
Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Paul Van Delst\[1\]](#) on Tue, 14 Dec 2010 15:35:18 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

Paulo Penteado wrote:

```
> On Dec 13, 7:48 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:
>> But I got the "LIST::REMOVE: Index is out of range" error instead. I'm still trying to figure out
how to best handle it
>> (it does make using lists a little bit more complicated). The HASH remove method works as I
expected:
>>
>> IDL> h=hash("a", 3.14)
>> IDL> help, h
>> H          HASH <ID=1 NELEMENTS=1>
>> IDL> help, h.remove()
>> <Expression>  FLOAT    =    3.14000
>> IDL> help, h.remove()
>> <Expression>  UNDEFINED = !NULL
>
> Besides the bad error message for list::remove(), I would say that the
> bug is in hash::remove(). Unless I am missing something, it should
> throw an error on an empty hash. Returning !null with no errors makes
> the empty list/hash look like it has elements (which happen to have !
> null as a value).
```

I guess it depends on one's perspective - in my case my first exposure to lists and hashes was via ruby:

```
irb(main):001:0> l=[nil,2,nil]
=> [nil, 2, nil]
irb(main):002:0> l.pop
=> nil
irb(main):003:0> l.pop
=> 2
irb(main):004:0> l.pop
=> nil
irb(main):005:0> l.pop
=> nil
irb(main):006:0> l.pop
=> nil
irb(main):007:0> ...etc...
```

Thinking about it a little more, I think I'm biased against IDL and thus making bad design decisions (sorry!). If there was a list "Empty" or "IsEmpty" method (like ruby's "empty?" method...for both arrays/lists and hashes), I would be more

tempted to use it,

```
if ( l isempty() ) then ...return, handle error, whatever...
```

as opposed to the current IDL idiom,

```
if ( n_elements(l) eq 0 ) then ...return, handle error, whatever...
```

For some reason I find this type of mixture of functional and object coding jarring - and this is from someone who has spent most of his life writing Fortran code! (I think it's also why ruby appeals to me more than python).

Oh well.

cheers,

paulv
