Subject: Re: hashmap in idl
Posted by b.a on Tue, 21 Jul 2009 06:18:09 GMT
View Forum Message <> Reply to Message

On Jul 20, 10:21 pm, David Fanning <n...@dfanning.com> wrote:
> b.a writes:
>> I have some 2D arrays that I want to give each of them, kind of index
>> so I can call them in my program by their indexes. Maybe something
>> like Hashmap in java and the number of arrays is not constant. I mean
>> in the middle of the program, new 2D arrays are created or eliminated
>> and I want to keep track of them.
>
> You could use Craig Markwardt's HashTable, or even my LinkedList
> object. Or, even a simple pointer array, for that matter.
>
> http://cow.physics.wisc.edu/~craigm/idl/arrays.html
> http://www.dfanning.com/programs/linkedlist__define.pro
>
>> I am also thinking of associating each 2D array to its relevant
>> widget_base which makes my program more efficient. But I don't know
>> how!!!
>
> I typically make an image object in these cases, with which window
> to draw into part of the "state" of an image, along with which color
> table to use to display it, how to scale it, etc.
>
> 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,

Thanks. HashTable is exactly what I want and I gone through the relevant page for that. But still I can't write even a simple code to actually use hashTable. I mean I am confused about the syntax. Is it like I have to define each method in a separate program and then call them from main?
I also tried the following which is the example provided, but no progress.

pro test

```
  mylist = Obj_New("LINKEDLIST", 5)
  mylist->Add, 10
  mylist->Add, 7, 1, /Before
  mylist->Add, 12
  print, mylist->Get_Item(/All, /Deref)
  mylist->Replace_Item, 1, 'Bob'
  mylist->Help
  mylist->Delete
  mylist->Help, /Print

end
```

Cheers

Subject: Re: hashmap in idl
Posted by b.a on Tue, 21 Jul 2009 06:21:09 GMT
View Forum Message <> Reply to Message

On Jul 20, 10:21 pm, David Fanning <n...@dfanning.com> wrote:
> b.a writes:
>> I have some 2D arrays that I want to give each of them, kind of index
>> so I can call them in my program by their indexes. Maybe something
>> like Hashmap in java and the number of arrays is not constant. I mean
>> in the middle of the program, new 2D arrays are created or eliminated
>> and I want to keep track of them.
>
> You could use Craig Markwardt's HashTable, or even my LinkedList
> object. Or, even a simple pointer array, for that matter.
>
> http://cow.physics.wisc.edu/~craigm/idl/arrays.html
> http://www.dfanning.com/programs/linkedlist__define.pro
>
>> I am also thinking of associating each 2D array to its relevant
>> widget_base which makes my program more efficient. But I don't know
>> how!!!
>
> I typically make an image object in these cases, with which window
> to draw into part of the "state" of an image, along with which color
> table to use to display it, how to scale it, etc.
>
> 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.")

Sorry not HashTable, LinkedList is what I have to use and I tried
linked list example.

Thanks

Subject: Re: hashmap in idl
Posted by David Fanning on Tue, 21 Jul 2009 12:30:43 GMT
View Forum Message <> Reply to Message

b.a writes:

> Thanks. HashTable is exactly what I want and I gone through the
> relevant page for that. But still I can't write even a simple code to
> actually use hashTable. I mean I am confused about the syntax. Is it
> like I have to define each method in a separate program and then call
> them from main?
> I also tried the following which is the example provided, but no
> progress.
>
> pro test
>
> mylist = Obj_New("LINKEDLIST", 5)
> mylist->Add, 10
> mylist->Add, 7, 1, /Before
> mylist->Add, 12
> print, mylist->Get_Item(/All, /Deref)
> mylist->Replace_Item, 1, 'Bob'
> mylist->Help
> mylist->Delete
> mylist->Help, /Print
>
> end

I'm not sure I understand you question. This example
works perfectly, as far as I can tell. Is something else
happening for you?

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: hashmap in idl
Posted by b.a on Tue, 21 Jul 2009 22:26:19 GMT
View Forum Message <> Reply to Message

On Jul 21, 10:30 pm, David Fanning <n...@dfanning.com> wrote:
> b.a writes:
>> Thanks. HashTable is exactly what I want and I gone through the
>> relevant page for that. But still I can't write even a simple code to
>> actually use hashTable. I mean I am confused about the syntax. Is it
>> like I have to define each method in a separate program and then call
>> them from main?
>> I also tried the following which is the example provided, but no
>> progress.
>
>> pro test
>
>> mylist = Obj_New("LINKEDLIST", 5)
>> mylist->Add, 10
>> mylist->Add, 7, 1, /Before
>> mylist->Add, 12
>> print, mylist->Get_Item(/All, /Deref)
>> mylist->Replace_Item, 1, 'Bob'
>> mylist->Help
>> mylist->Delete
>> mylist->Help, /Print
>
>> end
>
> I'm not sure I understand you question. This example
> works perfectly, as far as I can tell. Is something else
> happening for you?
>
> 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.")- Hide quoted text -
>

> - Show quoted text -

Hi David,

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.

Thank you

Subject: Re: hashmap in idl
Posted by David Fanning on Wed, 22 Jul 2009 03:09:59 GMT
View Forum Message <> Reply to Message

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.")

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

Subject: Re: hashmap in idl
Posted by David Fanning on Thu, 23 Jul 2009 02:10:00 GMT
View Forum Message <> Reply to Message

b.a writes:

> 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?

No, it is not true. In fact, it is so far from being true
it isn't even wrong. It's...I don't know. Nonsense, probably. :-)

But, clearly, you must have some reason for believing this.
What I have been trying to understand, so I can help you,
is what this reason is. Do you have some *evidence* you
would like to present that supports your idea?

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: hashmap in idl
Posted by b.a on Thu, 23 Jul 2009 03:30:14 GMT
View Forum Message <> Reply to Message

On Jul 23, 12:10 pm, David Fanning <n...@dfanning.com> wrote:
> b.a writes:
>> 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?
>
> No, it is not true. In fact, it is so far from being true
> it isn't even wrong. It's...I don't know. Nonsense, probably. :-)
>
> But, clearly, you must have some reason for believing this.
> What I have been trying to understand, so I can help you,
> is what this reason is. Do you have some *evidence* you
> would like to present that supports your idea?
>
> 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.")- Hide quoted text -
>
> - Show quoted text -

Hi David,

I didn't say I believe this, I am totally in doupt!!! But that was my understanding of linkedlist from what I searched on the web. The very first problem that I have is that I do not know how to define a linkedlist and add elements to it.
For other data structure, for example arrays, I can write a code which means create a 2D array. Add (x,y) to it or delete[m,n] from it, but I dont know how to write "create a linkedlist" and then add this to the linked list:

key1 data1
key2 data2
key3 data3
.
.
.
and then delete keyN dataN, etc.
for example in this code:

```
pro test


   mylist = Obj_New("LINKEDLIST", 5)
   mylist->Add, 10
   mylist->Add, 7, 1, /Before
   mylist->Add, 12
   print, mylist->Get_Item(/All, /Deref)
   mylist->Replace_Item, 1, 'Bob'
   mylist->Help
   mylist->Delete
   mylist->Help, /Print


end
```

when I just copy and paste it and compile it, it complains:

```
% Attempt to call undefined procedure/function: 'LINKEDLIST__DEFINE'.
% Execution halted at: TEST , ....
```

thats why I thought I have to write other programs in addition to
above to define the structure, Add, get-item and so on.

Cheers

Subject: Re: hashmap in idl
Posted by David Fanning on Thu, 23 Jul 2009 12:33:34 GMT
View Forum Message <> Reply to Message

b.a writes:

> when I just copy and paste it and compile it, it complains:
>
> % Attempt to call undefined procedure/function: 'LINKEDLIST__DEFINE'.
> % Execution halted at: TEST , ...
>

> thats why I thought I have to write other programs in addition to
> above to define the structure, Add, get-item and so on.

Ah, now we are getting somewhere. :-)

Your problem is that the file linkedlist__define.pro is not
on your IDL path. (Or, it is not spelled with all lowercase
letters if you are on a UNIX OS.) To use these programs,
they must be saved in lowercase letters and located in a folder
that is on your IDL path.

Here is an article that describes how to set up your IDL path
for the Coyote and Catalyst Libraries. You would do the same
thing for whatever folder holds this file.

http://www.dfanning.com/catalyst/howtoinstall.html

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: hashmap in idl
Posted by b.a on Fri, 24 Jul 2009 04:56:27 GMT
View Forum Message <> Reply to Message

On Jul 23, 10:33 pm, David Fanning <n...@dfanning.com> wrote:
> b.a writes:
>> when I just copy and paste it and compile it, it complains:
>
>> % Attempt to call undefined procedure/function: 'LINKEDLIST__DEFINE'.
>> % Execution halted at: TEST , ...
>
>> thats why I thought I have to write other programs in addition to
>> above to define the structure, Add, get-item and so on.
>
> Ah, now we are getting somewhere. :-)
>
> Your problem is that the file linkedlist__define.pro is not
> on your IDL path. (Or, it is not spelled with all lowercase

> letters if you are on a UNIX OS.) To use these programs,
> they must be saved in lowercase letters and located in a folder
> that is on your IDL path.
>
> Here is an article that describes how to set up your IDL path
> for the Coyote and Catalyst Libraries. You would do the same
> thing for whatever folder holds this file.
>
> http://www.dfanning.com/catalyst/howtoinstall.html
>
> 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,

Thanks a lot, now I konw what to do in order to fix this on my computer. Another thing is that my program should also work on other computers. Is it like that each computer which gonna use my program has to download the required library?

Cheers

Subject: Re: hashmap in idl
Posted by David Fanning on Fri, 24 Jul 2009 13:14:20 GMT
View Forum Message <> Reply to Message

b.a writes:

> Thanks a lot, now I konw what to do in order to fix this on my
> computer. Another thing is that my program should also work on other
> computers. Is it like that each computer which gonna use my program
> has to download the required library?

That would be the general idea, yes. :-)

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.")
