Heinz Stege wrote, On 2015-02-02, 1:40pm:
> On Fri, 30 Jan 2015 17:24:33 -0800 (PST), Jim P wrote:
>
>> The key to understanding the problem is that the IDL compiler *must* restore the hash.sav at compile time if a structure tag is defined as LIST(), HASH(), etc. At that point, all is lost* with respect to IDL "forgetting" about the compiled definitions.
>>
> Aah, that brings some light into the darkness. IDL 8.4 restores the
> hash.sav file when compiling a routine with LIST() as a structure tag?
> I didn't know this, since I'm still working with Version 8.0.1.* My
> version is different. It does _not_ restore hash.sav or list.sav when
> compiling a routine with a LIST() tag in the class structure
> definition.
>
> Do you think, that it may be interesting for the news group, why this
> change was done?
>
> Cheers, Heinz
>
> * My company does not have the money to upgrade. :-(

I think I was caught by this issue, hoping to compile a .sav in 8.4 for a client to run in 8.3. Some problems were
relieved by:

1. In my project's Build, replacing the standard RESOLVE_ALL with post-process command:
   RESOLVE_ALL, skip=['list','hash','orderedhash']

2. Avoiding the use of myList = List(myArray, /EXTRACT)

But I'm seeing another problem I can't seem to avoid, where I'm adding an item to a list, and I get:
   IDL_CONTAINER::ADD: Object reference type required in this context: INDEXITEM.
   (INDEXITEM was my variable that I was Adding to the list)

I confirm that neither IDL 8.3 nor 8.4 actually require this of a List() object.

I tried adding 'idl_container' to the "skip" list in RESOLVE_ALL, to no avail.

Any suggestions of how to get around this?

--

Cheers,
-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada
www.d-jackson.com