Subject: Re: Who called that procedure?
Posted by Michael Galloy on Tue, 03 Feb 2009 17:23:56 GMT
View Forum Message <> Reply to Message

wlandsman wrote:
> This is a minor problem but it has cost me a half-hour a couple of
> times.
>
> I run RESOLVE_ALL to make sure that I am including all necessary
> procedures when distributing software. One procedure may yield ~200
> compiled supporting procedures, and I see one -- say obsolete.pro --
> that should not be being used any more. So I want to know which of
> the 200 procedures is still calling obsolete.pro. There does not
> seem to be any pattern to the order of procedures displayed by
> RESOLVE_ALL so that does not help. What would be nicest I suppose
> would be a tree diagram of all the dependencies.
>
> In the end, I can simply grep the 200 procedures to see which one has
> a call to obsolete.pro. But is there a better way? Thanks, -- Wayne

I would like to add the creation of such dependency trees to IDLdoc at some point. In particular, they would be handy when giving the minimal amount of source to someone. Concerns about method calls, call_procedure/call_function/execute, and distinguishing a function call from an array indexing have always made me delay implementing it.

I usually end up just searching (although I have replaced grep with ack, it conveniently ignores .svn directories and some other niceties to reduce the false positives).

Another strategy is to put a "compile_opt obsolete" statement in MY_OBSOLETE_ROUTINE and then set:

```
IDL> !warn.obs_routines = 1
```

Now, calls to MY_OBSOLETE_ROUTINE will generate syntax errors when compiled (identifying their exact location in the error message).

Mike
--
www.michaelgalloy.com
Tech-X Corporation
Associate Research Scientist