Subject: Re: LIST "bug": .Remove on an empty list
Posted by penteado on Tue, 14 Dec 2010 03:18:39 GMT
View Forum Message <> Reply to Message

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).

For instance,

```
IDL> l=list(length=1)
IDL> help,l.remove()
<Expression>    UNDEFINED = !NULL
IDL> help,l.remove()
% LIST::REMOVE: Index is out of range.
% Error occurred at: LIST::REMOVE
%                    $MAIN$
% Execution halted at: $MAIN$
```

If the second remove() returned !null without errors, it would have looked like the first call, which was properly returning (and removing) the !null which was the list element. List is doing things properly, it is hash that is confusing:

```
IDL> h=hash('a')
IDL> help,h
H               HASH  <ID=1  NELEMENTS=1>
IDL> help,h.remove()
<Expression>    UNDEFINED = !NULL
IDL> help,h
H               HASH  <ID=1  NELEMENTS=0>
IDL> help,h.remove()
```

<Expression> UNDEFINED = !NULL