Subject: Re: Function referencing/automatic defintion question.
Posted by JD Smith on Fri, 30 May 2003 07:41:21 GMT
View Forum Message <> Reply to Message

On Thu, 29 May 2003 12:40:32 -0700, Paul van Delst wrote:

> If the result of
> PRINT, ROUTINE_INFO(/FUNCTIONS)
> contains the name of the function in question, then it has been
> compiled, right? And the fact that using
> COMILE_OPT STRICTARR
> makes eveything work means the same thing. (Right? I think so.)

There are routines which IDL knows about but hasn't compiled. They're called "unresolved". Try this:

```
print,routine_info(/FUNCTIONS,/SOURCE)
```

Anything listed without source is unresolved. When IDL compiles files, it records any procedures or functions it finds there, and only later actually goes looking for them. So, just being listed in routine info doesn't indicate a routine has been compiled (or even that it exists). Try compiling a file with a=mycrazyfunctionwhichwillneverexist() and you'll see it nonetheless.

I don't think there's anything wrong with your setup. I can put:

```
function stfunction,a
  return,a^2
end

pro stprocedure,b
  return
end

pro st__define,a
  a={ST,b:0}
end
```

in st__define.pro, and then:

```
IDL> a={st}
```

compiles the listed procedure and function by side-effect, and they work fine. The place this technique can go quite wrong, as it can for objects, is if the structure in question is already defined by some other means.

Then IDL does not feel compelled to compile your __define fine, and your utility routines remain hidden. Any chance you use a full structure definition in creating a struct of this type anywhere else? You obviously can't mix the two methods for this to work.

JD
