
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

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
```

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

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 15:46:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> 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...

I can't stop to look into this today (18 pages yesterday!!),
but why can't you just write this ISEmpty method for yourself?

```
Function List::IsEmpty  
  IF N_Elements(self) EQ 0 THEN RETURN, 0 ELSE RETURN, 1  
END
```

That would work, que no?

Cheers,

David

--

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

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [penteado](#) on Tue, 14 Dec 2010 15:49:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 14, 1:35 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:

> 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).

I agree there should be an empty() method. Besides the reason you mentioned, it is also more direct and expressive to just test for empty() than to do a comparison with a function call. I will add that to the class I am writing to support assignment in arrays inside lists.

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 15:50:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning writes:

```
> Function List::IsEmpty  
>   IF N_Elements(self) EQ 0 THEN RETURN, 0 ELSE RETURN, 1  
>   END
```

BTW, this will have to be saved as list__isempty.pro

David

--

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

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Paul Van Delst\[1\]](#) on Tue, 14 Dec 2010 16:06:55 GMT

David Fanning wrote:

> I can't stop to look into this today (18 pages yesterday!!),
> but why can't you just write this ISEmpty method for yourself?

I have. But I believe it should be part of the standard library in IDL. A user shouldn't have to extend a class for such basic functionality.

Philosophically, I'm a "just because you can extend a class doesn't mean you should" type of guy. Pragmatically, I'm a "just extend the damn class!" fellow. :o)

```
> Function List::IsEmpty  
>   IF N_Elements(self) EQ 0 THEN RETURN, 0 ELSE RETURN, 1  
>   END  
>  
> That would work, que no?
```

Well, no. I would do something like

```
IF N_Elements(self) EQ 0 THEN RETURN, 1 ELSE RETURN, 0
```

:o)

Or, more likely:

```
Function List::IsEmpty  
  @boolean_codes  
  IF N_Elements(self) EQ 0 THEN RETURN, TRUE ELSE RETURN, FALSE  
END
```

where "TRUE" and "FALSE" are defined (correctly :o) in the boolean_codes.pro include file.

cheers,

paulv

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 16:17:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> David Fanning wrote:
>> I can't stop to look into this today (18 pages yesterday!!),

>> but why can't you just write this ISEmpty method for yourself?
>
> I have. But I believe it should be part of the standard library in IDL. A user shouldn't have to extend a class for such
> basic functionality.
>
> Philosophically, I'm a "just because you can extend a class doesn't mean you should" type of guy.
> Pragmatically, I'm a "just extend the damn class!" fellow. :o)

I can tell you haven't been working with IDL very long. :-)

```
>> Function List::IsEmpty  
>>   IF N_Elements(self) EQ 0 THEN RETURN, 0 ELSE RETURN, 1  
>> END  
>>  
>> That would work, que no?  
>  
> Well, no. I would do something like  
>   IF N_Elements(self) EQ 0 THEN RETURN, 1 ELSE RETURN, 0  
>  
> :o)
```

Uh, right. I don't program well under pressure. :-(

Cheers,

David

P.S. Let's just say if we had to wait for ITTVIS to get it together, we would still be working with the TV and CONTOUR commands. ;-)

--

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

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 16:33:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> I have. But I believe it should be part of the standard

> library in IDL. A user shouldn't have to extend a class
> for such basic functionality.

Just a note for the young programmers out there who are trying to figure out how to make a living with out moving to Bangalore or some place like that. Pick a piece of software, any software, and start adding the basic functionality that the original programmers missed. You can make a long career and a pretty good living* out of it. :-)

Cheers,

David

*Especially if by "pretty good living" you mean
"play lots of tennis" instead of "make lots of money." :-)

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [penteado](#) on Tue, 14 Dec 2010 16:49:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Dec 14, 2:06 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:

> Philosophically, I'm a "just because you can extend a class doesn't mean you should" type of guy.
> Pragmatically, I'm a "just extend the damn class!" fellow. :o)

Similarly, for such a small change, I might make a derived class for local use, but would be reluctant to use it in general, stand-alone routines (as opposed to programs that already comprise many files), as the benefit of the method would be outweighed by the hassle of carrying the extra file around.

Which is why I mentioned I intend to put it into the class I started writing (have not had the time to touch it in weeks) to provide several missing things - most importantly, the ability to assign to elements in arrays contained in lists and hashes (http://groups.google.com/group/comp.lang.idl-pvwave/browse_thread/thread/89ad055656bf6798).

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Paul Van Delst\[1\]](#) on Tue, 14 Dec 2010 17:19:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

hello,

Paulo Penteado wrote:

> On Dec 14, 2:06 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:
>> Philosophically, I'm a "just because you can extend a class doesn't mean you should" type of guy.
>> Pragmatically, I'm a "just extend the damn class!" fellow. :o)
>
> Similarly, for such a small change, I might make a derived class for
> local use, but would be reluctant to use it in general, stand-alone
> routines (as opposed to programs that already comprise many files), as
> the benefit of the method would be outweighed by the hassle of
> carrying the extra file around.

Yes, the approach I usually take is to put the desired functionality in a subclass, and then use the subclass for everything. But, as you said, for such a small change I'm not sure what the best approach is. Put in a feature request to ITTVIS? :o)

> Which is why I mentioned I intend to put it into the class I started
> writing (have not had the time to touch it in weeks) to provide
> several missing things - most importantly, the ability to assign to
> elements in arrays contained in lists and hashes (http://groups.google.com/group/comp.lang.idl-pvwave/browse_thread/thread/89ad055656bf6798).

I know I'm taking the other scripting language analogies waaaay too far, but it would neat if IDL had a better mechanism for sharing user developed code -- like one does with ruby gems (see <http://rubygems.org>). Or python eggs.

cheers,

paulv

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 17:39:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> I know I'm taking the other scripting language analogies

> waaaaay too far, but it would neat if IDL had a better mechanism
> for sharing user developed code -- like one does with ruby gems >
> (see <http://rubygems.org>). Or python eggs.

Good idea. I've already talked to Paulo a little bit about some ideas I have for creating the New New Graphics routines. (Like the New Graphics routines but simple enough to work the way people expect them to.) We could put them here! ;-)

Cheers,

David

P.S. Does anyone beside me and Dick Jackson see the danger in calling a new graphics system "new"? Coyote suggests the name Newest New Old Graphics System for the work I've been doing lately, but I don't know... :-(

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Michael Galloy](#) on Tue, 14 Dec 2010 22:34:51 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/14/10 9:19 AM, Paul van Delst wrote:

> I know I'm taking the other scripting language analogies waaaaay too far, but it would neat if IDL had a better mechanism
> for sharing user developed code -- like one does with ruby gems (see <http://rubygems.org>). Or python eggs.

I'm interested in this even beyond eggs, but to the equivalent of (the Python tools) distutils, easy_install, distribute, etc.

Mike

--

www.michaelgalloy.com

Research Mathematician

Tech-X Corporation

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Michael Galloy](#) on Tue, 14 Dec 2010 22:34:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/14/10 9:39 AM, David Fanning wrote:

- > P.S. Does anyone beside me and Dick Jackson see the
- > danger in calling a new graphics system "new"? Coyote
- > suggests the name Newest New Old Graphics System for
- > the work I've been doing lately, but I don't know... :-)

Yes, that's why I thought "function graphics" would be a good name.
Maybe some kind of numbered system? I think we would be up to new 4.0
graphics now? With "old" graphics being direct graphics, then object
graphics, Live Tools, iTools, and "new graphics"? Am I missing some
other graphics systems?

Mike

--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 23:19:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Michael Galloy writes:

- > Yes, that's why I thought "function graphics" would be a good name.
- > Maybe some kind of numbered system? I think we would be up to new 4.0
- > graphics now? With "old" graphics being direct graphics, then object
- > graphics, Live Tools, iTools, and "new graphics"? Am I missing some
- > other graphics systems?

Gosh, I never thought I would end up a taxonomist. I may be missing
something, but aren't there five items on your list? Are we
starting the count from zero, like in IDL?

What in the world would you call these new FSC_*** routines
I've been working on? They aren't direct graphics, and they
are a hell of a lot better than Live Tools. Probably better
than iTools, too. Maybe we should number them by the year
they were created. Then you could look back on them like
you look back on failed marriages or bad relationships.
You might even feel better about the new stuff. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [penteado](#) on Tue, 14 Dec 2010 23:20:44 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 14, 8:34 pm, Michael Galloy <mgal...@gmail.com> wrote:

> On 12/14/10 9:39 AM, David Fanning wrote:

>

>> P.S. Does anyone beside me and Dick Jackson see the
>> danger in calling a new graphics system "new"? Coyote
>> suggests the name Newest New Old Graphics System for
>> the work I've been doing lately, but I don't know... :-(

>

> Yes, that's why I thought "function graphics" would be a good name.
> Maybe some kind of numbered system? I think we would be up to new 4.0
> graphics now? With "old" graphics being direct graphics, then object
> graphics, Live Tools, iTools, and "new graphics"? Am I missing some
> other graphics systems?

Though NG is just what we have been calling them, instead of Graphics,
as the documentation does. I tend to use NG just so that those
unfamiliar with the term will not read "Graphics" as "graphics".

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Paul Van Delst\[1\]](#) on Tue, 14 Dec 2010 23:29:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Michael Galloy writes:

>

>> Yes, that's why I thought "function graphics" would be a good name.
>> Maybe some kind of numbered system? I think we would be up to new 4.0
>> graphics now? With "old" graphics being direct graphics, then object
>> graphics, Live Tools, iTools, and "new graphics"? Am I missing some
>> other graphics systems?

>

- > Gosh, I never thought I would end up a taxonomist. I may be missing
- > something, but aren't there five items on your list? Are we
- > starting the count from zero, like in IDL?

I reckon just three: Direct, Object, and Function (to use Michael's term) graphics.

Let's simply not mention LiveTools (I'm sure RSI/ITTVIS would like that too :o) and, IMO, iTools just never had sufficient mojo (with apologies to all the iToolistas out there....)

- > What in the world would you call these new FSC_*** routines
- > I've been working on? They aren't direct graphics, and they
- > are a hell of a lot better than Live Tools. Probably better
- > than iTools, too. Maybe we should number them by the year
- > they were created. Then you could look back on them like
- > you look back on failed marriages or bad relationships.
- > You might even feel better about the new stuff. :-)

I reckon you've more than earned your own category. So let's expand the list to four:

Direct Graphics
Object Graphics
Function Graphics
Coyote Graphics (or would you prefer Fanning Graphics?)

Hmm?

cheers,

paulv

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [David Fanning](#) on Tue, 14 Dec 2010 23:47:58 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

- > I reckon you've more than earned your own category. So let's expand the list to four:
- >
- > Direct Graphics
- > Object Graphics
- > Function Graphics
- > Coyote Graphics (or would you prefer Fanning Graphics?)

OK, I could live with Coyote Graphics. That way I could add the odd object graphics tool designed to look like something else (e.g. FSC_Surface) into the mix and still feel certain

no one would mistake it for something complicated. :-)

I still have a problem with Direct Graphics, though. I know we have called Direct Graphics Direct Graphics for 30+ years, but what does it mean!? "Raster Graphics" I think I understand. "Traditional Graphics" gives it a sort of stately, oldish sort of feel I kind of like. But I'm getting no traction whatsoever getting anyone but Coyote to call them this, and I have to twist his tail to get him to do it.

What would make someone who was told by his boss he needs to learn IDL to keep his job pop open his wallet and buy my book?

Maybe I should put a Coyote Story between every chapter. :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Michael Galloy](#) on Wed, 15 Dec 2010 04:47:03 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/14/10 3:19 PM, David Fanning wrote:

> Michael Galloy writes:

>

>> Yes, that's why I thought "function graphics" would be a good name.

>> Maybe some kind of numbered system? I think we would be up to new 4.0

>> graphics now? With "old" graphics being direct graphics, then object

>> graphics, Live Tools, iTools, and "new graphics"? Am I missing some

>> other graphics systems?

>

> Gosh, I never thought I would end up a taxonomist. I may be missing

> something, but aren't there five items on your list? Are we

> starting the count from zero, like in IDL?

Well, direct graphics would be the old graphics leaving four different "new" graphics systems. So this would basically be counting from zero, yes.

Mike

--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [Adam Lefkoff](#) on Wed, 15 Dec 2010 17:15:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/14/2010 3:34 PM, Michael Galloy wrote:

> On 12/14/10 9:39 AM, David Fanning wrote:

>> P.S. Does anyone beside me and Dick Jackson see the
>> danger in calling a new graphics system "new"? Coyote
>> suggests the name Newest New Old Graphics System for
>> the work I've been doing lately, but I don't know... :-(

>
> Yes, that's why I thought "function graphics" would be a good name. Maybe some
> kind of numbered system? I think we would be up to new 4.0 graphics now? With
> "old" graphics being direct graphics, then object graphics, Live Tools, iTools,
> and "new graphics"? Am I missing some other graphics systems?

>
> Mike

I seem to recall that there was a precursor to Live Tools named "Insight"
(circa 94/95). And if you're counting *all* the graphic systems, shouldn't
"Watsyn" make the cut ??

cheers,

Adam

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [chris_torrence@NOSPAM](#) on Wed, 15 Dec 2010 22:46:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 13, 8:18 pm, Paulo Penteado <pp.pente...@gmail.com> wrote:

>
> 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).
>

Hi Paulo,

I would agree. I'm going to change both the List::Remove() and Hash::Remove() *functions* to throw an error if the list/hash is empty. The List and Hash::Remove *procedures* will just quietly return (no need to throw an error), unless you actually specify an index (or key) in which case it will throw an error.

I'll also try to add an IsEmpty() method for IDL 8.1.

Cheers,
Chris
ITTVIS

Subject: Re: LIST "bug": .Remove on an empty list
Posted by [penteado](#) on Fri, 17 Dec 2010 17:35:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Dec 14, 1:35 pm, Paul van Delst <paul.vande...@noaa.gov> wrote:
> 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...

I just remembered that there is not much need for IsEmpty, as the lists currently already overload their truth value (to the opposite; an empty list evaluates to false):

```
IDL> l=list()
IDL> if (~l) then print,'empty list' else print,'list has
',n_elements(l),' elements'
empty list
IDL> l.add,9
IDL> if (~l) then print,'empty list' else print,'list has
',n_elements(l),' elements'
list has      1 elements
```

And I find "if (~l) then ..." much more convenient than "if (l.isempty()) then ...".
