Subject: Re: A simple DLM question
Posted by Xin Tao on Tue, 28 Aug 2012 21:16:53 GMT
View Forum Message <> Reply to Message

I did check the flags of IDL_V_CONST and IDL_V_TEMP. Both failed, and then I posted the question. For example, I'm not sure how to explain this results.

// Code here.

```
 void simple(int argc, IDL_VPTR argv[])
{
  IDL_VPTR v;

  v = IDL_BasicTypeConversion(1, &argv[0], IDL_TYP_DOUBLE);

  printf("const = %d\n", v->flags & IDL_V_CONST);
  printf("temp = %d\n", v->flags & IDL_V_TEMP);

  if (v != argv[0]) IDL_DELTMP(v);
}
```

Now results:

```
IDL> simple, 3.0d
const = 1
temp = 0
IDL> simple, -3.0d
const = 0
temp = 2
```

This sounds really strange to me. But if there is a good explanation of this, please let me know. Thanks.

Xin Tao

On Tuesday, August 28, 2012 11:06:44 AM UTC-5, jimmylee...@gmail.com wrote:
> On Monday, August 27, 2012 4:04:43 PM UTC-6, Xin Tao wrote:
>
>> Thanks Jimmy. That indeed solved my problem. It was so confusing to me, because I found from the External Development Guide that IDL_DELTMP should check it first. :)
>
>>
>
>>
>

>>
>
>> On Monday, August 27, 2012 3:48:27 PM UTC-5, jimmylee...@gmail.com wrote:
>
>>
>
>>> On Monday, August 27, 2012 11:13:57 AM UTC-6, Xin Tao wrote:
>
>>
>
>>>
>
>>
>
>>>> Hi,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> I'm having trouble figuring out the problem of the following DLM code:
>

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> /* The c routine */
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>

```
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> void simple(int argc, IDL_VPTR argv[])
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> {
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>   IDL_VPTR v;
>
```

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> v = IDL_BasicTypeConversion(1, &argv[0], IDL_TYP_DOUBLE);
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>   IDL_DELTMP(v);
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> }
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> This routine just takes its input and convert it to double. After converting it to a DLM, however, I seem to see strange results.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> IDL> simple, 1.0d

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> % Loaded DLM: TESTMODULE.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> IDL> simple, -1.0d
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Bus error

>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> That is: if I give it 1.0d as input, then the code is fine. However, if I use -1.0d, then there is a BUS error, presumably from IDL_DELTMP(v). I really don't understand why this is the case. Isn't IDL_DELTMP supposed to decide first whether v is a temporary variable or not? If I remove IDL_DELTMP, of course, I'll frequently get the annoying warning message "% Temporary variables are still checked out - cleaning up...".
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>

>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Please give me some help. Thanks.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Try this:
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>

>
>>
>
>>> if (v != argv[0]) IDL_DELTMP(v);
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> That is, no conversion was necessary.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> The macro (in idl_export.h, if you're interested) doesn't do extensive checking, and you should only free variables that are temps, not expressions or constants.
>
>
>
> The docs are correct, but are confusing if you're not aware of the difference between IDL's temporary variables, constants, and named variables. It's not stated explicitly in this section that a constant like 1.0D is a different sort of data type internally than an expression or named variable, though that topic is discussed earlier in the docs.
>
>
>
> IDL_DELTMP doesn't check if the IDL_VARIABLE has the contant flag set (IDL_V_CONST),

only the temporary flag (IDL_V_TEMP).
>
>
>
> As a matter of habit, I always check the equality of the argv[] used as input against the output from any type conversion routine call before calling IDL_DELTMP. You can't predict when a user has entered an explicit constant value, rather than a variable name or expression.
