Subject: Re: Broken hash init()
Posted by penteado on Thu, 28 Oct 2010 23:49:15 GMT
View Forum Message <> Reply to Message

On Oct 27, 9:18 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
> On Oct 27, 8:57 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:
>
>
>
>
>
>> I just noticed that the init method is not properly creating hashes
>> from structures on some situations:
>
>> IDL> a={b:list(1,2)}
>> IDL> print,a
>> {<ObjHeapVar1(LIST)>}
>> IDL> print,a.b
>> 1
>> 2
>> IDL> h=hash(a)
>> IDL> print,h
>> B: 1
>
>> Or,
>
>> IDL> a={b:[1,2]}
>> IDL> print,a
>> { 1 2}
>
>> IDL> h=hash(a)
>> IDL> print,h
>> B: 1
>
>> But this works:
>
>> IDL> a={b:9,c:[5,8]}
>> IDL> print,a
>> { 9 5 8}
>
>> IDL> h=hash(a)
>> IDL> print,h
>> B: 9
>> C: 5 8
>
> Trying to get around that problem, I encountered the same problem
> because I forgot about the bug with foreach on string arrays. So
> perhaps that is also causing the problem with hash::init():

>
> IDL> a={a:[1,2]}
> IDL> h=hash()
> IDL> foreach tag,tag_names(a),i do h[tag]=a.(i)
> IDL> print,h
> A: 1
>
> Using a for loop instead of the foreach, it works normally:
>
> IDL> tn=tag_names(a)
> IDL> for i=0,n_elements(tn)-1 do h[tn[i]]=a.(i)
> IDL> print,h
> A: 1   2

I just verified that this is fixed in 8.0.1. Which makes this the
shortest duration I ever experienced a bug.
