Subject: Re: LIST "bug": .Remove on an empty list
Posted by Paul Van Delst[1] on Mon, 13 Dec 2010 21:48:18 GMT
View Forum Message <> Reply to Message

Hello,

I encountered similar behaviour in some code I wrote the other day too, except I did something like

```
IDL> l=list("a",3.14)
IDL> help, l
L               LIST  <ID=1  NELEMENTS=2>
IDL> help, l.remove()
<Expression>    FLOAT     =       3.14000
IDL> help, l.remove()
<Expression>    STRING    = 'a'
IDL> help, l.remove()
% LIST::REMOVE: Index is out of range.
% Error occurred at: LIST::REMOVE
%                   $MAIN$
% Execution halted at: $MAIN$
```

While the error message is not cryptic as in the OP's case, I had assumed that trying to remove something from an empty
list using the remove method would return !null -- similar to how ruby returns "nil" in a similar situation:

```
irb(main):003:0> l=["a",3.14]
=> ["a", 3.14]
irb(main):004:0> l.pop
=> 3.14
irb(main):005:0> l.pop
=> "a"
irb(main):006:0> l.pop
=> nil
```

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
```

Matt Haffner wrote:
> Although this is easy to code around by checking the length before
> calling .Remove, I was surprised this just didn't return silently:
>
> IDL> l=list(1, length=100)
> IDL> help,l
> L LIST <ID=1424588 NELEMENTS=100>
> IDL> l.remove,/all
> IDL> help,l
> L LIST <ID=1424588 NELEMENTS=0>
> IDL> l.remove,/all
> % PTR_FREE: Pointer type required in this context: P.
> % Error occurred at: LIST::REMOVE
> % LIST::REMOVE
> % $MAIN$
> % Execution halted at: $MAIN$
>
> Passing it along to help the diagnosing of cryptic errors ;)
>
> mh

cheers,

paulv
