.

Note that I never have a problem if I instead write the structure definition string to a temporary procedure file and compile the file. (This is the workaround used by Tom McGlynn's MRD_STRUCT procedure at http://idlastro.gsfc.nasa.gov/ftp/pro/structure) So it does seem that problem is caused by (machine-dependent?) limits to the EXECUTE statement.

Any Comments?

Wayne Landsman landsman@mpb.gsfc.nasa.gov

```
IDL> print,!VERSION
{ alpha OSF unix 5.1 Apr 13 1998}
IDL> a = 'd = {name:0'
IDL> for i=0,99 do a = a + ',name' + strtrim(i,2) + ':0'
IDL> a = a + '}'
IDL> res = execute(a)

d =  {name:0,name0:0,name1:0,name2:0,name3:0,name4:0,name5:0,name 6:0,name7:0,
 name8:0,name9:0,name10:0,name11:0,name12:0,name13:0,name14:0 ,name15:0,name16:0,
 name17:0,name18:0,name19:0,name20:0,name21:0,name22:0,name23 :0,name24:0,
 name25:0,name26:0,name27:0,name28:0,name29:0,name30:0,name31 :0,name32:0,
 name33:0,name34:0,name35:0,name36:0,name37:0,name38:0,name39 :0,name40:0,
 name41:0,name42:0,name43:0,name44:0,name45:0,name46:0,name47 :0,name48:0,
 name49:0,name50:0,name51:0,name52:0,name53:0,name54:0,name55 :0,name56:0,
 name57:0,name58:0,name59:0,name60:0,name61:0,name62:0,name63 :0,name64:0,
 name65:0,name66:0,name67:0,name68:0,name69:0,name70:0,name71 :0,name72:0,
 name73:0,name74:0,name75:0,name76:0,name77:0,name78:0,name79 :0,name80:0,
 name81:0,name82:0,name83:0,name84:0,name85:0,name86:0,name87 :0,name88:0,
 name89:0,name90:0,name91:0,name92:0,name93:0,name94:0,name95 :0,name96:0,
name97:0,name98:0,name99:0}
```

% Program code area full.
0