Subject: More LIST weirdness?
Posted by Paul Van Delst[1] on Tue, 14 Dec 2010 19:55:26 GMT
View Forum Message <> Reply to Message

Hello,

Based on the other thread about list access errors, does the following make sense to anyone:

```
IDL> q = list()
IDL> help, q
Q               LIST  <ID=1  NELEMENTS=0>
IDL> help, q[0]
<Expression>    UNDEFINED = !NULL
IDL> help, q[1]
<Expression>    UNDEFINED = !NULL
IDL> help, q[2]
<Expression>    UNDEFINED = !NULL
IDL> x=q[5]
IDL> help, x
X               UNDEFINED = !NULL
```

??

I was expecting the "index out of range" type of error. Or e.g.

```
IDL> y=q[*]
% Illegal subscript range.
% Error occurred at: LIST::_OVERLOADBRACKETSRIGHTSIDE
%                    $MAIN$
% Execution halted at: $MAIN$
```

Now I'm confused. Removing an item causes the "LIST::REMOVE: Index is out of range" error. Why wouldn't simply accessing
an invalid index do a similar thing? The behaviour seems inconsistent. Or am I?

I guess the lesson is to always check the list length before removing from it, or referencing in it. That way there is
no special case and if/when ITTVIS changes the behaviour, no harm done. Right?

cheers,

paulv