Subject: Re: hashmap in idl
Posted by b.a on Wed, 22 Jul 2009 23:35:56 GMT
View Forum Message <> Reply to Message

On Jul 22, 1:09 pm, David Fanning <n...@dfanning.com> wrote:
> b.a writes:
>> After reading your code again, I realized what my problem was and I
>> hope I am right. I used to think that same as in Java, linkedlist is
>> predefined and it has all the methods relevant to it. But as I
>> understood, I have to define the structure of linkedlist and all the
>> methods I need, in separate programs and call them from my main
>> program.
>
> No, not at all. Why don't you tell us exactly
> what is happening when you try to use LinkedList
> and we can sort you out. :-)
>
> 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.")

Hi David,

Sorry for being so confusing :(here is what happens to my program:

I want to have a linkedlist that has several pairs of "key"(LONG) and "data"(a 2D array). each time I want to add something to the linkedlist, I will specify the key(which would be an id of one of the new created widgets in my program), and the data which is read from a file and be kept as 2D array. number of elements added to or deleted from the linkedlist is not fixed.

I used to think that if I just write for example:

```
  key1 = 197
  data1 = data
  mylist = Obj_New("LINKEDLIST")
  mylist->Add, key1, data1
```

it is enough and it should work. But it seems that first I have to define several methods or functions - such as defining the linkedlist structure, pro add-after, pro add-before, delete , ...- and then the compiler would recognize what "mylist->Add, key1, data1" means and so on. I mean before my main program I have to implement at least these:

```
PRO LINKEDLIST__DEFINE
PRO LINKEDLIST::ADD, item, index, Before=before, After=after
PRO LINKEDLIST::ADD_AFTER, item, index
PRO LINKEDLIST::ADD_BEFORE, item, index
PRO LINKEDLIST::ADD_TO_END, item
PRO LINKEDLIST::DELETE_NODE, index, DESTROY=destroy
FUNCTION LINKEDLIST::GET_NODE, index
FUNCTION LINKEDLIST::GET_ITEM, index, Dereference=dereference, ALL=all
```

here my key is actually the index, but I define it myself. I allocate a number to each data. Is it true?

Thank you