Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by markb77 on Mon, 04 Mar 2013 14:19:31 GMT
View Forum Message <> Reply to Message

ps. my system is running IDL Version 8.0.1, Microsoft Windows (Win32 x86_64 m64)

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by markb77 on Mon, 04 Mar 2013 14:39:35 GMT
View Forum Message <> Reply to Message

I've updated this a bit. It seems that the HELP command can distinguish between the two cases, but SIZE(), even with the /STRUCTURE keyword specified, cannot.

```
pro test

   a = hash('first', 1, 'second', 2)
   b = hash('third', 3, 'fourth', 4, 'fifth', 5)

   print, 'Output of SIZE() for a HASH varible:'
   print, size(a, /STRUCTURE)

   c = [a,b]
   print, 'Output of SIZE() for an array of HASHes:'
   print, size(c, /STRUCTURE)

   help, a, output=ahelp
   help, c, output=chelp

   print, ahelp
   print, chelp

end

IDL> test

Output of SIZE() for a HASH varible:

{ OBJREF      11      0       0       2
1         2         0         0         0
0         0         0         0
}

Output of SIZE() for an array of HASHes:

{ OBJREF      11      0       0       2
```

```
1       2       0       0       0
0       0       0       0
}

A           HASH  <ID=18  NELEMENTS=2>
C           OBJREF    = Array[2]
```

so I could solve this by parsing the output of HELP, but this is strongly discouraged in the documentation for the IDL Help command itself, as the text formatting of the output from this command my change...

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by markb77 on Mon, 04 Mar 2013 16:05:03 GMT
View Forum Message <> Reply to Message

surprisingly, the ISA function doesn't help here either...

```
pro test

   a = hash('first', 1, 'second', 2)
   b = hash('third', 3, 'fourth', 4, 'fifth', 5)

   print, 'Output of SIZE() for a HASH varible:'
   print, size(a, /STRUCTURE)

   c = [a,b]
   print, 'Output of SIZE() for an array of HASHes:'
   print, size(c, /STRUCTURE)

   print, 'Output of ISA(/ARRAY) for scalar hash:'
   print, isa(a, /array)

   print, 'Output of ISA(/ARRAY) for array of hashes:'
   print, isa(c, /array)

   help, a, output=ahelp
   help, c, output=chelp

   print, 'Help for scalar hash:'
   print, ahelp

   print, 'Help for array of hashes:'
   print, chelp

end
```

Output of SIZE() for a HASH varible:
{ OBJREF 11 0 0 2
1 2 0 0 0
0 0 0 0
}
Output of SIZE() for an array of HASHes:
{ OBJREF 11 0 0 2
1 2 0 0 0
0 0 0 0
}
Output of ISA(/ARRAY) for scalar hash:
1
Output of ISA(/ARRAY) for array of hashes:
1
Help for scalar hash:
A HASH <ID=154 NELEMENTS=2>
Help for array of hashes:
C OBJREF = Array[2]

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by wlandsman on Mon, 04 Mar 2013 16:11:53 GMT
View Forum Message <> Reply to Message

You could use the TYPENAME function to distinguish the two variables

IDL> print,typename(a)
HASH
IDL> print,typename(c)
OBJREF

although I'm not sure why SIZE(/TNAME) doesn't give the same results. --Wayne

On Monday, March 4, 2013 11:05:03 AM UTC-5, superchromix wrote:
> surprisingly, the ISA function doesn't help here either...
>
>
>
>
>
> pro test
>
>
>
> a = hash('first', 1, 'second', 2)
>
> b = hash('third', 3, 'fourth', 4, 'fifth', 5)

```
>
>
>
>     print, 'Output of SIZE() for a HASH varible:'
>
>     print, size(a, /STRUCTURE)
>
>
>
>     c = [a,b]
>
>     print, 'Output of SIZE() for an array of HASHes:'
>
>     print, size(c, /STRUCTURE)
>
>
>
>     print, 'Output of ISA(/ARRAY) for scalar hash:'
>
>     print, isa(a, /array)
>
>
>
>     print, 'Output of ISA(/ARRAY) for array of hashes:'
>
>     print, isa(c, /array)
>
>
>
>     help, a, output=ahelp
>
>     help, c, output=chelp
>
>
>
>     print, 'Help for scalar hash:'
>
>     print, ahelp
>
>
>
>     print, 'Help for array of hashes:'
>
>     print, chelp
>
>
>
> end
```

>
>
>
> Output of SIZE() for a HASH varible:
>
> { OBJREF 11 0 0 2
>
> 1 2 0 0 0
>
> 0 0 0 0
>
> }
>
> Output of SIZE() for an array of HASHes:
>
> { OBJREF 11 0 0 2
>
> 1 2 0 0 0
>
> 0 0 0 0
>
> }
>
> Output of ISA(/ARRAY) for scalar hash:
>
> 1
>
> Output of ISA(/ARRAY) for array of hashes:
>
> 1
>
> Help for scalar hash:
>
> A HASH <ID=154 NELEMENTS=2>
>
> Help for array of hashes:
>
> C OBJREF = Array[2]

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by David Fanning on Mon, 04 Mar 2013 16:16:09 GMT
View Forum Message <> Reply to Message

wlandsman writes:

> You could use the TYPENAME function to distinguish the two variables
>

> IDL> print,typename(a)
> HASH
> IDL> print,typename(c)
> OBJREF
>
> although I'm not sure why SIZE(/TNAME) doesn't give the same results.

Oh, dear! Really!?

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by David Fanning on Mon, 04 Mar 2013 16:24:49 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>
> wlandsman writes:
>
>> You could use the TYPENAME function to distinguish the two variables
>>
>> IDL> print,typename(a)
>> HASH
>> IDL> print,typename(c)
>> OBJREF
>>
>> although I'm not sure why SIZE(/TNAME) doesn't give the same results.
>
> Oh, dear! Really!?

I guess this is right, though, since there are no "type" codes for hash
or list. They are objects, I guess, not real data types.

Still, awfully confusing to explain to a new user not familiar with the
"IDL Way". ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by wlandsman on Mon, 04 Mar 2013 16:28:55 GMT
View Forum Message <> Reply to Message

The documentation for TYPENAME() explains that it is not exactly the same as SIZE(/TNAME)

"If Variable is a scalar object or a one-element object array, Result will be the object class name. If Variable is an object array with two or more elements, Result will be the IDL basic type name “OBJECT” [because each element of the object array could have a different class name]."

So with a scalar hash, TYPENAME() can return "HASH" but for an array it will return OBJREF (not "OBJECT" as the documentation wrongly says). --Wayne

On Monday, March 4, 2013 11:24:49 AM UTC-5, David Fanning wrote:
> David Fanning writes:
>
>
>
>>
>
>> wlandsman writes:
>
>>
>
>>> You could use the TYPENAME function to distinguish the two variables
>
>>>
>
>>> IDL> print,typename(a)
>
>>> HASH
>
>>> IDL> print,typename(c)
>
>>> OBJREF
>

>>>
>
>>> although I'm not sure why SIZE(/TNAME) doesn't give the same results.
>
>>
>
>> Oh, dear! Really!?
>
>
>
> I guess this is right, though, since there are no "type" codes for hash
>
> or list. They are objects, I guess, not real data types.
>
>
>
> Still, awfully confusing to explain to a new user not familiar with the
>
> "IDL Way". ;-)
>
>
>
> Cheers,
>
>
>
> David
>
>
>
>
>
> --
>
> David Fanning, Ph.D.
>
> Fanning Software Consulting, Inc.
>
> Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
>
> Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by markb77 on Mon, 04 Mar 2013 16:37:26 GMT
View Forum Message <> Reply to Message

thanks. Even for someone familiar with IDL, this seems somewhat

backwards. For the rest of the IDL data types we can distinguish
between array and scalar objects using SIZE(x, /N_Dimensions)... no?
Now for lists and hashes we have to check the typename() for HASH vs
OBJREF ? that can't be how it was intended..?

-Mark

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by markb77 on Mon, 04 Mar 2013 16:44:26 GMT
View Forum Message <> Reply to Message

oops I meant "array and scalar variables" above...

to make one other point, the ISA(/ARRAY) function is unambiguous in
its meaning.. it should return true for arrays and false for scalars,
regardless of whether the scalar is an object reference.

unless.. of course, the object refererence refers to an array-like
object I suppose... but in the case of hash objects if you try to do
this:

a=hash('first',1)

and then try to index it with a[0], this throws an error.

confusion! :)
Mark

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by wlandsman on Mon, 04 Mar 2013 16:47:53 GMT
View Forum Message <> Reply to Message

I agree. I hope someone from Exelis will eventually chime in as to the best way to distinguish between scalar and array hashes or lists. --Wayne

On Monday, March 4, 2013 11:37:26 AM UTC-5, superchromix wrote:
> thanks. Even for someone familiar with IDL, this seems somewhat
>
> backwards. For the rest of the IDL data types we can distinguish
>
> between array and scalar objects using SIZE(x, /N_Dimensions)... no?
>
> Now for lists and hashes we have to check the typename() for HASH vs
>
> OBJREF ? that can't be how it was intended..?

>
>
>
> -Mark

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by Lajos Foldy on Mon, 04 Mar 2013 16:55:36 GMT
View Forum Message <> Reply to Message

On Monday, March 4, 2013 5:37:26 PM UTC+1, superchromix wrote:
> thanks. Even for someone familiar with IDL, this seems somewhat
>
> backwards. For the rest of the IDL data types we can distinguish
>
> between array and scalar objects using SIZE(x, /N_Dimensions)... no?
>
> Now for lists and hashes we have to check the typename() for HASH vs
>
> OBJREF ? that can't be how it was intended..?
>
>
>
> -Mark

Except structures. There is no scalar structure, s={i:0} and a=replicate(s,2) are both arrays.

regards,
Lajos

Subject: Re: bug? or how to distinguish between a Hash and an array of Hashes ?
Posted by markb77 on Mon, 04 Mar 2013 17:00:31 GMT
View Forum Message <> Reply to Message

> Except structures. There is no scalar structure, s={i:0} and a=replicate(s,2) are both arrays.
>
> regards,
> Lajos

good point. At least if you index a scalar structure as s[0], this
doesn't throw an error, unlike the case with the hash.
